



**UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA CIVIL MECÁNICA**

**MODELAMIENTO COMPUTACIONAL DE LAS REACCIONES EN LOS APOYOS DE
UN ESPESADOR PARA LA MINERÍA**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL MECÁNICO

SVEN EDVARD HARFAGAR MANDIOLA

**PROFESOR GUÍA:
ALEJANDRO ORTÍZ BERNARDÍN**

**MIEMBROS DE LA COMISIÓN:
WILLIAMS CALDERÓN MUÑOZ
VIVIANA MERUANE NARANJO**

**SANTIAGO DE CHILE
2014**

**RESUMEN DE LA MEMORIA PARA OPTAR AL
TÍTULO DE: Ingeniero Civil Mecánico
POR: Sven Edvard Harfagar Mandiola
FECHA: 29/08/2014
PROFESOR GUÍA: Alejandro Ortiz Bernardín**

MODELAMIENTO COMPUTACIONAL DE LAS REACCIONES EN LOS APOYOS DE UN ESPESADOR PARA LA MINERÍA

Los espesadores son elementos que se utilizan en el proceso de extracción de múltiples minerales, siendo su función la de separar una parte sólida de una líquida del fango proveniente de las pilas de lixiviación. En general son estructuras estacionarias cuyas dimensiones y contenidos (densidades relativas) pueden variar bastante, por lo cual los cálculos asociados a los apoyos se hacen para cada caso en particular, aun cuando estos sean a través de un software, actividad que incurre en un gasto de tiempo y recursos.

Por este motivo el tema propuesto busca encontrar alternativas de automatización en el proceso de cálculo de la etapa de ingeniería temprana y considerar una opción para la programación de un prototipo que en base a las variables más relevantes del diseño, entregue una respuesta concreta con un error aceptable.

La mecánica de estructuras posee una amplia variedad de modelos para aproximar las deformaciones y estados de esfuerzos a los que está sujeto un fenómeno del mundo real. En este trabajo se detallan algunos de estos modelos utilizados por el método de elementos finitos para discretizar el problema propuesto.

En base a las condiciones de diseño encontradas entre los antecedentes de los espesadores y las características que favorecen un modelo de elementos finitos eficiente, se constituye una estructura simplificada que en base a tiempos bajos de cálculo podría entregar una buena aproximación al cuadro de reacciones.

Para elaborar el modelo de elementos finitos propuesto se trabaja en base a la metodología clásica de la literatura, que consiste básicamente en escoger los tipos de elementos, determinar sus rigideces, sus cargas y su disposición en el espacio (mallado), ensamblar todo en un sistema de ecuaciones, establecer las condiciones de borde del dicho sistema, y por último encontrar la solución.

Una vez encontrada la solución, la última instancia corresponde a la validación de los resultados. Como este trabajo no persigue fines estadísticos de la industria o un nivel de detalle avanzado, se puede aproximar la validez numérica de los elementos utilizados a partir de geometrías más sencillas. Y la validez cualitativa de la respuesta en base a diversos ensayos sobre un espesador.

Debido a que la geometría y los tipos de esfuerzos del problema son más bien sencillos, se estima que las respuestas encontradas por el modelo tengan un nivel de error aceptable para su objetivo, predecir en una etapa de ingeniería temprana.

Finalmente, este trabajo no es una versión comercial e incluye una variedad de mejoras propuestas por el autor.

Dedicatoria

Este trabajo está dedicado especialmente a la memoria de mi abuelo Gregorio, a su siembra familiar de magia lógica, números, lúdica y pacifismo. A su corazón abundante y la paciencia infinita para enseñarnos y guiarnos en todos los tiempos. Y al recuerdo de mi abuela Ana, a su voluntad y tesón de hierro, capaz de levantar a las familias del polvo teñido, en cuya mesa siempre comimos todos, de todos orígenes, colores y olores.

Dedico este trabajo a la ciencia, la investigación y el conocimiento acumulado de la humanidad. A la educación y el entendimiento como núcleo de la sociedad en contraste de la violencia y la economía. Al proceso revolucionario del cual formamos parte como generación y del cual espero que se beneficien las venideras.

Y por último, se lo dedico a mis hermanos Francisco, Gregorio, Gustavo y Alonso, a su afecto incondicional y a su palabra de sabiduría de las cuales forjé mis modelos.

Agradecimientos

Agradezco a mi familia, en primer lugar a mi madre por haberme dejado ser libre desde pequeño, por haber confiado siempre en mí, por haberme prestado apoyo en todo momento y por esa tradición de calor epidérmico, marca inconfundible en todos sus hijos.

A mi padre por haber formado parte de mi proceso de aprendizaje desde la infancia temprana, por haberse proyectado en mí, por su interés constante en cada actividad que he realizado, incluyendo este trabajo y por haber alimentado esa hambre de cultura y conocimiento que hizo posible haber estudiado esta carrera.

Agradezco también al departamento de Mecánica, a sus profesores, en especial a mi profesor guía Alejandro Ortiz por haberme recibido mil veces en su oficina y haberme explicado pacientemente muchos de los conocimientos comprendidos en este trabajo.

Agradezco a mis amigos, a los mecánicos, industriales, estudiantes y no estudiantes, por todos los momentos que compartimos, por las risas, los llantos, las locuras y la pasión que tienen todos de vivir cada minuto.

Por último agradezco a Katalina Díaz, por haberme acompañado en estos años, por haberme traspasado parte de esa lógica racional que hace todo calzar a la perfección. Y a su familia, por haberme recibido con los brazos abiertos.

Tabla de contenido

1	INTRODUCCIÓN.....	1
1.1	OBJETIVO GENERAL.....	1
1.2	OBJETIVOS ESPECÍFICOS.....	1
2	ANTECEDENTES.....	2
2.1	CONTEXTO EN LA HIDROMETALURGIA.....	2
2.2	ESPEADORES: FUNCIÓN, CARGAS Y GEOMETRÍA.....	3
2.3	LAS ECUACIONES MECÁNICAS.....	6
2.3.1	Viga con deformación unidimensional.....	8
2.3.2	Viga de Euler-Bernoulli.....	8
2.3.3	Placa con deformación debido a esfuerzos planos.....	10
2.3.4	Placa de Reissner-Mindlin.....	10
2.4	PRINCIPIO DE HAMILTON.....	11
2.5	MODELAMIENTO DE LA GEOMETRÍA Y DEFINICIÓN DEL MALLADO (1).....	12
2.5.1	Elemento Frame.....	14
2.5.2	Elemento Shell.....	16
2.6	DEFINICIÓN DE LAS PROPIEDADES DEL MATERIAL.....	17
2.6.1	Matriz de rigidez de un elemento finito (1).....	17
2.6.2	Integración numérica de la matriz de rigidez – Integral de Gauss.....	19
2.6.3	Coordenadas globales – Matriz de transformación.....	20
2.6.4	Ensamble de la matriz de rigidez.....	22
2.6.5	Especificación de las condiciones de borde y esfuerzos.....	22
3	PROCEDIMIENTO.....	25
3.1	ANÁLISIS DE LA GEOMETRÍA Y LOS TIPOS DE ESFUERZO.....	25
3.2	SELECCIÓN DE LOS ELEMENTOS.....	26
3.3	GENERACIÓN DE LOS NODOS Y LA MALLA.....	26
3.3.1	Conectividad del modelo.....	28
3.3.2	Mallado del manto.....	30
3.3.3	Mallado de los pilares y vigas.....	34
3.4	OBTENCIÓN DE LA MATRIZ DE RIGIDEZ PARA CADA ELEMENTO.....	35
3.4.1	Integración numérica para el elemento Shell.....	35

3.5	GENERACIÓN DEL VECTOR DE ESFUERZOS PARA EL ELEMENTO FRAME...	41
3.6	ENSAMBLE DE LAS MATRICES DE RIGIDEZ Y VECTORES DE ESFUERZOS...	41
3.7	CONSTITUCIÓN DE LAS CONDICIONES DE BORDE.....	42
3.8	RESOLUCIÓN DEL SISTEMA DE ECUACIONES	43
3.9	PRE-VALIDACIÓN DE LAS MATRICES DE RIGIDEZ.....	43
4	RESULTADOS	45
4.1	PRE-VALIDACIÓN DE ELEMENTO SHELL.....	45
4.1.1	Resultados para viga horizontal empotrada en un extremo y fuerza puntual aplicada en extremo libre.....	45
4.1.2	Resultados para viga vertical empotrada en un extremo y fuerza puntual aplicada en extremo libre.....	49
4.1.3	Resultados para viga horizontal empotrada en un extremo y esfuerzo uniforme en aplicado en todo su largo.....	51
4.2	PRE-VALIDACIÓN DE ELEMENTO FRAME.....	52
4.3	RESULTADOS ESPESADOR.....	53
4.3.1	Aplicación de las fuerzas de peso.....	53
4.3.2	Aplicación de fuerza puntual.....	54
4.3.3	Aplicación de esfuerzo debido a presión.....	56
4.4	SOFTWARE FINAL.....	58
4.4.1	Malla y estado de deformación.....	59
4.4.2	Cuadro de reacciones.....	61
4.4.3	Interfaz gráfica.....	64
5	DISCUSIÓN.....	65
5.1	ANÁLISIS DE PRE-VALIDACIÓN.....	65
5.2	ANÁLISIS DE RESULTADOS ESPESADOR.....	66
5.3	ANÁLISIS DE SOFTWARE FINAL	67
6	CONCLUSIONES.....	69
6.1	CONCLUSIONES RESPECTO AL TRABAJO DE TÍTULO	69
6.2	RECOMENDACIONES PARA OTROS PROGRAMADORES (DESARROLLOS POSIBLES).....	70
7	BIBLIOGRAFÍA.....	71
8	ANEXOS Y APÉNDICES	72
8.1	CÓDIGOS	72
8.1.1	Clase FEM_espesador	72
8.1.2	Función Contribución.....	74

8.1.3	Función RE_Shell.....	74
8.1.4	Función R_Frame.....	76
8.1.5	Función Mat_T3.....	77
8.1.6	Clase Espesador.....	77
8.1.7	Función Homogen.....	82
8.1.8	Clase FEM_barra_Shell_horizontal.....	83
8.1.9	Clase FEM_barra_Shell_vertical.....	85
8.1.10	Clase FEM_barra_Shell_distribuido.....	87
8.1.11	Clase FEM_barra_Frame.....	90

Índice de tablas

Tabla 3.1 Parametrización cuerpos geométricos	27
Tabla 3.2 Puntos evaluados para la integral de Gauss y factores de peso	38
Tabla 4.1 Nodos y coordenadas para ensayo 1, viga empotrada horizontal.....	45
Tabla 4.2 Elementos y sus nodos para ensayo 1, viga empotrada horizontal.....	45
Tabla 4.3 Valores ensayo 1 viga empotrada horizontal	47
Tabla 4.4 Valores ensayo 2 viga empotrada horizontal	48
Tabla 4.5 Resultados viga empotrada 20 elementos Shell	48
Tabla 4.6 Valores ensayo viga empotrada vertical.....	49
Tabla 4.7 Resultados viga empotrada vertical.....	49
Tabla 4.8 Reacciones en empotramiento viga voladizo vertical	50
Tabla 4.9 Valores ensayo viga horizontal con esfuerzo aplicado en todo su largo.....	51
Tabla 4.10 Resultados viga horizontal empotrada de 20 elementos Frame	52
Tabla 4.11 Parámetros simulación de espesador sujeto a fuerzas de peso	53
Tabla 4.12 Parámetros simulación de espesador con aplicación de fuerza puntual	54
Tabla 4.13 Geometría espesador pequeño.....	58
Tabla 4.14 Geometría espesador grande.....	58
Tabla 4.15 Geometría espesador lleno con fluido denso.....	58

1 INTRODUCCIÓN

Los espesadores son elementos que se utilizan en el proceso de extracción de múltiples minerales, siendo su función la de separar una parte sólida de una líquida del fango proveniente de las pilas de lixiviación. En general son estructuras estacionarias cuyas dimensiones y contenidos (densidades relativas) pueden variar bastante, por lo cual los cálculos asociados a los apoyos se hacen para cada caso en particular, aun cuando estos sean a través de un software, actividad que incurre en un gasto de tiempo y recursos.

Como referencia, en la actualidad, si un ingeniero desea encontrar las reacciones en las bases de los pilares del espesador, puede tardar semanas en diseñar el modelamiento mediante un software comercial de elementos finitos (genérico), mientras que a través de un software diseñado para este fin (especializado), podría tardar tan sólo segundos o minutos.

En términos del cliente del ingeniero, a la industria minera le interesa que los tiempos sean los más breves posibles, siendo en general, tan solo un par de días el plazo que pone en primera instancia para este tipo de cálculos.

Por este motivo el tema propuesto busca encontrar alternativas de automatización en el proceso de cálculo de la etapa de ingeniería temprana y considerar una opción para la programación de un prototipo que en base a las variables más relevantes del diseño, entregue una respuesta concreta con un error aceptable.

El objeto de este software sería reducir costos a través de la disminución de los tiempos para la industria minera, y en particular hacer uso de varias herramientas teóricas y prácticas, como lo son los métodos de análisis de elementos finitos o los modelos de esfuerzo y deformación tradicionales.

1.1 OBJETIVO GENERAL

- Desarrollar un prototipo funcional que trabaje con modelamiento por elementos finitos para predecir las reacciones en las columnas (a partir de cargas estáticas en un espesador para la minería).

1.2 OBJETIVOS ESPECÍFICOS

- Encontrar una forma de programación que entregue un resultado en un tiempo considerablemente menor al que tomaría hacerlo por otros medios (mediante un software comercial de elementos finitos en el cual el usuario debe modelar caso a caso la geometría del espesador).
- Validar que la solución encontrada para el estado de esfuerzos guarde cierta relación con la realidad y no se aleje demasiado de los resultados encontrados mediante los medios anteriormente mencionados.
- Establecer una metodología y una estructura programática que permita futuros avances para otros trabajos de título u algún otro actor interesado en el desarrollo del software.
- Generar una interfaz que sea amigable y suficientemente completa (en relación a las variables de entrada) para el usuario.

2 ANTECEDENTES

2.1 CONTEXTO EN LA HIDROMETALURGIA

En la industria de la minería existen diversos tipos de procesos de extracción y la hidrometalurgia es aquella rama particular de la minería que hace uso de soluciones líquidas, acuosas u orgánicas para separar el metal requerido del resto del mineral.

En general, un proceso de extracción por hidrometalurgia considera los siguientes pasos:

- Disolución del componente deseado presente en la fase sólida
- Concentración y/o purificación de la solución deseada
- Precipitación del metal deseado o sus compuestos

Durante la primera etapa se lleva a cabo la extracción de la roca que contiene el mineral, el chancado de las rocas, el apilamiento o transporte del material chancado y la lixiviación. Para efectos de este trabajo de título, la lixiviación es el único de particular interés.

La lixiviación consiste en la disolución en solución acuosa del metal deseado y existen distintos tipos, los cuales se pueden clasificar en dos grandes subgrupos:

- De lecho fijo:
 - In situ
 - En botaderos
 - En pilas
 - En bateas
- De pulpas:
 - Por agitación a presión ambiente
 - En autoclaves

El primer segmento de técnicas corresponde a aquel en que el material chancado está apilado de cierta forma y se ataca con el solvente de forma que la solución escurre a la etapa siguiente. Mientras que el segundo corresponde a aquel en que el material chancado es depositado en un contenedor agitado continuamente con el solvente, dentro del cual se produce la solución que será guiada al resto de las etapas del proceso.

Dentro del segundo conjunto de técnicas destaca como objetivo de este informe la lixiviación de pulpas por agitación a presión ambiente. Ésta técnica se utiliza principalmente en minas con una alta ley del mineral, pues en contraste con las otras técnicas, la disolución del material ocurre tan solo en horas, frente a los meses que puede tardar en las de lecho fijo. Además, es un proceso continuo altamente automatizable.

Sin embargo, este procedimiento también tiene desventajas frente a las otras técnicas. En primer lugar, su costo es bastante superior al resto debido a las instalaciones y tecnología necesaria. Y además, ocurre que la pulpa transportada desde los agitadores contiene una fase sólida (que no se ha disuelto aún), lo cual obliga a la integración de un proceso adicional que separe la fase líquida de la sólida.

En la industria, para llevar a cabo esta tarea, actualmente existen dos métodos, el espesamiento (hasta 55% de sólidos separados) y la filtración (hasta un 92% de sólidos separados). El espesamiento consiste en la decantación natural o forzada en grandes estanques. Estos estanques son los llamados espesadores.

En cada uno de estos estanques, el sólido decantado tiene aproximadamente un 40% de solución de impregnación, que posee alguna parte del metal a extraer, por lo tanto debe ser reprocesado. Para conseguir un nivel mínimo de pérdidas, el proceso de espesamiento se separa en varias etapas, como se puede ver en la Figura 2.1.

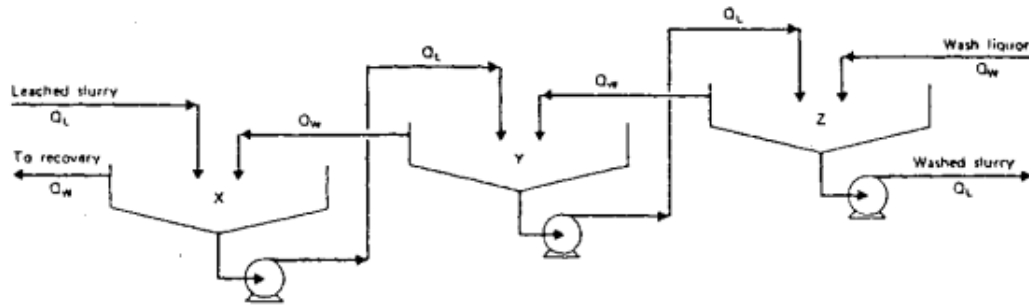


Figura 2.1 Circuito de decantación mediante espesadores

2.2 ESPESADORES: FUNCIÓN, CARGAS Y GEOMETRÍA

Como se comentó anteriormente, los espesadores consisten en estanques dentro de los cuales se lleva a cabo un proceso de decantación. Por un lado el espesador se alimenta de pulpa y por otro recibe un solvente para arrastrar el líquido de la parte superior. Los sólidos caen al fondo del estanque donde son enviados a una etapa anterior del ciclo para ser reprocesados. En la Figura 2.2 se visualiza el espesador de forma funcional.

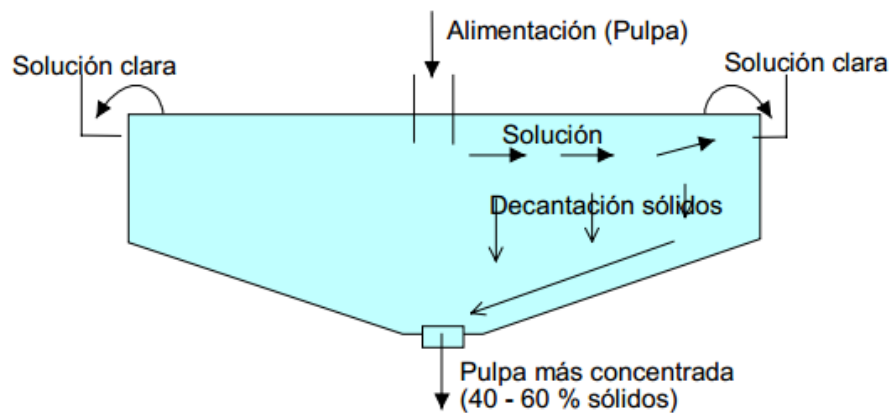


Figura 2.2 Esquema de un espesador [1]

Existen diversos diseños y tamaños de espesadores (pueden tener desde un par hasta cientos de metros de diámetro). Incluso para un mismo circuito, podrían requerirse espesadores distintos, debido a las concentraciones y flujos del ciclo. Visto de otra forma, cada espesador debe ser diseñado para las condiciones de operación que le corresponden.

Para efectos de este trabajo, los espesadores se pueden clasificar entre aquellos que usan una estructura metálica para soportar el peso de los fluidos y aquellos que son construidos bajo el nivel del suelo, como una excavación. Los dos tipos se pueden ver en la Figura 2.3.



Figura 2.3 Tipos de espesadores según tipos de apoyo

Para los espesadores soportados por una estructura metálica, los cálculos estructurales pueden tomar un buen tiempo en llevarse a cabo. Y si bien las condiciones de diseño pueden variar para cada ciclo de decantación, las cargas a las que están sometidas estas estructuras son básicamente estáticas y la mayor dificultad consiste en conocer cuáles son los esfuerzos a los que están sometidos los apoyos de la estructura.

Dada la condición de los espesadores de estar sujetos a cargas principalmente estáticas, la simulación mediante elementos finitos parece ser una alternativa viable. Pero más importante aún, a esto se suma el hecho de que la geometría de los espesadores es bastante sencilla, y a pesar de que pueden hacerse varios diseños sobre la estructura de soporte, en sus principios básicos (estanque y pilares) son bastantes similares. En la Figura 2.4 se muestra la estructura simplificada de un espesador.

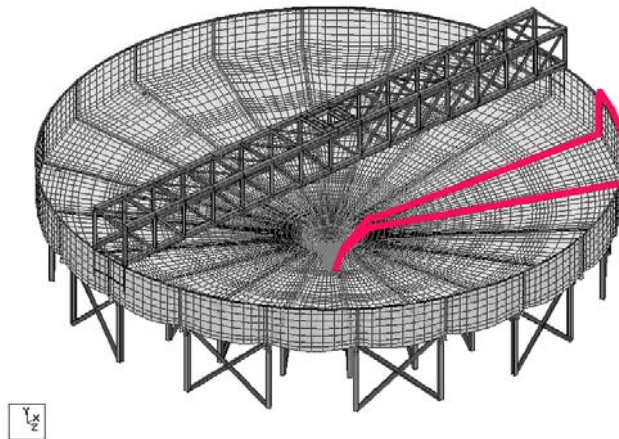


Figura 2.4 Estructura simplificada espesador, construida en elementos finitos.

De la figura anterior, se puede ver que el estanque tiene un esqueleto, formado por las distintas secciones del manto (como la que está resaltada). Cada una de las separaciones radiales que forman la sección corresponde a una viga. A su vez en cada una de estas vigas contiene varios pilares que van desde el cono hasta el suelo.

A grandes rasgos, la geometría del manto se compone de un cilindro y un cono que cambia su generatriz hacia la punta y al final está recortado. En la punta del cono hacia abajo podría haber un último apoyo en forma de viga que no se alcanza a distinguir (depende del tipo de espesador).

Por esta razón, de modo previo a una investigación más profunda, se puede suponer que las variables más relevantes de diseño para un espesador podrían ser las siguientes:

- El diámetro del cilindro
- La altura del cilindro
- La cantidad de separaciones radiales que tiene
- La cantidad de vigas verticales por cada separación
- El ángulo de las generatrices
- La altura desde la punta del cono pequeño hasta el suelo

2.3 LAS ECUACIONES MECÁNICAS

La mecánica de estructuras posee una amplia variedad de modelos para aproximar las deformaciones y estados de esfuerzos a los que está sujeto un fenómeno del mundo real.

Los sólidos y estructuras se deforman cuando son sometidas a algún esfuerzo o carga. Las ecuaciones mecánicas son aquellas que relacionan la magnitud de las fuerzas con las tensiones internas que experimenta un material y así mismo con su deformación.

Por tensión (del inglés *stress*), se entenderá a la cantidad física que indica las fuerzas internas que experimentan las partículas vecinas en un material continuo cuando por acción de un esfuerzo externo son sacadas de su posición de equilibrio.

Por deformación, se entenderá a la cantidad física que indica cómo cambia la geometría de un cuerpo desde un estado inicial a otro estado final.

Las fuerzas a las que son sometidos los cuerpos pueden ser estáticas o dinámicas, siendo las primeras aquellas que se mantienen constantes en el tiempo y las segundas las que involucran algún cambio, ya sea en su magnitud o dirección.

En términos generales, los sistemas sometidos a fuerzas estáticas pueden ser derivados como un caso particular de las ecuaciones que gobiernan los sistemas dinámicos, así, las ecuaciones para sistemas estáticos, usualmente se extraen obviando la parte dinámica del sistema.

Así, las teorías de deformación, son aquellas que asumen una cierta forma para la cual cambiará la geometría de los cuerpos que son sometidos a un esfuerzo.

La primera ecuación constitutiva, que relaciona el esfuerzo con la deformación (lineal), corresponde a la ley de Hooke:

$$E\varepsilon = \sigma \quad (2.1)$$

Dónde E es el módulo de Young, que representa la resistencia del material a la deformación. Además, ε corresponde la deformación (en una dimensión) y σ al esfuerzo o la tensión a la que está sujeto el material.

Este es el caso particular para deformación unidimensional, si luego esta ecuación se lleva al caso bidimensional o tridimensional, entonces aparecen esfuerzos en otras direcciones además de aquella en que se está aplicando la fuerza externa.

Ahora, para el caso general, se tiene que dado un punto dentro de un cuerpo sólido, cualquier esfuerzo se puede descomponer en seis componentes, a menudo llamado tensor de esfuerzos, escrito de forma vectorial en la (2.2).

$$\sigma^T = \{\sigma_{xx} \sigma_{yy} \sigma_{zz} \sigma_{zy} \sigma_{zx} \sigma_{xy}\} \quad (2.2)$$

Que a su vez están asociados a las seis componentes en las que se puede expresar una deformación, expresados de forma vectorial en la (2.3).

$$\varepsilon^T = \{\varepsilon_{xx} \varepsilon_{yy} \varepsilon_{zz} \varepsilon_{yz} \varepsilon_{xz} \varepsilon_{xy}\} \quad (2.3)$$

Luego, matemáticamente, la deformación corresponde a la derivada del desplazamiento del punto en relación a la cantidad de desplazamiento, es decir:

$$\begin{aligned} \varepsilon_{xx} &= \frac{\partial u}{\partial x}; \varepsilon_{yy} = \frac{\partial v}{\partial y}; \varepsilon_{zz} = \frac{\partial w}{\partial z}; \\ \varepsilon_{xy} &= \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}; \varepsilon_{xz} = \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}; \varepsilon_{yz} = \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \end{aligned} \quad (2.4)$$

Que se puede expresar de forma matricial como:

$$\varepsilon = LU \quad (2.5)$$

Donde L es la matriz de operadores diferenciales parciales y U es el vector con los desplazamientos.

$$L = \begin{Bmatrix} \partial/\partial x & 0 & 0 \\ 0 & \partial/\partial y & 0 \\ 0 & 0 & \partial/\partial z \\ 0 & \partial/\partial z & \partial/\partial y \\ \partial/\partial z & 0 & \partial/\partial x \\ \partial/\partial x & \partial/\partial x & 0 \end{Bmatrix}; U = \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \quad (2.6)$$

La matriz de operadores diferenciales es la que varía según el modelo de deformación que se asuma. Puesto que las magnitudes de deformación en una dirección determinada, variarán según las deformaciones en otras direcciones, según lo determine cada modelo.

La misma generalización es posible para el módulo de Young. Esto quiere decir que la (2.1) puede ser reescrita como:

$$\sigma = c\varepsilon \quad (2.7)$$

En donde la matriz c es la resistencia material según cada componente del plano cartesiano. Como se espera que $c_{ij} = c_{ji}$, resulta evidente que la matriz c es simétrica. Además, cabe recalcar que si bien para un material anisotrópico, todas las componentes son distintas, para uno anisotrópico, la matriz se simplifica en la (2.9).

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{xz} \\ \sigma_{yz} \end{Bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} & c_{16} \\ & c_{22} & c_{23} & c_{24} & c_{25} & c_{26} \\ & & c_{33} & c_{34} & c_{35} & c_{36} \\ & & & c_{44} & c_{45} & c_{46} \\ & sim & & & c_{55} & c_{56} \\ & & & & & c_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \varepsilon_{xy} \\ \varepsilon_{xz} \\ \varepsilon_{yz} \end{Bmatrix} \quad (2.8)$$

$$c_{isotropico} = \begin{bmatrix} c_{11} & c_{12} & c_{12} & 0 & 0 & 0 \\ & c_{11} & c_{12} & 0 & 0 & 0 \\ & & c_{11} & 0 & 0 & 0 \\ & sim & & (c_{11} - c_{12})/2 & 0 & 0 \\ & & & & (c_{11} - c_{12})/2 & 0 \\ & & & & & (c_{11} - c_{12})/2 \end{bmatrix} \quad (2.9)$$

Para este último caso, las ecuaciones de las constantes aparecen en múltiples fuentes bibliográficas. [2] [3] [4].

$$c_{11} = \frac{E(1 - \nu)}{(1 - 2\nu)(1 + \nu)}; \quad c_{12} = \frac{E\nu}{(1 - 2\nu)(1 + \nu)}; \quad \frac{c_{11} - c_{12}}{2} = G \quad (2.10)$$

En donde E es el Módulo de Young del material, ν el Módulo de Poisson y G es el Módulo de corte, que según las relaciones de la (2.10), queda determinado por la (2.11).

$$G = \frac{E}{2(1 + \nu)} \quad (2.11)$$

2.3.1 Viga con deformación unidimensional

Probablemente más simple de los modelos de deformación. No existen muchas suposiciones acerca de este modelo, puesto que se basa en la ley de Hook unidimensional. La dirección admisible de deformación y desplazamiento es sólo una y es en el sentido axial de la viga. La ecuación de deformación corresponde a:

$$\epsilon_{xx} = \frac{\partial u}{\partial x} \rightarrow L = \frac{\partial}{\partial x}; \quad U = \{u\} \quad (2.12)$$

Para este caso no hay más esfuerzos que el normal σ_{xx} .

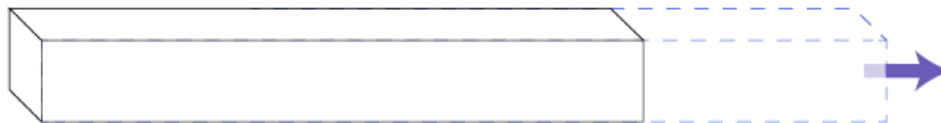


Figura 2.5 Viga con deformación unidimensional

2.3.2 Viga de Euler-Bernoulli

Si al modelo anterior ahora se le suma el hecho de que la viga puede deformarse además en sentido perpendicular a su eje principal (dado que exista deflexión), entonces el problema se vuelve más complejo. Uno de los modelos más usados para asumir la forma en que se deforma la viga es el de Euler-Bernoulli.

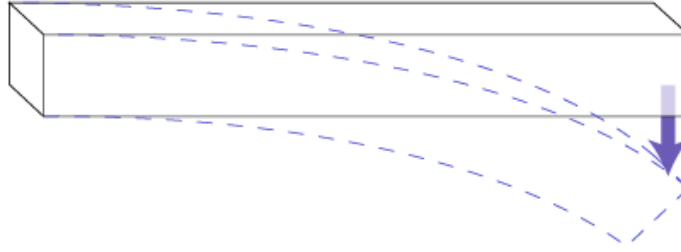


Figura 2.6 Viga sujeta a flexión

Se define como eje neutral de la viga, la línea que atraviesa la estructura y que no se alarga ni comprime luego de una deflexión. La presunción principal de este problema consiste en asumir que los planos de corte al eje neutral de la viga que son inicialmente perpendiculares, se mantendrán perpendiculares luego de ocurrida la deflexión.

Matemáticamente, esto quiere decir que se cumple la relación explicitada en la (2.13 con la correspondiente aproximación de seno por el ángulo para deformaciones pequeñas.

$$u = -z \cdot \text{sen}(\theta) \cong -z \cdot \theta \quad (2.13)$$

Esto quiere decir que no hay movimiento de material en el plano normal a x en dirección al eje z, con respecto al eje neutro, dicho de otra forma, lo que a su vez quiere decir que los esfuerzos de corte para ese plano son despreciables.

$$\varepsilon_{xz} = 0 \rightarrow \sigma_{xz} = 0 \quad (2.14)$$

Por otra parte, el ángulo θ puede ser escrito de forma diferencial como:

$$\theta = \frac{\partial w}{\partial x} \quad (2.15)$$

Luego, reemplazando la deformación en sentido axial puede ser reescrita como:

$$\varepsilon_{xx} = \frac{\partial u}{\partial x} = \frac{\partial}{\partial x} \left(-z \frac{\partial w}{\partial x} \right) = -z \frac{\partial^2 w}{\partial x^2} = -z L w \rightarrow L = \frac{\partial^2}{\partial x^2}; U = \{w\} \quad (2.16)$$

Esta última ecuación relaciona la deformación longitudinal (en sentido axial) con la deflexión. Como la viga tiene simetría cilíndrica se puede aplicar el mismo principio si existiera deflexión en más de un sentido.

2.3.3 Placa con deformación debido a esfuerzos planos

En términos mecánicos, se puede entender como placa, aquel cuerpo cuyo espesor es relativamente pequeño en relación a sus otras dimensiones, y que además está sujeto a esfuerzos externos principalmente normales.

En particular una placa sometida a esfuerzos planos es muy similar al caso de la viga con deformación unidimensional, pero con una dirección adicional, lo que además de añadir otra componente de deformación, añade una deformación en corte.

En este modelo, la placa sólo se deforma en direcciones dentro del plano que conforma. Vale decir, no considera deflexión.

La deformación para este caso queda determinada por los desplazamientos \vec{u} y \vec{v} , y no requiere relaciones adicionales, más que las constitutivas.

$$\varepsilon_{xx} = \frac{\partial u}{\partial x}; \varepsilon_{yy} = \frac{\partial v}{\partial y}; \varepsilon_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}; \rightarrow L = \begin{bmatrix} \partial/\partial x & 0 \\ 0 & \partial/\partial y \\ \partial/\partial y & \partial/\partial x \end{bmatrix}; U = \begin{Bmatrix} u \\ v \end{Bmatrix} \quad (2.17)$$

2.3.4 Placa de Reissner-Mindlin

Este es un modelo de deformación para deflexión de placas y sugiere que la deformación está determinada por el desplazamiento perpendicular al plano (\vec{w}), y las rotaciones según los ejes x e y (θ_x y θ_y respectivamente)

Similar al caso de la viga de Bernoulli, este modelo asume que el desplazamiento horizontal en los sentidos x e y , puede ser aproximado por el ángulo formado en la deflexión descompuesta en esos dos ejes. Esto se puede expresar según la (2.18).

$$\begin{aligned} u &= z\theta_y \\ v &= -z\theta_x \end{aligned} \quad (2.18)$$

Luego, la deformación queda determinada como:

$$\varepsilon = -zL\theta$$

En la literatura en general se separan las deformaciones debido a tensiones normales ((2.19) y las deformaciones debido a tensiones de corte ((2.20).

$$\varepsilon = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{Bmatrix} = L_1 \{w, \theta_x, \theta_y\} \rightarrow L_1 = \begin{bmatrix} 0 & 0 & -\partial/\partial x \\ 0 & \partial/\partial y & 0 \\ 0 & \partial/\partial x & -\partial/\partial y \end{bmatrix}; U = \begin{Bmatrix} w \\ \theta_x \\ \theta_y \end{Bmatrix} \quad (2.19)$$

$$\gamma = \begin{Bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} = L_2 \{w, \theta_x, \theta_y\} \rightarrow L_2 = \begin{bmatrix} \partial/\partial x & 0 & 1 \\ \partial/\partial y & -1 & 0 \end{bmatrix}; U = \begin{Bmatrix} w \\ \theta_x \\ \theta_y \end{Bmatrix} \quad (2.20)$$

2.4 PRINCIPIO DE HAMILTON

Para encontrar la solución de un problema, la base que explica cómo opera el método de elementos finitos es el llamado principio de Hamilton, que se enuncia de la siguiente forma:

“De todo el historial de desplazamientos posibles, la solución más exacta es aquella que minimiza la función del Lagrangiano” [2]

Un desplazamiento se dice posible si cumple con las siguientes condiciones:

- (a) Las ecuaciones de compatibilidad
- (b) Las ecuaciones esenciales o condiciones cinéticas de borde
- (c) Las condiciones en un tiempo inicial y un tiempo final

La condición (a) establece la compatibilidad entre los desplazamientos en el dominio. Para este caso, la compatibilidad geométrica entre los elementos. La (b) establece que se deben cumplir las condiciones de borde, y la (c) que dicho desplazamiento cumpla con las otras condiciones para un tiempo inicial y uno final.

Matemáticamente, el principio de Hamilton establece que:

$$\delta \int_{t_1}^{t_2} L dt = 0 \quad (2.21)$$

La función lagrangiano L , es obtenida usando el historial de desplazamientos posibles:

$$L = T - \Pi - W_f \quad (2.22)$$

En donde T es la energía cinética, Π es la energía potencial y W_f es el trabajo realizado por fuerzas externas. Si se desarrolla la ecuación 2.1 para un sistema mecánico se obtiene que¹:

$$KU + M\ddot{U} = F \quad (2.23)$$

En donde K es la matriz de rigidez, que representa el ensamble de todas las rigideces existentes para cada elemento, D es la matriz de desplazamiento, que representa el ensamble de todos los desplazamientos de cada elemento, M es la matriz asociada a la masa de cada elemento, \ddot{D} es la aceleración de cada elemento y F es la fuerza externa aplicada a cada elemento.

Luego, para un problema estático, la aceleración es nula, por lo tanto se tiene que:

$$KU = F \quad (2.24)$$

¹ Como no es motivo de este trabajo de título, se obvia el procedimiento a partir del cual se obtiene esta ecuación matricial, sin embargo, en su fuente, está bastante detallado.

Esta es la forma clásica de fuerza (proveniente de la energía elástica) asociada a un resorte lineal, sólo que de forma matricial. La condición más relevante para modelar un problema como elasticidad lineal, es que se trate de pequeñas deformaciones.

2.5 MODELAMIENTO DE LA GEOMETRÍA Y DEFINICIÓN DEL MALLADO [2]

Para constituir la matriz de rigidez mencionada en el capítulo anterior y encontrar los desplazamientos y así mismo el estado de esfuerzos del sistema, primero se debe conocer la rigidez de los elementos. A su vez, para determinar la ecuación que rige la rigidez de cada elemento, primero se debe constituir la geometría del modelo y hacer una selección de los elementos más adecuados.

La construcción de la geometría consiste básicamente en localizar puntos en el espacio con una forma deseada. La distribución de los puntos dependerá en gran medida del mallado que se quiera hacer y este a su vez dependerá de la forma que se escoja para los elementos.

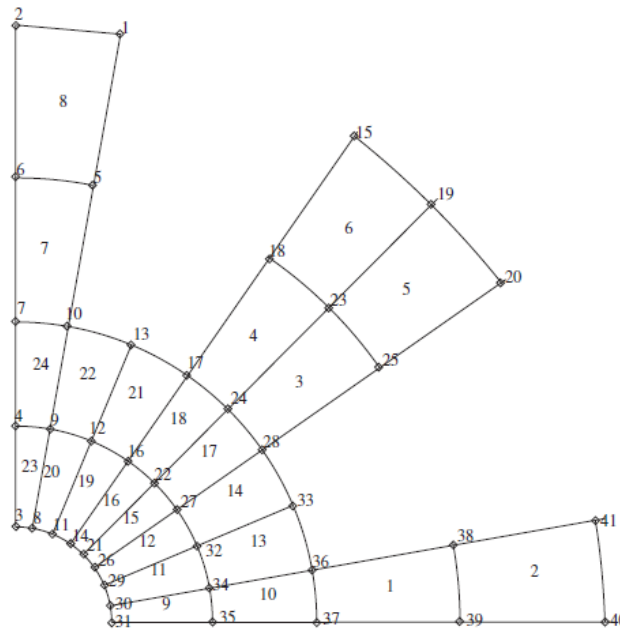


Figura 2.7 Enumeración de puntos y mallado para una geometría en 2D

Dicho esto, un elemento es básicamente un lugar geométrico compuesto por los puntos que el modelo geométrico defina. Un elemento puede ser una línea de puntos, un plano o un volumen. En la Figura 2.7 se puede ver en el centro de cada paralelogramo un número que corresponde a la designación del número del elemento, y los números en los puntos que demarcan la numeración de los nodos. En la Figura 2.8 se puede ver una clasificación de los tipos comunes de elementos que existen según la biblioteca del software para análisis por modelamiento de elementos finitos ANSYS.












Element Order	2D Solid	3D Solid	3D Shell	Line Elements
Linear	 PLANE42 PLANE182	 SOLID45 SOLID185	 SHELL63 SHELL181	 BEAM3/44  BEAM188
Quadratic	 PLANE82/183  PLANE2	 SOLID95/186  SOLID92/187	 SHELL93	 BEAM189

Figura 2.8 Tipos comunes de elementos para modelación por elementos finitos

Otro punto a considerar en el posicionamiento de los nodos, son las uniones entre los diversos elementos que conforman el cuerpo. Esto quiere decir que para que dos elementos tengan conectividad, deben compartir por lo menos un nodo.

En el análisis estructural de la mecánica clásica, se hacen simplificaciones al modelo geométrico dependiendo de qué fenómeno se requiera estudiar, por ejemplo, cuando se estudian los esfuerzos que soporta una placa metálica en sentido normal, usualmente se asume que dicha placa es plana, debido a que su grosor es de una magnitud despreciable frente a sus otras dimensiones.

Esto mismo ocurre en los elementos finitos, el tipo de elemento óptimo para un fenómeno en particular, dependerá de las simplificaciones que caigan a lugar debido la geometría, la naturaleza del fenómeno en cuestión y la teoría física elegida.

Luego de que se escogen los tipos de elementos que se van a usar, se tiene que generar una serie de coherencias en los nodos (puntos) de forma que haya continuidad entre la geometría y los elementos. Para hacer esto, es necesario diseñar un método de segmentación del cuerpo y de enumeración de los nodos.

El tamaño de los elementos es una variable especialmente importante a considerar en el proceso de mallado. Es lógico que mientras más pequeños los elementos (más fina la malla), mayor el número de estos y así mismo más lenta la obtención de los resultados, y posiblemente con un error menor. Por esta razón, la malla no es necesariamente homogénea a lo largo de una geometría, y puede ser recomendable que sea más fina en los lugares del cuerpo en que se requiere un mayor grado de exactitud.

Sin embargo, esto nuevamente dependerá del tipo de cuerpo y fenómeno que se desee analizar, puesto que hay geometrías para las que un mallado relativamente grueso puede resultar en un error aceptable. Esto ocurre sobre todo en geometrías sencillas y cuerpos simples. Como por ejemplo, para una viga en voladizo con esfuerzo en el extremo libre y empotramiento en el extremo contrario, se espera que si dicho fenómeno es modelada por elementos Frame, baste con cuatro o cinco elementos para que la respuesta a la deformación sea casi exactamente la misma que la predicha por el modelo teórico de Euler-Bernoulli (error menor al 0,01%).

2.5.1 Elemento Frame

El elemento Frame es un simple pero muy usado componente estructural para el análisis mediante elementos finitos. Geométricamente es una línea recta de sección arbitraria, formada por dos nodos, que puede ser sometida a deformaciones perpendiculares y normales a su eje. Existen otros elementos lineales, tales como el Truss, o Beam sin embargo, este es el que tiene más grados de libertad en cuanto a deformación.

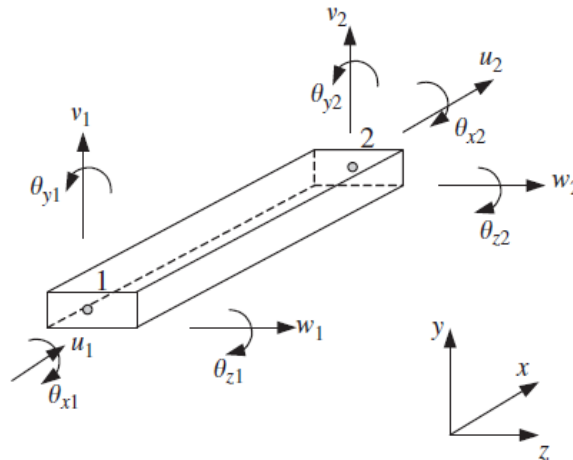


Figura 2.9 Elemento Frame planar y sus grados de libertad [2]

Clásicamente, los elementos Frame se utilizan para modelar vigas o pilares delgados. Los elementos se unen entre sí mediante soldadura lo que quiere decir que se transmiten las fuerzas y momentos entre ellos. Para esta descripción, las secciones de corte a través de elementos son definidas constantes.

Los grados de libertad de cada nodo del elemento son seis, el desplazamiento en la dirección x , (u), el desplazamiento en la dirección y , (v), el desplazamiento en la dirección z , (w) y la rotación con respecto a cada eje, (θ_i). Esto da origen a 12 grados de libertad por elemento, debido a que son 2 nodos.

Esto quiere decir que la matriz de rigidez poseerá una dimensión de 12×12 . En términos comparativos, este tamaño es relativamente bajo, por lo cual se extiende que el tiempo y la capacidad requerida para obtener los desplazamientos de este tipo de elemento, debería ser baja.

El procedimiento para obtener la matriz de rigidez de este elemento es el documentado en la Sección 2.7.1, utilizando los modelos de deformación de viga con deformación unidimensional y la viga de Euler-Bernoulli, en conjunto. Sin embargo en la literatura se puede encontrar la matriz de rigidez para este elemento expresada de forma analítica, como sugiere la Figura 2.10.

$$\mathbf{k}_e = \begin{bmatrix}
\begin{matrix} u_1 & v_1 & w_1 & \theta_{x1} & \theta_{y1} & \theta_{z1} & u_2 & v_2 & w_2 & \theta_{x2} & \theta_{y2} & \theta_{z2} \\ \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \end{matrix} \\
\frac{AE}{2a} & 0 & 0 & 0 & 0 & 0 & -\frac{AE}{2a} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{3EI_z}{2a^3} & 0 & 0 & 0 & \frac{3EI_z}{2a^2} & 0 & -\frac{3EI_z}{2a^3} & 0 & 0 & 0 & \frac{3EI_z}{2a^2} \\
0 & \frac{3EI_y}{2a^3} & 0 & -\frac{3EI_y}{2a^2} & 0 & 0 & 0 & 0 & -\frac{3EI_y}{2a^3} & 0 & -\frac{3EI_y}{2a^2} & 0 \\
0 & 0 & \frac{GJ}{2a} & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{GJ}{2a} & 0 & 0 \\
0 & 0 & 0 & \frac{2EI_y}{a} & 0 & 0 & 0 & 0 & \frac{3EI_y}{2a^2} & 0 & \frac{EI_y}{a} & 0 \\
0 & 0 & 0 & 0 & \frac{2EI_z}{a} & 0 & -\frac{3EI_z}{2a^2} & 0 & 0 & 0 & 0 & \frac{EI_z}{a} \\
-\frac{AE}{2a} & 0 & 0 & 0 & 0 & 0 & \frac{AE}{2a} & 0 & 0 & 0 & 0 & 0 \\
0 & -\frac{3EI_z}{2a^3} & 0 & 0 & 0 & 0 & 0 & \frac{3EI_z}{2a^3} & 0 & 0 & 0 & -\frac{3EI_z}{2a^2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{3EI_y}{2a^3} & 0 & \frac{3EI_y}{2a^2} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{GJ}{2a} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2EI_y}{a} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2EI_z}{a}
\end{bmatrix}$$

sy.

Figura 2.10 Matriz de rigidez para un elemento frame [2]

A es el área de la sección del elemento, E su módulo de Young y a es la mitad del largo del elemento, I_y e I_z son los segundos momentos de inercia respecto al eje y y z respectivamente y por último G es el módulo de corte y J es el momento polar de inercia.

Esta matriz de rigidez se encuentra dispuesta según las coordenadas locales del elemento, vale decir, las componentes de la matriz se encuentran orientadas según un sistema situado en uno de los extremos de la misma barra.

En el Capítulo 2.6.3, se explica cómo a través de una matriz de transformación la matriz de rigidez pasa a estar orientada según el origen, el sistema de coordenadas definido por la geometría global del sistema.

2.5.2 Elemento Shell

Un elemento Shell, es en esencia una estructura que puede ser derivada de una Placa, formando el plano medio como una superficie simple o doblemente curvada. [4].

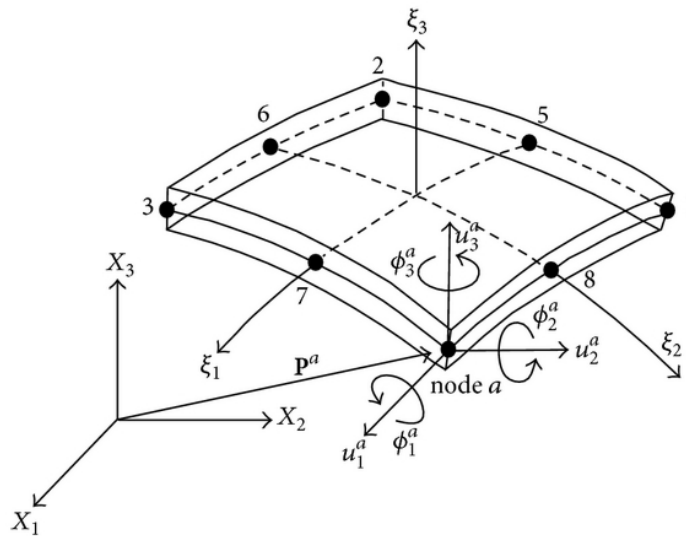


Figura 2.11 Esquema representativo de un elemento Shell de 8 nodos [5]

Este es un elemento que admite esfuerzos y deformaciones en tres direcciones para cada nodo, y torsiones y momentos en tres direcciones para cada nodo. Esto se traduce en esfuerzos de flexión debido a torsiones o flexiones y a esfuerzos planares debido a tensiones en alguna dirección del plano. Se usa principalmente para modelar fuselajes, techos con formas especiales, estanques de agua, cilíndricos o semiesféricos, y en general, problemas en que la pared del cuerpo a analizar sea relativamente delgada en relación a las otras dimensiones del cuerpo, de tal forma que los esfuerzos más relevantes sean los normales a la pared.

Si bien por definición, estos elementos admiten curvaturas, se ha observado que si son planos y rectos pueden dar un buen resultado en geometrías sencillas, con una fineza del mallado que permita simular la curvatura del cuerpo.

Existen elementos Shell de un número indefinido de nodos, sin embargo, los más recurrentes en la literatura, son el elemento de 4 y el de 8 nodos. En ambos casos, se considera que cada uno de los nodos tiene seis grados de libertad, tres de desplazamiento y tres de rotación, como se puede ver en la Figura 2.11.

La elección entre uno y otro tipo de elemento dependerá del grado de exactitud que se requiera en los resultados y en su contraparte, del tiempo y la capacidad de procesamiento que pudiera requerir el modelo. Mientras que el de 4 nodos considera invertir matrices de 24×24 , para el de 8 son matrices de 48×48 .

A diferencia del elemento Frame, cuya matriz de rigidez se suele expresar de forma analítica, para el caso de los elementos Shell, lo que se sugiere en la literatura es que se haga mediante una integración numérica. Un procedimiento para obtener la matriz de deformación y así la de rigidez es el que se describe en la Sección 2.7.1.

2.6 DEFINICIÓN DE LAS PROPIEDADES DEL MATERIAL

La sección anterior muestra como modelar una geometría, dependiendo de las características del problema y así el tipo de elemento que se escoja. Luego de que esa etapa está definida, el paso siguiente es obtener las matrices que caracterizan a cada elemento, para así aplicar las características mecánicas del material y ensamblar todo en un solo problema (sistema de ecuaciones).

Como se menciona en la Sección 2.5, los equilibrios mecánicos del sistema están representados por la (2.24). Por lo general lo que se busca en un análisis estático es obtener los desplazamientos de ese sistema de ecuaciones, a partir de una cierta matriz de rigidez y de una cierta matriz de esfuerzos.

En el capítulo anterior, también se menciona la matriz de rigidez de cada elemento, sin embargo para resolver el sistema, hay que ensamblar todas las matrices de todos los elementos en lo que se conoce como matriz global de rigidez K .

2.6.1 Matriz de rigidez de un elemento finito [2]

Un elemento finito puede ser caracterizado a partir de sus matrices de rigidez, deformación y esfuerzo. Para determinar dichas matrices generalmente se realiza el siguiente procedimiento:

1. Determinación de las funciones de forma
2. Obtención de la matriz de deformación
3. Integración en la matriz de rigidez

Como se menciona en el capítulo anterior, un elemento está determinado en el espacio por los nodos que lo conforman. Sin embargo, para medir el desplazamiento y la deformación del elemento se recurre a una técnica llamada *interpolación de movimiento*, que dice que se puede aproximar el desplazamiento del elemento como una función de forma por el desplazamiento de cada nodo.

$$U^h(x, y, z) = \sum_{i=1}^{n_d} N_i(x, y, z) d_i = N(x, y, z) d_e = \begin{bmatrix} N_1 & 0 & 0 & \dots & N_1 & 0 & 0 \\ 0 & \ddots & 0 & \dots & 0 & \ddots & 0 \\ 0 & 0 & N_n & \dots & 0 & 0 & N_n \end{bmatrix} \cdot \begin{Bmatrix} d_1 \\ \vdots \\ d_n \end{Bmatrix} \quad (2.25)$$

En la (2.25), U^h denota el desplazamiento aproximado del elemento, N_i la función de forma asociada a cada nodo, n_d es el número de nodos, d_i el desplazamiento en cada grado de libertad de cada nodo. El lóbulo del extremo derecho es la forma matricial en donde N es la matriz que reúne las funciones de forma y d_e el desplazamiento en todos los grados de libertad de todos los nodos del elemento.

Para los casos más sencillos, como el elemento Shell de 4 nodos, las funciones de forma descritas en la literatura corresponde a polinomios. En general las funciones de forma deben cumplir con una serie de condiciones como ser linealmente independientes, coherentes entre sí y en total sumar

la unidad². En este trabajo, las funciones de forma para el elemento Shell se pueden resumir en la (3.20).

Para el elemento Frame, las funciones de forma son las siguientes:

$$\begin{aligned} N_1(\xi) &= \frac{1}{4}(2 - 3\xi + \xi^3); & N_2(\xi) &= \frac{1}{4}a(1 - \xi - \xi^2 + \xi^3); \\ N_3 &= \frac{1}{4}(2 + 3\xi + \xi^3); & N_4(\xi) &= \frac{a}{4}(-1 - \xi + \xi^2 + \xi^3) \end{aligned} \quad (2.26)$$

A partir de esta aproximación se puede obtener la matriz de deformación B como la derivada del desplazamiento en relación a las direcciones de deformación.

$$B = LN \quad (2.27)$$

En donde L es el operador diferencial, que varía según el número de dimensiones y de grados de libertad del problema, sin embargo, basta con decir que se obtiene a partir de las deformaciones admitidas por el modelo de deformación. Para el caso de un cuerpo en 2D que admite deformación planar en los ejes x e y , una matriz L posible se muestra en la (2.28).

$$L = \begin{bmatrix} \partial/\partial x & 0 \\ 0 & \partial/\partial y \\ \partial/\partial y & \partial/\partial x \end{bmatrix} \quad (2.28)$$

Luego, del principio de Hamilton, se puede desprender que la matriz de rigidez del elemento corresponde a³ [2]:

$$k_e = \int_{V_e} B^T c B dV \quad (2.29)$$

En donde c es la matriz del material, que contiene todas las rigideces a los distintos tipos de esfuerzos (flexión, elongación, torsión) en la dirección correspondiente. Por consiguiente, la forma de dicha matriz variará según el número de nodos y grados de libertad del problema.

² La literatura utilizada para esta introducción hace una buena descripción de estas condiciones, además de entregar una alternativa confiable de los polinomios.

³ No es objetivo de este trabajo de título exponer el desarrollo matemático teórico detrás de esta expresión, pero se puede encontrar con bastante detalle en su fuente.

2.6.2 Integración numérica de la matriz de rigidez – Integral de Gauss

Como se explica en la sección anterior, la matriz de rigidez, finalmente se desprende de la integración sobre la matriz de deformación y la matriz c proveniente del material. A su vez, la matriz de deformación se compone de las derivadas de las funciones de forma respecto a las distintas direcciones admitidas por el problema.

Si bien las funciones de forma podrían ser sencillas y el resultado final ser resuelto de forma analítica, suele suceder que ocurra el caso contrario, en que la matriz de rigidez no obedece a funciones sencillas de integrar, o que su resultado analítico no es amigable para trabajar (contiene demasiados términos, y algunos poco significativos).

Por esta razón, en muchas situaciones, la matriz de rigidez de un elemento suele ser expresada como una integración numérica. Esta integración proviene de la (2.29), pero se hace sobre un espacio normalizado (sin dimensiones físicas) que va de -1 a 1 en todas las direcciones, y es llamada integración de Gauss. El espacio normalizado también recibe el nombre de sistema de coordenadas naturales.

El traspaso de un sistema al otro es bastante sencillo, simplemente se sitúa el nuevo sistema de coordenadas en algún punto del elemento (por ejemplo, su centro, o cualquiera de los vértices) y se normalizan las coordenadas por las dimensiones del elemento. En la (2.30) se expresan las coordenadas naturales para un elemento plano, con dimensiones $2a$ y $2b$ en las direcciones x e y , respectivamente.

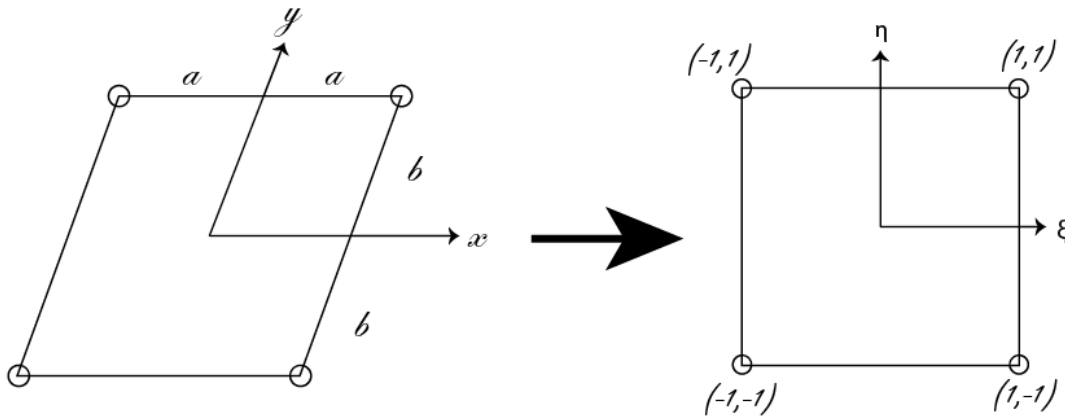


Figura 2.12 Transformación a sistema Gaussiano

$$\xi = \frac{x}{a}; \eta = \frac{y}{b}; \quad (2.30)$$

Básicamente, una integral de Gauss es una sumatoria evaluada en una serie de m puntos gaussianos ξ_i , multiplicados por un coeficiente de peso w_i .

$$I = \int_{-1}^{+1} f(\xi) d\xi = \sum_{i=1}^m w_i f(\xi_i) \quad (2.31)$$

Los coeficientes de peso dependen del número de puntos y se pueden encontrar en la bibliografía [2]. Por lo general mientras se incrementa el número de puntos evaluados m , el resultado de la integral será más certero, sin embargo, incrementará los recursos de procesamiento de datos o el tiempo de cálculo.

Como una observación importante, cabe recalcar que el sistema de coordenadas según el cual se lleva a cabo la integración numérica, por lo general está centrado en el elemento, por lo tanto la matriz resultante obtenida, tendrá esa orientación. Pero, para la etapa de ensamble de la matriz de rigidez, las matrices deben estar orientadas según el mismo eje de referencia, el sistema de coordenadas globales.

2.6.3 Coordenadas globales – Matriz de transformación

Como coordenadas globales se entiende al sistema de coordenadas situado en el origen del espacio que contiene a todos los elementos finitos del modelo. Para que las ecuaciones que rigen el problema mecánico (mencionadas en la Sección 2.5) sean congruentes entre sí, vale decir, para que la conectividad sea la indicada y para que las matrices de rigidez estén correctamente orientadas y ensambladas, es necesario que estén construidas a partir de la misma referencia, es decir, el sistema de coordenadas globales.

Para cambiar las componentes de una matriz de rigidez al sistema de coordenadas globales, se hace uso de una matriz de transformación. Esta matriz de transformación cambiará sus dimensiones y sus componentes según la geometría del elemento. Sin embargo para los elementos Shell y Frame, está compuesta por la sub-matriz de transformación T_3 .

$$T_3 = \begin{bmatrix} l_x & m_x & n_x \\ l_y & m_y & n_y \\ l_z & m_z & n_z \end{bmatrix} \quad (2.32)$$

En donde l_i, m_i, n_i son los cosenos de los ángulos formados entre el sistema de coordenadas natural y el físico. Para encontrar dichos parámetros, existe un procedimiento estándar bastante sencillo, en el que se obtienen las direcciones $\vec{x}, \vec{y}, \vec{z}$ de las coordenadas naturales, según el sistema de coordenadas global $\vec{X}, \vec{Y}, \vec{Z}$.

Primero se definen los vectores V_1, V_2, V_3 que apuntan a tres puntos situados en el sistema de coordenadas natural. Estos tres puntos deben definir las direcciones de este sistema, por lo tanto es conveniente que se encuentren a lo largo de alguno de los tres ejes.

Estos vectores pueden ser expresados según el sistema de coordenadas globales según la (2.33)

$$\vec{V}_j = X_j \vec{X} + Y_j \vec{Y} + Z_j \vec{Z} \quad (2.33)$$

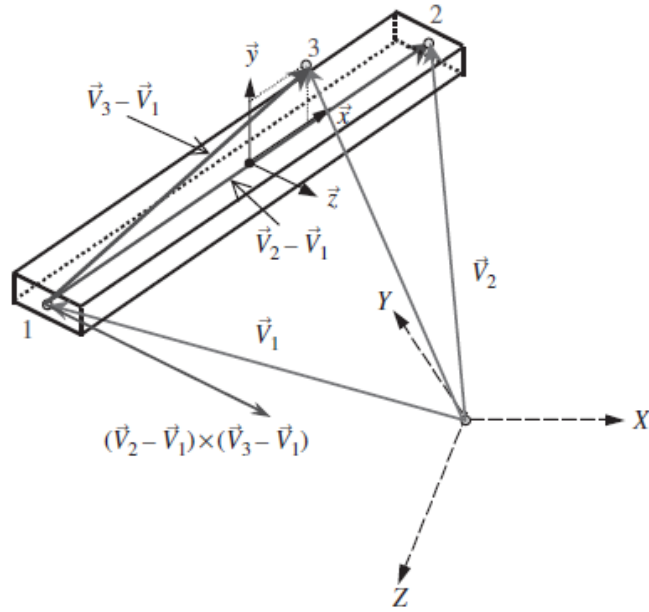


Figura 2.13 Sistema de coordenadas globales y naturales para un elemento Frame [2]

Luego a partir de las diferencias entre dichos vectores y su posterior normalización, se pueden encontrar las direcciones del sistema natural de coordenadas, expresadas según el sistema de coordenadas global.

$$\vec{x} = \frac{\vec{V}_2 - \vec{V}_1}{|\vec{V}_2 - \vec{V}_1|}; \vec{y} = \frac{\vec{V}_3 - \vec{V}_1}{|\vec{V}_3 - \vec{V}_1|}; \vec{z} = \vec{x} \times \vec{y}; \quad (2.34)$$

Por último, debido a que las direcciones son vectores escalares, los cosenos pueden determinarse a partir del producto punto entre ambos sistemas de coordenadas.

$$l_x = \vec{x} \cdot \vec{X}; m_x = \vec{x} \cdot \vec{Y}; n_x = \vec{x} \cdot \vec{Z} \quad (2.35)$$

Luego, para poder expresar las matrices de rigidez en coordenadas globales, basta con aplicar la matriz como sigue.

$$K_e = T^T k_e T \quad (2.36)$$

En donde la matriz T tendrá distinta forma según sea el caso (elemento Shell o Frame), pero compuesta de la misma sub-matriz T_3 , repetida el número de veces que sea necesario.

$$T = \begin{bmatrix} T_3 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & T_3 \end{bmatrix} \quad (2.37)$$

2.6.4 Ensamble de la matriz de rigidez

Como se menciona anteriormente, para obtener el sistema de ecuaciones mecánicas que rige el modelo escogido, se debe construir una matriz global de rigidez. Esta matriz incluye las rigideces de todos los elementos del modelo.

El procedimiento de ensamble por lo general consiste en sumar las rigideces asociadas a cada grado de libertad de cada nodo, de elementos distintos.

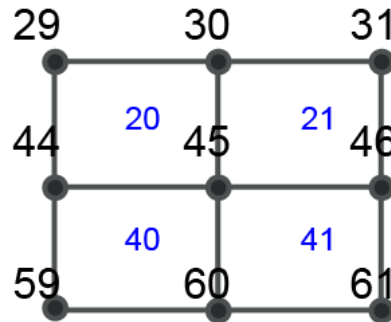


Figura 2.14 Mallado simple de 4 elementos. Números en grande corresponden a los nodos y los pequeños a los elementos.

Los elementos están formados por nodos en común, por ejemplo, en la Figura 2.14 se puede ver como el elemento 20 y 21 tienen en común los nodos 30 y 45. Esto quiere decir que en la matriz de rigidez global, los grados de libertad asociados a los nodos 30 y al 45 son la suma de las rigideces provenientes de los elementos 20 y 21.

Para que esta suma sea posible, las rigideces tienen que estar expresadas con respecto a las coordenadas globales, como se explica en la sección anterior.

Cuando este ejercicio se ha realizado para todos los nodos y elementos, entonces la matriz resultante debería ser simétrica y tener dimensiones de $n \times n$, donde n es igual a la cantidad de nodos multiplicado por la cantidad de grados de libertad de cada nodo en caso de que todos los nodos tengan la misma cantidad de grados de libertad. O en su forma general:

$$n = \sum_{i=1}^{\#nodos} \#Grados\ de\ libertad\ (i) \quad (2.38)$$

2.6.5 Especificación de las condiciones de borde y esfuerzos

Hasta este punto ya se tiene ensamblada la matriz de rigidez, ahora hace falta establecer las fuerzas asociadas a cada grado de libertad de cada nodo y las condiciones de borde para cada desplazamiento de cada nodo.

Al igual que para la aproximación de la deformación de un elemento como un polinomio de funciones, el esfuerzo al cual está sometido un elemento también debe ser integrado a partir de las funciones de forma.

$$f_e = \int_{S_e} N^T f_s dS \quad (2.39)$$

Dónde S_e es la superficie sobre la cual se aplica el esfuerzo f_s , y N^T es la matriz que contiene las funciones de forma.

El resultado de la (2.39) es un vector cuyo tamaño es igual a la cantidad de grados de libertad del elemento (12 para el elemento Frame, y 24 para el elemento Shell). Al igual que las matrices de rigidez, debe estar expresado en las coordenadas locales y debe ser expresado en las coordenadas absolutas, y también se tiene que ensamblar junto con todos los otros vectores de esfuerzos.

La matriz de transformación es exactamente la misma que para el elemento, sólo que se utiliza como dicta la (2.40)

$$F = T^T f_e \quad (2.40)$$

A su vez, el ensamble se trabaja de la misma forma que para las matrices de rigidez, vale decir, se itera a través de cada elemento y se suma a los nodos que lo constituyen, los esfuerzos propios del elemento. Así, cuando hay un nodo en común para dos o más elementos, se suman los esfuerzos que aporta cada elemento en cada dirección.

Por último, resulta intuitivo que el tamaño del vector de esfuerzos ensamblados debe tener el mismo tamaño que el vector de matrices de rigidez ensambladas (en una de sus dimensiones). Vale decir, si la matriz global de rigideces es de 300×300 elementos, el vector de esfuerzos global debería ser de 300×1 elementos.

Para las condiciones de borde, resulta más simple y aclaratorio utilizar un sistema de ecuaciones como ejemplo.

Un sistema de ecuaciones como el descrito en la (2.24), puede ser expresado de forma matricial como:

$$\begin{bmatrix} k_{11} & k_{12} & k_{13} & \dots \\ k_{21} & k_{22} & k_{23} & \dots \\ k_{31} & k_{32} & k_{33} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{pmatrix} u_1 \\ v_1 \\ w_1 \\ \vdots \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \end{pmatrix} \quad (2.41)$$

Si uno de los desplazamientos del nodo está determinado como condición de borde, por ejemplo, si se considera el desplazamiento $w_1 = U_0$, entonces se debe cumplir necesariamente que:

$$[0 \quad 0 \quad 1 \quad 0 \quad \dots] \begin{pmatrix} u_1 \\ v_1 \\ w_1 \\ \theta_{x1} \\ \vdots \end{pmatrix} = U_0 \quad (2.42)$$

Debido a que k_{33} es el único que término que multiplica exclusivamente a w_1 . Si se incorpora esta condición a la (2.41), entonces resulta en:

$$\begin{bmatrix} k_{11} & k_{12} & k_{13} & \dots \\ k_{21} & k_{22} & k_{23} & \dots \\ 0 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{pmatrix} u_1 \\ v_1 \\ w_1 \\ \vdots \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ U_0 \\ \vdots \end{pmatrix} \quad (2.43)$$

Pero debido a que la matriz de rigidez es simétrica, entonces se debe hacer la eliminación de los factores verticales, mediante la suma de dichos factores al lóbulo derecho, con lo que se tendría:

$$\begin{bmatrix} k_{11} & k_{12} & 0 & \dots \\ k_{21} & k_{22} & 0 & \dots \\ 0 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{pmatrix} u_1 \\ v_1 \\ w_1 \\ \vdots \end{pmatrix} = \begin{pmatrix} f_1 - k_{13}U_0 \\ f_2 - k_{23}U_0 \\ U_0 \\ \vdots \end{pmatrix} \quad (2.44)$$

El proceso debe ser repetido para cada grado de libertad de cada nodo que considere alguna condición de borde. Con esto ya se tiene una matriz K y un vector F completos con constantes conocidas, y un vector D con todas las variables incógnitas del problema.

3 PROCEDIMIENTO

Para llevar a cabo el desarrollo del software que encuentra las reacciones en los apoyos de un espesador para la minería, se pretende seguir el siguiente plan de trabajo:

- (a) Análisis de la geometría y los tipos de esfuerzos
- (b) Selección de los elementos
- (c) Generación de los nodos y la malla
- (d) Obtención de la matriz de rigidez y vector de esfuerzos de cada elemento
- (e) Ensamble de la matriz de rigidez global y vector de esfuerzos global
- (f) Constitución de las condiciones de borde
- (g) Resolución del sistema de ecuaciones

3.1 ANÁLISIS DE LA GEOMETRÍA Y LOS TIPOS DE ESFUERZO

Asumiendo que el espesador es una estructura sujeta principalmente a cargas estáticas, lo primero que queda determinado es la constancia de los esfuerzos en el tiempo.

Además, como se observa en el capítulo 2, este tipo de estructura está sometida principalmente a dos tipos de carga:

- El gradiente de presión producido por el fluido contenido
- El peso mismo de la estructura

En términos geométricos, un espesador tipo se puede construir a partir de 3 tipos de figuras, como se ve en la Figura 3.1. Además, los cuerpos volumétricos, vale decir los conos y el cilindro, son de pared delgada en relación al resto de sus dimensiones y sobre todo considerando que los esfuerzos principales corresponden a los de presión, que son normales a la superficie.

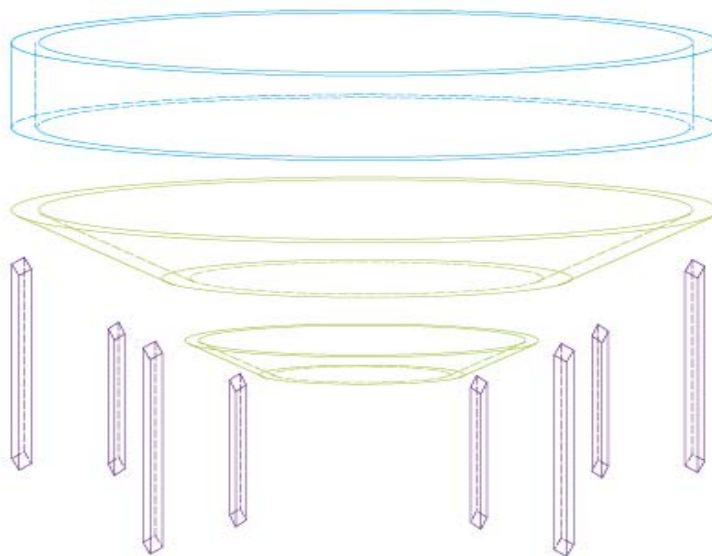


Figura 3.1 Simplificación geometría del espesador

Como el objetivo del software a desarrollar no apunta a un nivel de detalle sino más bien a un resultado aproximado, se han eliminado las vigas que no son verticales para la estructura de soporte.

Además, se observa que mientras que el cono de mayor tamaño es el que transmite las cargas a los pilares, el segundo y más pequeño tan sólo sirve como embudo para guiar los sólidos hacia el resto del ciclo. Esto quiere decir, que como condición de diseño, se considera que los pilares sólo intersectan con el cono de mayor superficie. En algunos casos observados de catálogos, el segundo cono de menor tamaño no existe.

3.2 SELECCIÓN DE LOS ELEMENTOS

A partir de la descripción de la sección anterior se puede suponer que los cilindros y los conos están sujetos principalmente a un gradiente de presión. Como se menciona, su espesor es bastante menor que sus otras dimensiones, y finalmente, son geoméricamente sencillos (no tienen muchas variaciones en el contorno ni curvas complejas). Por este motivo, es que para el manto del espesador, se escogió el elemento Shell de 4 nodos.

Por un lado este elemento es bastante sencillo de programar en relación a otros elementos, y además responde a la naturaleza principal del problema, que son los esfuerzos planos y pared delgada. Por otra parte, se desestimó el uso de elementos tipo Plate, debido a que estos sólo consideran deformaciones de flexión, mientras que el Shell considera además las deformaciones planares, que podrían tener relevancia dada la naturaleza de los esfuerzos de presión y la tridimensionalidad del problema.

En cuanto al sistema de vigas y estructura de soporte, se consideró que la alternativa que respondía mejor a la naturaleza del problema es el elemento Frame, que ofrece deformaciones longitudinales y normales a la barra, además de los tres tipos de rotación.

Otra ventaja de trabajar con estos dos tipos de elementos es que los grados de libertad por nodo son coincidentes, lo que hace más sencillo el ensamble de la matriz de rigidez global.

3.3 GENERACIÓN DE LOS NODOS Y LA MALLA

Esta es la etapa en que comienza la programación en el software Matlab.

Como se explica anteriormente, para modelar, el software debe recibir como entrada las principales medidas del espesador y construir una geometría simplificada como la descrita en el capítulo 3.1. Debido a esta división en conos, cilindro y elementos de línea, resulta más fácil tomar como entradas, las dimensiones constitutivas de estos cuerpos. La forma en que se parametrizó la geometría queda especificada en la Tabla 3.1.

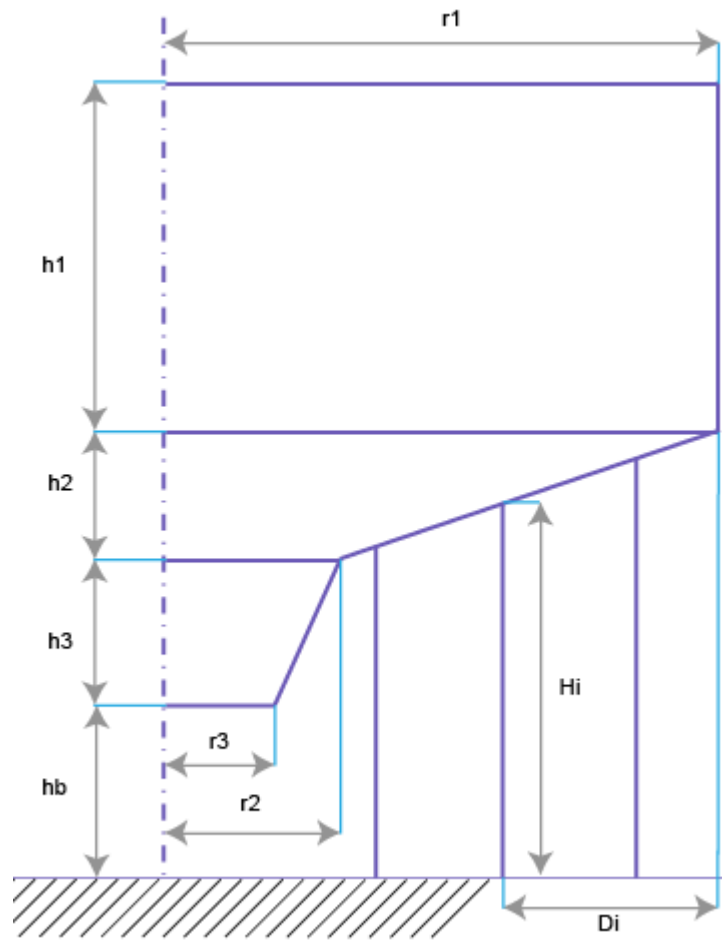


Figura 3.2 Dimensiones constitutivas del espesor modelado

Tabla 3.1 Parametrización cuerpos geométricos

Cuerpo	Dimensiones constitutivas
Cilindro	h_1 : altura, r_1 : radio
Cono 1	h_2 : altura, R_2 : radio mayor, r_2 : radio menor
Cono 2	h_3 : altura, R_3 : radio mayor, r_3 : radio menor
Pilares	N_s : Número de separaciones radiales N_p : Número de pilares por separación H_i : Altura i-ésimo pilar D_i : Distancia entre i-ésimo y el límite externo del cono

Además, se debe notar que para que haya continuidad en el manto y los pilares intersecten justo con el cono de mayores radios, estas dimensiones deben cumplir con una serie de condiciones, tal y como se expresa a continuación.

- Para que haya continuidad entre el cilindro y el cono 1, se requiere que $r_1 = R_2$
- Para que haya continuidad entre el cono 1 y el cono 2, se requiere que $r_2 = R_3$

Para que los pilares intersecten justo con el manto, el procedimiento es un poco más complejo, y se puede explicar de la siguiente forma:

- La altura del pilar se puede dividir entre el espacio que queda debajo del manto del espesador, y sobre la base del manto. Así se tiene que:

$$H_i = h_b + h_{is} \quad (3.1)$$

Donde h_b es la separación entre el manto y el nivel del suelo, y h_{is} la altura sobre la base del manto. Luego, la altura que intercepta con el manto es h_{is} y debe cumplir con que su longitud debe corresponder a la altura del cono a una distancia D_i de su borde exterior.

$$h_{is} = (r_1 - r_2 - D_i) \tan(\arctan(h_2/(r_1 - r_2))) \quad (3.2)$$

3.3.1 Conectividad del modelo

Como se explica en la Sección 2.6.4, para que un elemento se conecte a otro, estos deben tener por lo menos un nodo en común. En el esquema de geometrías que se presenta, existen tres tipos de conectividad distinta. Entre dos elementos Shell a lo largo de todo el manto, entre elementos Frame, a lo largo de cada pilar, y entre elementos Shell y Frame, en las conexiones entre muchos y pilares.

Estas conectividades dan origen a otro tipo de condiciones, que tiene que ver con el número y posicionamiento de nodos:

- El número de nodos en cada circunferencia debe ser un múltiplo del número de separaciones radiales que posea la estructura de vigas del espesador.
- El número de nodos en cada circunferencia debe ser constante en toda la altura del mallado, esto quiere decir que conos y cilindros tienen que tener la misma cantidad de nodos en el perímetro de sus circunferencias.

Estando determinadas estas condiciones del mallado, ya hay información suficiente para comenzar la localización de los nodos en el espacio.

Los nodos constituyen un arreglo en que el índice indica el número del nodo, y el valor guardado son las coordenadas globales que le corresponden en el espacio.

$$N_{nodo} \rightarrow \{x \quad y \quad z\} \quad (3.3)$$

A su vez los elementos son un arreglo en que el índice indica el número del elemento y el valor almacenado corresponde a los números de los nodos.

$$N_{elemento} \rightarrow \{N_{nodo1} \quad \dots \quad N_{nodoi} \quad \dots \quad N_{nodoN}\} \quad (3.4)$$

El orden en que aparecen los nodos dentro de cada elemento debe ser en sentido contra-reloj los elementos Shell, para el caso de los elementos Frame no tiene mayor incidencia.

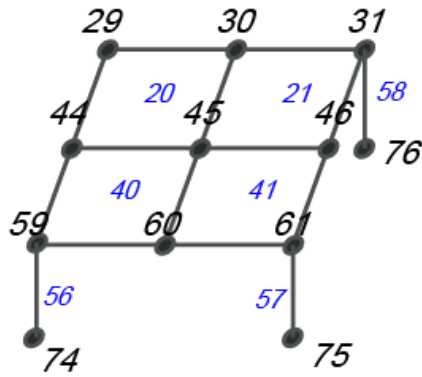


Figura 3.3 Mallado simple de 7 elementos. Números en grande corresponden a los nodos y los pequeños a los elementos.

Como se puede ver en la Figura 3.3, una correcta enumeración de los elementos siguiendo la regla establecida, sería la siguiente:

$$\begin{aligned}
 20 &\rightarrow \{44 \quad 45 \quad 30 \quad 29\} \\
 21 &\rightarrow \{45 \quad 46 \quad 31 \quad 30\} \\
 40 &\rightarrow \{59 \quad 60 \quad 45 \quad 44\} \\
 41 &\rightarrow \{60 \quad 61 \quad 46 \quad 45\} \\
 56 &\rightarrow \{59 \quad 74\} \\
 57 &\rightarrow \{61 \quad 75\} \\
 58 &\rightarrow \{31 \quad 76\}
 \end{aligned}
 \tag{3.5}$$

3.3.2 Mallado del manto

Lo primero para conseguir la malla del modelo es situar a los nodos en el espacio considerando las geometrías y condiciones establecidas anteriormente. Debido a la simetría cilíndrica del problema una buena forma de generar el volumen del manto es por revolución.

Para generar un volumen por revolución, antes se debe construir el perfil del manto, esto es, dicho de otra forma, ordenar una serie de alturas y radios en un plano de acuerdo a las variables introducidas por el usuario.

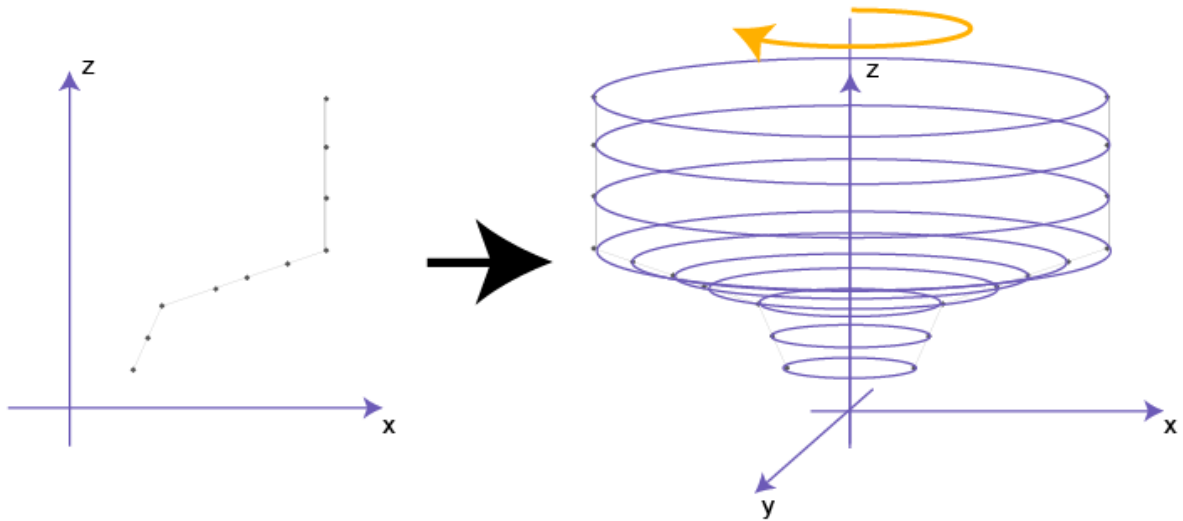


Figura 3.4 Mallado por revolución de un perfil

Antes de determinar las coordenadas para el perfil (Figura 3.4, izquierda), se debe establecer la cantidad de puntos que habrá para una revolución, debido a que este valor pre-define la relación de aspecto para los elementos. Luego dicha relación terminará por definirse con las distancias verticales (según \hat{z}).

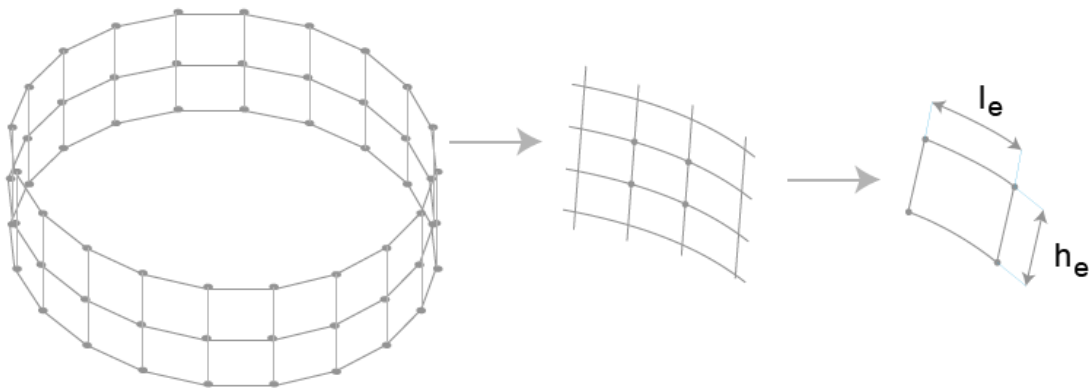


Figura 3.5 Relación de aspecto para un nodo

$$R_{aspecto} = \frac{l_e}{h_e} = (2\pi r_i / n_r) / h_e \quad (3.6)$$

Dónde r_i es la distancia promedio de los nodos del elemento al eje de revolución (\hat{z}), y n_r es el número de puntos que hay para una revolución.

Este último número entero (n_r), tiene que ser divisor del número de separaciones que se especificaron para el espesor, es decir la estructura de vigas que envuelve al manto en dónde intersectan los pilares.

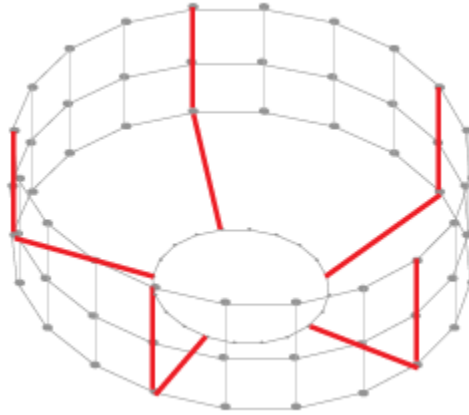


Figura 3.6 Separaciones radiales del espesor

Una vez definido el término n_r , entonces se procede a determinar las coordenadas del perfil.

Para el cilindro esta labor es bastante sencilla, simplemente se hace una distribución de puntos para la misma distancia al eje \hat{z} .

$$(x_i, z_i) = (r_1, h_i) \quad (3.7)$$

Donde h_i es el i -ésimo punto una distribución uniforme entre h_{total} y $h_{total} - h_1$. En donde el número de puntos determina la relación de aspecto para los elementos del cilindro, por lo que se recomienda que el número de puntos siga la (3.8), para que la relación de aspecto sea lo más parecida posible a la unidad.

$$n_{cilindro} = \lfloor h_1 / (2\pi r_1 / n_r) \rfloor + 1 \quad (3.8)$$

Luego, el paso siguiente es definir los puntos para el perfil del primer cono. Como se menciona anteriormente, es en esta línea en que intersectan los pilares, por lo tanto es la parte del perfil que requiere más trabajo para satisfacer los requerimientos de la geometría.

Para comenzar, lo primero que resulta útil calcular es el ángulo de la generatriz.

$$\alpha_1 = \arctan(h_2 / (r_1 - r_2)) \quad (3.9)$$

A partir de este punto, si no existiera la intersección con los pilares, se podría hacer una distribución uniforme de puntos al igual que para el cilindro, pero variando el radio para cada una de las alturas de acuerdo al ángulo calculado en la (3.9). El número de nodos para una distribución de esta naturaleza responde al mismo principio que del cilindro, salvo que en vez del radio constante, se puede calcular con el radio promedio del cono.

$$n_{zcono1} = \lfloor h_2 / (2\pi((r_1 + r_2)/2)/n_r) \rfloor + 1 \quad (3.10)$$

Ahora sólo falta que exista un nodo en todos los lugares donde los pilares intersectan con el manto. Sea D el vector que recoge todas las distancias ingresadas por el usuario, sea el vector R la colección de coordenadas x para los nodos del cono y el vector H la colección de coordenadas z .

Si se define el vector R , para que los puntos estén en la generatriz del cono, el vector H se puede conseguir por trigonometría simple.

$$H = R \cdot \tan(\alpha_1) - r_2 \cdot \tan(\alpha_1) + h_3 + h_b \quad (3.11)$$

La decisión a tomar en este punto es cómo incorporar el vector D al vector R , o dicho de otra forma, construir el vector R a partir del vector D .

Una alternativa posible es utilizar el número de nodos calculados en la (3.10) como referente, ya que de esa forma los elementos tendrán una relación de aspecto cercana a la unidad.

El vector R por lo menos debería contener el radio mayor r_1 , el menor r_2 y al vector D , entonces, sólo hace falta agregar nodos entre las distancias de los pilares hasta alcanzar el referente n_{zcono1} .

$$R = [r_1 \quad \dots \quad D_1 \quad \dots \quad D_i \quad \dots \quad D_{i+1} \quad \dots \quad D_n \quad \dots \quad r_2] \quad (3.12)$$

Para insertar los nodos faltantes entre las distancias de la (3.12), se puede hacer una distribución uniforme en que el número de nodos es un proporcional entre la diferencia de las distancias, el largo total $r_1 - r_2$ y el referente n_{zcono1} .

$$\#Nodos_{i,i+1} = \left\lfloor \frac{D_{i+1} - D_i}{r_1 - r_2} n_{zcono1} \right\rfloor + 1 \quad (3.13)$$

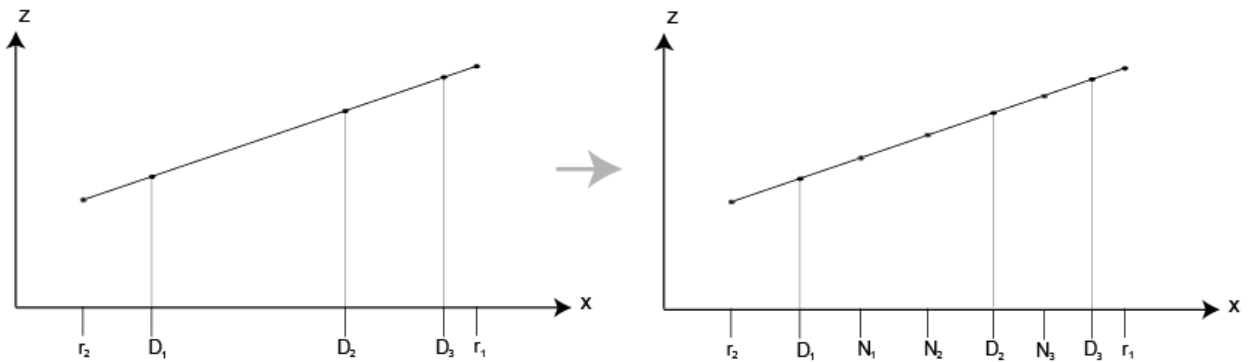


Figura 3.7 Añadimiento de nodos al vector R

Finalmente, para el cálculo de las alturas correspondientes para esos radios, se usa la (3.11).

La función auxiliar que ayuda a esta homogenización de nodos corresponde al Anexo 8.1.7.

Finalmente, las coordenadas del perfil correspondientes al cono quedan definidas como:

$$(x_i, y_i) = (R_i, H_i) \quad (3.14)$$

La única parte faltante del perfil corresponde al cono menor. Para localizar sus nodos en el espacio, primero se procede a calcular el número de nodos para que la relación de aspecto de los elementos sea cercana a la unidad, al igual que los casos anteriores.

$$n_{zcono2} = \lfloor h_3 / (2\pi((r_2 + r_3)/2)/n_r) \rfloor + 1 \quad (3.15)$$

Luego, el paso siguiente es generar una distribución uniforme R de radios entre r_2 y r_3 , y a partir de esa distribución calcular las alturas.

$$\alpha_2 = \arctan(h_3/(r_2 - r_3)) \quad (3.16)$$

$$H = R \cdot \tan(\alpha_2) - r_3 \cdot \tan(\alpha_2) + h_b \quad (3.17)$$

Las coordenadas quedan definidas exactamente igual que para el cono mayor.

Luego de que están definidos los nodos del perfil, entonces hay que generar la revolución mostrada en la Figura 3.4. Para lo cual se itera entre 0 y 2π , con un número de pasos igual a n_r . La metodología que posiciona los nodos en cada iteración y constituye todos los nodos del manto corresponde al Anexo 8.1.6.

Hasta este punto ya se hallan posicionados los nodos, el último paso para determinar la malla es recorrerlos hasta la penúltima fila, para generar los elementos.

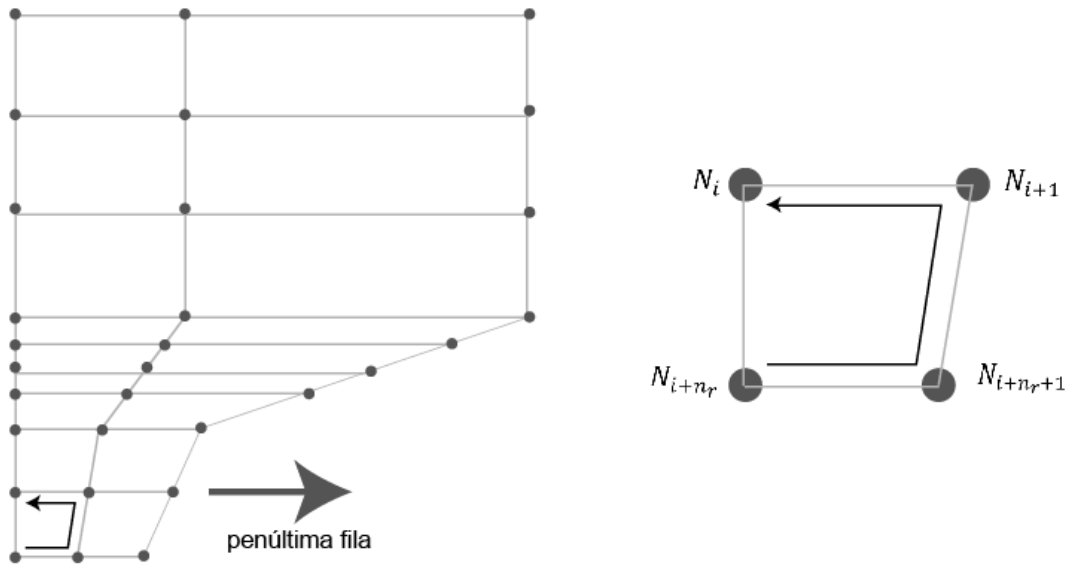


Figura 3.8 Aclaración penúltima fila

Como se puede ver en la Figura 3.8, si el arreglo de nodos se recorre desde arriba hasta abajo, el último elemento queda determinado a partir del nodo de la penúltima fila.

$$\text{Último elemento de la columna} \rightarrow \{N_{i+n_r}, N_{i+n_r+1}, N_{i+1}, N_i\} \quad (3.18)$$

3.3.3 Mallado de los pilares y vigas

Hasta este punto ya se encuentra listo el mallado del manto del espesador, lo que hace falta ahora es constituir los elementos de las vigas que refuerzan el manto y los pilares, debido a que corresponden al mismo tipo de elemento, Frame.

Sea A el vector que contiene todos los ángulos donde corresponde que vaya una viga, dichos ángulos pueden ser calculados fácilmente a partir del número de separaciones determinado por el usuario. Además, sea $N_{común}$ el vector que contiene todos los nodos en que se conecta un pilar con el manto.

Este último vector puede ser construido dentro de la generación de los nodos del manto de la sección anterior, en que simplemente se pregunta si el nodo que se está posicionando proviene del vector D y el ángulo en el que se está iterando, está contenido en el vector A .

Los nodos que conforman los elementos que refuerzan el manto son los mismos que los del manto, por lo tanto la construcción de estos elementos se hace al mismo tiempo que la generación del manto, como se indica en la Figura 3.9.

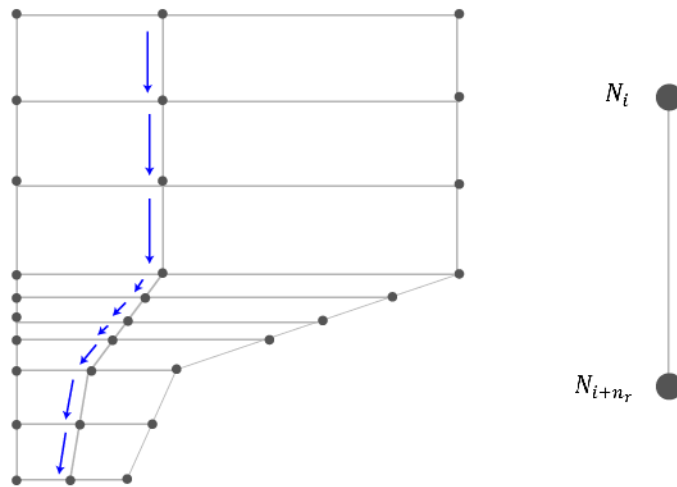


Figura 3.9 Generación de vigas de refuerzo

En términos de algoritmo, sólo es necesario preguntar dentro de la iteración que genera los elementos del manto, si el ángulo en el que se está iterando está contenido en A .

A partir de este punto, lo siguiente es recorrer el vector $N_{común}$ y generar los nodos de los pilares como una distribución uniforme entre la altura del nodo iterado y el suelo. La cantidad de puntos de la distribución se puede calcular según la (3.19) para que el promedio de tamaño de elementos sea igual para los pilares y del cono donde se conectan. Es conveniente generar los elementos como un recorrido de esta distribución uniforme, para ahorrar recursos.

$$n_{zpillar-i} = z_i / (2\pi(\frac{r_1 + r_2}{2}) / nr) \quad (3.19)$$

Donde z_i es la altura del i -ésimo nodo de $N_{común}$. Es importante recordar que estos nuevos nodos son adicionados al final de los nodos generados para el manto. Y lo mismo ocurre para los elementos.

3.4 OBTENCIÓN DE LA MATRIZ DE RIGIDEZ PARA CADA ELEMENTO

El objetivo de esta etapa es desarrollar un algoritmo que a partir de las coordenadas de cada elemento, el tipo de material y otras variables geométricas (como el espesor o el perfil) entregue la matriz de rigidez.

Para el caso del elemento Frame, la literatura entrega una versión analítica de dicha matriz, por lo que no merece mayor explicación, salvo mencionar que el código correspondiente a este procedimiento es el Anexo 8.1.4 y la matriz corresponde a la (2.10). Con la anotación de que dicha matriz está expresada en función de las coordenadas locales del elemento, y debe transformarse a las coordenadas globales a través del procedimiento expresado en la Sección 2.6.3.

3.4.1 Integración numérica para el elemento Shell

El código generado en Matlab que incluye los procedimientos detallados en esta sección corresponde al Anexo 8.1.3.

Lo primero que hay que considerar, es que el elemento Shell se construye a partir de 4 nodos, con 6 grados de libertad cada uno. Esto quiere decir, que la matriz de rigidez esperada es de 24×24 términos.

A su vez, de los 6 grados de libertad que admite cada nodo, hay dos que están asociados a las deformaciones planares (u, v), otros tres que están asociados a las deformaciones de flexión fuera del plano (w, θ_u, θ_v) y el último, que no aplica a este problema, debido a que representa una torsión planar según el eje- z , que por lo general se obvia para casos como este, salvo algunas excepciones.

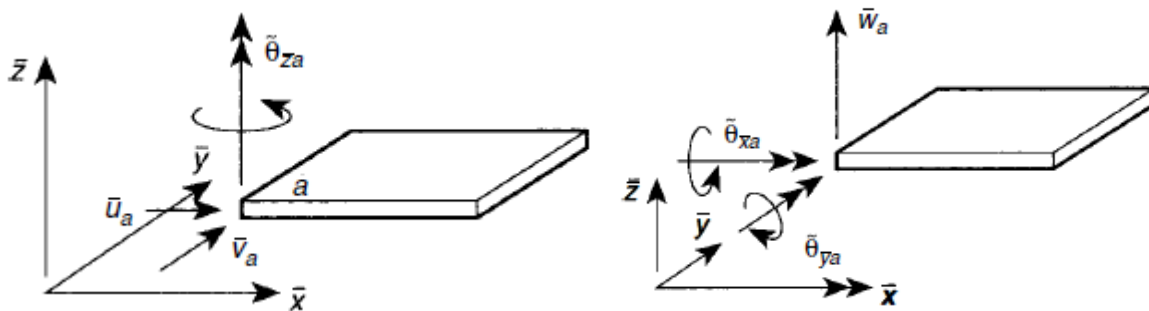


Figura 3.10 (izquierda) Grados de libertad asociados a la deformación planar (derecha) Grados de libertad asociados a la deformación en flexión.

Dicho esto, en distintas fuentes bibliográficas [2] [3] se menciona que es recomendable obtener una matriz de rigidez asociada a la deformación por flexión, y otra para la deformación planar del elemento.

3.4.1.1 Matriz de rigidez para la flexión de un elemento Shell

Como se explica en el capítulo anterior, se consideran sólo tres grados de libertad asociados a la flexión, por lo tanto, la matriz de rigidez esperada será de 12×12 términos (4 nodos \times 3 grados de libertad).

Resulta necesario recordar del capítulo 2.7.2, que la integral de la (2.29) se hace en las coordenadas naturales, en que las coordenadas ξ, η se obtienen a partir del cambio de variable sugerido en la (2.30).

Luego, el procedimiento a ejecutar es el descrito en la Sección 2.7.1, y el primer paso es la obtención de la matriz de deformación B .

Para obtener la matriz B , se deben calcular las derivadas de las funciones de forma, una por cada nodo. Para este caso, las funciones de forma sugeridas en las fuentes bibliográficas [2] y [4] se pueden generalizar en la (3.20), expresadas en función de las coordenadas naturales.

$$N_i(\xi, \eta) = \frac{1}{4}(1 + \xi_i \xi)(1 + \eta_i \eta) \quad (3.20)$$

En donde ξ_i y η_i son las coordenadas del i -ésimo nodo en el sistema de coordenadas naturales.

Como las deformaciones de los nodos son expresadas como una interpolación de las funciones de forma (sumatoria (2.25)), entonces las derivadas de la deformación también pueden ser expresadas como las derivadas de las funciones de forma.

Sin embargo, como se está operando en las coordenadas naturales, entonces se ocupa el siguiente procedimiento algebraico para encontrar las derivadas de las funciones de forma según las coordenadas globales en términos de las naturales.

La (3.21) es la aplicación de la regla de la cadena a las derivadas de las funciones de forma.

$$\begin{aligned} \frac{\partial N}{\partial \xi} &= \frac{\partial N}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial N}{\partial y} \frac{\partial y}{\partial \xi} \\ \frac{\partial N}{\partial \eta} &= \frac{\partial N}{\partial x} \frac{\partial x}{\partial \eta} + \frac{\partial N}{\partial y} \frac{\partial y}{\partial \eta} \end{aligned} \quad (3.21)$$

Escrito de forma matricial, se puede expresar como:

$$\begin{pmatrix} \partial N_i / \partial \xi \\ \partial N_i / \partial \eta \end{pmatrix} = \begin{bmatrix} \partial x / \partial \xi & \partial y / \partial \xi \\ \partial x / \partial \eta & \partial y / \partial \eta \end{bmatrix} \begin{pmatrix} \partial N_i / \partial x \\ \partial N_i / \partial y \end{pmatrix} \quad (3.22)$$

$$\begin{pmatrix} \partial N_i / \partial \xi \\ \partial N_i / \partial \eta \end{pmatrix} = J \begin{pmatrix} \partial N_i / \partial x \\ \partial N_i / \partial y \end{pmatrix} \rightarrow \begin{pmatrix} \partial N_i / \partial x \\ \partial N_i / \partial y \end{pmatrix} = J^{-1} \begin{pmatrix} \partial N_i / \partial \xi \\ \partial N_i / \partial \eta \end{pmatrix} \quad (3.23)$$

En donde la matriz cuadrada del lóbulo derecho corresponde a la matriz Jacobiana de la (2.29). Además en la (3.22) se agregó un índice, para indicar el i -ésimo término de la matriz N , como la matriz que multiplica a los desplazamientos de la i -ésimo nodo.

Finalmente, como se expone en la (3.23), las derivadas de las funciones de forma según (x, y) pueden ser expresadas como las derivadas de las funciones de forma según las coordenadas naturales y el inverso de la matriz Jacobiana.

Para obtener los términos de la matriz Jacobiana, resulta útil recordar que las coordenadas de cualquier elemento, son expresadas como una interpolación de las funciones de forma,

$$x = \sum_{i=1}^4 N_i x_i; y = \sum_{i=1}^4 N_i y_i \quad (3.24)$$

Esto quiere decir que las componentes de la matriz Jacobiana se pueden definir como sigue:

$$\frac{\partial x}{\partial \xi} = \sum_{i=1}^4 \frac{\partial N_i}{\partial \xi} x_i; \frac{\partial N_i}{\partial \xi} = \frac{1}{4}(\xi_i)(1 + \eta_i \eta) \rightarrow \frac{\partial x}{\partial \xi} = \sum_{i=1}^4 \frac{1}{4}(\xi_i)(1 + \eta_i \eta)x_i \quad (3.25)$$

$$\frac{\partial x}{\partial \eta} = \sum_{i=1}^4 \frac{\partial N_i}{\partial \eta} x_i; \frac{\partial N_i}{\partial \eta} = \frac{1}{4}(\eta_i)(1 + \xi_i \xi) \rightarrow \frac{\partial x}{\partial \eta} = \sum_{i=1}^4 \frac{1}{4}(\eta_i)(1 + \xi_i \xi)x_i \quad (3.26)$$

$$\frac{\partial y}{\partial \xi} = \sum_{i=1}^4 \frac{\partial N_i}{\partial \xi} y_i; \frac{\partial N_i}{\partial \xi} = \frac{1}{4}(\xi_i)(1 + \eta_i \eta) \rightarrow \frac{\partial y}{\partial \xi} = \sum_{i=1}^4 \frac{1}{4}(\xi_i)(1 + \eta_i \eta)y_i \quad (3.27)$$

$$\frac{\partial y}{\partial \eta} = \sum_{i=1}^4 \frac{\partial N_i}{\partial \eta} y_i; \frac{\partial N_i}{\partial \eta} = \frac{1}{4}(\eta_i)(1 + \xi_i \xi) \rightarrow \frac{\partial y}{\partial \eta} = \sum_{i=1}^4 \frac{1}{4}(\eta_i)(1 + \xi_i \xi)y_i \quad (3.28)$$

Y finalmente la matriz Jacobiana escrita de forma matricial y compacta se puede expresar como:

$$J = \sum_{i=1}^4 \frac{1}{4} \begin{bmatrix} (\xi_i)(1 + \eta_i \eta)x_i & (\xi_i)(1 + \eta_i \eta)y_i \\ (\eta_i)(1 + \xi_i \xi)x_i & (\eta_i)(1 + \xi_i \xi)y_i \end{bmatrix} \quad (3.29)$$

Como se está calculando la matriz de rigidez para la deformación asociada a la flexión, es necesario considerar solamente los tres grados de libertad mencionados w , θ_x y θ_y . Al igual que en la Sección 2.3.4, las matrices de deformación también pueden ser tratadas de forma separada, en la matriz de deformación debido a esfuerzos de corte y debido a esfuerzos normales.

Resulta útil recordar que la matriz Jacobiana queda determinada como una función de un punto (ξ, η) , por lo que debe ser evaluada para cada nodo de un elemento.

$$B_j^I(\xi, \eta) = \begin{bmatrix} 0 & 0 & -\frac{\partial N_j}{\partial x} \\ 0 & \frac{\partial N_j}{\partial y} & 0 \\ 0 & \frac{\partial N_j}{\partial x} & -\frac{\partial N_j}{\partial y} \end{bmatrix}; B^I = [B_1^I \quad B_2^I \quad B_3^I \quad B_4^I] \quad (3.30)$$

$$B_j^O(\xi, \eta) = \begin{bmatrix} \frac{\partial N_j}{\partial x} & 0 & N_j \\ \frac{\partial N_j}{\partial y} & -N_j & 0 \end{bmatrix}; B^O = [B_1^O \quad B_2^O \quad B_3^O \quad B_4^O] \quad (3.31)$$

La matriz B^I está asociada a los esfuerzos normales y la matriz B^O a los esfuerzos de corte.

Luego, la matriz de rigidez se puede constituir como la suma de ambas integrales como se puede observar en la (3.32). Luego en la (3.33) se encuentran las integrales expresadas en las coordenadas naturales (con el correspondiente determinante del jacobiano, explicado en la Sección 2.6.2) para luego llevar a cabo la integración de Gauss.

$$k_e^b = \int_{A_e} \frac{h^3}{12} [B^I]^T c B^I dA + \int_{A_e} kh [B^O]^T c_s B^O dA \quad (3.32)^4$$

$$k_e^b = \int_{-1}^1 \int_{-1}^1 \frac{h^3}{12} [B^I]^T c B^I \det(J) d\xi d\eta + \int_{-1}^1 \int_{-1}^1 kh [B^O]^T c_s B^O \det(J) d\xi d\eta \quad (3.33)$$

$$= \sum_{i=1}^2 \sum_{j=1}^2 w_i w_j \frac{h^3}{12} [B^I(\xi_i, \eta_j)]^T c B^I(\xi_i, \eta_j) \det(J) + \sum_{i=1}^1 \sum_{j=1}^1 w_i w_j kh [B^O(\xi_i, \eta_j)]^T c_s B^O(\xi_i, \eta_j) \det(J) \quad (3.34)$$

En la (3.32) la matriz c corresponde a la resistencia del material en las direcciones normales, y la matriz c_s corresponde a la resistencia del material a la torsión. La constante k corresponde a un término de corrección, su valor es $\pi^2/12$, y h es el espesor del elemento.

En la literatura de la fuente, se recomienda que la integral de Gauss para obtener la matriz asociada a los esfuerzos de corte se haga sobre un solo punto en cada dirección, debido que para dos puntos podría ocurrir un fenómeno llamado *shear locking*, en que el adelgazamiento de la pared (h), podría incurrir en un incremento en la rigidez, algo contradictorio. Mientras que para el primer término, asociado a la rigidez normal, se establece que un buen número de puntos es dos en cada dirección (x, y) y que los resultados son bastante cercanos al modelo teórico del cual proviene.

Tabla 3.2 Puntos evaluados para la integral de Gauss y factores de peso

2 puntos	ξ_i, η_i	$[-1/\sqrt{3}, 1/\sqrt{3}]$	w_i	$[1, 1]$
1 punto	ξ, η	0	w_i	2

3.4.1.2 Matriz de rigidez para la deformación planar de un elemento Shell

Las funciones de forma son propias del elemento, por lo tanto para las siguientes explicaciones, se utilizan las mismas que para obtener la matriz de rigidez a la flexión. Sin embargo las funciones de forma provenientes son las asociadas al modelo especificado en la Sección 2.3.3, por lo tanto se utiliza la matriz L ahí establecida.

Esto quiere decir que la función de forma asociada a la deformación planar tendrá la siguiente forma:

⁴ Ecuación sustraída de [2](capítulo 8.2)

$$B_j(\xi, \eta) = \begin{bmatrix} \partial N_j / \partial x & 0 \\ 0 & \partial N_j / \partial y \\ \partial N_j / \partial y & \partial N_j / \partial x \end{bmatrix}; B = [B_1 \quad B_2 \quad B_3 \quad B_4] \quad (3.35)$$

La rigidez para este caso tendrá la forma:

$$k_e^m = \int_{A_e} h[B]^T c B dA \quad (3.36)$$

$$= \sum_{i=1}^2 \sum_{j=1}^2 w_i w_j h B(\xi_i, \eta_i)^T c B(\xi_i, \eta_i) \det(J) \quad (3.37)$$

En donde los ponderadores de peso son los mencionados en la Tabla 3.2.

3.4.1.3 Matriz de rigidez completa para el elemento Shell

Dado que la matriz de rigidez contempla seis grados de libertad por nodo, y en las matrices de rigidez para la flexión y la deformación planar en total contemplan sólo cinco, la rigidez asociada al sexto grado de libertad será nula y no interactuará con los otros grados. Sin embargo, para que la matriz sea invertible, la diagonal no puede contener valores nulos, por lo tanto se le asigna un valor arbitrario (que no interactúa con los resultados), que para este caso es 1.

$$k_e = \begin{bmatrix} k_{11}^m & 0 & 0 & k_{12}^m & 0 & 0 & k_{31}^m & 0 & 0 & k_{41}^m & 0 & 0 \\ 0 & k_{11}^b & 0 & 0 & k_{12}^b & 0 & 0 & k_{31}^b & 0 & 0 & k_{41}^b & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ k_{21}^m & 0 & 0 & k_{22}^m & 0 & 0 & k_{32}^m & 0 & 0 & k_{42}^m & 0 & 0 \\ 0 & k_{21}^b & 0 & 0 & k_{22}^b & 0 & 0 & k_{32}^b & 0 & 0 & k_{42}^b & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ k_{31}^m & 0 & 0 & k_{23}^m & 0 & 0 & k_{33}^m & 0 & 0 & k_{43}^m & 0 & 0 \\ 0 & k_{31}^b & 0 & 0 & k_{11}^b & 0 & 0 & k_{33}^b & 0 & 0 & k_{43}^b & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ k_{41}^m & 0 & 0 & k_{24}^m & 0 & 0 & k_{34}^m & 0 & 0 & k_{44}^m & 0 & 0 \\ 0 & k_{41}^b & 0 & 0 & k_{24}^b & 0 & 0 & k_{34}^b & 0 & 0 & k_{44}^b & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.38)$$

Dónde k_{ij}^m corresponde a la sub-matriz de 2×2 elementos proveniente de la matriz de rigidez para la deformación planar, y k_{ij}^b corresponde a la sub-matriz de 3×3 elementos proveniente de la matriz de rigidez para la flexión. Los índices identifican la interacción entre los nodos, es decir, por ejemplo, el valor de k_{32}^m corresponde a la rigidez del nodo 3 respecto al nodo 2.

Cabe recordar que k_e está expresada según el sistema de coordenadas local. Por lo que para llevarla a las coordenadas globales, se debe obtener la matriz de transformación del elemento. El procedimiento es exactamente el explicado en la Sección 2.6.3 y el código que obtiene la matriz T_3 corresponde al Anexo 8.1.5.

Como resultado de esta etapa se obtiene la matriz de rigidez de un elemento Shell expresado en función de las coordenadas globales.

3.4.1.4 Generación del vector de esfuerzos para el elemento Shell

Debido a que el trabajo de obtener los esfuerzos para cada nodo depende de las funciones de forma del elemento, como se indica en la (2.39), resulta más rápido y sencillo si se genera dicho vector al mismo tiempo que la matriz de rigidez.

Al igual que para la matriz de rigidez, para el vector de esfuerzos resulta conveniente utilizar una integral de Gauss, con dos puntos en cada dirección. Esto quiere decir que la (2.39) queda expresada de la siguiente forma:

$$f_e = \sum_{i=1}^2 \sum_{j=1}^2 w_i w_j N^t f_s \det(J); f_s = [f_x \quad f_y \quad f_z \quad M_x \quad M_y \quad M_z] \quad (3.39)$$

Dónde f_s corresponde al vector de esfuerzos y momentos aplicados sobre el elemento. El resultado f_e será un vector que contiene las fuerzas sobre cada grado de libertad de cada nodo del elemento, vale decir, será un vector de 24×1 valores por cada elemento.

Para este problema en particular, sólo se contemplan las tres componentes de la fuerza, que corresponden al peso propio de cada elemento para todos los elementos, y el gradiente de presión para los elementos del manto.

En coordenadas locales, la presión que soporta cara elemento Shell está expresado en la (3.40), en donde Z_i es la columna de fluido que está sobre el elemento, H_{total} la altura total del espesor y \bar{z}_i es el promedio de las alturas (coordenada z) de los nodos que componen el elemento.

$$f_{presión} = [0 \quad 0 \quad \rho_{fluido} g Z_i \quad 0 \quad 0 \quad 0] \quad (3.40)$$

$$Z_i = H_{total} - \bar{z}_i \quad (3.41)$$

Para el peso por unidad de área del elemento se multiplica por la densidad del material por el espesor del elemento. Sin embargo, este valor tiene dirección $-\hat{z}$ en las coordenadas globales por lo que primero hay que expresarla en las coordenadas locales. Este ejercicio es muy sencillo y consiste en multiplicar el vector en coordenadas globales por el inverso de la matriz de transformación T .

$$F_{area}^g = [0 \quad 0 \quad \rho_{elemento} h_{elemento} g \quad 0 \quad 0 \quad 0] \quad (3.42)$$

$$f_{peso} = T^{-1} F_{area}^g \quad (3.43)$$

Por último, el vector local de esfuerzos será:

$$f_s = f_{presión} + f_{peso} \quad (3.44)$$

Luego se calcula f_e en base a la (3.39) y se multiplica nuevamente por la matriz de transformación del elemento para expresar la fuerza aplicada sobre cada nodo en función de las coordenadas globales.

3.5 GENERACIÓN DEL VECTOR DE ESFUERZOS PARA EL ELEMENTO FRAME

Para el vector de esfuerzos del elemento Frame en coordenadas globales, sólo se considera la mitad del peso del elemento para cada nodo y se expresa como sigue:

$$F_{frame}^g = \left[0 \quad 0 \quad \frac{\rho_{viga} A_{viga} l_{elem} g}{2} \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \frac{\rho_{viga} A_{viga} l_{elem} g}{2} \quad 0 \quad 0 \quad 0 \right] \quad (3.45)$$

Dónde A_{elem} y ρ_{viga} son la sección y la densidad de la viga respectivamente, l_{elem} es el largo del elemento y g es la constante de aceleración gravitacional.

3.6 ENSAMBLE DE LAS MATRICES DE RIGIDEZ Y VECTORES DE ESFUERZOS

La sección anterior describe cómo obtener la matriz de rigidez para cada elemento. Lo siguiente es ensamblar esas matrices en la matriz global.

El procedimiento para conseguir esto es bastante estándar y resulta más sencillo de entender si se explica mediante un ejemplo.

Sea el vector E la colección de todos los elementos del sistema y N la colección de los nodos.

Sea el elemento E_p definido como sigue⁵:

$$E_p = [74 \quad 75 \quad 63 \quad 62] \quad (3.46)$$

Y sea K_p la matriz de rigidez local del elemento.

Los nodos que componen el elemento indican su interacción, para este caso, la matriz del elemento indican que los nodos 74,75,63 y 62 interactúan entre sí de acuerdo a la matriz de rigidez del elemento.

Los grados de libertad del elemento de forma global, serán los siguientes:

$$[u_{74} \quad v_{74} \quad w_{74} \quad \theta_{x74} \quad \theta_{y74} \quad \theta_{z74} \quad \dots \quad u_{62} \quad v_{62} \quad w_{62} \quad \theta_{x62} \quad \theta_{y62} \quad \theta_{z62}] \quad (3.47)$$

Por ejemplo, el valor $K_p(2; 21)$, indica la rigidez del grado de libertad v_{74} respecto al grado de libertad w_{62} . Esta es la información que aparecerá en la matriz global.

Siguiendo con el ejemplo, la posición del valor $K_p(3,21)$ en la matriz global será la expresada en la Figura 3.11.

⁵ El número de nodos del elemento no tiene relevancia en el proceso de ensamble, para los elementos Frame el proceso es exactamente el mismo.

$$\begin{array}{c|ccc}
 & u_1 & \dots & v_{74} \\
 \hline
 u_1 & K_{11} & & \vdots \\
 \vdots & & \ddots & \\
 w_{62} & \dots & & K_p(3,21)
 \end{array}$$

Figura 3.11 Posicionamiento del valor $K_p(3,21)$ en la matriz de rigidez global

El mismo proceso debe ser realizado para el vector de esfuerzos, sin embargo esto es bastante más sencillo puesto que para la matriz de rigidez se tenía la interacción entre distintos nodos, lo que daba origen a un sistema de dos dimensiones. Para el vector de esfuerzos el problema es en una dimensión.

Sea F_p el vector de esfuerzos correspondiente al elemento E_p definido como sigue:

$$F_p = [f_{u1} \quad f_{v1} \quad f_{w1} \quad f_{\theta_{x1}} \quad f_{\theta_{y1}} \quad f_{\theta_{z1}} \quad \dots \quad f_{\theta_{z4}}] \quad (3.48)$$

En que el término $F_p(4)$ corresponde a la posición del grado de libertad θ_{x74} en el vector de esfuerzos global.

Este ejercicio debe ser resuelto para todos los nodos de todos elementos, por lo tanto se itera a través de E y de cada nodo perteneciente a cada E_i , asignando los valores de la matriz de rigidez local y del vector de esfuerzos local del elemento al lugar correspondiente de la matriz y vector global respectivamente.

La función que cumple este rol para este trabajo de título recibe como parámetro el vector E_i y entrega las posiciones de cada elemento local en la matriz global y corresponde al Anexo 8.1.2.

3.7 CONSTITUCIÓN DE LAS CONDICIONES DE BORDE

Hasta este punto ya se tiene un sistema de ecuaciones completo, sin embargo, el número de soluciones es infinito. Para que el sistema tenga una única solución, se deben ingresar las condiciones de borde al sistema.

Las condiciones de borde pueden ser de múltiples formas, pero para este caso consiste en condicionamiento de la respuesta, es decir con valores conocidos para función objetivo. Este tipo de condición de borde se conoce como *de primer tipo* o de Dirichlet.

La función objetivo de este problema es el desplazamiento de los nodos luego de la aplicación de ciertos esfuerzos, por lo que adicionar las condiciones de borde al problema, se trata de asignarle valores a algunos de los desplazamientos de los nodos. El procedimiento para llevar a cabo esta acción se explica detalladamente en la Sección 2.6.5.

Para este fenómeno en particular, del espesador, se utiliza la condición de anclaje en la base de los pilares. Esto quiere decir que el desplazamiento para esos nodos es nulo.

Sea $N_{anclados}$ la colección de los nodos del extremo libre de los pilares y sea U el conjunto de desplazamientos para todos los nodos del problema (la función objetivo).

$$U_i = (u_i \quad v_i \quad w_i \quad \theta_{xi} \quad \theta_{yi} \quad \theta_{zi}) = 0 ; \forall i \in N_{anclados} \quad (3.49)$$

Luego se itera dentro de $N_{anclados}$ y se sigue el procedimiento de la Sección 2.6.5.

Los esfuerzos aplicados también pueden ser considerados como una condición de borde del mismo tipo, en que la función objetivo corresponde a los esfuerzos a los que está sometido el material.

El código que hace este trabajo es muy breve para el problema en cuestión, sólo se requiere conocer los grados de libertad de la matriz global de rigidez y del vector global de esfuerzos que correspondiente a los nodos anclados.

3.8 RESOLUCIÓN DEL SISTEMA DE ECUACIONES

Para encontrar la solución que satisface las condiciones de borde primero se deben ensamblar las matrices de rigidez y el vector de esfuerzos, y luego aplicar las condiciones de borde.

Sea $K^{g'}$ la matriz de rigidez global que considera las condiciones de borde y sea $F^{g'}$ el vector global de esfuerzos que considera los valores de las condiciones de borde.

Luego obtener los desplazamientos U consiste básicamente en invertir la matriz $K^{g'}$

$$U = F^{g'} \cdot (K^{g'})^{-1} \quad (3.50)$$

Si esas deformaciones se aplican al sistema de ecuaciones original (sin condiciones de borde), entonces se estará calculando los esfuerzos internos y reacciones del modelo.

$$F = K^g U \quad (3.51)$$

Desde la aplicación de las condiciones de borde hasta este punto, los códigos que ejecutan dichos procedimientos corresponden al Anexo 8.1.1.

3.9 PRE-VALIDACIÓN DE LAS MATRICES DE RIGIDEZ

Estimar si los resultados obtenidos por los programas desarrollados para la geometría y las condiciones de borde propuestas es algo complejo, pues conlleva conocer datos estadísticos inusuales y muy precisos, lo que se aleja de los objetivos de este trabajo de título.

Sin embargo, se puede establecer la calidad de algunos procedimientos y algoritmos mediante una pre-validación. Este proceso consiste en utilizar las mismas funciones para simular un fenómeno con una respuesta teórica ya conocida y medir el error de la respuesta simulada respecto a la esperada.

Posiblemente, el caso más básico, común y estudiado sea el de una viga empotrada en voladizo, y ese es el fenómeno que se utiliza en este trabajo.

El objetivo de la pre-validación, para este caso, es comprobar si las matrices de rigidez y los vectores de esfuerzo están bien generados, si las metodologías de ensamblaje y obtención de la matriz de transformación son correctas.

Para verificar si los modelos son correctos, además es necesario que la fineza del mallado sea una variable, pues esto permite verificar que la respuesta tenga una tendencia al valor teórico esperado.

Todas estas verificaciones deberían quedar satisfechas simulando los siguientes escenarios:

- Viga en voladizo con emplazamiento según \hat{x} y fuerza puntual en su extremo libre. Simulado mediante elementos Shell
- Viga en voladizo con emplazamiento según \hat{x} y fuerza puntual en su extremo libre. Simulado mediante elementos Frame
- Viga en voladizo con emplazamiento según \hat{x} y esfuerzo uniforme en toda su extensión. Simulado mediante elementos Shell
- Viga en voladizo con emplazamiento según \hat{z} y fuerza puntual en su extremo libre. Simulado mediante elementos Shell

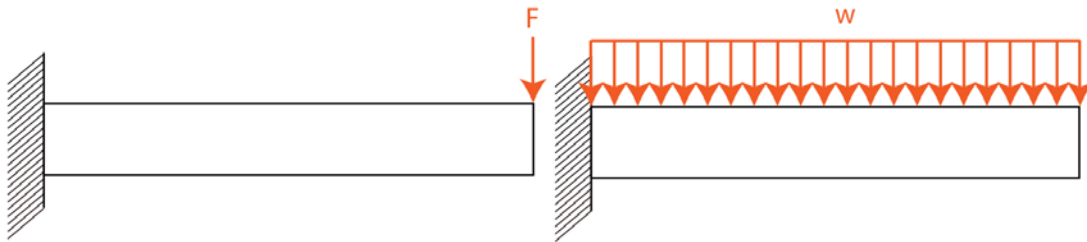


Figura 3.12 Viga empotrada, con carga puntual y carga distribuida

En la Figura 3.12 se muestra la representación gráfica en dos dimensiones del problema a analizar. Para validar los procesos mencionados, se busca comparar la deformación estimada por el sistema de elementos finitos con la respuesta teórica, además de las reacciones en el empotramiento.

La reacción corresponderá a la suma de los esfuerzos o fuerzas aplicadas sobre la barra. Y las deformaciones se rigen por las siguientes ecuaciones:

$$y(x) = \frac{Fx^2}{6EI}(3L - x); y_{max} = \frac{FL^4}{3EI} \quad (3.52)$$

$$y(x) = \frac{Fx^2}{24EI}(x^2 - 4Lx + 6L^2); y_{max} = \frac{FL^4}{8EI} = \frac{(wA)L^4}{8EI} \quad (3.53)$$

En donde y es la deflexión, F es la fuerza aplicada, E es el módulo de Young, I es el segundo momento de área del perfil y L es el largo total de la barra. Aparte, la primera ecuación es para el caso de fuerza puntual en el extremo libre, y la segunda para esfuerzo homogéneo a lo largo de la viga. Para el segundo caso la fuerza corresponde al esfuerzo w por el área de aplicación A de la misma.

Para cada tipo de escenario, es recomendable probar varias configuraciones geométricas y tamaño de malla para verificar el nivel de error que acarrea el sistema.

Además también se puede hacer análisis por inspección de cada matriz global, rigideces, esfuerzos, deformación y reacciones.

La metodología de construcción de los ensayos no difiere mucho del caso del espesador, salvo que la geometría del mallado es distinta, sin embargo la base matemático-física es exactamente la

misma, los métodos programáticos son los mismos, por lo tanto no hace falta hacer una descripción más a fondo de dichos ensayos.

Los códigos asociados a estos procedimientos corresponden a los Anexos 8.1.8, 8.1.9, 8.1.10 y 8.1.11.

4 RESULTADOS

4.1 PRE-VALIDACIÓN DE ELEMENTO SHELL

4.1.1 Resultados para viga horizontal empotrada en un extremo y fuerza puntual aplicada en extremo libre

En primer lugar se probó un caso único y se revisó por inspección que la matriz de rigidez global tuviera la forma correcta. Esto es indicador que el algoritmo de ensamble cumple correctamente su función. Además, de la misma forma se puede verificar las condiciones que las condiciones de borde fueron bien aplicadas.

Para chequear esto, basta con probar para dos elementos, es decir, 6 nodos. Como se muestra en la Figura 4.1.

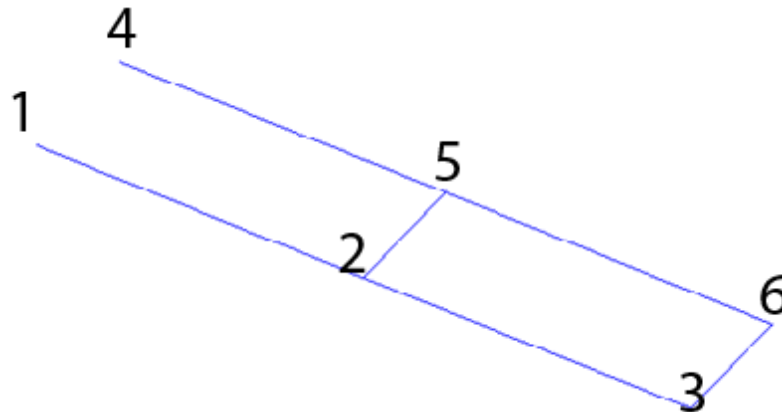


Figura 4.1 Ensayo viga empotrada con 2 elementos Shell

Para este ensayo, el arreglo de los nodos tiene los siguientes valores:

Tabla 4.1 Nodos y coordenadas para ensayo 1, viga empotrada horizontal

1	[0 0 0]	4	[0 0,08 0]
2	[0,2 0 0]	5	[0,2 0,08 0]
3	[0,4 0 0]	6	[0,4 0,08 0]

Además, el arreglo de los elementos se conforma como sigue:

Tabla 4.2 Elementos y sus nodos para ensayo 1, viga empotrada horizontal

1	[1 2 5 4]
2	[2 3 6 5]

A partir de esta información, se puede inferir que en la matriz de rigidez, para las filas correspondientes a los grados de libertad del nodo 1, deben existir valores para las columnas asociadas a los grados de libertad de los nodos 1, 2, 5 y 4. De la misma como los nodos 2 y 5 interactúan con todo el resto de los nodos pues son comunes entre los dos elementos.

En la Figura 4.2 se muestra el ensamble, correspondiente, en verde claro se muestran los valores de la diagonal, o las rigideces de los grados de libertad de cada nodo respecto a sí mismo. Y en naranja se muestran los grados de libertad cruzados, es decir, de un nodo respecto a otro. Los números representan los nodos y los bloques son los grados de libertad agrupados por nodo.

A su vez, las condiciones de borde se deberían ver reflejadas en las filas y columnas asociadas a los nodos 1 y 4, con unos en la diagonal y ceros en el resto de los valores. En la segunda figura, las celdas destacadas en celeste muestran si esto se cumple o no. Para verificar esto no es necesario ver los valores con detalle, sino tener una apreciación de su forma.

	1	2	3	4	5	6	
1	8E+07 3E+07 0 0 0 0 3E+07 2E+08 0 0 0 0 0 0 2E+09 8E+07 8E+07 0 0 8E+07 5E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 8E+07 0 0 0 0 8E+07 2E+08 0 0 0 0 0 0 5E+09 2E+08 0 0 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	0 0	8E+07 3E+07 0 0 0 0 3E+07 2E+08 0 0 0 0 0 0 2E+09 8E+07 8E+07 0 0 8E+07 5E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 8E+07 0 0 0 0 8E+07 2E+08 0 0 0 0 0 0 5E+09 2E+08 0 0 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 8E+07 0 0 0 0 8E+07 2E+08 0 0 0 0 0 0 5E+09 2E+08 0 0 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 8E+07 0 0 0 0 8E+07 2E+08 0 0 0 0 0 0 5E+09 2E+08 0 0 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0
2	2E+06 2E+06 0 0 0 0 8E+07 7E+07 0 0 0 0 0 0 2E+09 8E+07 3E+07 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 8E+07 0 0 0 0 8E+07 2E+08 0 0 0 0 0 0 5E+09 2E+08 0 0 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 2E+06 0 0 0 0 8E+07 7E+07 0 0 0 0 0 0 2E+09 8E+07 3E+07 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 2E+06 0 0 0 0 8E+07 7E+07 0 0 0 0 0 0 2E+09 8E+07 3E+07 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 2E+06 0 0 0 0 8E+07 7E+07 0 0 0 0 0 0 2E+09 8E+07 3E+07 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 2E+06 0 0 0 0 8E+07 7E+07 0 0 0 0 0 0 2E+09 8E+07 3E+07 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 2E+06 0 0 0 0 8E+07 7E+07 0 0 0 0 0 0 2E+09 8E+07 3E+07 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0
3	0 0	2E+06 2E+06 0 0 0 0 8E+07 7E+07 0 0 0 0 0 0 2E+09 8E+07 3E+07 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 2E+06 0 0 0 0 8E+07 7E+07 0 0 0 0 0 0 2E+09 8E+07 3E+07 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 2E+06 0 0 0 0 8E+07 7E+07 0 0 0 0 0 0 2E+09 8E+07 3E+07 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 2E+06 0 0 0 0 8E+07 7E+07 0 0 0 0 0 0 2E+09 8E+07 3E+07 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 2E+06 0 0 0 0 8E+07 7E+07 0 0 0 0 0 0 2E+09 8E+07 3E+07 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 2E+06 0 0 0 0 8E+07 7E+07 0 0 0 0 0 0 2E+09 8E+07 3E+07 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0
4	8E+07 3E+07 0 0 0 0 3E+07 2E+08 0 0 0 0 0 0 2E+09 8E+07 8E+07 0 0 8E+07 5E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 8E+07 0 0 0 0 8E+07 2E+08 0 0 0 0 0 0 5E+09 2E+08 0 0 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	0 0	8E+07 3E+07 0 0 0 0 3E+07 2E+08 0 0 0 0 0 0 2E+09 8E+07 8E+07 0 0 8E+07 5E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 8E+07 0 0 0 0 8E+07 2E+08 0 0 0 0 0 0 5E+09 2E+08 0 0 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 8E+07 0 0 0 0 8E+07 2E+08 0 0 0 0 0 0 5E+09 2E+08 0 0 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 8E+07 0 0 0 0 8E+07 2E+08 0 0 0 0 0 0 5E+09 2E+08 0 0 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0
5	2E+06 2E+06 0 0 0 0 8E+07 7E+07 0 0 0 0 0 0 2E+09 8E+07 3E+07 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 8E+07 0 0 0 0 8E+07 2E+08 0 0 0 0 0 0 5E+09 2E+08 0 0 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 2E+06 0 0 0 0 8E+07 7E+07 0 0 0 0 0 0 2E+09 8E+07 3E+07 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 2E+06 0 0 0 0 8E+07 7E+07 0 0 0 0 0 0 2E+09 8E+07 3E+07 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 2E+06 0 0 0 0 8E+07 7E+07 0 0 0 0 0 0 2E+09 8E+07 3E+07 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 2E+06 0 0 0 0 8E+07 7E+07 0 0 0 0 0 0 2E+09 8E+07 3E+07 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 2E+06 0 0 0 0 8E+07 7E+07 0 0 0 0 0 0 2E+09 8E+07 3E+07 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0
6	0 0	2E+06 8E+07 0 0 0 0 8E+07 2E+08 0 0 0 0 0 0 5E+09 2E+08 0 0 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 2E+06 0 0 0 0 8E+07 7E+07 0 0 0 0 0 0 2E+09 8E+07 3E+07 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 2E+06 0 0 0 0 8E+07 7E+07 0 0 0 0 0 0 2E+09 8E+07 3E+07 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 2E+06 0 0 0 0 8E+07 7E+07 0 0 0 0 0 0 2E+09 8E+07 3E+07 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 2E+06 0 0 0 0 8E+07 7E+07 0 0 0 0 0 0 2E+09 8E+07 3E+07 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0	2E+06 2E+06 0 0 0 0 8E+07 7E+07 0 0 0 0 0 0 2E+09 8E+07 3E+07 0 0 8E+07 4E+06 30048 0 0 8E+07 30048 3E+06 0 0 0 0 0 0

Figura 4.2 Matriz de rigidez global para 2 elementos previa la aplicación de condiciones de borde

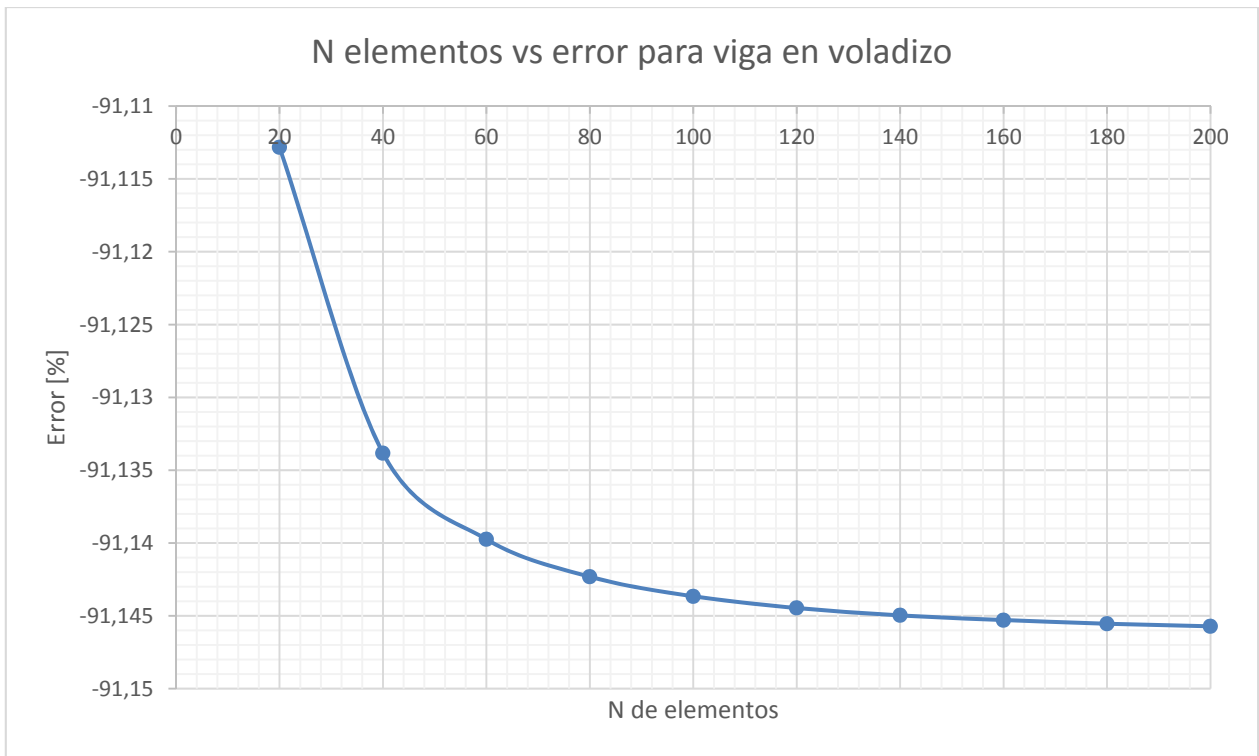


Figura 4.4 Número de elementos versus error entre deformación teórica y obtenida

En tercer lugar, se probó con el mismo tamaño de malla para distintas configuraciones de sección/fuerza puntual. Para todos los casos se probó con una sección cuadrada y acero como material.

Tabla 4.4 Valores ensayo 2 viga empotrada horizontal

L	1,76 m	E	210 GPa	ν	0,3
-----	--------	-----	---------	-------	-----

El número de elementos se fijó en 20, con un tamaño y distribución uniforme como se puede ver en la Figura 4.5.

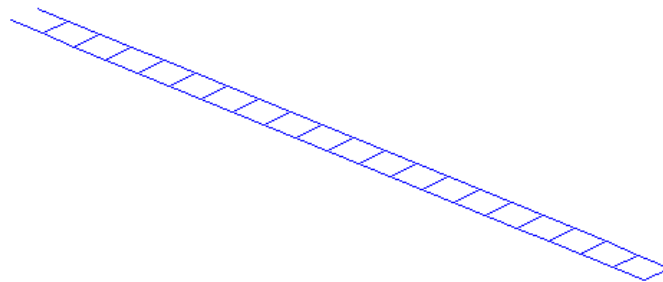


Figura 4.5 Ensayo viga empotrada con 20 elementos Shell

Los resultados fueron los siguientes:

Tabla 4.5 Resultados viga empotrada 20 elementos Shell

F[N]	b [m]	a [m]	flexión teórica [m]	flexión FEM [m]	Error [%]
100	0,04	0,04	0,004056381	0,004038614	0,439915061

100	0,02	0,06	0,002403781	0,002394732	0,377889857
100	0,08	0,05	0,001038434	0,001033714	0,456550512
2000	0,04	0,04	0,081127619	0,08077229	0,439915061
2000	0,02	0,06	0,048075626	0,047894637	0,377889857
2000	0,08	0,05	0,02076867	0,020674282	0,456550512
30000	0,04	0,04	1,216914286	1,211584344	0,439915061
30000	0,02	0,06	0,721134392	0,718419557	0,377889857
30000	0,08	0,05	0,311530057	0,310114229	0,456550512
400000	0,04	0,04	16,22552381	1,62E+01	0,439915061
400000	0,02	0,06	9,61512522	9,58E+00	0,377889857
400000	0,08	0,05	4,153734095	4,134856387	0,456550512

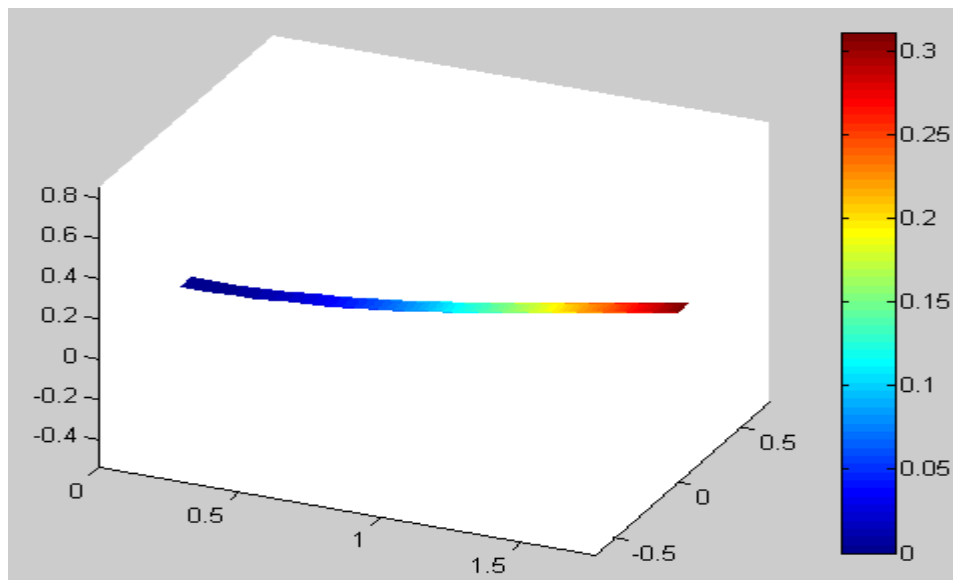


Figura 4.6 Viga deformada para $F=30000$ N, $b=0,08$ y $a=0,05$

4.1.2 Resultados para viga vertical empotrada en un extremo y fuerza puntual aplicada en extremo libre

Para este ensayo se comparó los valores de la deformación de un caso en particular en una distribución de puntos a lo largo de la barra, con la deformación sugerida por la teoría en los mismos puntos. Se recalca que para este caso la viga se encuentra dispuesta de forma vertical, para chequear si la matriz de transformación está correctamente constituida.

Los valores fijados para el ensayo fueron los siguientes:

Tabla 4.6 Valores ensayo viga empotrada vertical

L	2,85 m	F	1000 N	a	0,03 m	b	0,03 m	E	210 GPa	ν	0,3
-----	--------	-----	--------	-----	--------	-----	--------	-----	---------	-------	-----

La cantidad de elementos se fijó en 60, distribuidos homogéneamente con tamaño uniforme, y los resultados fueron los siguientes:

Tabla 4.7 Resultados viga empotrada vertical

N nodo	Distancia z [m]	flexión teórica [m]	flexión FEM [m]	Error [%]
----------	-------------------	---------------------	-----------------	-----------

5	0,19	0,003548454	3,50E-03	1,366832964
10	0,4275	0,017453705	1,73E-02	0,651426372
15	0,665	0,040998755	4,08E-02	0,436264283
20	0,9025	0,073238525	7,30E-02	0,333185495
25	1,14	0,113227937	1,13E-01	0,273055856
30	1,3775	0,160021912	1,60E-01	0,233924823
35	1,615	0,212675372	2,12E-01	0,206647082
40	1,8525	0,270243239	2,70E-01	0,186737516
45	2,09	0,331780435	3,31E-01	0,17174177
50	2,3275	0,396341881	3,96E-01	0,160207163
55	2,565	0,4629825	4,62E-01	0,151222427
60	2,8025	0,530757212	5,30E-01	0,14419151

Además, otra información interesante a analizar son las reacciones en el empotramiento⁶:

Tabla 4.8 Reacciones en empotramiento viga voladizo vertical

Nodo	$R_x[N]$	$R_y[N]$	$R_z[N]$	$M_x[Nm]$	$M_y[Nm]$	$M_z[Nm]$
1	-500,00000139588	0	0	0	-1425	435
62	-500,000001407183	0	0	0	-1425	-435

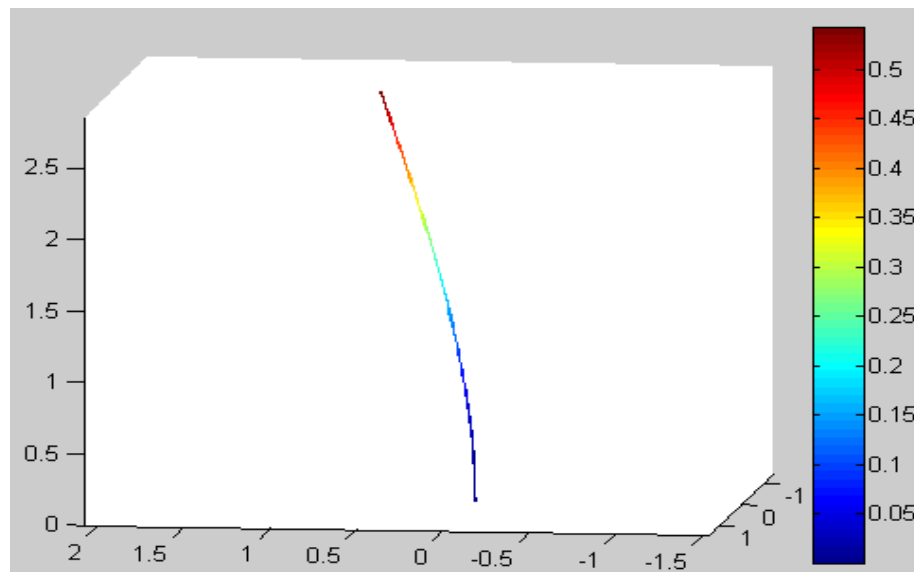


Figura 4.7 Deflexión en x para viga empotrada vertical

⁶ Estas reacciones están expresadas según las coordenadas globales y no locales de los nodos.

4.1.3 Resultados para viga horizontal empotrada en un extremo y esfuerzo uniforme en aplicado en todo su largo

Para este ensayo el interés se centra en que la matriz de esfuerzos esté bien generada y ensamblada para el problema de viga en voladizo con carga aplicada en todo su largo. Para verificar esto se realizó la modelación con los siguientes parámetros.

Tabla 4.9 Valores ensayo viga horizontal con esfuerzo aplicado en todo su largo

L	1 m	w	100000 N/m ²	a	0,06 m	b	0,07 m	E	210 GPa	ν	0,3
-----	-----	-----	-------------------------	-----	--------	-----	--------	-----	---------	-------	-----

El número de elementos se fijó en 64 de largo por 3 de ancho, como se puede ver en la Figura 4.8, en donde además se grafican los esfuerzos nodales.

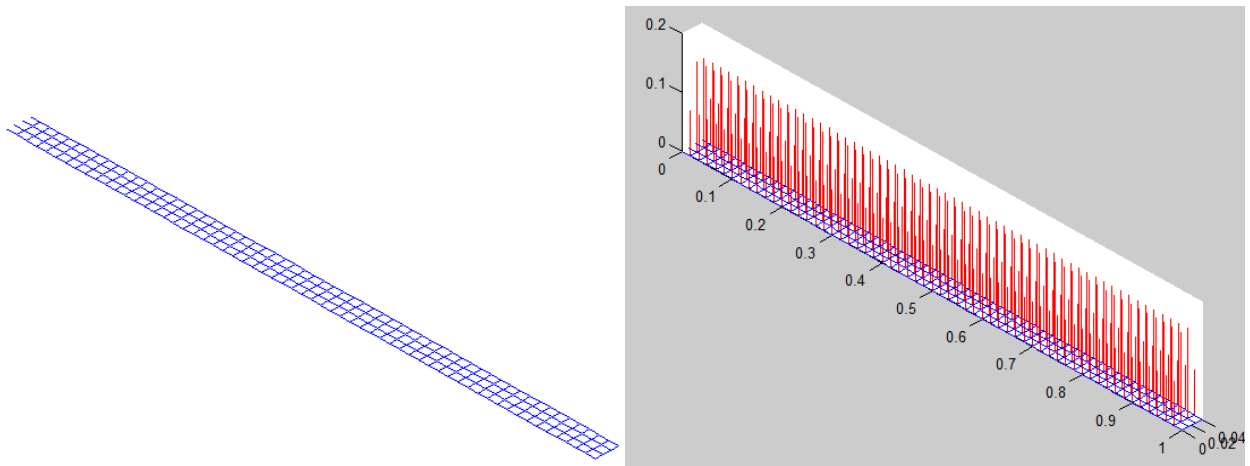


Figura 4.8 Malla y esfuerzos viga empotrada horizontal con esfuerzo aplicado en todo su largo

Los resultados y su comparación con el modelo teórico son los siguientes:

N nodo	Distancia z [m]	flexión teórica [m]	flexión FEM [m]	Error [%]
5	0,0667	1,7719E-05	1,80E-05	1,60667229
10	0,15	8,4691E-05	8,60E-05	1,51820006
15	0,2333	0,00019349	0,00019565	1,1183644
20	0,3167	0,00033652	0,00033925	0,80969581
25	0,4	0,00050646	0,00050977	0,65356335
30	0,4833	0,00069726	0,00070102	0,53916473
35	0,5667	0,00090382	0,00090764	0,421721
40	0,65	0,00112107	0,00112503	0,35333372
45	0,7333	0,00134545	0,00134942	0,29460613
50	0,8167	0,00157429	0,00157783	0,2248829
55	0,9	0,00180487	0,00180811	0,17946122
60	0,9833	0,0020361	0,00203888	0,13682221

4.2 PRE-VALIDACIÓN DE ELEMENTO FRAME

Para verificar la matriz de rigidez y el ensamble de este tipo de elementos se probó para un caso en particular con los siguientes parámetros:

L	2,85 m	F	1000 N	a	0,03 m	b	0,03 m	E	210 GPa	ν	0,3
-----	--------	-----	--------	-----	--------	-----	--------	-----	---------	-------	-----

El número de elementos se fijó en 20, y la distribución homogénea y uniforme se puede ver en la Figura 4.9.

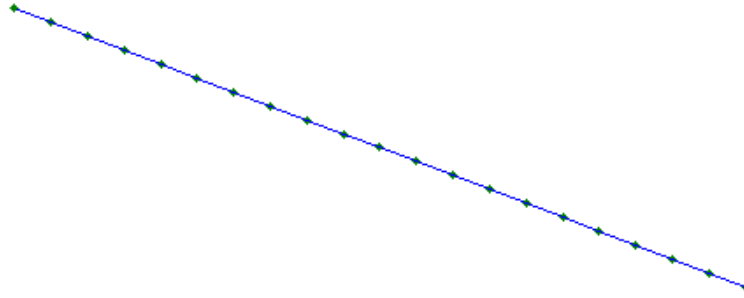


Figura 4.9 Viga empotrada horizontal con elementos Frame

Los resultados arrojados fueron los siguientes:

Tabla 4.10 Resultados viga horizontal empotrada de 20 elementos Frame

N Nodo	Distancia x [m]	flexión teórica [m]	flexión FEM [m]	Error [%]
2	0,1425	0,002007346	0,002007346	1,69429E-09
4	0,4275	0,017453705	0,017453705	1,67104E-09
6	0,7125	0,046781374	0,046781374	1,64622E-09
8	0,9975	0,088357257	0,088357257	1,61906E-09
10	1,2825	0,140548259	0,140548259	1,58984E-09
12	1,5675	0,201721285	0,201721285	1,5582E-09
14	1,8525	0,270243239	0,270243239	1,52556E-09
16	2,1375	0,344481027	0,344481027	1,49396E-09
18	2,4225	0,42280	0,422801553	1,46434E-09
20	2,7075	0,503571721	0,503571721	1,43909E-09

4.3 RESULTADOS ESPESADOR

Antes de los resultados finales, se probó el mismo algoritmo que genera el mallado para el software final, pero con distintos tipos de esfuerzos a modo de verificar si las respuestas del sistema tienen un sentido coherente para distintos escenarios, este análisis es más bien cualitativo en la mayoría de los casos, por la misma razón que se realizó el test de pre-validación para acreditar las magnitudes numéricas.

4.3.1 Aplicación de las fuerzas de peso.

Para este ensayo se probó una configuración geométrica determinada sin ningún esfuerzo más que el peso mismo de los elementos.

La configuración usada fue la siguiente:

Tabla 4.11 Parámetros simulación de espesor sujeto a fuerzas de peso

Geometría espesor				Materiales Vigas: acero		Materiales Manto: acero	
h_1	2 m	r_1	3 m	E	$210 \times 10^9 Pa$	h	0,03 m
h_2	1 m	r_2	2 m	ν	0,3 m	ρ	$7900 kg/m^3$
h_3	0,8 m	r_3	1,5 m	I_x	$6,75E - 08 m^4$		
h_4	0,98 m	D	[0,25; 0,7]m	I_y	$6,75E - 08 m^4$		
N_s	4			J	$1,08397E - 07 m^4$		
N_p	2			A	$0,0009 m^2$		

Los resultados arrojados por el programa de elementos finitos fueron los siguientes:

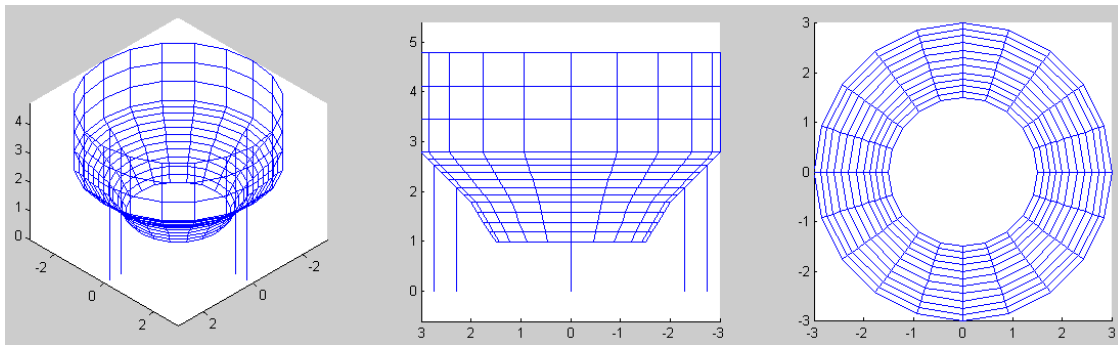


Figura 4.10 Mallado de espesor sujeto a fuerzas de peso

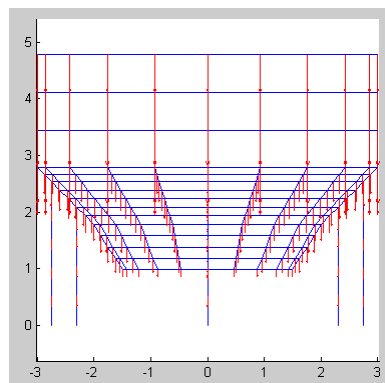


Figura 4.11 Fuerzas nodales del peso de cada elemento

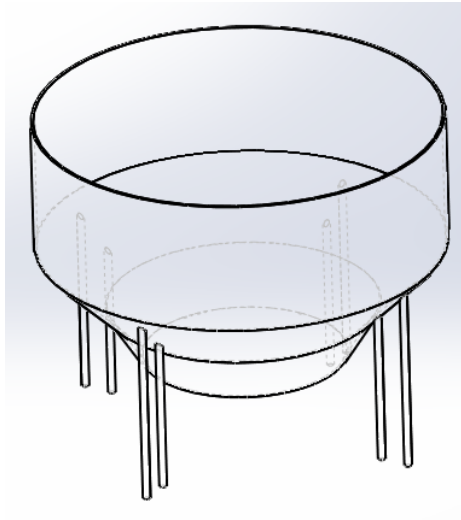


Figura 4.12 Espesador generado en 3D mediante software SolidWorks

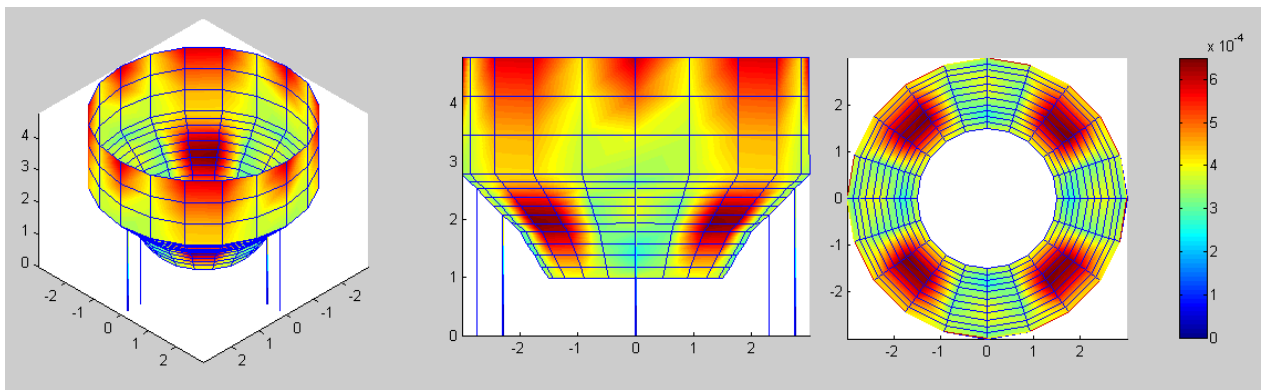


Figura 4.13 Espesador deformado por el peso de los elementos

4.3.2 Aplicación de fuerza puntual

Este ensayo consiste en la aplicación radial de una fuerza igual a 10.000 [N] en la máxima altura del espesador. Su objetivo es hacer evidente el correcto ensamblaje de la geometría y corroborar la forma de las reacciones en la base. Para este caso no se consideró el peso de los elementos.

Tabla 4.12 Parámetros simulación de espesador con aplicación de fuerza puntual

Geometría espesador				Materiales Vigas: acero		Materiales Manto: acero	
h_1	6 m	r_1	10 m	E	$210 \times 10^9 Pa$	h	0,02 m
h_2	5 m	r_2	5 m	ν	0,3 m	ρ	$7890 kg/m^3$
h_3	4 m	r_3	2 m	I_x	$6,75E - 06 m^4$		
h_4	3 m	D	[1,5; 3; 5]m	I_y	$6,75E - 06 m^4$		
N_s	5			J	$1,08397E - 05 m^4$		
N_p	3			A	$0,0009 m^2$		

Los resultados arrojados por el programa de elementos finitos fueron los siguientes:

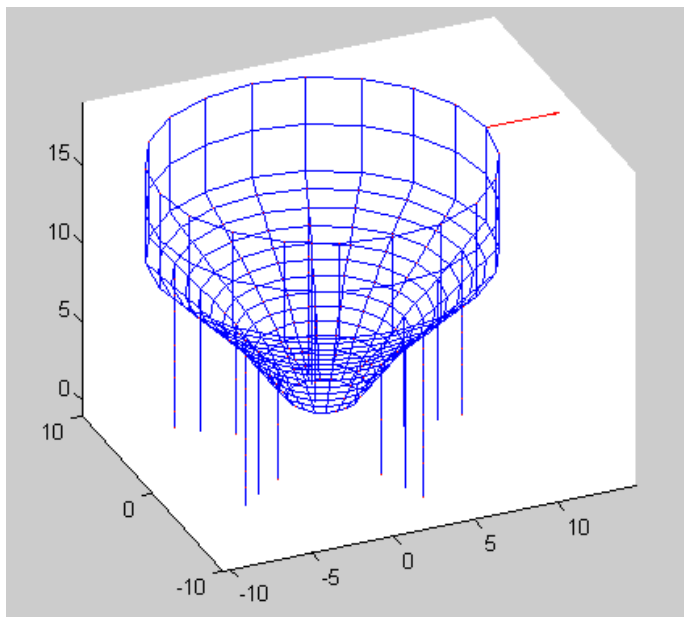


Figura 4.14 Fuerza puntual aplicada al espesor

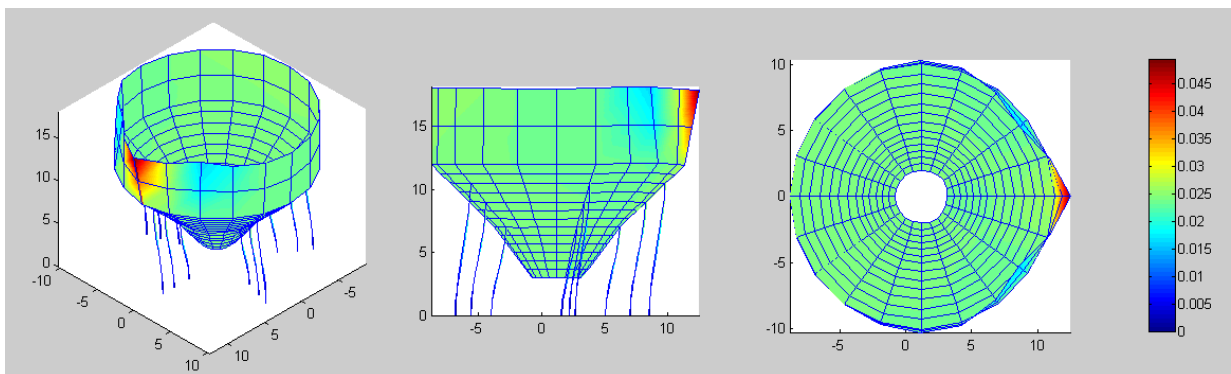


Figura 4.15 Espesor deformado debido a fuerza puntual

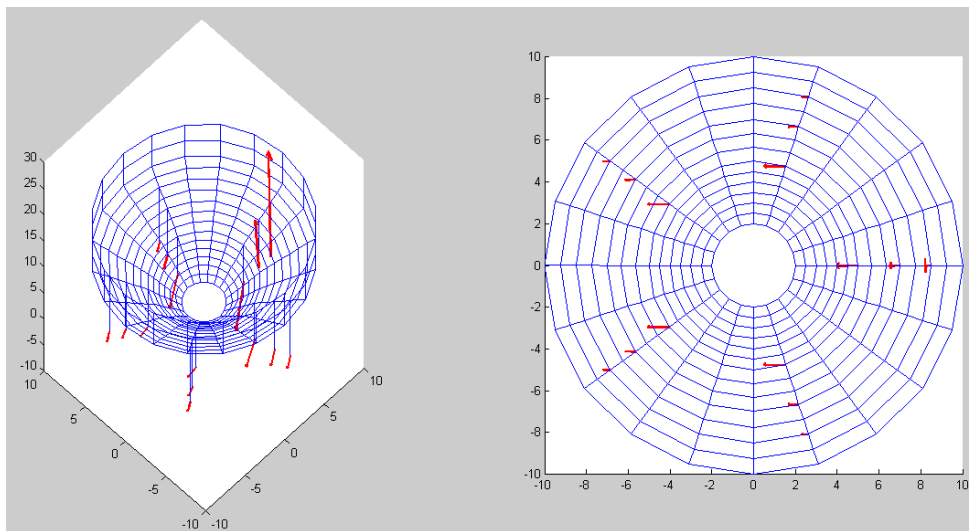


Figura 4.16 Reacciones en la base de los pilares debido a fuerza puntual

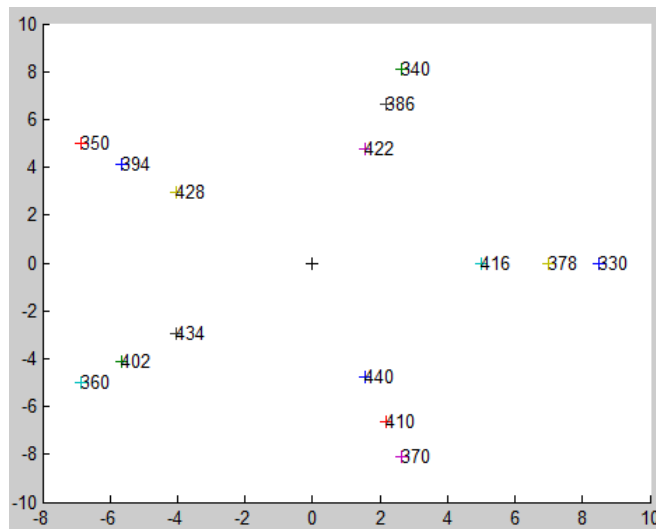


Figura 4.17 Nodos de los pilares ensayo fuerza puntual

Además las reacciones en la base de cada pilar fueron las siguientes:

N	R_x	R_y	R_z	N	R_x	R_y	R_z
330	-323,98	-1,209298	23013,66	402	-539,34	3,3606106	-1069,908
340	-339,42	-1,459756	-1499,678	410	-538,11	1,8328666	-2769,441
350	-340,35	-2,162292	-1580,033	416	-1083,4	-7,261376	-8904,078
360	-340,45	1,525429	-1501,909	422	-1141,7	6,779671	-5840,852
370	-338,94	0,7375644	-2188,301	428	-1139,6	-6,272036	-560,6163
378	-513,46	-2,780714	10706,529	424	-1139,2	4,3662395	-439,4265
386	-538,13	-0,199968	-2136,059	440	-1144,5	7,0545493	-4141,457
394	-539,37	-4,311494	-1088,431				

La suma de las reacciones en la dirección x para los pilares da como resultado $-9999,99999219591$, que es casi exactamente la misma magnitud que la fuerza aplicada. Para las otras sumas en la dirección y y z , respectivamente, el resultado es $-2,32621181250892e -06$ y $-6,76812305755448e -07$.

4.3.3 Aplicación de esfuerzo debido a presión

El último ensayo previo a la prueba del software mismo, consiste en la verificación del correcto cálculo de la presión hidrostática en las paredes del manto. Para este caso no se consideró el peso de los elementos y se utilizó la misma configuración geométrica y mallado que para el caso anterior. Se usó agua como fluido, con una densidad de 1000 kg/m^3 .

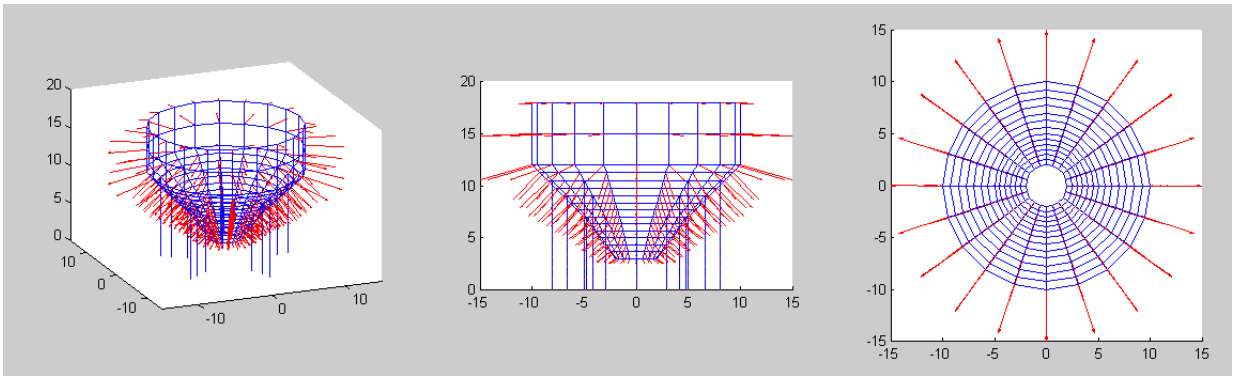


Figura 4.18 Fuerzas nodales del gradiente de presión sobre el manto

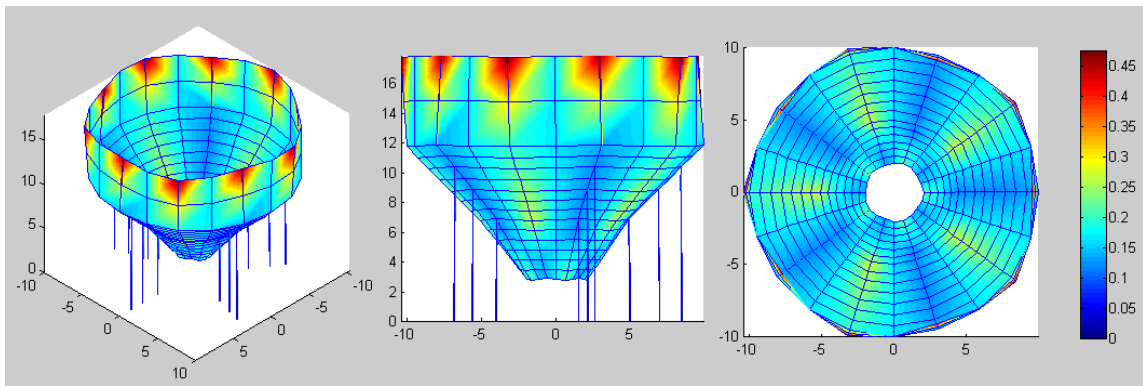


Figura 4.19 Espesador deformado debido a gradiente de presión

4.4 SOFTWARE FINAL

Los últimos resultados corresponden a los arrojados por la última versión escrita del software a la fecha de este trabajo de título. Considera los esfuerzos de peso de los elementos y el gradiente de presión debido al fluido.

Para esta fase se probaron varias configuraciones, con el objetivo de verificar la coherencia de las respuestas encontradas, la versatilidad del software, su usabilidad y por último el tiempo de respuesta. Para conseguir estas comprobaciones, se sometió el programa a los siguientes ensayos:

a) Espesador de dimensiones pequeñas

Tabla 4.13 Geometría espesador pequeño

Geometría espesador				Materiales Vigas: acero		Materiales Manto: acero	
h_1	2 m	r_1	1,5 m	E	$210 \times 10^9 Pa$	h	0,01 m
h_2	1 m	r_2	1 m	ν	0,3 m	ρ	$7890 kg/m^3$
h_3	0,7 m	r_3	0,3 m	I_x	$6,75E - 06 m^4$	Fluido: Pulpa	
h_4	1 m	D	[0; 0,4]m	I_y	$6,75E - 06 m^4$	ρ_f	$2.000 kg/m^3$
N_s	4			J	$1,08397E - 05 m^4$		
N_p	2			A	$0,0009 m^2$		

b) Espesador de dimensiones grandes

Tabla 4.14 Geometría espesador grande

Geometría espesador				Materiales Vigas: acero		Materiales Manto: acero	
h_1	6 m	r_1	30 m	E	$210 \times 10^9 Pa$	h	0,1 m
h_2	5 m	r_2	15 m	ν	0,3 m	ρ	$7890 kg/m^3$
h_3	3 m	r_3	3 m	I_x	$6,75E - 05 m^4$	Fluido: Pulpa	
h_4	3 m	D	[0.5; 5; 10; 15]m	I_y	$6,75E - 05 m^4$	ρ_f	$2.000 kg/m^3$
N_s	6			J	$1,08397E - 04 m^4$		
N_p	4			A	$0,0025 m^2$		

c) Espesador grande lleno con fluido denso

Tabla 4.15 Geometría espesador lleno con fluido denso

Geometría espesador				Materiales Vigas: acero		Materiales Manto: acero	
h_1	4 m	r_1	10 m	E	$210 \times 10^9 Pa$	h	0,03 m
h_2	2 m	r_2	4 m	ν	0,3 m	ρ	$7890 kg/m^3$
h_3	2 m	r_3	2 m	I_x	$6,75E - 05 m^4$	Fluido: Pulpa densa	
h_4	3 m	D	[0; 3; 6]m	I_y	$6,75E - 05 m^4$	ρ_f	$5.000 kg/m^3$
N_s	5			J	$1,08397E - 04 m^4$		
N_p	3			A	$0,0025 m^2$		

4.4.1 Malla y estado de deformación

Los resultados (mallado y deformación) obtenidos a partir del software para cada uno de los casos se exponen a continuación.

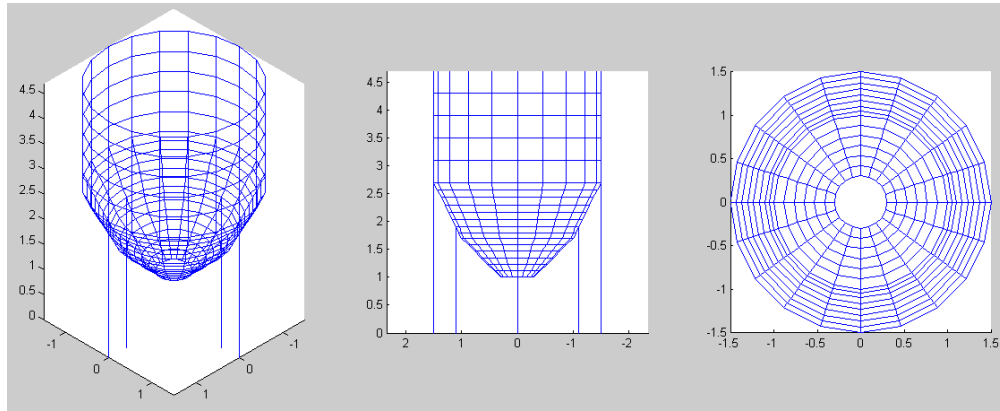


Figura 4.20 Mallado espesor pequeño

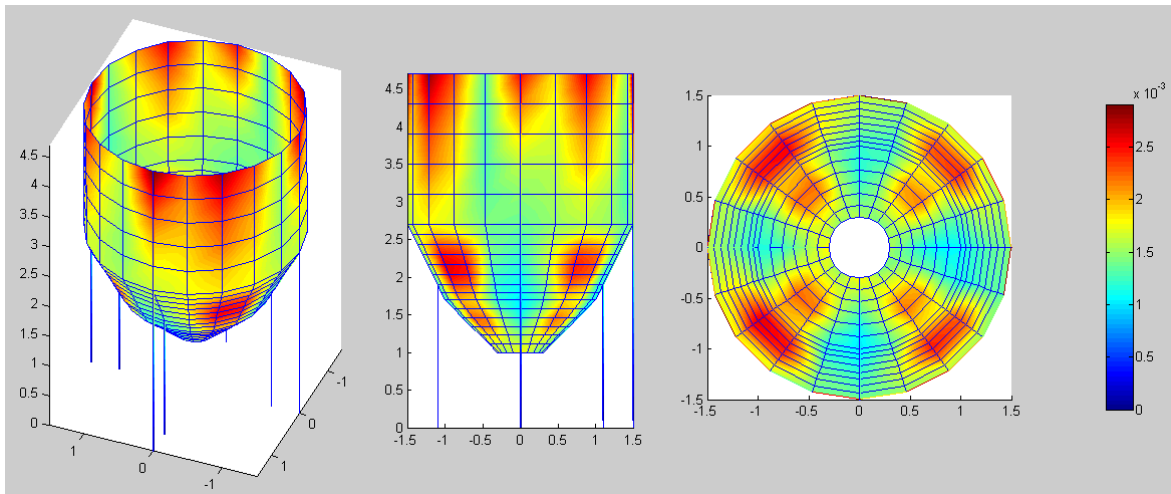


Figura 4.21 Espesor pequeño deformado

Tiempo de cálculo	9,937380 [s]	# Elementos	542
-------------------	--------------	-------------	-----

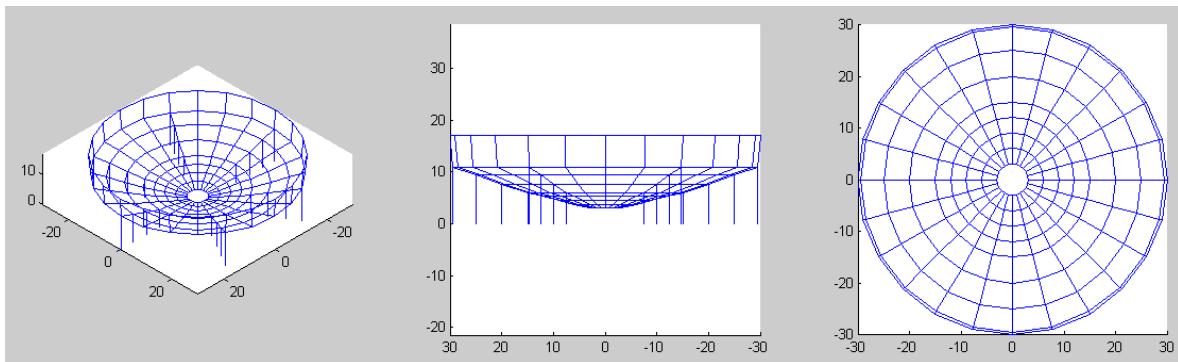


Figura 4.22 Mallado espesor grande

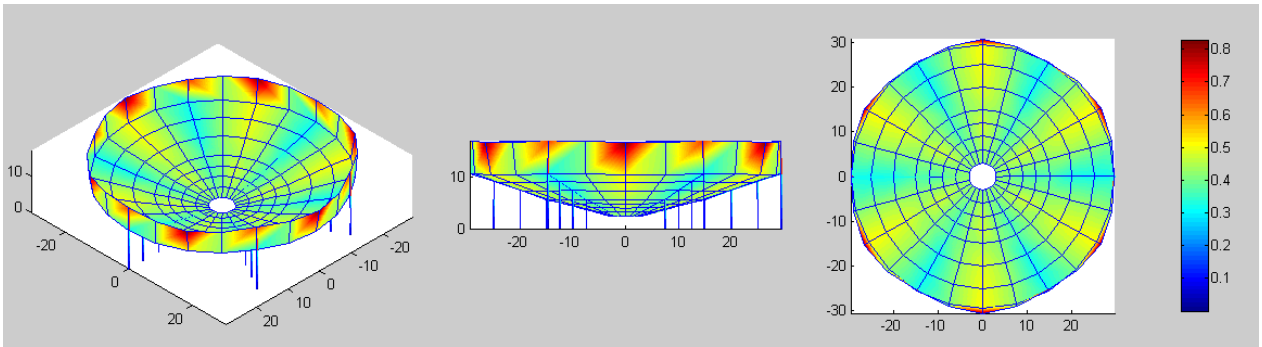


Figura 4.23 Espesador grande deformado

Tiempo de cálculo	2,675091 [s]	# Elementos	330
-------------------	--------------	-------------	-----

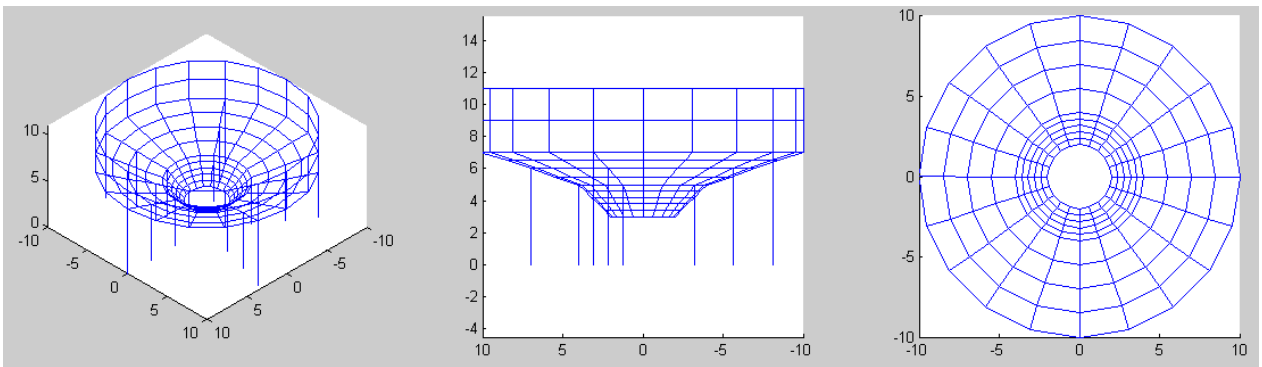


Figura 4.24 Mallado espesador lleno de fluido denso

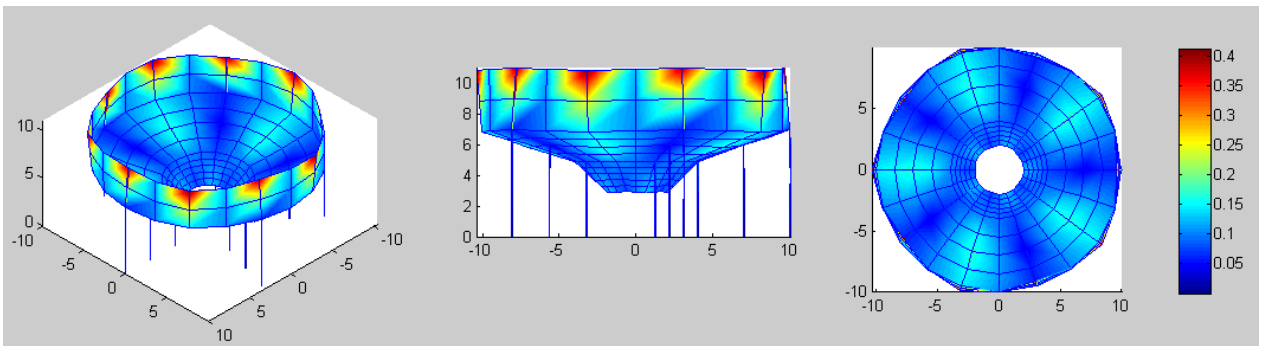


Figura 4.25 Espesador deformado debido a un fluido muy denso

Tiempo de cálculo	3,437628 [s]	# Elementos	365
-------------------	--------------	-------------	-----

4.4.2 Cuadro de reacciones

Los cuadros de reacciones predichos por el software para cada uno de los ensayos fueron los siguientes:

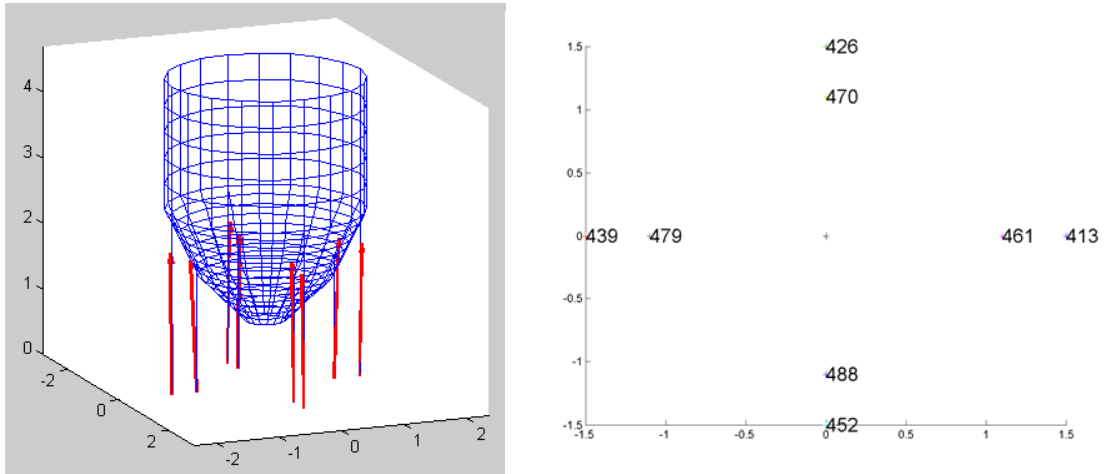


Figura 4.26 Cuadro de reacciones y nodos de los pilares para espesor pequeño

Nodo	Rx	Ry	Rz	Mx	My	Mz
413	337,26684	-801,342317	48063,8303	1220,47328	561,978091	-127,882001
426	996,918312	334,893718	48051,8316	-557,433005	1502,15488	-142,831631
439	-125,735113	985,085309	50516,2038	-1488,71758	-244,595101	-133,424986
452	-793,866471	-126,486455	50532,0173	247,131876	-1210,68069	-123,659656
461	1160,37791	-918,681292	51204,2189	1286,6637	1357,34831	-199,342347
470	1412,42591	1159,11529	51123,5963	-1353,38744	1792,29261	-236,222432
479	-2026,35889	1393,94208	47615,6546	-1771,53627	-1789,03306	-204,523453
488	-961,028496	-2026,52633	47694,4511	1791,08767	-1309,24461	-178,154607

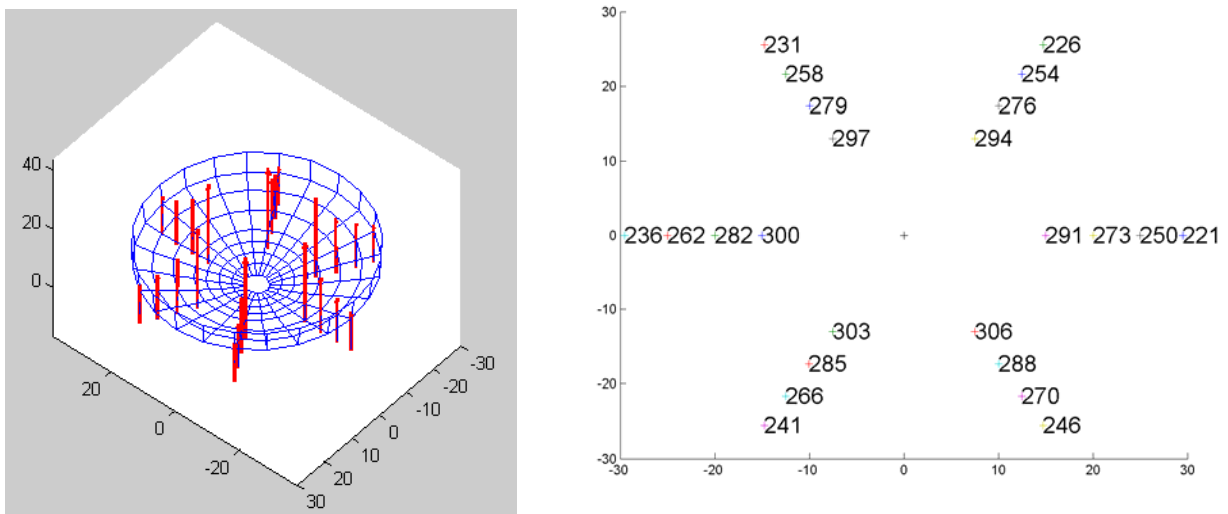


Figura 4.27 Cuadro de reacciones y nodos de los pilares para espesor grande

Nodo	Rx	Ry	Rz	Mx	My	Mz
221	-63158,8921	2398,71885	4240997,3	-6741,27385	-136133,682	-2643,71527
226	-21798,4882	-59326,431	4240997,3	127387,657	-48478,9538	-2643,71527
231	49686,6855	-39064,4697	4240997,3	85471,1759	106172,041	-2643,71527
236	52506,5486	35183,261	4240997,3	-74563,5657	114096,884	-2643,71527
241	-17235,8538	60808,9208	4240997,3	-131553,994	-35656,2889	-2643,71527
246	6377,77701	1360,20902	2936071,64	-3724,80876	34339,6246	8719,19273
250	677,20583	6485,95408	2936071,64	-33809,9529	7069,02393	8719,19273
254	-5959,24079	2648,33105	2936071,64	-17170,8913	-29970,7275	8719,19273
258	-4360,21918	-4849,19548	2936071,64	23197,7585	-25591,9522	8719,19273
262	3264,47713	-5645,29868	2936071,64	31507,8945	14154,0312	8719,19273
266	-40809,7025	-48,4303907	7980222,28	-868,413319	-55778,6575	1539,27663
270	-12564,8316	-38827,2993	7980222,28	52780,3012	-18062,4632	1539,27663
273	33044,2095	-23948,1603	7980222,28	33488,4334	44615,4413	1539,27663
276	32987,2762	24026,5223	7980222,28	-32083,3111	45636,3224	1539,27663
279	-12656,9516	38797,3676	7980222,28	-53317,0102	-16410,6429	1539,27663
282	-63158,8921	2398,71885	4240997,3	-6741,27385	-136133,682	-2643,71527
285	-21798,4882	-59326,431	4240997,3	127387,657	-48478,9538	-2643,71527
288	49686,6855	-39064,4697	4240997,3	85471,1759	106172,041	-2643,71527
291	52506,5486	35183,261	4240997,3	-74563,5657	114096,884	-2643,71527
294	-17235,8538	60808,9208	4240997,3	-131553,994	-35656,2889	-2643,71527
297	6377,77701	1360,20902	2936071,64	-3724,80876	34339,6246	8719,19273
300	677,20583	6485,95408	2936071,64	-33809,9529	7069,02393	8719,19273
303	-5959,24079	2648,33105	2936071,64	-17170,8913	-29970,7275	8719,19273
306	-4360,21918	-4849,19548	2936071,64	23197,7585	-25591,9522	8719,19273

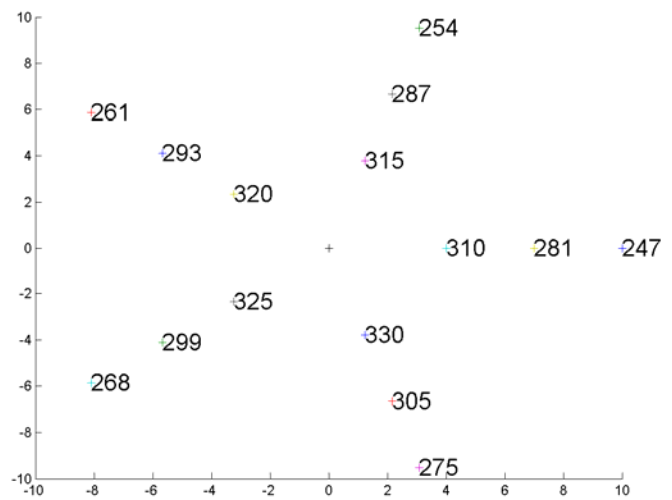
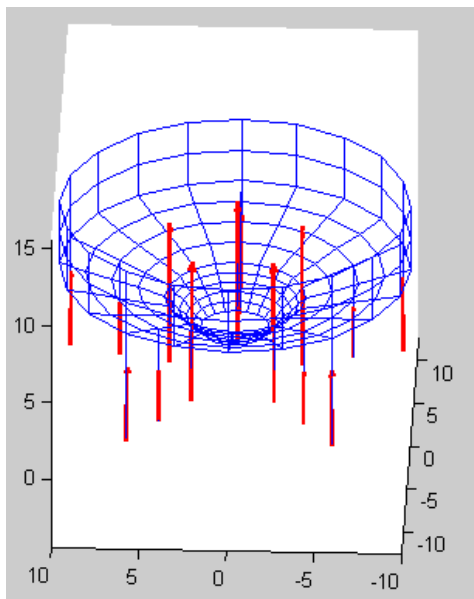


Figura 4.28 Cuadro de reacciones y nodos de los pilares para espesador lleno de fluido denso

Nodo	Rx	Ry	Rz	Mx	My	Mz
247	-58151,9276	2417,27884	4128472,63	-6698,30949	-125455,756	-2634,70133
254	-20268,9027	-54558,7894	4128472,63	117245,623	-45138,4316	-2634,70133
261	45625,0568	-36136,4651	4128472,63	79160,0894	97558,6712	-2634,70133
268	48466,7385	32225,2258	4128472,63	-68321,997	105433,006	-2634,70133
275	-15670,9651	56052,7499	4128472,63	-121385,406	-32397,4898	-2634,70133
281	4559,18889	1354,29841	2904368,53	-3638,6082	28294,5111	8852,39637
287	120,852525	4754,54753	2904368,53	-28034,0709	5282,96273	8852,39637
293	-4484,49792	1584,17357	2904368,53	-13687,4005	-25029,4605	8852,39637
299	-2892,42466	-3775,47442	2904368,53	19574,7922	-20752,0201	8852,39637
305	2696,88117	-3917,54508	2904368,53	25785,2874	12204,0068	8852,39637
310	-37789,0071	-151,742523	7978874,49	-626,898831	-52104,9049	1594,4102
315	-11533,1297	-35986,3724	7978874,49	49360,987	-16697,5173	1594,4102
320	30661,1409	-22089,0588	7978874,49	31133,6665	41785,2717	1594,4102
325	30482,7569	22334,5833	7978874,49	-30119,3229	42522,2355	1594,4102
330	-11821,7611	35892,5904	7978874,49	-49748,4318	-15505,0849	1594,4102

Para los tiempos de cálculo acá presentes, se estimó la curva a través de un polinomio, representada a continuación:

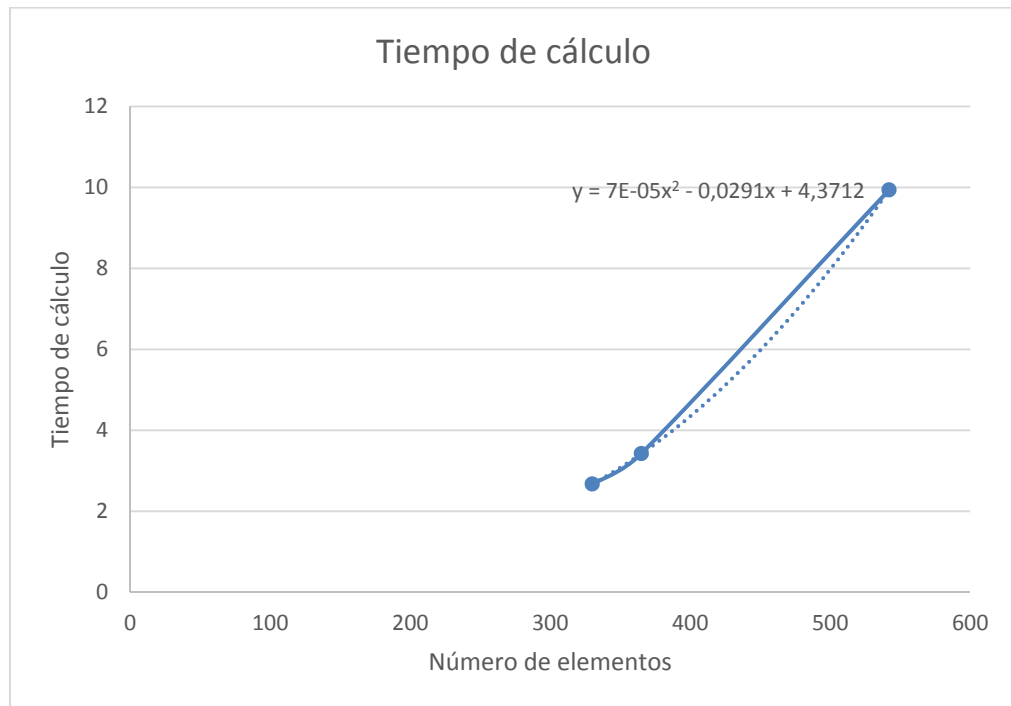


Figura 4.29 Tiempo cálculo en relación al número de elementos

4.4.3 Interfaz gráfica

Por último, la interfaz gráfica en que el usuario puede ingresar y obtener las respuestas ofrecidas por el software, es la siguiente:

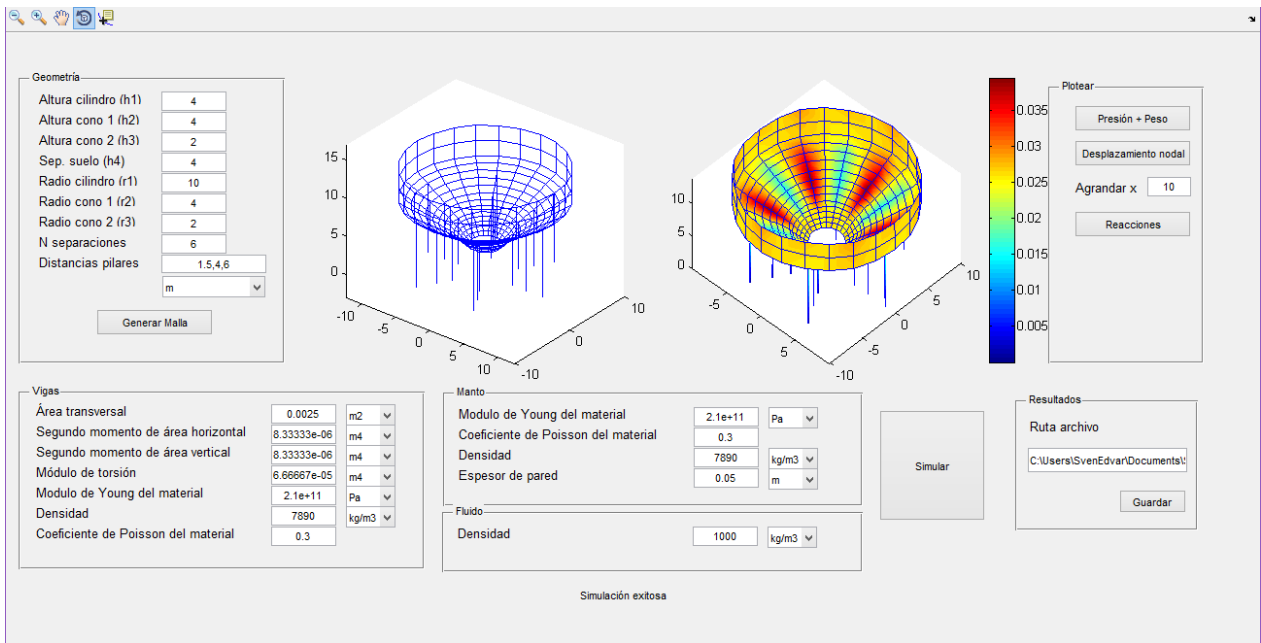


Figura 4.30 Interfaz de usuario

En esta interfaz se permite cambiar las unidades (MKS, mgs, sistema inglés, etc), e incluye su propia función de conversión de unidades. Además, el usuario puede guardar una imagen de los nodos de los pilares que están anclados y el cuadro de reacciones en un archivo Excel.

Los pasos para trabajar con el software son: Llenar los datos necesarios > Generar Malla > Simular > Graficar o guardar archivos.

5 DISCUSIÓN

5.1 ANÁLISIS DE PRE-VALIDACIÓN

El primer ensayo muestra que los elementos fueron correctamente generados a partir de la matriz de los nodos, como se puede ver en la Tabla 4.1 y Tabla 4.2.

Luego se puede verificar mediante la Figura 4.2 que el ensamble de la matriz de rigidez tiene la forma esperada, ya que están ocupadas las filas y columnas que corresponden a los nodos que interactúan entre sí. Dicho de otra forma, se corresponde con la contribución de cada elemento.

En la Figura 4.3 se muestra además, que las condiciones de empotramiento están bien localizadas en la matriz global de rigidez.

El segundo ensayo demuestra en primer lugar que el análisis mediante elementos finitos arroja un resultado bastante cercano al teórico, lo que refuerza el hecho de que la matriz de rigidez esté bien ensamblada, y las condiciones de borde bien estructuradas.

En la Figura 4.4 además se demuestra que existe una tendencia entre la deflexión supuesta por la simulación y la deflexión teórica, en función del número de elementos. Es decir, mientras mayor la fineza de la malla, menor el error de la respuesta. Además se observa que entre 20 y 60 nodos la caída del error es exponencial, mientras que a partir de 100 elementos, el error casi no evoluciona.

El tercer ensayo muestra que sin importar la configuración geométrica, el modelo obtiene una deflexión con un error bastante bajo (menor al 1%), sobre la deflexión teórica. Además, se observa que el error no evoluciona con el nivel de deformación, a pesar de que el modelo de deformaciones lineales supone una buena correlación con la realidad para pequeñas deformaciones (bajo el 1%). Esto supone que la respuesta del modelo tendría un error considerablemente bajo dentro del dominio aceptable por el modelo de deformación mecánica considerado para este trabajo.

Para el ensayo de la viga vertical, la deflexión en cada posición es bastante similar a la propuesta por el modelo, lo que indica que la matriz de transformación está bien constituida. Además se observa que el error crece a medida que los puntos se alejan del lugar de aplicación de la fuerza, lo cual tiene sentido si se piensa que la deformación es información que viaja a través de la malla.

A pesar de los errores encontrados para la deformación (que son bastante pequeños), para las reacciones en la pared se tiene que el error es casi nulo. Para los nodos anclados equivale casi exactamente a la fuerza aplicada en el extremo de la viga, y lo mismo ocurre para el momento, en que la suma de M_y es igual a las fuerzas aplicadas por el brazo, que en este caso equivale al largo de la viga.

En cuanto a la aplicación de esfuerzos superficiales, en la Figura 4.8 se puede ver gráficamente que los esfuerzos están siendo bien aplicados, en este caso normales a superficie. Se puede ver que en los nodos de los bordes estos son menores, esto se debe a que cada nodo aparece en menos contribuciones (dos en vez de cuatro), es decir pertenece a menos elementos. Sin embargo la deformación para los nodos de los bordes es casi idéntica que los del centro debido a que su rigidez se comporta de la misma forma (este último resultado, que compara la deformación en el centro con la de los bordes, se puede ver directo sobre el software, debido a que no es tan relevante y es gráficamente incómoda de insertar).

Por último el ensayo para elementos Frame tiene los mejores resultados, esto proviene de que en los primeros (ensayos para elementos Shell) se estaba utilizando un elemento de placa para simular un fenómeno que se comporta más bien como una línea (en el modelo mecánico). Los resultados no dejan lugar a duda de que la rigidez y el ensamble de estos elementos están bien desarrollados.

A lo sumo, los test de pre-validación tienen una respuesta que por lo menos valida numéricamente los algoritmos (con un error porcentual menor al 1% en promedio en relación al modelo teórico). Si se produce un cambio de geometría, pero se utilizan los mismos algoritmos, los resultados variarán dependiendo de la calidad del mallado y de la complejidad del problema como un todo, sin embargo, si se tiene una buena relación de aspecto para los elementos contiguos, y para cada elemento, el resultado debería ser una buena extrapolación de los modelos de deformación sugeridos en los antecedentes.

5.2 ANÁLISIS DE RESULTADOS ESPESADOR

Para el primer ensayo lo primero que se puede hacer notar es el mallado. Dadas las condiciones geométricas, se genera un mallado que es más fino hacia el segundo cono. Los elementos del cilindro mucho más grandes. Esto ocurre porque el algoritmo que define el tamaño de los elementos, hace homogéneos los tamaños dentro de cada sub-estructura, vale decir, para el cono grande los elementos son similares entre sí y lo más cuadrados posibles, así como los del cono pequeño y del cilindro.

En la segunda figura en que se grafican las fuerzas para cada nodo cabe resaltar que estas apuntan en la dirección correcta (había abajo), se puede verificar en el software, que numéricamente los valores son acertados, de acuerdo al tamaño de los elementos. Esto hace que los vectores graficados sean más grandes en los nodos del cilindro, que son mucho más grandes que los del manto, sin embargo, como en los conos hay más elementos, la densidad de esfuerzos se mantiene en el manto.

Numéricamente el error en la sumatoria del peso de todos elementos es despreciable y está relacionada principalmente con la discretización de las figuras cilíndricas por poliedros.

Otro elemento a destacar es que el peso de los elementos de las vigas es despreciable en relación a los elementos del manto, en las figuras no alcanzan a aparecer, sin embargo si se hace inspecciona el vector en el programa, se puede verificar que existen.

En cuanto al estado de deformación, los mayores desplazamientos se encuentran donde no existe intersección con los pilares. Lo cual tiene mucho sentido considerando que los pilares distribuyen la carga hacia los anclajes y aportan rigidez en los puntos donde se cruzan con el manto.

Para el segundo ensayo, el mallado es exactamente el mismo que para el primero, por lo que no recibe comentarios. En la Figura 4.14 se puede ver cómo el esfuerzo es absolutamente radial, lo cual es consistente con la deformación mostrada en la Figura 4.15. De esta imagen cabe destacar que además ocurre un achatamiento en la zona de la fuerza, un desplazamiento de la estructura completa en la dirección de la fuerza, y un perfil en los pilares que indica que está todo bien conectado. El color predominante del manto corresponde al desplazamiento de la estructura como un todo.

En la Figura 4.16 se muestra que las reacciones también tienen una forma consistente de acuerdo a la fuerza aplicada. En el pilar que está justo sobre el esfuerzo, la reacción es de compresión y a medida que los pilares se alejan de ese punto, se vuelven hacia una reacción de tracción. Esto es

coherente con el modelo de deformación que se adoptó, comparable con una viga flexionada, tracción y flexión en caras opuestas.

Por último para este ensayo, el cuadro de reacciones muestra que numéricamente el modelo tiene un error mínimo en la suma de las reacciones en cada dirección. Vale decir, en la dirección x , la suma de las reacciones es casi idéntica a la fuerza aplicada, y en las otras direcciones, la suma distinta de cero se debe posiblemente al error residual que acarrea utilizar un sistema discreto. A pesar de esto, considerando mallado tan grueso, el cuadro de reacciones parece tener un nivel de detalle aceptable.

En el tercer ensayo, nuevamente el mismo mallado que los dos anteriores, la Figura 4.18 la dirección correcta de los esfuerzos, normales a la superficie donde corresponde y una ponderación de las normales en los cambios de dirección del manto. En cuanto a las magnitudes, ocurre lo mismo que para el peso de los elementos, mientras más grande el elemento mayor será su superficie, por lo tanto mayor será el gradiente de presión.

También ocurre lo mismo que con el peso de los elementos, se puede ver que la densidad de vectores crece a medida que los elementos se hacen más pequeños, lo que quiere decir que la densidad de esfuerzos debería ser consistente con el gradiente de presión.

Adicional a esto, cabe destacar que el cálculo de las presiones se hace con la altura promedio de los nodos del elemento. Para los elementos del cilindro esta aproximación es poco precisa, pues existe una diferencia de 5 m verticales entre dos nodos contiguos.

En la Figura 4.19 se ve nuevamente un comportamiento coherente con los reforzamientos de las vigas del manto. En relación al ensayo anterior, la deformación es mayor, lo que quiere decir que el esfuerzo debido a la presión hidrostática, para esta configuración geométrica, hace un aporte relativo superior al peso de los elementos en relación a la deformación generada.

En general estos resultados demuestran que la generación de la malla tiene bastantes aspectos perfectibles, tanto para el cálculo de las fuerzas de presión, como para la relación de aspecto de elementos contiguos. Aun cuando esto signifique que la fineza de la malla aumente considerablemente, lo cual podría traducirse en tiempos más largos de cálculo.

Por otra parte, el cuadro de reacciones parece tener una lógica apropiada y magnitudes que en conjunto parecen ser acertadas, tal vez se podría realizar otro tipo de ensayo para verificar esto con mayor exactitud.

5.3 ANÁLISIS DE SOFTWARE FINAL

Acerca de las mallas generadas, se puede ver que mientras más grande el espesor, más gruesa es la distribución y el tamaño de los elementos. Otro aspecto a considerar es la relación entre las alturas del cilindro y los conos. Mientras alto el cilindro, el número de elementos verticales aumenta, con lo que disminuye la diferencia de tamaños en el cambio de dirección, entre cono y cilindro, lo cual mejorará la calidad de los resultados.

En cuanto a los tiempos de cálculo, el resultado también es esperable, mientras mayor el número de elementos, más se demora en encontrar la solución al sistema de ecuaciones. A partir de los tiempos calculados (considerando que son propios del equipo portátil en que se llevó a cabo el

ensayo), se puede aproximar la relación a una curva cuadrática, sin embargo esta relación podría cambiar dependiendo de la tecnología utilizada.

A pesar de lo anterior, se estima que para unos 10000 elementos, el programa se tardaría aproximadamente una hora en determinar las reacciones. Esto en relación a otros programas es un mal número, sin embargo esto se debe principalmente a que Matlab es un programa que no fue diseñado para resolver problemas de elementos finitos, y no cuenta con algoritmos de optimización para este trabajo.

No obstante, si se aproxima la calidad de la respuesta en comparación con el modelo teórico de los test de pre-validación, y además se considera que la geometría del espesador no contiene mucha complejidad, los resultados obtenidos por el software podrían ser una buena primera aproximación para el cálculo de las reacciones.

Otro elemento a considerar en esta discusión es la forma de los cuadros de reacciones. En las figuras de la Sección 4.4.2, se puede ver que la distribución de las reacciones puede cambiar bastante dependiendo de la geometría del espesador. Por ejemplo, para el caso del espesador pequeño se tiene que las reacciones son muy similares y con una componente horizontal un tanto más marcada que para los otros modelos. Mientras que para el más grande la distribución es creciente hacia el centro del espesador, y para el último caso las magnitudes son menores en el pilar medio (entre el centro y la periferia).

Esto es una muestra de la utilidad del programa en cuanto al número de escenarios posibles y la forma que adquiere la respuesta en cada uno de ellos. Basándose en esa diferencia en la forma de las respuestas, se puede estimar que por otros medios, la metodología de cálculo podría cambiar entre una configuración y otra.

Algo curioso que ocurre en cuanto a la deformación predicha es que tiene una cierta componente tangencial al manto para la mayoría de los casos. Sin embargo, esto es algo intuitivo desde el punto de vista de las ecuaciones mecánicas.

Teóricamente el modelo de deformación que da origen al método de elementos finitos hace uso de la minimización de la energía interna de la estructura (potencial igual a cero). Luego, a partir de esta afirmación, resulta evidente que la forma más fácil de deformar un cilindro no es de forma radial, sino tangencial (como cuando se aplasta una lata).

Esto querría decir que en términos cualitativos, la respuesta tiene una forma que intuitivamente es correcta, sin embargo hacen falta más ensayos para verificarlo con mayor exactitud numérica. Como el objetivo de este trabajo de título no es obtener una respuesta demasiado exacta, sino más bien rápida, la respuesta encontrada por el sistema se considera como suficiente.

Como último elemento, la interfaz gráfica ofrecida por el programa Matlab es más que suficiente para un usuario con habilidades intermedias en uso de computadores. No requiere saber nada de programación y muestra las respuestas tanto de forma gráfica como numérica.

El cuadro de reacciones generado en el archivo Excel junto con la imagen de los nodos de los pilares anclados ofrece una respuesta concreta para el futuro cálculo de los apoyos, que es para lo que fue diseñado este software.

6 CONCLUSIONES

6.1 CONCLUSIONES RESPECTO AL TRABAJO DE TÍTULO

Las siguientes conclusiones fueron elaboradas en base a los objetivos que se pusieron en primer lugar para este trabajo de título:

- El software diseñado cumple con los requisitos expuestos, entrega el cuadro de reacciones de un espesador sujeto a su propio peso y al gradiente de presión debido al fluido que sostiene.
- El tiempo de cálculo del programa es bastante bajo (en promedio unos 10 segundos) en relación al cálculo por otros medios (2-3 semanas según profesor guía).
- Si bien los códigos no se encuentran completamente explicados, la metodología propuesta detalla la teoría en base a la cual fueron escritos. Aparte, la estructura de programación se dividió en clases y funciones bastante compartimentadas, que favorecen la comprensión de los algoritmos. Por último, dichas clases y funciones fueron desarrolladas con el mayor grado de generalidad posible (por el estudiante), lo que permite su uso para otras aplicaciones.
- La validación del programa se realizó en varias instancias, primero de forma numérica para geometrías más sencillas, y luego de forma más cualitativa con la geometría del espesador mismo. Los resultados son bastante positivos y se consideran como satisfactorios.

Este software aún no es apropiado para el uso comercial, todavía tiene aspectos por pulir, sin embargo, como una primera versión es un buen punto de partida para la realización de un software comercial.

En los objetivos se estimó que el tiempo de cálculo de las reacciones mediante métodos clásicos, como puede ser el cálculo estructural, podía tardar hasta tres semanas y se debía hacer para cada espesador por separado. Con esta utilidad, se puede resolver el cálculo para múltiples diseños y tan sólo se tarda unos segundos, lo cual indica que su uso podría ahorrar tiempo y costos en las etapas de ingeniería temprana de un proyecto de minería.

A modo de resumen, se concluye que este trabajo de título cumple con las expectativas propuestas en una primera instancia, siendo su mayor déficit la falta de validación numérica con un mejor nivel de detalle.

6.2 RECOMENDACIONES PARA OTROS PROGRAMADORES (DESARROLLOS POSIBLES)

Tomando en cuenta las dificultades y elementos perfectibles del software, a continuación se ofrecen algunos puntos que no fueron desarrollados y que podrían agregarle un valor significativo a lo actualmente desarrollado:

- Simulación simplificada de esfuerzos debido a actividad tectónica: Uno de los desarrollos más útiles podría constituirlo un añadido de las reacciones debido a movimientos sísmicos. El problema desde el punto de vista mecánico debería incluir las aceleraciones y una matriz de masas, sin embargo si se hace una simulación dinámica, podría tomar tiempos bastante mayores (varias horas), por lo tanto se recomienda hacer un análisis simplificado. Esto quiere decir por ejemplo, tomar las vigas como elemento aislado y hacer la simulación posterior la estimación del modelo estático.
- Algoritmo de homogeneización de la malla: El código propuesto sólo mejora la relación de aspecto dentro de una geometría en particular (cono o cilindro), sin embargo no compara estas formas globalmente. Como recomendación se propone calcular los parámetros de la malla y realizar un trabajo de optimización numérica antes de obtener la última versión de dichos parámetros, sin quitar las restricciones consideradas en este código.
- Algoritmo mejorado de gradiente de presión: Si la malla es suficientemente fina, este ítem no es tan necesario, sin embargo, para agregarle mayor exactitud a la distribución de esfuerzos para elementos grandes, puede ser una buena idea utilizar una función que dentro de la generación del vector global de esfuerzos, aproxime de mejor forma las fuerzas nodales (no tan sólo en base a su función de deformación).
- Condiciones de borde distintas: En este trabajo la condición de borde en los pilares es de anclaje, pero podría no ser así, es decir, el usuario podría seleccionar que tipo de condición de borde quiere en los pilares, la explicación de este procedimiento se encuentra presente en este trabajo en forma bastante precisa.
- Soporte central: Algunos espesadores tienen una estructura de soporte en su centro, algunas veces puede ser un pilar adicional y otras una continuación del cono menor. En términos conceptuales, esto se trata de añadir más elementos al manto o a las vigas del manto. A pesar que las mayores deformaciones son en la periferia, resulta complejo estimar cual es el impacto de estas sub-estructuras en el espesador.

7 BIBLIOGRAFÍA

- [1] G. S. Cáceres Arenas, «Apunte Hidrometalurgia,» Universidad de Atacama.
- [2] G. L. & S. Quek, The finite Element Method - A Practical course, Singapore: Butterworth Heinemann, 2003.
- [3] O. Z. & R. Taylor, The finite element method for solid and structural mechanics, Elsevier Butterworth-Heinemann.
- [4] J. F. & T. Belytschko, A first course in finite elements, Wiley, 2007.
- [5] W.-Y. J. & S.-C. Han, Mathematical Problems in Engineering, 2013.

8 ANEXOS Y APÉNDICES

8.1 ANEXO A: CÓDIGOS

8.1.1 Clase FEM_espesador

```
classdef FEM_espesador
    %Clase que define el sistema de elementos finitos para calculos de un
    %espesador sometido al peso propio de los elementos y el gradiente de
    %presión

    properties
        malla;
        g=9.8;
        esp=0.05;
        matrices_rigidez={};
        rigidez_global;
        copia_rigidez_global;
        esfuerzos_global;
        esf={};
        deformaciones;
        reacciones;
    end

    methods
        function
obj=FEM_espesador(mallado,rho_vigas,rho_manto,rho_fluido,E_vigas,nu_vigas,E_manto,nu_manto,A_viga
,Iz_viga,Iy_viga,J_viga,h_manto)
            tic;
            %mallado : objeto de la clase Espesador
            %rho_vigas : densidad de material de las vigas
            %rho_manto : densidad de material del manto
            %rho_fluido : densidad del fluido
            %E_vigas : Modulo de Young del material de las vigas
            %nu_vigas : Coeficiente de Poisson del material de las vigas
            %E_manto : Modulo de Young del material del manto
            %nu_manto : Coeficiente de Poisson del material del manto
            %A_viga : Área transversal del manto
            %Iz : Segundo momento de área según z de las vigas
            %Iy : Segundo momento de área según y de las vigas
            %J_viga : Momento de torsión para vigas
            %h_manto : Espesor del manto

            %Mallado del espesador #####

            obj.malla=mallado;
            tipo=1;

            Ndof=6;
            N=Ndof*length(obj.malla.nodos);

            obj.rigidez_global=zeros(N,N);
            obj.esfuerzos_global=zeros(1,N);

            %rigideces del manto #####
            k=1;
            while(tipo==1)
                elem_aux=obj.malla.elementos{k};

                if(length(elem_aux)~=4)
                    tipo=2;
                    continue;
                end
                xs=zeros(1,4);ys=zeros(1,4);zs=zeros(1,4);
                for j=1:4
                    nodo=obj.malla.nodos{elem_aux(j)};
                    xs(j)=nodo(1);
                    ys(j)=nodo(2);
                    zs(j)=nodo(3);
                end
            end
        end
    end
end
```



```

h_elem=mean(zs);
h=obj.malla.h_total-h_elem;
presion=rho_fluido*obj.g*h;

[rigidez,esfuerzos]=RE_shell(xs,ys,zs,E_manto,nu_manto,h_manto,[0,0,presion],rho_manto);

rigidez_global_aux=zeros(Ndof*length(obj.malla.nodos));
contrib=contribucion2(elem_aux,size(rigidez),size(rigidez_global_aux));
rigidez_global_aux(contrib)=rigidez(1:end);
obj.rigidez_global=obj.rigidez_global+rigidez_global_aux;
%obj.rigidez_global=ensamble2(obj.rigidez_global,rigidez,contribucion(elem_aux));

for j=1:4
    aux=obj.esfuerzos_global(6*(elem_aux(j)-1)+1:6*(elem_aux(j)-1)+6);
    obj.esfuerzos_global(6*(elem_aux(j)-1)+1:6*(elem_aux(j)-
1)+6)=aux+esfuerzos(6*(j-1)+1:6*(j-1)+6)';
end
k=k+1;
end

%rigideces de los pilares #####
while(tipo==2)
    if(k>length(obj.malla.elementos))
        tipo=0;
        continue;
    end
    elem_aux=obj.malla.elementos{k};
    xs=zeros(1,4);ys=zeros(1,4);zs=zeros(1,4);
    for j=1:2
        nodo=obj.malla.nodos{elem_aux(j)};
        xs(j)=nodo(1);
        ys(j)=nodo(2);
        zs(j)=nodo(3);
    end

rigidez=rigidez_frame_express2(xs,ys,zs,E_vigas,nu_vigas,A_viga,Iz_viga,Iy_viga,J_viga);

rigidez_global_aux=zeros(Ndof*length(obj.malla.nodos));
contrib=contribucion2(elem_aux,size(rigidez),size(rigidez_global_aux));
rigidez_global_aux(contrib)=rigidez(1:end);
obj.rigidez_global=obj.rigidez_global+rigidez_global_aux;

l = pdist([xs(1),ys(1),zs(1);xs(2),ys(2),zs(2)], 'euclidean');
peso_elem=l*A_viga*rho_vigas*obj.g;
peso_nodo=[0,0,-peso_elem/2,0,0,0];

for j=1:2
    aux=obj.esfuerzos_global(6*(elem_aux(j)-1)+1:6*(elem_aux(j)-1)+6)+peso_nodo;
    obj.esfuerzos_global(6*(elem_aux(j)-1)+1:6*(elem_aux(j)-1)+6)=aux;
end
k=k+1;
end
%condiciones de borde #####
obj.copia_rigidez_global=obj.rigidez_global;
for i=1:length(obj.malla.nodos_anclados)
    aux_dofs=obj.dofs(obj.malla.nodos_anclados(i));
    aux_esfuerzos=obj.esfuerzos_global(aux_dofs);
    obj.esfuerzos_global(aux_dofs)=0;

    aux_rigideces=obj.rigidez_global(aux_dofs,aux_dofs);
    obj.rigidez_global(aux_dofs,:)=0;
    obj.rigidez_global(:,aux_dofs)=0;
    for j=1:length(aux_dofs)
        obj.rigidez_global(aux_dofs(j),aux_dofs(j))=1;
    end
end
obj.deformaciones=obj.rigidez_global\obj.esfuerzos_global';

obj.reacciones=obj.copia_rigidez_global*obj.deformaciones;

for i=1:length(obj.malla.nodos)
    aux=obj.deformaciones((i-1)*6+1:(i-1)*6+6);

```

```

        aux_r=obj.reacciones((i-1)*6+1:(i-1)*6+6);
        obj.malla.deformaciones{i}=aux;
        obj.malla.reacciones{i}=aux_r;
        obj.malla.modulo_deformaciones(i)=norm(obj.deformaciones((i-1)*6+1:(i-1)*6+3));
    end
    obj.malla.def_max=max(abs(obj.malla.modulo_deformaciones));
    obj.malla.def_min=min(abs(obj.malla.modulo_deformaciones));
    toc;
end
function plotesfuerzos(obj)
    %función que grafica las fuerzas de cada nodo en el espacio
    obj.malla.plotelements;
    hold on
    for i=1:length(obj.malla.nodos)
        nodo=obj.malla.nodos{i};
        fuerza=obj.esfuerzos_global(6*(i-1)+1:6*(i-1)+3);
quiver3(nodo(1),nodo(2),nodo(3),fuerza(1)/100000,fuerza(2)/100000,fuerza(3)/100000);
    end
    hold off
end
function resp=dofs(obj,nodo)
    %función que devuelve los grados de libertad de la matriz
    %global de rigidez o de esfuerzos asociados a un número de nodo en particular
    N_dof=6;
    resp=zeros(1,N_dof);
    for i=1:N_dof
        resp(i)=N_dof*nodo-(N_dof-i);
    end
end
end
end
end
end

```

8.1.2 Función Contribución

```

function [ resp ] = contribucion( contrib , porte, porte2)
    %Posición en matriz global #####
    mat_nodos=zeros(porte(1)*porte(2),1);
    N_dof=6;
    for nodo_i = 1:length(contrib)
        for nodo_j = 1:length(contrib)
            for dof_i = 1:N_dof
                g_row = N_dof*contrib(nodo_i)-(N_dof-dof_i);
                l_row= N_dof*nodo_i-(N_dof-dof_i);
                for dof_j = 1:N_dof
                    g_col = N_dof*contrib(nodo_j)-(N_dof-dof_j);
                    l_col= N_dof*nodo_j-(N_dof-dof_j);
                    mat_nodos(l_row+(l_col-1)*porte(1))=g_row+(g_col-1)*porte2(1);
                end
            end
        end
    end
    resp=mat_nodos;
end

```

8.1.3 Función RE_Shell

```

function [rigidez,esf] = RE_shell(x_e,y_e,z_e,E,nu,h,f,rho)
    ka=pi^2/12;
    chis=[-1,1,1,-1];
    etas=[-1,-1,1,1];
    chis2=[-1/3^0.5,1/3^0.5];
    etas2=[-1/3^0.5,1/3^0.5];
    w=[1,1];
    N_dof=6;
    % matrices del material #####
    G = E/(2*(1+nu));
    matriz_Cs=[G,0;0,G];
    matriz_C=E/(1-nu^2)*[1,nu,0;nu,1,0;0,0,(1-nu)/2];
    %largo y ancho, coordenadas locales #####
    a = pdist([x_e(1),y_e(1),z_e(1);x_e(2),y_e(2),z_e(2)], 'euclidean')/2;
    b = pdist([x_e(2),y_e(2),z_e(2);x_e(3),y_e(3),z_e(3)], 'euclidean')/2;

```

```

c = pdist([x_e(3),y_e(3),z_e(3);x_e(4),y_e(4),z_e(4)], 'euclidean')/2;
d = (((2*b)^2-((2*a-2*c)/2)^2)^.5)/2;
x=[-a,a,c,-c];
y=[-d,-d,d,d];
T32=mat_T32(x_e,y_e,z_e);

peso_global=[0;0;-polyarea(x,y)*h*rho];
peso_local=inv(T32)*peso_global;
f=f+peso_local''';

matriz_flexion_normal=zeros(12,12);
matriz_flexion_corte=zeros(12,12);
matriz_planar=zeros(8,8);
matriz_final=zeros(24,24);

% INTEGRACIÓN NUMÉRICA #####
esf=zeros(24,1);

for i=1:2 %iteración en chi
    for j=1:2 %iteración en eta

        dxdchi=zeros(4,1);
        dxdeta=zeros(4,1);
        dydchi=zeros(4,1);
        dydeta=zeros(4,1);

        for k=1:4
            dxdchi(k)=1/4*chis(k)*(1+etas(k)*etas2(j))*x(k);
            dydchi(k)=1/4*chis(k)*(1+etas(k)*etas2(j))*y(k);
            dxdeta(k)=1/4*etas(k)*(1+chis(k)*chis2(i))*x(k);
            dydeta(k)=1/4*etas(k)*(1+chis(k)*chis2(i))*y(k);
        end
        mat_jacob=[sum(dxdchi),sum(dydchi);sum(dxdeta),sum(dydeta)];
        det_jacob=det(mat_jacob);
        inv_jacob=1/det_jacob*[sum(dydeta),-sum(dxdeta);-sum(dydchi),sum(dxdchi)];

        N1=(1-chis2(i))*(1-etas2(j));
        N2=(1+chis2(i))*(1-etas2(j));
        N3=(1+chis2(i))*(1+etas2(j));
        N4=(1-chis2(i))*(1+etas2(j));

        N=1/4*[
            N1,0,0,0,0,0,N2,0,0,0,0,0,N3,0,0,0,0,0,N4,0,0,0,0,0;
            0,N1,0,0,0,0,0,N2,0,0,0,0,0,N3,0,0,0,0,0,N4,0,0,0,0,0;
            0,0,N1,0,0,0,0,0,N2,0,0,0,0,0,N3,0,0,0,0,0,N4,0,0,0,0,0;
            0,0,0,N1,0,0,0,0,0,N2,0,0,0,0,0,N3,0,0,0,0,0,N4,0,0,0,0,0;
            0,0,0,0,N1,0,0,0,0,0,N2,0,0,0,0,0,N3,0,0,0,0,0,N4,0,0,0,0,0;
            0,0,0,0,0,N1,0,0,0,0,0,N2,0,0,0,0,0,N3,0,0,0,0,0,N4];
    ];

    BI=zeros(3,12);
    for k=1:4
        dN=inv_jacob*[1/4*chis(k)*(1+etas(k)*etas2(j));1/4*etas(k)*(1+chis(k)*chis2(i))];
        dNdx=dN(1);dNdy=dN(2);

        BI(1:3,(k-1)*3+1:k*3)=[
            0,0,-dNdx;
            0,dNdy,0;
            0,dNdx,-dNdy
        ];
        BI(1:3,(k-1)*2+1:k*2)=[
            dNdx,0;
            0,dNdy;
            dNdy,dNdx;
        ];
    end

    esf=esf+N'*[f(1),f(2),f(3),0,0,0]*det_jacob;

matriz_flexion_normal=matriz_flexion_normal+w(i)*w(j)*(h^3/12)*(BI'*matriz_C*BI)*det_jacob;
matriz_planar=matriz_planar+w(i)*w(j)*polyarea(x,y)*h*B'*matriz_C*B*det_jacob;
end
end
end

```

```

dxdchi=zeros(4,1);
dxdeta=zeros(4,1);
dydchi=zeros(4,1);
dydeta=zeros(4,1);

for k=1:4
    dxdchi(k)=1/4*chis(k)*(1+etas(k)*0)*x(k);
    dydchi(k)=1/4*chis(k)*(1+etas(k)*0)*y(k);
    dxdeta(k)=1/4*etas(k)*(1+chis(k)*0)*x(k);
    dydeta(k)=1/4*etas(k)*(1+chis(k)*0)*y(k);
end

mat_jacob=[sum(dxdchi),sum(dydchi);sum(dxdeta),sum(dydeta)];
det_jacob=det(mat_jacob);
inv_jacob=1/det_jacob*[sum(dydeta),-sum(dxdeta);-sum(dydchi),sum(dxdchi)];

BO=zeros(2,12);
for k=1:4
    dN=inv_jacob*[1/4*chis(k)*(1+etas(k)*0);1/4*etas(k)*(1+chis(k)*0)];
    dNdx=dN(1);dNdy=dN(2);
    BO(1:2,(k-1)*3+1:k*3)=[
        dNdx,0,1/4*(1+chis(k)*0)*(1+etas(k)*0);
        dNdy,-1/4*(1+chis(k)*0)*(1+etas(k)*0),0;
    ];
end

matriz_flexion_corte=matriz_flexion_corte+(2*2)*(ka*h)*(BO'*matriz-Cs*BO)*det_jacob;
%flexion_corte=matriz_flexion_corte;
matriz_flexion=matriz_flexion_normal+matriz_flexion_corte;

% Ensemble matriz elemento #####

for i=1:4
    for j=1:4
        matriz_final(N_dof*(i-1)+1:N_dof*(i-1)+2,N_dof*(j-1)+1:N_dof*(j-1)+2)=matriz_planar(2*(i-1)+1:2*(i-1)+2,2*(j-1)+1:2*(j-1)+2);
        matriz_final(N_dof*(i-1)+3:N_dof*(i-1)+5,N_dof*(j-1)+3:N_dof*(j-1)+5)=matriz_flexion(3*(i-1)+1:3*(i-1)+3,3*(j-1)+1:3*(j-1)+3);
    end
end

for i=1:4
    matriz_final(6*i,6*i)=1;
end

%Obtención de matriz de transformación y matriz en coordenadas
for i=1:8
    trans2(3*(i-1)+1:3*(i-1)+3,3*(i-1)+1:3*(i-1)+3)=T32;
end

esf=trans2'*esf;
rigidez=trans2'*matriz_final*trans2;

end

```

8.1.4 Función R_Frame

```

function rigidez = R_frame(x_e,y_e,z_e,E,nu,A,Iz,Iy,J)
a = pdist([x_e(1),y_e(1),z_e(1);x_e(2),y_e(2),z_e(2)], 'euclidean')/2;
G = E/(2*(1+nu));
aux=[
A*E/(2*a),0,0,0,0,0,-A*E/(2*a),0,0,0,0,0;
0,3*E*Iz/(2*a^3),0,0,0,3*E*Iz/(2*a^2),0,-3*E*Iz/(2*a^3),0,0,0,3*E*Iz/(2*a^2);
0,0,3*E*Iy/(2*a^3),0,-3*E*Iy/(2*a^2),0,0,0,-3*E*Iy/(2*a^3),0,-3*E*Iy/(2*a^2),0;
0,0,0,G*J/(2*a),0,0,0,0,-G*J/(2*a),0,0;
0,0,0,0,2*E*Iy/a,0,0,0,3*E*Iy/(2*a^2),0,E*Iy/a,0;
0,0,0,0,0,2*E*Iz/a,0,-3*E*Iz/(2*a^2),0,0,0,E*Iz/a;
0,0,0,0,0,0,A*E/(2*a),0,0,0,0,0;
0,0,0,0,0,0,0,3*E*Iz/(2*a^3),0,0,0,-3*E*Iz/(2*a^2);
0,0,0,0,0,0,0,0,3*E*Iy/(2*a^3),0,3*E*Iy/(2*a^2),0;
0,0,0,0,0,0,0,0,0,0,G*J/(2*a),0,0;
0,0,0,0,0,0,0,0,0,0,2*E*Iy/a,0;
0,0,0,0,0,0,0,0,0,0,0,2*E*Iz/a
];

```

```

];
for j=1:size(aux,2)
    for k=j:size(aux,1)
        aux(k,j)=aux(j,k);
    end
end
xt=x_e;yt=y_e;zt=z_e;
xt(4)=0;
yt(4)=0;
zt(4)=z_e(1);
T3=mat_T32(xt,yt,zt);
trans=zeros(12);
for i=1:4
    trans(3*(i-1)+1:3*(i-1)+3,3*(i-1)+1:3*(i-1)+3)=T3;
end
rigidez = trans'*aux*trans;
end

```

8.1.5 Función Mat_T3

```

function resp=mat_T3(X1,Y1,Z1)
    x=[1 0 0];
    y=[0 1 0];
    z=[0 0 1];
    v1=[X1(1),Y1(1),Z1(1)];
    v2=[X1(2),Y1(2),Z1(2)];
    v3=[X1(4),Y1(4),Z1(4)];
    x_uni=(v2-v1)/norm(v2-v1);
    y_uni=(v3-v1)/norm(v3-v1);
    z_uni=cross(v2-v1,v3-v1)/norm(cross(v2-v1,v3-v1));
    resp=[
        dot(x_uni,x), dot(x_uni,y), dot(x_uni,z);
        dot(y_uni,x), dot(y_uni,y), dot(y_uni,z);
        dot(z_uni,x), dot(z_uni,y), dot(z_uni,z);
    ];
end

```

8.1.6 Clase Espesador

```

classdef espesador
    %Clase que genera un mallado y grafica diversas características y
    %funciones de un espesador
    properties
        nodos={};
        elementos={};
        nr;
        nz;
        nz_cil;
        nz_con1;
        nz_con2;
        nz_pilares;
        h_total;
        h_manto;
        radios;
        alturas;
        conexiones;
        nodos_anclados=[];
        deformaciones={};
        modulo_deformaciones=[];
        def_max;
        def_min;
        nodos_vigas;
        reacciones;
    end
    methods
        function obj=espesador2(hcil,hcon1,hcon2,despl,r1,r2,r3,nsep,distancias)
            %Función generadora de la malla #####
            n=20;
            obj.h_manto=hcil+hcon1+hcon2;
            while rem(n,nsep)~=0
                n=n+1;
            end
            obj.nr=n;
        end
    end
end

```

```

%obj.conexiones=zeros(1,nsep*length(distancias));
obj.h_total=hcil+hcon1+hcon2+despl;

%Cilindro #####
obj.nz_cil=(ceil(hcil/(2*pi*r1/obj.nr))+1);
obj.radios(1:obj.nz_cil)=r1;
obj.alturas(1:obj.nz_cil)=linspace(obj.h_total,obj.h_total-hcil,obj.nz_cil);

%Cono1 #####
aux=(ceil(hcon1/(2*pi*((r1+r2)/2)/obj.nr))+1)*2;
if length(distancias)>aux
    obj.nz_con1=length(distancias)*2;
else
    obj.nz_con1=aux;
end
aux=homogen2(obj.nz_con1,r1,r2,distancias);
obj.nz_con1=length(aux);
pri=obj.nz_cil;
ult=obj.nz_cil+obj.nz_con1-1;
ang=atan(hcon1/(r1-r2));
obj.radios(pri:ult)=aux;
obj.alturas(pri:ult)=obj.radios(pri:ult)*tan(ang)-r2*tan(ang)+despl+hcon2;

%Cono2 #####
obj.nz_con2=(ceil(hcon2/(2*pi*((r2+r3)/2)/obj.nr))+1);
pri=obj.nz_cil+obj.nz_con1-1;
ult=obj.nz_cil+obj.nz_con1+obj.nz_con2;
obj.radios(pri:ult)=linspace(r2,r3,obj.nz_con2+2);
obj.alturas(pri:ult)=linspace(obj.h_total-hcil-hcon1,obj.h_total-hcil-hcon1-
hcon2,obj.nz_con2+2);

obj.nz=obj.nz_cil+obj.nz_con1+obj.nz_con2;

angulos_aux=linspace(0,2*pi,nsep+1);

%generación nodos del manto #####
for i=1:length(obj.alturas)
    for a=linspace(0,2*pi,obj.nr+1)
        if a~=2*pi

            x=obj.radios(i)*cos(a);
            y=obj.radios(i)*sin(a);
            z=obj.alturas(i);
            if i~=1
                if (obj.alturas(i)~=obj.alturas(i-1))
                    obj.nodos{length(obj.nodos)+1,1}=[x,y,z];
                end
            else
                obj.nodos{length(obj.nodos)+1,1}=[x,y,z];
            end
            % Encontrar los nodos donde están conectados los
            % pilares y los nodos de las vigas

            if any(angulos_aux==a)
                obj.nodos_vigas(end+1)=length(obj.nodos);
                if any(distancias==round2(r1-obj.radios(i),0.0001))~=0 &&
obj.alturas(i)<=obj.h_total-hcil
                    obj.conexiones(end+1)=length(obj.nodos);
                end
            end
        end
    end
end
end
%generación de los elementos del manto #####
for i=linspace(1,obj.nz-1,obj.nz-1) %recorro en vertical hasta la penúltima fila de
puntos
    for j=linspace(1,obj.nr,obj.nr) %recorro en horizontal hasta el penúltimo punto
de la fila
        indice=(i-1)*obj.nr+j; %índice del punto
        %encontramos los índices de los puntos que forman al
        %elemento
        derecha=obj.derecha(indice); derecha=derecha(1);
        abajo=obj.abajo(indice); abajo=abajo(1);
    end
end
end

```

```

        abajo_derecha=obj.abajo(derecha);abajo_derecha=abajo_derecha(1);
        elemento=[abajo,abajo_derecha,derecha,indice];
        %indexamos el elemento y sus respectivos puntos
        obj.elementos{end+1,1}=elemento;
    end
end
%generación nodos_vigas #####
for i=1:length(obj.nodos_vigas)
    if obj.abajo(obj.nodos_vigas(i))~-=-1
        abajo=obj.abajo(obj.nodos_vigas(i));abajo=abajo(1);
        elemento=[obj.nodos_vigas(i), abajo];
        obj.elementos{end+1,1}=elemento;
    end
end

%generación nodos de los pilares #####
%obj.nodos_anclados=zeros(nsep*length(distancias),1);
for i=1:length(obj.conexiones)
    nodo_aux=obj.nodos{obj.conexiones(1,i)};
    nz_aux=round(nodo_aux(3)/((2*pi*((r2+r3)/2)/obj.nr));
    viga_aux=zeros(nz_aux+1,3);
    viga_aux(:,1)=nodo_aux(1);
    viga_aux(:,2)=nodo_aux(2);
    viga_aux(:,3)=linspace(nodo_aux(3),0,nz_aux+1);
    for j=1:length(viga_aux)
        if j==1
            obj.elementos{end+1}=[obj.conexiones(i),length(obj.nodos)+1];
        end
        if j~=1
            obj.nodos{end+1}=viga_aux(j,:);
            if viga_aux(j,3)~=0
                obj.elementos{end+1}=[length(obj.nodos),length(obj.nodos)+1];
            else
                obj.nodos_anclados(end+1)=length(obj.nodos);
            end
        end
    end
end
end

function num=fila(obj,point)
% Devuelve la fila del 'point-ésimo' punto
num=floor((point-1)/(obj.nr))+1;
end

function num=izquierda(obj,point)
% Devuelve el punto a la izquierda del 'point-ésimo' punto
fila=obj.fila(point);
if(point-1>(fila-1)*obj.nr)
    num=[point-1,obj.nodos{point-1}];
else
    num=[point+obj.nr-1,obj.nodos{point+obj.nr-1}];
end

end

function num=derecha(obj,point)
% Devuelve el punto a la derecha del 'point-ésimo' punto
fila=obj.fila(point);
if(point+1<(fila)*obj.nr+1)
    num=[point+1,obj.nodos{point+1}];
else
    num=[point-obj.nr+1,obj.nodos{point-obj.nr+1}];
end

end

function num=arriba(obj,point)
fila=obj.fila(point);
if(fila>1)
    num=[point-obj.nr, obj.nodos{point-obj.nr}];
else
    num=-1;
end
end

```

```

end

function num=abajo(obj,point)
    fila=obj.fila(point);
    if(fila<obj.nz)
        num=[point+obj.nr, obj.nodos{point+obj.nr}];
    else
        num=-1;
    end
end
function plot(obj)
    aux=cell2mat(obj.nodos);
    x=aux(:,1);
    y=aux(:,2);
    z=aux(:,3);
    scatter3(x,y,z, '.');
end
function plotelement(obj,elem)
    %grafica la posición y los nodos de un elemento
    obj.plot();
    malla=obj.elementos;
    elemento=malla{elem};
    aux={};
    hold on;
    for i=1:length(elemento)
        punto=obj.nodos{elemento(i)};
        aux{i,1}=punto;
        scatter3(punto(1),punto(2),punto(3), 'fill', 'MarkerFaceColor', [.75 0 0]);
    end
    c=cell2mat(aux);
    fill3(c(:,1),c(:,2),c(:,3),1);
    hold off;
end
function plotelements(obj)
    %grafica todos los elementos del espesador
    malla=obj.elementos;
    hold on;
    axis equal;
    view([1 1 1]);
    set(gcf, 'Renderer', 'zbuffer');
    angulo=linspace(0,2*pi,720);
    aux2=campos;
    radio=(aux2(1)^2+aux2(2)^2)^.5*20;
    j=1;
    for elem=linspace(1,length(malla),length(malla))
        campos([radio*cos(angulo(j)),radio*sin(angulo(j)),100]);
        elemento=malla{elem};
        aux={};
        for i=linspace(1,length(elemento),length(elemento))
            punto=obj.nodos{elemento(i)};
            aux{i,1}=punto;
        end
        c=cell2mat(aux);
        plot3(c(:,1),c(:,2),c(:,3));
        j=j+1;
        if j==length(angulo)
            j=1;
        end
    end
    hold off;
end
function plot_vistas_def(obj,ampliacion)
    %grafica 4 vistas de la deformación, requiere la alimentación
    %de la clase FEM_espesador
    h1=subplot(1,3,1);
    obj.plot_deformacion(2,ampliacion);
    hold on;
    obj.plot_deformacion(0,ampliacion);
    axis 'equal';
    view([1,1,1]);
    h2=subplot(1,3,2);
    obj.plot_deformacion(2,ampliacion);
    hold on;
    obj.plot_deformacion(0,ampliacion);

```



```

view([0,-1,0]);
axis 'equal';
h3=subplot(1,3,3);
obj.plot_deformacion(2,ampliacion);
hold on;
obj.plot_deformacion(0,ampliacion);
view([0,0,1]);
axis 'equal';
xlim('manual');ylim('manual');
%xlim([-max(obj.radios), max(obj.radios)]);
%ylim([-max(obj.radios), max(obj.radios)]);
end
function plot_vistas_malla(obj)
%grafica 3 vistas de el mallado del espesor
h1=subplot(1,3,1);
obj.plotelements;
axis 'equal';
view([1,1,1]);
h2=subplot(1,3,2);
obj.plotelements;
view([0,1,0]);
axis 'equal';
h3=subplot(1,3,3);
obj.plotelements;
view([0,0,1]);
axis 'equal';
xlim('manual');ylim('manual');
xlim([-max(obj.radios), max(obj.radios)]);
ylim([-max(obj.radios), max(obj.radios)]);
%axis 'equal';
end
function plot_reacciones(obj)
%grafica las reacciones en los pilares del espesor
obj.plotelements;
hold on;
factor=max(max(cell2mat(obj.reacciones))/max(obj.radios));
for i=1:length(obj.nodos_anclados)
    indice=obj.nodos_anclados(i);
    punto=obj.nodos{indice};
    reaccion=obj.reacciones{indice};
    %aux{i,1}=punto;
    %aux_def{i,1}=deformacion'';
quiver3(punto(1),punto(2),punto(3),reaccion(1)/factor,reaccion(2)/factor,reaccion(3)/factor,'color
r','red','linewidth',2);
end
hold off;
end
function plot_nodos_pilares(obj)
%grafica los nodos anclados de los pilares del espesor y sus
%respectivos números nodales
hold on;
for i=1:length(obj.nodos_anclados)
    indice=obj.nodos_anclados(i);
    punto=obj.nodos{indice};
    scatter3(punto(1),punto(2),punto(3),'+');
    text(punto(1),punto(2),punto(3),{indice},'FontSize',18);
end
plot(0,0,'k+-');
hold off;
view([0 0 1]);
end
function plot_deformacion(obj,opc,factor)
%grafica la deformación del espesor, el factor es una
%ampliación de la deformación. Opc indica que tipo de gráfico
%crea la función, 0: tipo malla, 1:flechas de desplazamiento,
%2: mapa de colores.
malla=obj.elementos;
hold on;
axis equal;
view([1 1 1]);
set(gcf,'Renderer','zbuffer');
colorbar;
%set(gcf,'CLim',[obj.def_min obj.def_max]);

```

```

angulo=linspace(0,2*pi,720);
aux2=campos;
radio=(aux2(1)^2+aux2(2)^2)^.5*20;
j=1;
for elem=linspace(1,length(malla),length(malla))
    campos([radio*cos(angulo(j)),radio*sin(angulo(j)),100]);
    elemento=malla{elem};
    aux={};
    aux_def={};
    for i=1:length(elemento)
        punto=obj.nodos{elemento(i)};
        deformacion=obj.deformaciones{elemento(i)};deformacion=deformacion(1:3);
        aux{i,1}=punto;
        aux_def{i,1}=deformacion'';
        if opc==1
            quiver3(punto(1),punto(2),punto(3),deformacion(1)*100,deformacion(2)*100,deformacion(3)*100);
        end
    end
    c=cell2mat(aux);
    if opc==0
        c2=cell2mat(aux_def);
        c=c+c2*factor;
    end
    if opc==2
        c2=cell2mat(aux_def);
        c=c+c2*factor;
        if(length(elemento)==4)
            c21=norm(c2(1,:));
            c22=norm(c2(2,:));
            c23=norm(c2(3,:));
            c24=norm(c2(4,:));
        end
        if(length(elemento)==2)
            c21=norm(c2(1,:));
            c22=norm(c2(2,:));
        end
        if(length(elemento)==4)
            fill3(c(:,1),c(:,2),c(:,3),[c21 c22 c23 c24],'EdgeColor','flat');
        else
            color_line3(c(:,1),c(:,2),c(:,3),[c21 c22]);
        end
    else
        plot3(c(:,1),c(:,2),c(:,3));
    end
    j=j+1;
    if j==length(angulo)
        j=1;
    end
end
if opc==2
    obj.plot_deformacion(0, factor);
end
hold off;
end
end
end
end

```

8.1.7 Función Homogen

```

function resp = homogen(nr,r1,r2,puntos)
    resp=[];
    for i=1:length(puntos)
        if i==1
            aux=linspace(r1,r1-puntos(i),round(nr/(r1-r2)*puntos(i))+1);
            resp(1:length(aux))=aux;
        else
            dx=puntos(i)-puntos(i-1);
            aux=linspace(r1-puntos(i-1),r1-puntos(i),round(nr/(r1-r2)*dx)+1);
            resp(end:end+length(aux)-1)=aux;
        end
    end
end

```

```

end
if resp(end)>(r2)
    dx=(r1-r2)-puntos(end);
    aux=linspace(resp(end),r2,round(nr/(r1-r2)*dx)+1);
    if length(aux)<2
        aux=linspace(r1-puntos(end),r2,round(nr/(r1-r2)*dx)+2);
    end
    resp(end:end+length(aux)-1)=aux;
end
end
end

```

8.1.8 Clase FEM_barra_Shell_horizontal

```

classdef FEM_barra_Shell_horizontal
    %Clase que genera un sistema de elementos finitos tipo Shell para una viga
    %empotrada en voladizo con esfuerzo aplicado en su punta

    properties
        nodos={};
        elementos={};
        matrices={};
        rigideces={};
        E_acero = 210*10^9; %Módulo de young: N/m2
        nu_acero = 0.3; %Módulo de poisson: sin unidad
        orden={};
        rigidez_global;
        rigidez_global_copia;
        esfuerzos;
        deformacion;
        deformacion2;
        reacciones;
    end

    methods
        function obj = FEM_barra_Shell_horizontal(largo,ancho,alto,n_x,n_y,fuerza)
            j=1;
            n_nodos=n_x*n_y;
            n_elementos=(n_x-1)*(n_y-1);
            for y_i=linspace(0,ancho,n_y)
                for x_i=linspace(0,largo,n_x)
                    obj.nodos{j}=[x_i y_i 0];
                    j=j+1;
                end
            end

            for i=1:n_elementos
                n_fila=ceil(i/(n_x-1));
                obj.elementos{i}=[i+(n_fila-1) i+(n_fila) i+n_x+n_fila i+n_x+n_fila-1];
            end
            Ndof=6;
            obj.rigidez_global=zeros(Ndof*length(obj.nodos));
            obj.esfuerzos=zeros(Ndof*length(obj.nodos),1);
            for i=1:length(obj.elementos)
                xs=zeros(1,4);ys=zeros(1,4);zs=zeros(1,4);
                elem=obj.elementos{i};
                for j=1:4;
                    nodo=obj.nodos{elem(j)};
                    xs(j)=nodo(1);
                    ys(j)=nodo(2);
                    zs(j)=nodo(3);
                end
            end

            [rigidez,esfuerzo]=RE_shell(xs,ys,zs,obj.E_acero,obj.nu_acero,alto,[0,0,0],0);
            end
            rigidez_global_aux=zeros(Ndof*length(obj.nodos));
            contrib=contribucion2(elem,size(rigidez),size(rigidez_global_aux));
            rigidez_global_aux(contrib)=rigidez(1:end);
            obj.rigidez_global=obj.rigidez_global+rigidez_global_aux;
            %obj.esfuerzos=esfuerzo;
            for j=1:4
                esf_auxiliar=obj.esfuerzos(6*(elem(j)-1)+1:6*(elem(j)-1)+6);
                obj.esfuerzos(6*(elem(j)-1)+1:6*(elem(j)-1)+6)=esf_auxiliar+esfuerzo(6*(j-
1)+1:6*(j-1)+6);
            end
        end
    end
end

```

```

        end
end
n_dof=6;
obj.rigidez_global_copia=obj.rigidez_global;

for i=1:n_y
    %esfuerzos en la punta libre
    degrees=obj.dofs(i*n_x);
    obj.esfuerzos(degrees(3))=fuerza/n_y;
    %anclaje a la pared
    degrees2=obj.dofs(1+n_x*(i-1));
    obj.rigidez_global(degrees2,:)=0;
    obj.rigidez_global(:,degrees2)=0;
    for j=1:length(degrees2)
        obj.rigidez_global(degrees2(j),degrees2(j))=1;
    end
end

obj.deformacion=obj.rigidez_global\obj.esfuerzos;
obj.reacciones=obj.rigidez_global_copia*obj.deformacion;
for i=1:length(obj.nodos)
    aux=obj.deformacion((i-1)*6+1:(i-1)*6+6);
    obj.deformacion2{i}=aux;
end
end
function resp=dofs(obj,nodo)
    N_dof=6;
    resp=zeros(1,N_dof);
    for i=1:N_dof
        resp(i)=N_dof*nodo-(N_dof-i);
    end
end
function plotelements(obj)
    malla=obj.elementos;
    hold on;
    for elem=linspace(1,length(malla),length(malla))
        elemento=malla{elem};
        aux={};
        for i=linspace(1,length(elemento),length(elemento))
            punto=obj.nodos{elemento(i)};
            aux{i,1}=punto;
        end
        c=cell2mat(aux);
        plot3(c(:,1),c(:,2),c(:,3));
    end
    pbaspect('manual');
    axis equal;
    hold off;
end
function plotesfuerzos(obj)
    obj.plotelements;
    hold on
    for i=1:length(obj.nodos)
        nodo=obj.nodos{i};
        fuerza=obj.esfuerzos(6*(i-1)+1:6*(i-1)+3);
        quiver3(nodo(1),nodo(2),nodo(3),fuerza(1)/1000,fuerza(2)/1000,fuerza(3)/1000);
    end
    hold off
end
function plot_deformacion(obj,opc,factor)
    malla=obj.elementos;
    hold on;
    axis equal;
    view([1 1 1]);
    set(gcf,'Renderer','zbuffer');
    colorbar;
    angulo=linspace(0,2*pi,720);
    aux2=campos;
    radio=(aux2(1)^2+aux2(2)^2)^.5*20;
    j=1;
    for elem=linspace(1,length(malla),length(malla))
        campos([radio*cos(angulo(j)),radio*sin(angulo(j)),100]);
        elemento=malla{elem};
        aux={};
    end
end

```

```

        aux_def={};
        for i=1:length(elemento)
            punto=obj.nodos{elemento(i)};
            deformacion=obj.deformacion2{elemento(i)};deformacion=deformacion(1:3);
            aux{i,1}=punto;
            aux_def{i,1}=deformacion'';
            if opc==1
                quiver3(punto(1),punto(2),punto(3),deformacion(1)*100,deformacion(2)*100,deformacion(3)*100);
            end
        end
        c=cell2mat(aux);
        if opc==0
            c2=cell2mat(aux_def);
            c=c+c2*factor;
        end
        if opc==2
            c2=cell2mat(aux_def);
            c=c+c2*factor;
            if(length(elemento)==4)
                c21=norm(c2(1,:));
                c22=norm(c2(2,:));
                c23=norm(c2(3,:));
                c24=norm(c2(4,:));
            end
        end
        if(length(elemento)==4)
            fill3(c(:,1),c(:,2),c(:,3),[c21 c22 c23 c24],'EdgeColor','flat');
        end
        j=j+1;
        if j==length(angulo)
            j=1;
        end
    end
    hold off;
end
end
end

```

8.1.9 Clase FEM_barra_Shell_vertical

```

classdef FEM_barra_Shell_vertical
    %Clase que genera un sistema de elementos finitos para una viga
    %empotrada en voladizo, dispuesta de forma vertical con esfuerzo aplicado en su punta

    properties
        nodos={};
        elementos={};
        matrices={};
        rigideces={};
        E_acero = 210*10^9; %Módulo de young: N/m2
        nu_acero = 0.3; %Módulo de poisson: sin unidad
        orden={};
        rigidez_global;
        rigidez_global_copia;
        esfuerzos;
        deformacion;
        deformacion2;
        reacciones;
    end

    methods
        function obj = FEM_barra_Shell_vertical(largo,ancho,alto,n_x,n_y,fuerza)
            j=1;
            n_nodos=n_x*n_y;
            n_elementos=(n_x-1)*(n_y-1);
            for y_i=linspace(0,ancho,n_y)
                for x_i=linspace(0,largo,n_x)
                    obj.nodos{j}=[0 y_i x_i];
                    j=j+1;
                end
            end
        end
    end
end

```

```

for i=1:n_elementos
    n_fila=ceil(i/(n_x-1));
    obj.elementos{i}=[i+(n_fila-1) i+(n_fila) i+n_x+n_fila i+n_x+n_fila-1];
end
Ndof=6;
obj.rigidez_global=zeros(Ndof*length(obj.nodos));
obj.esfuerzos=zeros(Ndof*length(obj.nodos),1);
for i=1:length(obj.elementos)
    xs=zeros(1,4);ys=zeros(1,4);zs=zeros(1,4);
    elem=obj.elementos{i};
    for j=1:4;
        nodo=obj.nodos{elem(j)};
        xs(j)=nodo(1);
        ys(j)=nodo(2);
        zs(j)=nodo(3);
    end
end

[rigidez,esfuerzo]=RE_shell(xs,ys,zs,obj.E_acero,obj.nu_acero,alto,[0,0,0],0);
end
rigidez_global_aux=zeros(Ndof*length(obj.nodos));
contrib=contribucion2(elem,size(rigidez),size(rigidez_global_aux));
rigidez_global_aux(contrib)=rigidez(1:end);
obj.rigidez_global=obj.rigidez_global+rigidez_global_aux;
for j=1:4
    esf_auxiliar=obj.esfuerzos(6*(elem(j)-1)+1:6*(elem(j)-1)+6);
    obj.esfuerzos(6*(elem(j)-1)+1:6*(elem(j)-1)+6)=esf_auxiliar+esfuerzo(6*(j-
1)+1:6*(j-1)+6);
end
end

n_dof=6;
obj.rigidez_global_copia=obj.rigidez_global;

for i=1:n_y
    degrees=obj.dofs(i*n_x);
    obj.esfuerzos(degrees(1))=fuerza/n_y;
    degrees2=obj.dofs(1+n_x*(i-1));
    obj.rigidez_global(degrees2,:)=0;
    obj.rigidez_global(:,degrees2)=0;
    for j=1:length(degrees2)
        obj.rigidez_global(degrees2(j),degrees2(j))=1;
    end
end

obj.deformacion=obj.rigidez_global\obj.esfuerzos;
obj.reacciones=obj.rigidez_global_copia*obj.deformacion;
for i=1:length(obj.nodos)
    aux=obj.deformacion((i-1)*6+1:(i-1)*6+6);
    obj.deformacion2{i}=aux;
end
end
function resp=dofs(obj,nodo)
    N_dof=6;
    resp=zeros(1,N_dof);
    for i=1:N_dof
        resp(i)=N_dof*nodo-(N_dof-i);
    end
end
function plotelements(obj)
    malla=obj.elementos;
    hold on;
    for elem=linspace(1,length(malla),length(malla))
        elemento=malla{elem};
        aux={};
        for i=linspace(1,length(elemento),length(elemento))
            punto=obj.nodos{elemento(i)};
            aux{i,1}=punto;
        end
        c=cell2mat(aux);
        plot3(c(:,1),c(:,2),c(:,3));
    end
    pbaspect('manual');
    axis equal;

```

```

        hold off;
    end
    function plotesfuerzos(obj)
        obj.plotelements;
        hold on
        for i=1:length(obj.nodos)
            nodo=obj.nodos{i};
            fuerza=obj.esfuerzos(6*(i-1)+1:6*(i-1)+3);
            quiver3(nodo(1),nodo(2),nodo(3),fuerza(1)/1000,fuerza(2)/1000,fuerza(3)/1000);
        end
        hold off
    end
    function plot_deformacion(obj,opc,factor)
        malla=obj.elementos;
        hold on;
        axis equal;
        view([1 1 1]);
        set(gcf,'Renderer','zbuffer');
        colorbar;
        angulo=linspace(0,2*pi,720);
        aux2=campos;
        radio=(aux2(1)^2+aux2(2)^2)^.5*20;
        j=1;
        for elem=linspace(1,length(malla),length(malla))
            campos([radio*cos(angulo(j)),radio*sin(angulo(j)),100]);
            elemento=malla{elem};
            aux={};
            aux_def={};
            for i=1:length(elemento)
                punto=obj.nodos{elemento(i)};
                deformacion=obj.deformacion2{elemento(i)};deformacion=deformacion(1:3);
                aux{i,1}=punto;
                aux_def{i,1}=deformacion'';
                if opc==1
                    quiver3(punto(1),punto(2),punto(3),deformacion(1)*100,deformacion(2)*100,deformacion(3)*100);
                end
            end
            c=cell2mat(aux);
            if opc==0
                c2=cell2mat(aux_def);
                c=c+c2*factor;
            end
            if opc==2
                c2=cell2mat(aux_def);
                c=c+c2*factor;
                if(length(elemento)==4)
                    c21=norm(c2(1,:));
                    c22=norm(c2(2,:));
                    c23=norm(c2(3,:));
                    c24=norm(c2(4,:));
                end
            end
            if(length(elemento)==4)
                fill3(c(:,1),c(:,2),c(:,3),[c21 c22 c23 c24],'EdgeColor','flat');
            end
            j=j+1;
            if j==length(angulo)
                j=1;
            end
        end
        hold off;
    end
end
end
end

```

8.1.10 Clase FEM barra_Shell_distribuido

```

classdef FEM_barra_Shell_distribuido
    %Clase que genera un sistema de elementos finitos para una viga
    %empotrada en voladizo, dispuesta de forma vertical con esfuerzo
    %aplicado en todo su largo

```

```

properties
    nodos={};
    elementos={};
    rigideces={};
    E_acero = 210*10^9; %Módulo de young: N/m2
    nu_acero = 0.3; %Módulo de poisson: sin unidad
    rigidez_global;
    esfuerzos_global;
    esfuerzos_global_copia;
    rigidez_global_copia;
    esfuerzos;
    deformacion;
    deformacion2;
    reacciones;
    deformacion_z;
end

methods
    function obj = FEM_barra_Shell_distribuido(largo,ancho,alto,n_x,n_y,W_dist)
        j=1;
        n_nodos=n_x*n_y;
        n_elementos=(n_x-1)*(n_y-1);
        for y_i=linspace(0,ancho,n_y)
            for x_i=linspace(0,largo,n_x)
                obj.nodos{j}=[x_i y_i 0];
                j=j+1;
            end
        end

        for i=1:n_elementos
            n_fila=ceil(i/(n_x-1));
            obj.elementos{i}=[i+(n_fila-1) i+(n_fila) i+n_x+n_fila i+n_x+n_fila-1];
        end

        n_dof=6;
        obj.esfuerzos=zeros(n_dof*length(obj.nodos),1);
        obj.rigidez_global=zeros(n_dof*length(obj.nodos));
        obj.esfuerzos_global=zeros(n_dof*length(obj.nodos),1);
        for i=1:length(obj.elementos)
            xs=zeros(1,4);ys=zeros(1,4);zs=zeros(1,4);
            elem=obj.elementos{i};
            for j=1:4
                nodo=obj.nodos{elem(j)};
                xs(j)=nodo(1);
                ys(j)=nodo(2);
                zs(j)=nodo(3);
            end

[rigidez,esfuerzo]=RE_shell(xs,ys,zs,obj.E_acero,obj.nu_acero,alto,[0,0,W_dist],0);
            rigidez_global_aux=zeros(n_dof*length(obj.nodos));
            contrib=contribucion2(elem,size(rigidez),size(rigidez_global_aux));
            rigidez_global_aux(contrib)=rigidez(1:end);
            obj.rigidez_global=obj.rigidez_global+rigidez_global_aux;
            obj.esfuerzos=esfuerzo;
            for j=1:4
                esf_auxiliar=obj.esfuerzos_global(6*(elem(j)-1)+1:6*(elem(j)-1)+6);
                obj.esfuerzos_global(6*(elem(j)-1)+1:6*(elem(j)-
1)+6)=esf_auxiliar+obj.esfuerzos(6*(j-1)+1:6*(j-1)+6);
            end
        end
        obj.rigidez_global_copia=obj.rigidez_global;
        obj.esfuerzos_global_copia=obj.esfuerzos_global;
        %empotramiento en izquierda #####
        for i=1:n_y
            degrees2=obj.dofs(1+n_x*(i-1));
            obj.esfuerzos_global(degrees2)=0;
            obj.rigidez_global(degrees2,:)=0;
            obj.rigidez_global(:,degrees2)=0;
            for j=1:length(degrees2)
                obj.rigidez_global(degrees2(j),degrees2(j))=1;
            end
        end
        %empotramiento en derecha #####

```



```

%for i=1:n_y
%   degrees2=obj.dofs(n_x+n_x*(i-1));
%   obj.esfuerzos_global(degrees2)=0;
%   obj.rigidez_global(degrees2,:)=0;
%   obj.rigidez_global(:,degrees2)=0;
%   for j=1:length(degrees2)
%       obj.rigidez_global(degrees2(j),degrees2(j))=1;
%   end
%end

%obtención deformación y reacciones #####
obj.deformacion=obj.rigidez_global\obj.esfuerzos_global;
obj.reacciones=obj.rigidez_global_copia*obj.deformacion;
for i=1:length(obj.nodos)
    aux=obj.deformacion((i-1)*6+1:(i-1)*6+6);
    obj.deformacion2{i}=aux;
end
end

function resp=dofs(obj,nodo)
    N_dof=6;
    resp=zeros(1,N_dof);
    for i=1:N_dof
        resp(i)=N_dof*nodo-(N_dof-i);
    end
end

function plotelements(obj)
    %obj.plot();
    malla=obj.elementos;
    hold on;
    for elem=linspace(1,length(malla),length(malla))
        elemento=malla{elem};
        aux={};
        for i=linspace(1,length(elemento),length(elemento))
            punto=obj.nodos{elemento(i)};
            aux{i,1}=punto;
        end
        c=cell2mat(aux);
        plot3(c(:,1),c(:,2),c(:,3));
    end
    pbaspect('manual');
    axis equal;
    hold off;
end

function plotesfuerzos(obj)
    obj.plotelements;
    hold on
    for i=1:length(obj.nodos)
        nodo=obj.nodos{i};
        fuerza=obj.esfuerzos_global(6*(i-1)+1:6*(i-1)+3);
    end
    hold off
end

quiver3(nodo(1),nodo(2),nodo(3),fuerza(1)/130,fuerza(2)/130,fuerza(3)/130,'color','red');
end
hold off

end

function plot_deformacion(obj,opc,factor)
    malla=obj.elementos;
    hold on;
    axis equal;
    view([1 1 1]);
    set(gcf,'Renderer','zbuffer');
    colorbar;
    angulo=linspace(0,2*pi,720);
    aux2=campos;
    radio=(aux2(1)^2+aux2(2)^2)^.5*20;
    j=1;
    for elem=linspace(1,length(malla),length(malla))
        campos([radio*cos(angulo(j)),radio*sin(angulo(j)),100]);
        elemento=malla{elem};
        aux={};
        aux_def={};
        for i=1:length(elemento)
            punto=obj.nodos{elemento(i)};
            deformacion=obj.deformacion2{elemento(i)};deformacion=deformacion(1:3);
        end
    end
end

```

```

        aux{i,1}=punto;
        aux_def{i,1}=deformacion'';
        if opc==1
quiver3(punto(1),punto(2),punto(3),deformacion(1)*100,deformacion(2)*100,deformacion(3)*100);
        end
    end
    c=cell2mat(aux);
    if opc==0
        c2=cell2mat(aux_def);
        c=c+c2*factor;
        plot3(c(:,1),c(:,2),c(:,3));
    end
    if opc==2
        c2=cell2mat(aux_def);
        c=c+c2*factor;
        if(length(elemento)==4)
            c21=norm(c2(1,:));
            c22=norm(c2(2,:));
            c23=norm(c2(3,:));
            c24=norm(c2(4,:));
        end
        if(length(elemento)==4)
            fill3(c(:,1),c(:,2),c(:,3),[c21 c22 c23 c24],'EdgeColor','flat');
        end
    end
    j=j+1;
    if j==length(angulo)
        j=1;
    end
end
hold off;
end
end
end
end
end

```

8.1.11 Clase FEM_barra_Frame

```

classdef FEM_barra_frame
    properties
        nodos={};
        deformaciones2;
        elementos={};
        esfuerzos;
        deformaciones;
        rigidez_global;
        reacciones;
        copia_rigidez_global;
    end
    methods
        function obj=FEM_barra_frame(L,n,h,w)
            i=1;
            for dx=linspace(0,L,n)
                obj.nodos{i}=[dx,0,0];
                i=i+1;
            end
            Ndof=6;
            obj.rigidez_global=zeros(Ndof*length(obj.nodos));
            %Ensamble matrices de rigidez #####
            for j=1:n-1
                obj.elementos{j}=[j,j+1];
                nodo1 = obj.nodos{j};
                nodo2 = obj.nodos{j+1};
                xs=[nodo1(1),nodo2(1)];
                ys=[nodo1(2),nodo2(2)];
                zs=[nodo1(3),nodo2(3)];
                rigidez=rigidez_frame_express(xs,ys,zs,210*10^9,0.3,h,w);
                rigidez_global_aux=zeros(Ndof*length(obj.nodos));
                contrib=contribucion2(obj.elementos{j},size(rigidez),size(rigidez_global_aux));
                rigidez_global_aux(contrib)=rigidez(1:end);
                obj.rigidez_global=obj.rigidez_global+rigidez_global_aux;
            end
        end
    end
end

```

```

obj.copia_rigidez_global=obj.rigidez_global;
obj.esfuerzos=zeros(1,length(obj.nodos)*6);
obj.esfuerzos((length(obj.nodos)-1)*6+3)=1000;
%Condiciones de borde #####
for i=1:6
    obj.rigidez_global(i,:)=0;
    obj.rigidez_global(:,i)=0;
    obj.rigidez_global(i,i)=1;
end
obj.deformaciones=obj.esfuerzos/obj.rigidez_global;
obj.reacciones=obj.copia_rigidez_global*obj.deformaciones';
for i=1:length(obj.nodos)
    aux=obj.deformaciones((i-1)*6+1:(i-1)*6+6);
    obj.deformaciones2{i}=aux;
end

end
function plotelements(obj)
    malla=obj.elementos;
    hold on;
    for elem=linspace(1,length(malla),length(malla))
        elemento=malla{elem};
        aux={};
        for i=linspace(1,length(elemento),length(elemento))
            punto=obj.nodos{elemento(i)};
            aux{i,1}=punto;
        end
        c=cell2mat(aux);
        plot3(c(:,1),c(:,2),c(:,3));
    end
    pbaspect('manual');
    axis equal;
    hold off;
end
function plotesfuerzos(obj)
    obj.plotelements;
    hold on
    for i=1:length(obj.nodos)
        nodo=obj.nodos{i};
        fuerza=obj.esfuerzos_global(6*(i-1)+1:6*(i-1)+3);
        quiver3(nodo(1),nodo(2),nodo(3),fuerza(1)/100000,fuerza(2)/100000,fuerza(3)/100000);
    end
    hold off
end
function plot(obj)
    a=obj.nodos';
    aux=cell2mat(a);
    x=aux(:,1);
    y=aux(:,2);
    z=aux(:,3);
    scatter3(x,y,z, '.');
end
function plot_deformacion(obj,opc,factor)
    hold on;
    malla=obj.elementos;
    axis equal;
    view([1 1 1]);
    set(gcf,'Renderer','zbuffer');
    colorbar;
    angulo=linspace(0,2*pi,720);
    aux2=campos;
    radio=(aux2(1)^2+aux2(2)^2)^.5*20;
    j=1;
    for elem=linspace(1,length(malla),length(malla))
        campos([radio*cos(angulo(j)),radio*sin(angulo(j)),100]);
        elemento=malla{elem};
        aux={};
        aux_def={};
        for i=1:length(elemento)
            punto=obj.nodos{elemento(i)};
            deformacion=obj.deformaciones{elemento(i)};deformacion=deformacion(1:3);
            aux{i,1}=punto;
            aux_def{i,1}=deformacion'';
        end
    end
end

```

