



UNIVERSIDAD DE CUENCA

Facultad de Ingeniería

Carrera de Ingeniería Civil

Análisis de la viga de Timoshenko mediante el MEF a partir de la variante de Galerkin.

Trabajo de titulación previo a la obtención del título de Ingeniero Civil.

Autores:

Andrés Fernando Calle Nieto

CI:0106417017

Correo: andrescalleniet@gmail.com

Christopher William Orellana Sarmiento

CI:0106431109

Correo: christobal777@hotmail.com

Director:

Ing. Ángel Julver Pino Velázquez

CI: 0105928642

Cuenca-Ecuador

08-marzo-2022



Resumen:

La viga es uno de los elementos estructurales fundamentales en la Ingeniería Civil, que debido a las cargas que debe soportar presenta un desplazamiento vertical o deflexión como característica principal. Su estudio se basa en teorías tales como la de Euler-Bernoulli y la teoría de Timoshenko que ha sido de interés para el presente trabajo. Esta teoría se basa en la solución de ecuaciones diferenciales para lo cual se necesita el uso de métodos numéricos entre los cuales destaca el Método de los Elementos Finitos (MEF), no siendo el único, pero sí el más utilizado en la actualidad. Una de las variantes es la de Galerkin que destaca por su eficiencia y que es muy conocido en otros campos de la Físico – Matemática, no así en el Análisis Estructural, en donde la solución para el problema discreto se puede encontrar explícitamente como una combinación lineal de las funciones base o funciones de forma con coeficientes desconocidos, es decir, se obtiene un sistema de n ecuaciones lineales con n incógnitas que permite determinar los coeficientes. Se realiza por tanto la implementación de la variante de Galerkin para la formulación de elementos finitos de la teoría de Timoshenko en MATLAB para la solución de distintos casos de vigas sometidas a diferentes estados de carga (puntual, distribuida) y apoyo (empotrada, simplemente apoyada). Los resultados muestran la diferencia existente entre el valor de las deflexiones obtenidas por el método clásico de deformación de vigas y la teoría de Timoshenko, debido principalmente a la consideración de la acción del cortante que hace esta última y que es válida para vigas en donde los valores de la esbeltez son bajos.

Palabras claves: Viga. Galerkin. Timoshenko. Elemento finito. Funciones de forma. Deflexión. Radio de giro. Matriz de rigidez.



Abstract:

The beam is one of the fundamental structural elements in Civil Engineering, which due to the loads it must support presents a vertical displacement or deflection as its main characteristic. Its study is based on theories such as the Euler-Bernoulli theory and the Timoshenko theory, which has been of interest for the present work. This theory is based on the solution of a differential equation for which the use of numerical methods is required, among which the Finite Element Method (FEM) stands out, not being the only one, but the most used at present. One of the variants is that of Galerkin, which stands out for its efficiency and is well known in other fields of Physics-Mathematics, but not in Structural Analysis, where the solution to the discrete problem can be found explicitly as a linear combination of the basis functions or shape functions with unknown coefficients, i.e., a system of n linear equations with n unknowns is obtained, which allows determining the coefficients. Therefore, the implementation of the Galerkin variant for the finite element formulation of the Timoshenko theory in MATLAB is performed for the solution of different cases of beams subjected to different load (point, distributed) and support (embedded, simply supported) states. The results show the difference between the value of deflections obtained by the classical beam deformation method and the Timoshenko theory, mainly due to the consideration of the shear action by the latter, which is valid for beams where the slenderness values are low.

Keywords: Beam. Galerkin. Timoshenko. Finite element. Form functions. Deflection. Turning radius.



ÍNDICE

1. Introducción.....	10
1.1 Estado del arte.....	10
1.2 Planteamiento del problema.....	11
1.3 Justificación.....	12
1.4 Hipótesis.....	12
1.5 Objetivo General.....	13
1.6 Objetivos específicos.....	13
2. Marco Teórico.....	14
2.1 Material de estudio.....	14
Viga.....	14
2.1.1 Tipos de apoyo.....	14
2.2 Cargas.....	15
2.2.1 Carga Puntual.....	15
2.2.2 Carga distribuida.....	15
2.3 Método de Galerkin.....	16
2.4 Teoría de modelado de flexión de vigas.....	20
2.4.1 Teoría de Euler Bernoulli.....	20
2.4.2 Teoría de Timoshenko.....	20
2.4.3 Ecuaciones descriptivas de la línea de flexión.....	23
2.5 Método de los elementos finitos.....	25
2.6 Métodos de resolución matricial.....	26
2.6.1 Resolución por el método de eliminación de Gauss.....	26
2.6.2 Resolución por el método de eliminación de Gauss-Jordan.....	26
2.7 Análisis de elementos finitos.....	26
2.7.1 Formulación mediante elementos finitos.....	29
2.8 Integración Numérica.....	33
2.9 Uso del software Matlab.....	35
2.9.1 Explicación de código y uso de sentencias.....	35
2.9.2 Ingreso de datos e interfaz del programa.....	37
2.9.3 Descripción de Funciones propias.....	38
2.9.4 Entorno gráfico del programa.....	39
3. Ejemplos y análisis de resultados.....	42



3.1 Solución de una ecuación diferencial por el método de Galerkin.....42

3.2 Aplicación del método del MEF mediante la variante de Galerkin para la viga Timoshenko.....47

 Ejemplo 1: Viga en voladizo con carga puntual en el extremo libre.....47

 Ejemplo 2: Viga simple apoyada con carga distribuida q58

 Ejemplo 3. Viga simplemente apoyada con carga puntual a una distancia x de uno de los apoyos.....62

4. Conclusiones.....69

5. Referencias.....71



Yo **Andrés Fernando Calle Nieto**, autor/a del trabajo de titulación "**Análisis de la viga de Timoshenko mediante el MEF a partir de la variante de Galerkin.**", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor/a.

Cuenca, 08 de marzo de 2022

Andrés Fernando Calle Nieto

C.I: 0106417017



Yo **Andrés Fernando Calle Nieto** en calidad de autor/a y titular de los derechos morales y patrimoniales del trabajo de titulación "**Análisis de la viga de Timoshenko mediante el MEF a partir de la variante de Galerkin.**", de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 08 de marzo de 2022

Andrés Fernando Calle Nieto

C.I: 0106417017



Christopher William Orellana Sarmiento autor/a del trabajo de titulación "Análisis de la viga de Timoshenko mediante el MEF a partir de la variante de Galerkin.", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor/a.

Cuenca 8 de marzo del 2022

Christopher William Orellana Sarmiento

C.I: 0106431109



Yo **Christopher William Orellana Sarmiento** en calidad de autor/a y titular de los derechos morales y patrimoniales del trabajo de titulación "Análisis de la viga de Timoshenko mediante el MEF a partir de la variante de Galerkin.", de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca 8 de marzo del 2022

Christopher William Orellana Sarmiento

C.I: 0106431109



CAPÍTULO 1

1. Introducción.

La viga es una estructura horizontal que puede sostener carga entre dos apoyos sin crear empuje lateral en éstos. Es un elemento fundamental en la construcción, ya que, en función de su geometría, longitud y materiales, se puede determinar la capacidad de la misma para soportar las cargas a las que se encontrará sometida; se debe comprender además que es un elemento que se comporta a flexión y tensión. Debido a estos esfuerzos la viga se deforma curvándose, siendo el desplazamiento vertical de la curva elástica la deflexión de la misma. Para su estudio existen varias teorías, como la de Euler-Bernoulli (teoría clásica) y la de Timoshenko (primer orden). La teoría de vigas de Euler-Bernoulli hace la suposición de que la sección transversal perpendicular al eje neutro de la viga permanece de esta forma después de que ha sido deformada considerando que el esfuerzo cortante transversal es cero (en realidad se considera que no influye considerablemente). En la teoría de vigas de Timoshenko no se realiza esta suposición, ya que el plano de la sección transversal permanece recto, pero no necesariamente perpendicular al eje neutro de la viga (Oñate, 2013). Esto le otorga una ventaja al método, debido a que en vigas cortas los esfuerzos cortantes son considerables.

Los problemas de vigas son analizados por ecuaciones diferenciales que en ciertos casos resultan ser de dificultosa resolución por los métodos analíticos clásicos y se debe recurrir, entonces, a soluciones discretas aproximadas. Entre los distintos métodos que existen, cobra fuerza el Método de los Elementos Finitos que en la actualidad es de los más utilizados.

El Método del Elemento Finito (MEF) es uno de los métodos numéricos para aproximación de soluciones de problemas que necesitan resolver ecuaciones en derivadas ordinarias y parciales, más eficaz y utilizado en la actualidad, y, también, de más objeto de investigación tanto en la teoría como en su aplicación. Su potencial se basa en su sólida base matemática, su capacidad para ser aplicado a una gran variedad de problemas de diferente origen y su aptitud cuando los dominios de análisis son complejos, debido a que la discretización se adapta fácilmente a cualquier elemento de estudio, considerando no linealidades y distintos tipos de condiciones de contorno. Por tanto, a la hora de solucionar un problema, el MEF resulta ser una herramienta de análisis adecuada, óptima y en varias ocasiones la única solución.

Se pretende entonces dar un enfoque del uso del MEF en la solución de vigas, de acuerdo a la teoría de Timoshenko, que estén sometidas a distintas cargas y utilizando la variante de Galerkin.

1.1 Estado del arte.

El MEF es ampliamente utilizado en la Ingeniería de Estructuras donde aparecen trabajos de investigación que analizan deflexiones de vigas, tales como Turner, Clough, Martin y Topp, analizaron la rigidez y deflexión de estructuras complejas mostrando que al aumentar el número de nodos aumenta la precisión de los resultados, dentro de su investigación analiza una viga en cantiléver (Turner, Clough, Martin, & Topp, 1956). Cheng, Han y Huang, aplican el MEF



Petrov-Galerkin a la viga de Timoshenko, usan integración reducida para evitar el fenómeno de bloqueo y así aumentar la precisión de la solución (Cheng, Han, & Huang, 1997). Eisenberger, presenta un método para obtener funciones de forma exactas, directamente de las ecuaciones diferenciales de la teoría de la viga de Timoshenko. Éstas funciones de forma son cúbicas y parabólicas para la deflexión y la rotación respectivamente (Eisenberger, 1994). Friedman y Kosmatka, basan su enfoque en el Principio de Hamilton para la formulación del MEF con elemento de dos nodos, polinomios de Lagrange cúbicos y cuadráticos, además matriz rigidez libre de bloqueo de cortante; para analizar una viga corta de Timoshenko obteniendo resultados exactos para la deflexión (Friedman & Kosmatka, 1993). Mukherjee y Prathap, analizan el fenómeno de bloqueo de una viga de Timoshenko con elemento de tres nodos y llegando a la conclusión de que este tipo de elemento no es muy vulnerable al fenómeno de bloqueo (Mukherjee & Prathap, 2002). Lee, Koo y Choi, obtienen una formulación variacional de la viga de Timoshenko al separar el modo de deflexión por flexión y deflexión por cortante. Los desplazamientos se interpolan como una función de interpolación continua de Hermite para la deflexión por flexión y una interpolación lineal para la deflexión por cortante. El elemento desarrollado no sufre bloqueo de corte, y el resultado de la aplicación de este elemento es bueno de acuerdo con la solución analítica. La ventaja adicional del elemento desarrollado es que el momento se puede calcular exactamente en cualquier ubicación del elemento (Lee, Koo, & Choi, 1994). Edem, analiza con el MEF una viga de Timoshenko con elemento de dos nodos, pero utiliza un nuevo modelo para representar la rotación de la viga, funciones de forma interdependientes; esto conduce al uso de funciones de interpolación de Hermite y lineal para la deflexión y la rotación respectivamente. La nueva formulación que propone elude el fenómeno de bloqueo (Edem, 2006). Reddy, utiliza dos enfoques: El modelo de elementos finitos de deformación supuesta de la teoría de viga de Timoshenko convencional, y el modelo de elementos finitos de desplazamiento asumido de una teoría de viga de Timoshenko modificada. Para obtener la matriz de rigidez 4×4 (para el caso de flexión pura) del modelo de elementos finitos superconvergentes para problemas estáticos. Analiza vigas de Euler-Bernoulli y Timoshenko. Demuestra que el elemento de interpolación interdependiente proporciona valores nodales exactos para los desplazamientos. El elemento de interpolación interdependiente basado en la teoría de la viga de Timoshenko es ideal para el análisis estructural de marcos, ya que un elemento por miembro estructural todavía produce desplazamientos y fuerzas nodales exactos (Reddy, 1999). Paiva y Guimarães, hacen un análisis comparativo de las deflexiones entre las vigas Euler-Bernoulli y Timoshenko, tomando en cuenta el coeficiente de esbeltez, aplicando el Método de Diferencias Finitas (MDF). (Paiva & Guimarães, 2018)

1.2 Planteamiento del problema.

El comportamiento real de las estructuras ha sido foco de estudio durante muchos años por físicos e ingenieros civiles preocupados de entender la razón de la estabilidad de sus diseños basándose en las leyes de la física. Esto ha llevado a la formulación de diferentes hipótesis basadas en ecuaciones que puedan representar el modelo físico lo más aproximado a la realidad. La matemática detrás de dichos modelos en muchos casos ha requerido la implementación de



nuevas teorías, técnicas y métodos aproximados frente a la dificultad del desarrollo de una solución y en donde ha sido necesario el uso de herramientas computacionales que han servido de apoyo, facilitando el desarrollo de dichos métodos, en las últimas décadas. En la época actual las teorías que modelan de forma eficiente el comportamiento real pueden optimizar el diseño estructural logrando impactar en la evaluación de costos de proyectos de gran envergadura.

Existen varias teorías de análisis entre la cual destaca la de Timoshenko, por considerar el esfuerzo cortante en la sección transversal y otorga una ventaja en el análisis de vigas cortas en donde los efectos debido a esfuerzos cortantes son considerables. Al ser analizada de acuerdo a la técnica del método de los elementos finitos (MEF) se utiliza la variante de Galerkin, llegándose a obtener la matriz de rigidez correspondiente y con ello la solución aproximada del problema.

1.3 Justificación.

Debido a que el análisis asistido por computadora se ha convertido en una herramienta de gran utilidad, que permite conocer el comportamiento de la viga sin requerir ensayos previos en laboratorio, reduciendo así los costos y tiempos de prueba, este ha permitido mejorar características como resistencia, estabilidad y seguridad. Los problemas generales relacionados con el análisis de este elemento estructural responden al planteamiento de ecuaciones diferenciales de no fácil solución por vía exacta y que requieren de soluciones discretas a través de distintos métodos, como los de colocación, Galerkin, diferencias finitas, etc., pero, sin embargo, métodos tales como MEF (Método de los elementos finitos), MEC (Método de elementos de contorno), MED (Método de las diferencias finitas), etc. han encontrado más aceptación, probablemente, por ser más eficaces en cuanto a su programación, entre otras ventajas. Estos últimos métodos, más que sus capacidades para modelar las ecuaciones matemáticas que se derivan de estos problemas, modelan su comportamiento físico, de ahí su eficacia. Por tanto, el MEF (Método de los elementos finitos) no es el único método, sino el más usado hasta la actualidad. Siendo así, y, a manera de una simple aplicación para el presente trabajo, se desarrolla el método de Galerkin que destaca por su eficiencia con respecto a otros (por ejemplo, Ritz), y que es muy conocido en otros campos de la Físico – Matemática, no así en el Análisis Estructural, lo que justifica su interés.

1.4 Hipótesis.

La variante de Galerkin resuelve de manera más eficaz el problema de la viga Timoshenko, al utilizar como función de peso arbitraria la misma función para aproximar la ya conocida, y en donde la solución para el problema discreto se puede encontrar explícitamente como una combinación lineal de las funciones base con coeficientes desconocidos. El planteamiento de esta variante permitirá desarrollar la formulación de elementos finitos para la viga Timoshenko y la solución del sistema matricial. Debido a las consideraciones en las que se basa la teoría de Timoshenko el efecto de la fuerza cortante es más significativo en vigas menos esbeltas.



1.5 Objetivo General.

Obtención de la matriz de rigidez de la viga Timoshenko por el MEF a partir de la variante de Galerkin.

1.6 Objetivos específicos.

Selección de los tipos de vigas y cargas a analizar.

Descripción de las ecuaciones gobernantes de la viga Timoshenko.

Desarrollo del modelo de elemento finito.

Ensamble de la matriz de rigidez.

Obtención de deflexiones en diferentes vigas ejemplares.

Comparación con el método clásico de los resultados adquiridos con el método Galerkin mediante software, (MATLAB).

CAPÍTULO 2

2. Marco Teórico.

2.1 Material de estudio.

Viga.

La viga es un elemento estructural que se caracteriza por tener una longitud mucho mayor a su largo y ancho. Generalmente son usadas para soportar cargas perpendiculares entre dos apoyos sin crear empuje lateral en éstos. Comúnmente utilizadas en puentes y edificaciones.

Una viga estructural común tiene las siguientes características:

- Homogénea: Aquella que tiene las mismas propiedades elásticas en todos los puntos de la viga.
- Isotrópica: Que tiene las mismas propiedades elásticas en todas las direcciones en cada punto de la viga.
- Lineal: Cuando la relación entre las tensiones y deformaciones es lineal.
- Elástica: Porque tienen la capacidad de recobrar su forma y dimensiones primitivas cuando cesa el esfuerzo que había determinado su deformación.

Cuando la viga se encuentra sometida a esfuerzo de flexión, tanto por cargas externas como por el propio peso de la misma, se produce un desplazamiento vertical ΔZ (Figura 1), a la diferencia entre el posicionamiento final y el inicial se le denomina deflexión.

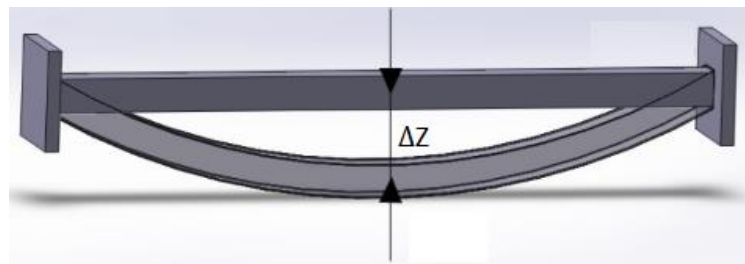


Figura 1 . Deflexión de la viga.

2.1.1 Tipos de apoyo.

Los apoyos son definidos por los grados de libertad de movimiento que presenta una viga horizontal frente a la acción de las fuerzas internas actuantes en una sección que pueden ser axiales, cortantes y de momento flector. Se le denomina **simple apoyo** cuando el desplazamiento a lo largo del eje longitudinal de la viga y el giro de ésta no están impedidos mientras que el desplazamiento transversal es impedido mediante una reacción en ese sentido.

Cuando el apoyo permite el giro de la viga, pero impide el desplazamiento en cualquier dirección mediante una reacción que se puede dividir en una componente a lo largo del eje longitudinal de la viga y otra a lo largo del eje transversal, se le denomina **apoyo fijo**. Para determinar estas dos componentes es necesario hacer uso de dos ecuaciones de la estática.

Cuando el apoyo impide el desplazamiento a lo largo de los ejes y el giro de la viga mediante una reacción que se puede dividir en una componente longitudinal, otra transversal y una reacción de momento hablamos de un **empotramiento**.

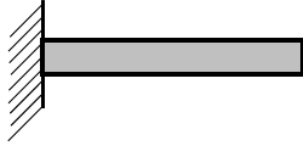
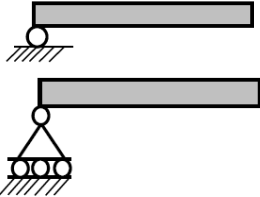
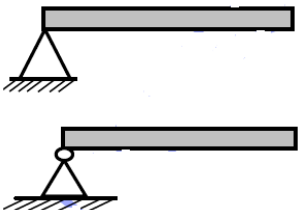
APOYOS, DESPLAZAMIENTOS Y REACCIONES PARA VIGAS			
Tipo de apoyo	Figura	Desplazamientos μ	Reacciones f_b
Empotrado		$\Delta_y = 0$ $\Delta_x = 0$ $\varphi = 0$	$V_y \neq 0$ $V_x \neq 0$ $M \neq 0$
Apoyo simple o rodillo		$\Delta_y = 0$ $\Delta_x \neq 0$ $\varphi \neq 0$	$V_y \neq 0$ V_x no existe M no existe
Apoyo fijo o pasador		$\Delta_y = 0$ $\Delta_x = 0$ $\varphi \neq 0$	$V_y \neq 0$ $V_x \neq 0$ M no existe

Tabla 1 .Apoyos, desplazamientos y reacciones para vigas.

2.2 Cargas.

Las cargas son las fuerzas que actúan a lo largo de la viga y producen cambios en los estados tensionales y deformaciones de la misma. Se pueden distinguir:

2.2.1 Carga Puntual.

Es la que actúa sobre una longitud tan pequeña de la viga que puede suponerse que lo hace sobre un punto.

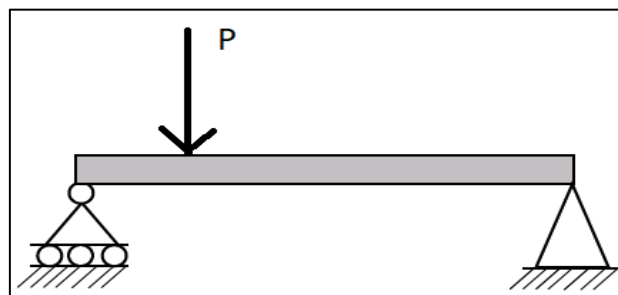


Figura 2 .Viga simplemente apoyada con carga puntual.

2.2.2 Carga distribuida.

Es la que actúa sobre una longitud finita considerable de la viga. Puede ser uniformemente distribuida en toda su longitud o sobre una parte de ella.

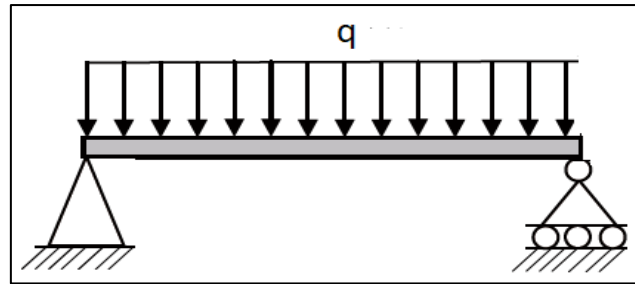


Figura 3 .Viga simplemente apoyada con carga distribuida.

2.3 Método de Galerkin.

El método de Galerkin se puede desarrollar a partir de la siguiente ecuación unidimensional diferencial (Ottosen & Petersson, 1992):

$$Lw(x) + g(x) = 0 \quad ; \quad a \leq x \leq b \quad (2.1)$$

Siendo L un operador diferencial, $w(x)$ es una función desconocida y $g(x)$ es la función conocida.

Considerando las cantidades conocidas de frontera a y b que se obtienen analíticamente según las formas de apoyo de la viga, tendremos las condiciones de frontera respectivas de la ecuación (2.1):

$$w(a) = w_a \quad ; \quad w(b) = w_b \quad (2.2)$$

Con el fin de aproximar una solución para la ecuación (2.1) que cumpla las condiciones de frontera indicadas en (2.2) se reformula la ecuación (2.1), multiplicando la misma por una función de peso arbitraria $v(x)$ donde se obtiene:

$$v(Lw + g) = 0$$

Integrando la expresión anterior entre los límites a y b obtenemos:

$$\int_a^b v(Lw + g)dx = 0 \quad (2.3)$$

Se debe demostrar que tanto (2.1) como (2.3) son formas equivalentes. Para esto la función arbitraria v se denota como:

$$v = Lw + g$$

Entonces la ecuación (2.3) llega a ser:

$$\int_a^b (Lw + g)^2 dx = 0$$



que llega a ser verdadero si cumple (2.1). Para probar que (2.1) y (2.3) son equivalentes es necesario integrar por partes (2.1) y de esta manera se obtiene:

$$\int_a^b v(Lw + g)dx = 0$$

Al buscar una solución numérica que sea aproximada, se debe establecer una aproximación para la función desconocida w . Se asume entonces la siguiente aproximación para cumplir las condiciones de frontera (2.2):

$$w^{app} = \psi_1 a_1 + \psi_2 a_2 + \dots + \psi_n a_n \quad (2.4)$$

Donde a_1, \dots, a_n son parámetros desconocidos y ψ_1, \dots, ψ_n son funciones de x y se denominan funciones de base o funciones de prueba. Buscando determinar los parámetros a_1, \dots, a_n , que dependerán de la naturaleza física del problema, la solución aproximada (2.4) se puede expresar entonces como:

$$w^{app} = \psi \mathbf{a} \quad (2.5)$$

donde:

$$\psi = [\psi_1, \psi_2 \dots \psi_n] \quad , \quad \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \quad (2.6)$$

Reemplazando w en (2.3) por w^{app} se obtiene:

$$\int_a^b v(Lw^{app} + g)dx = 0 \quad (2.7)$$

De la misma forma se puede reemplazar w por w^{app} en la ecuación diferencial (2.1) obteniendo:

$$Lw^{app} + g = e \quad (2.8)$$

Ya que w^{app} no satisface exactamente la ecuación, $e(x)$ es el término residual o medida de error. Con lo expresado en (2.8), la ecuación (2.7) resulta ser:

$$\int_a^b v e dx = 0 \quad (2.9)$$

Algunas características relacionadas con la ecuación 2.9 son:

- La Ortogonalidad entre las funciones $v(x)$ y $e(x)$ dentro del intervalo $a - b$
- La ecuación sirve como medio para determinar las incógnitas a_1, \dots, a_n

La función peso arbitraria se puede escribir generalmente como:



$$v = V_1 C_1 + V_2 C_2 + \dots + V_n C_n \quad (2.10)$$

donde V_1, \dots, V_n son funciones de x conocidas y C_1, \dots, C_n son parámetros.

La ecuación 2.10 puede expresarse como:

$$v = VC \quad (2.11)$$

en donde:

$$V = [V_1, V_2 \dots V_n] \quad ; \quad C = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_n \end{bmatrix} \quad (2.12)$$

Debido a que la función de peso v es arbitraria y V se conoce, se puede concluir que los parámetros C son arbitrarios. La ecuación (2.11) puede ser expresada también como:

$$v = V^T C^T \quad (2.13)$$

esto en consideración a que v es un número y su transpuesta resulta ser el mismo número, es decir $v = v^T$.

Reemplazando (2.13) en (2.9) y tomando en cuenta que C^T no depende de x se obtiene:

$$\int_a^b V^T e dx = 0 \quad (2.14)$$

La matriz V^T tiene dimensión $n \times 1$ según (2.12), la expresión (2.14) entonces comprende n ecuaciones y puede escribirse como:

$$\begin{aligned} \int_a^b V_1 e dx &= 0 \\ \int_a^b V_2 e dx &= 0 \\ \vdots \\ \int_a^b V_n e dx &= 0 \end{aligned} \quad (2.15)$$

De la expresión (2.15) se puede observar que el residuo depende de las n incógnitas a_1, \dots, a_n y sirve como un sistema de ecuaciones para determinar las mismas, con lo cual, si reemplazamos (2.5) en (2.8) y considerando que la matriz \mathbf{a} no depende de x se obtiene:

$$L(\psi)\mathbf{a} + g = e \quad (2.16)$$

Diferenciando la matriz ψ de la expresión (2.6) se obtiene que:

$$L(\psi) = [L(\psi_1), L(\psi_2) \dots L(\psi_n)] \quad (2.17)$$



Siendo $L(\psi)$ una matriz de $1 \times n$ y reemplazando (2.16) en (2.14)

$$\left(\int_a^b V^T L(\psi) dx \right) \mathbf{a} = - \int_a^b V^T g dx \quad (2.18)$$

Se puede definir entonces:

$$\begin{cases} K = \int_a^b V^T L(\psi) dx \\ f = - \int_a^b V^T g dx \end{cases} \quad (2.19)$$

Donde K y f son independientes de \mathbf{a} , quedando la expresión (2.18) como:

$$K \mathbf{a} = f \quad (2.20)$$

Esta expresión consiste en n ecuaciones lineales en donde \mathbf{a} puede ser determinada para luego ser sustituido en la expresión (2.5) y así obtener la solución aproximada.

Debido a que V^T tiene dimensión $n \times 1$ y $L(\psi)$ tiene dimensión $1 \times n$, K y f pueden expresarse como:

$$K = \begin{bmatrix} \int_a^b V_1 L(\psi_1) dx & \int_a^b V_1 L(\psi_2) dx & \cdots & \int_a^b V_1 L(\psi_n) dx \\ \int_a^b V_2 L(\psi_1) dx & \int_a^b V_2 L(\psi_2) dx & \cdots & \int_a^b V_2 L(\psi_n) dx \\ \vdots & \vdots & \ddots & \vdots \\ \int_a^b V_n L(\psi_1) dx & \int_a^b V_n L(\psi_2) dx & \cdots & \int_a^b V_n L(\psi_n) dx \end{bmatrix}; \quad f = - \begin{bmatrix} \int_a^b V_1 g dx \\ \int_a^b V_2 g dx \\ \vdots \\ \int_a^b V_n g dx \end{bmatrix} \quad (2.21)$$

siendo la matriz K una matriz cuadrada de $n \times n$.

En el método de Galerkin las componentes de la matriz V son iguales a las funciones de prueba ψ , es decir:

$$V_i = \psi_i ; \quad i = 1, \dots, n \quad (2.22)$$

Por tanto:

$$\int_a^b \psi_i dx = 0 ; \quad i = 1, \dots, n \quad (2.23)$$

Además, las funciones de prueba son ortogonales al residuo. Reemplazando (2.22) en (2.21) se puede obtener la matriz coeficiente K y el lado derecho f como:

$$K = \begin{bmatrix} \int_a^b \psi_1 L(\psi_1) dx & \int_a^b \psi_1 L(\psi_2) dx & \dots & \int_a^b \psi_1 L(\psi_n) dx \\ \int_a^b \psi_2 L(\psi_1) dx & \int_a^b \psi_2 L(\psi_2) dx & \dots & \int_a^b \psi_2 L(\psi_n) dx \\ \vdots & \vdots & \ddots & \vdots \\ \int_a^b \psi_n L(\psi_1) dx & \int_a^b \psi_n L(\psi_2) dx & \dots & \int_a^b \psi_n L(\psi_n) dx \end{bmatrix}; \quad f = - \begin{bmatrix} \int_a^b \psi_1 g dx \\ \int_a^b \psi_2 g dx \\ \vdots \\ \int_a^b \psi_n g dx \end{bmatrix} \quad (2.24)$$

2.4 Teoría de modelado de flexión de vigas.

Se considera una viga de longitud l dirigida a lo largo del eje (x), sección transversal de área A simétrica al eje (x, y), que tiene módulo de inercia I sobre la que actúan una serie de cargas verticales y momentos contenidos en el plano principal de inercia de la sección transversal (x, y) como se indica en la siguiente figura:

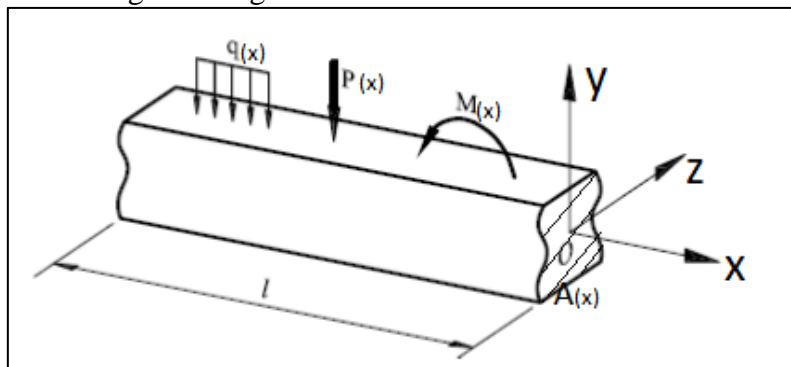


Figura 4 .Viga convencional sometida a cargas y momentos. Fuente: (Oñate, 2013)

2.4.1 Teoría de Euler Bernoulli.

La teoría clásica de vigas o llamada también de Euler Bernoulli se basa principalmente en las siguientes hipótesis:

1. Los desplazamientos verticales de todos los puntos de una sección transversal son pequeños e iguales a los del eje de la viga.
2. El desplazamiento lateral es nulo.
3. Las secciones transversales normales al eje de la viga antes de la deformación, permanecen planas y ortogonales a dicho eje después de la deformación. (Oñate, 2013)

2.4.2 Teoría de Timoshenko.

El interés central del trabajo actual se dirige hacia la teoría establecida por Timoshenko, que comparte las dos primeras hipótesis antes mencionadas, pero con respecto a la hipótesis 3 realiza una suposición cinemática fundamental estableciendo que **“las secciones transversales normales al eje de la viga permanecen planas, pero no necesariamente normales al eje de la viga después de la deformación”** (Oñate, 2013). Esta hipótesis representa una mayor aproximación a la deformación real de la sección transversal en vigas de gran grosor, ya que

mientras la relación entre la longitud y el canto disminuye, las secciones transversales dejan de ser planas una vez que se deforman.

Se puede decir, además, que la viga de Euler Bernoulli no considera que el esfuerzo cortante es nulo como tal, sino que dicho esfuerzo no influye significativamente en la deformación transversal del elemento, mientras que si lo hace la viga de Timoshenko, que por su parte, no considera la influencia de los esfuerzos normales en la dirección transversal de la viga, como lo hace por ejemplo la viga pared (viga-diafragma), la cual no acepta ninguna de las hipótesis de partida de las vigas anteriores.

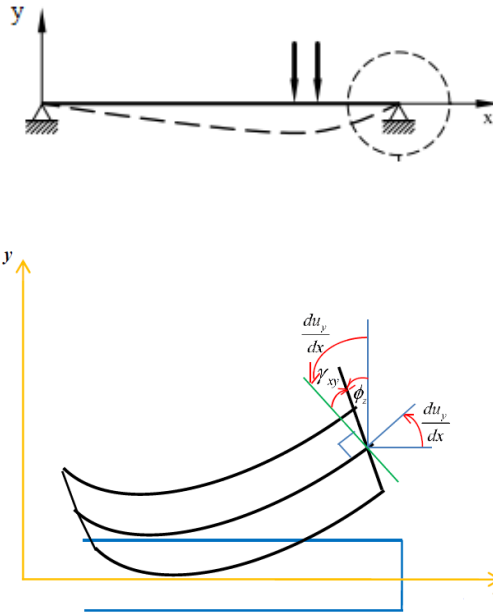


Figura 5. Giro de la sección normal. Fuente: (Piedra, 2019)

Analizando el giro de la sección normal a la fibra media, considerando el efecto de la deformación por cortante a la sección transversal se tiene las siguientes relaciones de la figura anterior:

$$\sin \varphi_z = \frac{u_x}{y} \approx \varphi_z$$

de donde:

$$u_x = -y\varphi_z \quad (2.25)$$

La expresión general para la deformación viene dada por:

$$\varepsilon_x = \frac{du_x}{dx}$$

que relacionada a la ecuación (2.25) resulta:

$$\varepsilon_x = -y \frac{d\varphi_z}{dx} \quad (2.26)$$



La teoría de Timoshenko por tanto considera el efecto de la deformación por cortante transversal, coincidiendo la magnitud de dicha deformación con el giro adicional de la normal (Piedra, 2019) y en base a la Figura 5 se puede decir que:

$$\varphi_z = \frac{du_y}{dx} - \gamma_{xy} \quad (2.27)$$

en donde:

γ_{xy} es un giro adicional debido a la deformación por cortante.

El esfuerzo normal σ_x y el esfuerzo cortante τ_{xy} se relacionan con las deformaciones correspondientes según las siguientes expresiones:

$$\sigma_x = E \varepsilon_x \quad (2.28)$$

$$\tau_{xy} = G \gamma_{xy} \quad (2.29)$$

Donde:

E : es el módulo de elasticidad

G : es el módulo cortante que se puede expresar en función del módulo de elasticidad

E y el coeficiente de Poisson ν :

$$G = \frac{E}{2(1+\nu)} \quad (2.30)$$

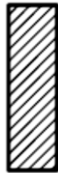
Además, el esfuerzo cortante puede ser expresado como la fuerza cortante Q_y que actúa en un área de sección transversal equivalente A_s , así:

$$\tau_{xy} = \frac{Q_y}{A_s} \quad (2.31)$$

y de la relación entre el área cortante A_s y el área de la sección transversal del elemento A se puede obtener el factor de corrección para el cortante K_s :

$$K_s = \frac{A_s}{A} \quad (2.32)$$

El valor de K_s depende de la sección de la viga; la Tabla 2 muestra algunos valores de K_s para distintas secciones

Sección transversal	Figura	Valor de K_s
Rectangular		$\frac{5}{6}$

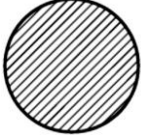
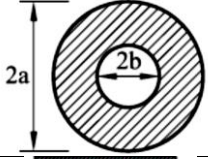

Circular		$\frac{6}{7}$
Circulo-hueco		$\frac{6(1+m^2)^2}{(1+m^2)^2+20m^2}; m = \frac{a}{b}$
Tubo cuadrado de pared delgada		$\frac{5}{12}$

Tabla 2 .Valor de Ks para distintas secciones de viga.

El momento flector puede ser escrito utilizando las expresiones (2.28) y (2.26) como:

$$M_z(x) = EI_z \frac{d\varphi_z(x)}{dx} \quad (2.33)$$

en donde el momento de inercia $I_z = \int y^2 dA$.

El esfuerzo cortante puede ser escrito utilizando las expresiones (2.29), (2.31) y (2.32) y de acuerdo a la ley de Hooke como:

$$Q_y(x) = K_s GA \gamma_{xy} \quad (2.34)$$

2.4.3 Ecuaciones descriptivas de la línea de flexión.

Diferenciando la ecuación 2.33 del momento flector, se obtiene la relación entre el momento interno y la rotación de la sección transversal:

$$\frac{dM_z(x)}{dx} = \frac{d}{dx} \left(EI_z \frac{d\varphi_z(x)}{dx} \right) \quad (2.35)$$

La relación entre la fuerza cortante y momento flector obtenidas a través de relaciones de equilibrio se expresa como:

$$-Q_y(x) = \frac{dM_z(x)}{dx} \quad (2.36)$$

Con la ayuda de esta relación reemplazando 2.34 y 2.35 en 2.36 se puede establecer que:

$$\frac{d}{dx} \left(EI_z \frac{d\varphi_z(x)}{dx} \right) = -K_s GA \gamma_{xy} \quad (2.37)$$

Reemplazando el giro adicional producida por la deformación a cortante γ_{xy} por su expresión en el sistema de referencia deducida en 2.27 tenemos:

$$\frac{d}{dx} \left(EI_z \frac{d\varphi_z(x)}{dx} \right) + K_s GA \left(-\varphi_z + \frac{du_y}{dx} \right) = 0 \quad (2.38)$$



Esta es la **ecuación diferencial de flexión**.

Considerando un elemento infinitesimal en el plano xy donde la reacción de cortante interno esta únicamente en función de la carga distribuida constante q_y (Piedra, 2019) se puede expresar la relación como:

$$-q_y = \frac{dQ_y(x)}{dx} \quad (2.39)$$

A partir de la diferenciación de (2.36) obtenemos la relación entre la carga distribuida y el momento flector que es:

$$q_y = \frac{-dQ_y(x)}{dx} = \frac{d^2M_z(x)}{dx^2} \quad (2.40)$$

Retomando el esfuerzo cortante de acuerdo a la ley de Hooke 2.34 después de la diferenciación y la consideración de equilibrio se obtiene:

$$\frac{d}{dx} (K_s GA \left(-\varphi_z + \frac{du_y}{dx} \right)) = -q_y \quad (2.41)$$

La expresión 2.41 representa la **ecuación diferencial del cortante** para el rango $0 < x < L$

La viga Timoshenko queda descrita por las ecuaciones **2.38** y **2.41**, que son dos ecuaciones diferenciales de segundo orden donde:

- u es la deflexión
- φ es la rotación,
- q es la carga transversal distribuida,
- G es el módulo de cortante,
- A es el área de la sección transversal,
- K_s es el factor de corrección por cortante,
- E es el módulo de Young,
- I es el momento de inercia.

Las condiciones de frontera esenciales son:

$$w(0) = u_1$$

$$\varphi(0) = u_2$$

$$w(L) = u_3$$

$$\varphi(L) = u_4$$

y las condiciones de fronteras naturales:

$$K_s GA \left(-\varphi_z + \frac{du_y}{dx} \right)_{x=0} = Q_1$$



$$K_s GA \left(-\varphi_z + \frac{du_y}{dx} \right)_{x=L} = Q2$$

$$EI_z \frac{d\varphi_z(x)}{dx} \Big|_{x=0} = M1$$

$$EI_z \frac{d\varphi_z(x)}{dx} \Big|_{x=L} = M2$$

2.5 Método de los elementos finitos.

El método del elemento finito es una técnica en la que el dominio está representado como un conjunto de dominios simples, llamados elementos finitos, tal que, es posible construir sistemáticamente las funciones aproximación necesitadas en una aproximación variacional o residuo pesado de la solución de un problema sobre cada elemento. Por lo tanto, el método del elemento finito difiere de los métodos tradicionales en la manera de cómo se construyen las funciones aproximación. El método del elemento finito tiene las siguientes características básicas:

1. División del todo en partes.

Se representa geoméricamente un dominio, como un conjunto geométrico de dominios simples, que dan lugar a una derivación sistemática de las funciones aproximación. Para un dominio unidimensional, el cual es una línea recta, los elementos son pequeños segmentos interconectados que forman la línea original.

2. Derivación de funciones aproximación sobre cada elemento.

Las funciones aproximación son por lo regular, polinomios algebraicos que se derivan usando la teoría de interpolación. Los polinomios de mayor orden, se aproximan más a la solución. Una vez que el orden del polinomio se seleccionó, podemos obtener una expresión para la incógnita en un elemento.

3. Discretización de la ecuación.

Se discretiza la ecuación diferencial hasta formular un sistema de ecuaciones algebraicas. Para este paso, se usa el método de Galerkin. Este proceso transforma las ecuaciones diferenciales en un sistema de ecuaciones algebraicas, que se pueden escribir en forma matricial. Este proceso tiene tres sub-pasos:

- Formulación de las ecuaciones elemento: Estas se obtiene utilizando las funciones forma locales y usando Galerkin sobre cada elemento. Con esto se obtendrá la matriz de rigidez, el vector frontero, el vector de carga uniforme y el vector de carga puntual de cada elemento



- Ensamble de elementos. usando la matriz de conectividad, se suma apropiadamente todos los sistemas elemento obteniendo un sistema de ecuaciones global. Con esto se obtendrá la matriz de rigidez global y el vector de fuerza global
- Imposición de las condiciones de contorno: el sistema obtenido, es singular mientras no se incorpore las condiciones de contorno, cuya implementación genera un sistema viable para resolverse.

4. Solución del sistema discreto (Implementación computacional).

El paso final en el proceso del elemento finito es resolver el sistema de ecuaciones algebraico, generalmente son matrices banda. (Piedra, 2019)

2.6 Métodos de resolución matricial.

El objetivo de la obtención de la matriz de rigidez global y su respectivo vector de fuerzas radica en la solución de la igualdad $\bar{k} \cdot \bar{a} = \bar{f}$ es decir encontrar los valores numéricos de los elementos de \bar{a} . Para esto es necesario la resolución de la matriz aumentada $\bar{k}^{-1}\bar{f}$, esto implica utilizar métodos ampliamente conocidos.

2.6.1 Resolución por el método de eliminación de Gauss.

El método de eliminación de Gauss consiste en ir eliminando los elementos bajo la diagonal en la columna del pivote, con el objeto de ir reduciendo los arreglos en forma triangular superior. Al hacer esto, la última ecuación tendrá una sola incógnita, y esta puede ser fácilmente determinada. La (n-1) ecuación contiene la incógnita anterior n y una nueva n-1 sustituyendo la primera incógnita de la última ecuación anterior se puede resolver la nueva y así sucesivamente.

2.6.2 Resolución por el método de eliminación de Gauss-Jordan.

El método de Gauss-Jordán consiste en ir eliminando las incógnitas en forma similar que, en el método de eliminación de Gauss, El primer paso es dividir el coeficiente del pivote en la ecuación y hasta dejaren términos de un último pivote n así para cuando estos finalicen ya se obtendrá la matriz en forma escalonada reducida.

2.7 Análisis de elementos finitos.

Partiendo del sistema de ecuaciones diferenciales que rigen el modelo de viga de Timoshenko, expresiones (2.38) y (2.41), establecidas en el apartado 2.4.3, y teniendo en cuenta solo las condiciones de contorno arbitrarias y multiplicando estas expresiones por una función de peso v e integrando por partes, se obtiene que:

$$-K_s GA \left(-\varphi + \frac{dw}{dx} \right) v_1 \Big|_a^b + \int_a^b K_s GA \left(-\varphi + \frac{dw}{dx} \right) \frac{dv_1}{dx} dx = \int_a^b q v_1 dx \quad (2.42)$$

y



$$EI \frac{d\varphi}{dx} v_2 \Big|_a^b - \int_a^b EI \frac{d\varphi}{dx} \frac{dv_2}{dx} dx + \int_a^b K_s GA \left(-\varphi + \frac{dw}{dx} \right) v_2 dx = 0 \quad (2.43)$$

La formulación débil del sistema de ecuaciones diferenciales (2.38) y (2.41) se puede obtener sumando las expresiones (2.42) y (2.43):

$$\int_a^b EI \frac{d\varphi}{dx} \frac{dv_2}{dx} dx + \int_a^b K_s GA \left(-\varphi + \frac{dw}{dx} \right) \left(\frac{dv_1}{dx} - v_2 \right) dx = \int_a^b q v_1 dx + K_s GA \left(-\varphi + \frac{dw}{dx} \right) v_1 \Big|_a^b + EI \frac{d\varphi}{dx} v_2 \Big|_a^b \quad (2.44)$$

Debido a que la rotación φ y la deflexión w son independientes se pueden aproximar mediante las siguientes expresiones:

$$\tilde{\varphi} = N_\varphi u \quad (2.45)$$

$$\tilde{w} = N_w u \quad (2.46)$$

en donde:

$$N_\varphi = [0 \quad N_{\varphi 1} \quad 0 \quad N_{\varphi 2} \quad \cdots \quad 0 \quad N_{\varphi n}] \quad ; \quad u = \begin{bmatrix} u_1 \\ \varphi_1 \\ u_2 \\ \varphi_2 \\ \vdots \\ u_n \\ \varphi_n \end{bmatrix} \quad (2.47)$$

$$N_w = [N_{w1} \quad 0 \quad N_{w2} \quad 0 \quad \cdots \quad N_{wn} \quad 0] \quad ; \quad u = \begin{bmatrix} u_1 \\ \varphi_1 \\ u_2 \\ \varphi_2 \\ \vdots \\ u_n \\ \varphi_n \end{bmatrix} \quad (2.48)$$

siendo u_i y φ_i los valores nodales y n el número de incógnitas. Diferenciado con respecto a x las expresiones (2.45) y (2.46), se puede obtener:

$$\frac{d\tilde{\varphi}}{dx} = \frac{dN_\varphi(x)}{dx} u \quad (2.49)$$

$$\frac{d\tilde{w}}{dx} = \frac{dN_w(x)}{dx} u \quad (2.50)$$

Hay que notar que tanto la rotación φ como la deflexión w son de clase C_0 , lo que significa que garantizan que el desplazamiento axial es continuo dentro del elemento y entre elementos. Además N_{wi} y $N_{\varphi i}$ son las denominadas *funciones de forma* o funciones de interpolación polinómicas definidas en el dominio del elemento. Estas funciones de forma interpolan dentro del elemento únicamente los desplazamientos y giros correspondientes al nodo i ; se puede

establecer entonces que la función de forma vale la unidad en el nodo i y cero en el resto de los nodos (Oñate, 2013), como se observa en la Figura 6.

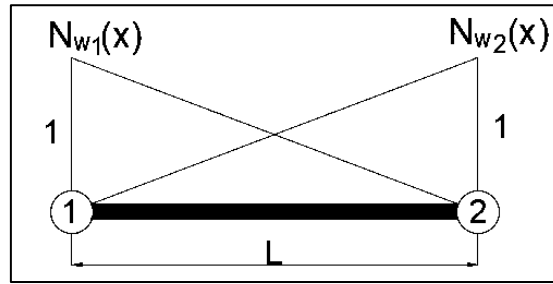


Figura 6. Funciones de forma.

La función de peso arbitraria, de acuerdo al método de Galerkin se puede expresar como:

$$v_1 = N_w \delta_u \quad (2.51)$$

$$v_2 = N_\varphi \delta_u \quad (2.52)$$

Como v es un número y su transpuesta resulta ser el mismo número, es decir $v = v^T$, las expresiones (2.51) y (2.52) pueden ser reescritas como:

$$v_1 = \delta_u^T N_w^T \quad (2.53)$$

$$v_2 = \delta_u^T N_\varphi^T \quad (2.54)$$

y diferenciando con respecto a x se obtiene que:

$$\frac{dv_1}{dx} = \delta_u^T \frac{dN_w^T(x)}{dx} \quad (2.55)$$

$$\frac{dv_2}{dx} = \delta_u^T \frac{dN_\varphi^T(x)}{dx} \quad (2.56)$$

Sustituyendo (2.53), (2.54), (2.55) y (2.56) en (2.44), y tomando en cuenta que δ_u^T es arbitrario por la existencia de las funciones arbitrarias v_1 y v_2 , se obtiene:

$$\int_a^b EI \frac{dN_\varphi^T(x)}{dx} \frac{dN_\varphi}{dx} dx + \int_a^b K_s GA \left(-N_\varphi^T + \frac{dN_w^T(x)}{dx} \right) \left(\frac{dN_w}{dx} - N_\varphi \right) dx = \int_a^b q N_w^T(x) dx + Q(x) N_w^T(x) \Big|_a^b + M(x) N_\varphi^T(x) \Big|_a^b \quad (2.57)$$

La expresión (2.57) es la **Formulación de Galerkin**, la cual se utilizará para la formulación de los elementos finitos.

2.7.1 Formulación mediante elementos finitos.

Se define a la viga como un elemento lineal horizontal que recibe una carga en la dirección transversal a su eje y se divide en un número finito de partes, las cuales se denominan elementos finitos, y que se encuentran unidas por los llamados nodos. Los desplazamientos generados en cada nodo son los grados de libertad, que para las deflexiones se expresan como u_1^e y u_2^e y para los giros como φ_1^e y φ_2^e . (Figura 7).

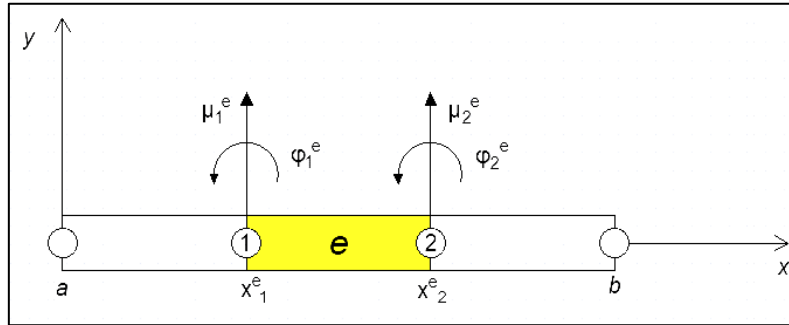


Figura 7. Elemento finito.

Se considera una aproximación local para un elemento de viga, en donde la deflexión w y la rotación φ son independientes, de clase C_0 y se pueden aproximar de acuerdo a las siguientes expresiones:

$$\tilde{\varphi}^e = N_{\varphi}^e u^e \quad (2.58)$$

$$\tilde{w}^e = N_w^e u^e \quad (2.59)$$

en donde:

$$N_{\varphi}^e = \begin{bmatrix} 0 & N_{\varphi 1}^e & 0 & N_{\varphi 2}^e \end{bmatrix} \quad ; \quad u^e = \begin{bmatrix} u_1^e \\ \varphi_1^e \\ u_2^e \\ \varphi_2^e \end{bmatrix} \quad (2.60)$$

$$N_w^e = \begin{bmatrix} N_{w1}^e & 0 & N_{w1}^e & 0 \end{bmatrix} \quad ; \quad u^e = \begin{bmatrix} u_1^e \\ \varphi_1^e \\ u_2^e \\ \varphi_2^e \end{bmatrix} \quad (2.61)$$

siendo u_i y φ_i los valores nodales. Diferenciado con respecto a x las expresiones (2.60) y (2.61), se puede obtener:

$$\frac{dN_{\varphi}^e}{dx} = \begin{bmatrix} 0 & \frac{dN_{\varphi 1}^e}{dx} & 0 & \frac{dN_{\varphi 2}^e}{dx} \end{bmatrix} \quad (2.62)$$



$$\frac{dN_w^e}{dx} = \left[\frac{dN_{w1}^e}{dx} \quad 0 \quad \frac{dN_{w2}^e}{dx} \quad 0 \right] \quad (2.63)$$

La ecuación (2.57) puede ser definida entonces sobre un subdominio $[x_1^e; x_2^e]$ del dominio $[a, b]$:

$$\int_{x_1^e}^{x_2^e} EI \frac{dN_\varphi^{eT}(x)}{dx} \frac{dN_\varphi^e}{dx} dx u^e + \int_{x_1^e}^{x_2^e} K_s GA \left(-N_\varphi^{eT} + \frac{dN_w^{eT}(x)}{dx} \right) \left(\frac{dN_w^e}{dx} - N_\varphi^e \right) dx u^e = \int_{x_1^e}^{x_2^e} q N_w^{eT}(x) dx + Q(x) N_w^{eT}(x) \Big|_{x_1^e}^{x_2^e} + M(x) N_\varphi^{eT}(x) \Big|_{x_1^e}^{x_2^e} \quad (2.64)$$

La expresión (2.64) es entonces la **Formulación de Elementos Finitos para la Viga Timoshenko**. La ecuación puede ser descrita de otra forma mediante la definición de las siguientes matrices:

- $K_b^e = \int_{x_1^e}^{x_2^e} EI \frac{dN_\varphi^{eT}(x)}{dx} \frac{dN_\varphi^e}{dx} dx$: Matriz de rigidez correspondiente a efectos de flexión.
- $K_\tau^e = \int_{x_1^e}^{x_2^e} K_s GA \left(-N_\varphi^{eT} + \frac{dN_w^{eT}(x)}{dx} \right) \left(\frac{dN_w^e}{dx} - N_\varphi^e \right) dx$: Matriz de rigidez correspondiente a efectos de cortante.
- $f_b^e = Q(x) N_w^{eT}(x) \Big|_{x_1^e}^{x_2^e} + M(x) N_\varphi^{eT}(x) \Big|_{x_1^e}^{x_2^e}$: Vector frontera
- $f_q^e = \int_{x_1^e}^{x_2^e} q N_w^{eT}(x) dx$: Vector de carga

De acuerdo a estas expresiones se puede definir la matriz de rigidez como:

$$K^e = K_b^e + K_\tau^e \quad (2.65)$$

y el vector fuerza como:

$$F^e = f_b^e + f_q^e \quad (2.66)$$

De esta manera la ecuación (2.64) puede ser escrita en forma matricial como:

$$K^e u^e = F^e \quad (2.67)$$

De acuerdo con las expresiones (2.58) y (2.59) la aproximación para la deflexión y la rotación pueden ser definidas como:

$$\tilde{\varphi}(x) = N_{\varphi 1}(x) \varphi_1 + N_{\varphi 2}(x) \varphi_2 \quad (2.68)$$

$$\tilde{w}(x) = N_{w1}(x) u_1 + N_{w2}(x) u_2 \quad (2.69)$$

Los nodos en los cuales se aplican los desplazamientos y rotaciones ($u; \varphi$), así como también los momentos y las fuerzas, se ubican en los extremos del elemento de viga y siguen lo designado en la Figura 8.

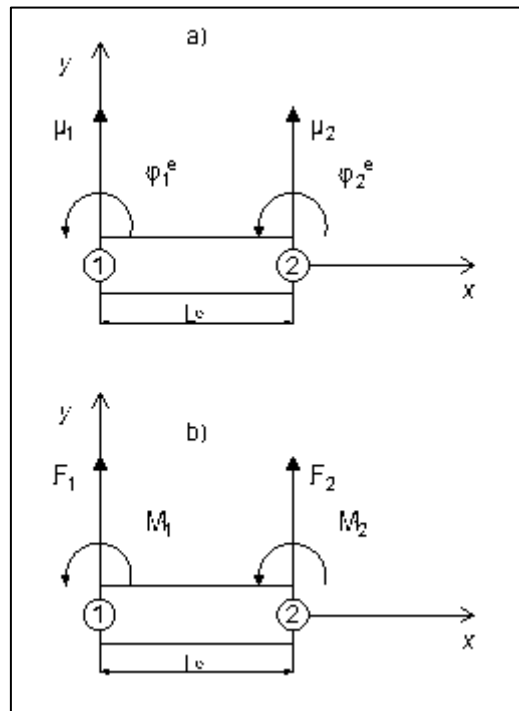


Figura 8. Definición de la dirección positiva para el elemento en el plano xy : a) parámetros de deformación b) parámetros de carga.

Se puede observar en las matrices de rigidez K_b^e y K_τ^e que aparecen las derivadas de primer orden de las funciones de forma, las cuales serán, como mínimo, polinomios lineales para la rotación y deflexión, es decir:

$$N_{w1}(x) = N_{\varphi1}(x) = 1 - \frac{x}{L} \quad (2.70)$$

$$N_{w2}(x) = N_{\varphi2}(x) = \frac{x}{L} \quad (2.71)$$

derivando las expresiones:

$$\frac{dN_{w1}}{dx} = \frac{dN_{\varphi1}}{dx} = -\frac{1}{L} \quad (2.72)$$

$$\frac{dN_{w2}}{dx} = \frac{dN_{\varphi2}}{dx} = \frac{1}{L} \quad (2.73)$$

Se calculan las integrales de K_b^e y K_τ^e :

$$K_b^e = EI \int_0^L \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{dN_{\varphi 1}}{dx} & \frac{dN_{\varphi 1}}{dx} & 0 & \frac{dN_{\varphi 1}}{dx} & \frac{dN_{\varphi 2}}{dx} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{dN_{\varphi 2}}{dx} & \frac{dN_{\varphi 1}}{dx} & 0 & \frac{dN_{\varphi 2}}{dx} & \frac{dN_{\varphi 2}}{dx} \end{bmatrix} dx$$

$$K_b^e = EI \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{1}{L} & 0 & -\frac{1}{L} \\ 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{L} & 0 & \frac{1}{L} \end{bmatrix} \quad (2.74)$$

$$K_\tau^e = K_s GA \int_0^L \begin{bmatrix} \frac{dN_{w1}}{dx} & \frac{dN_{w1}}{dx} & \frac{dN_{w1}}{dx} (-N_{\varphi 1}) & \frac{dN_{w1}}{dx} & \frac{dN_{w2}}{dx} & \frac{dN_{w1}}{dx} (-N_{\varphi 2}) \\ (-N_{\varphi 1}) & \frac{dN_{w1}}{dx} & (-N_{\varphi 1}) (-N_{\varphi 1}) & (-N_{\varphi 1}) & \frac{dN_{w2}}{dx} & (-N_{\varphi 1}) (-N_{\varphi 2}) \\ \frac{dN_{w2}}{dx} & \frac{dN_{w1}}{dx} & \frac{dN_{w2}}{dx} (-N_{\varphi 1}) & \frac{dN_{w2}}{dx} & \frac{dN_{w2}}{dx} & \frac{dN_{w2}}{dx} (-N_{\varphi 2}) \\ (-N_{\varphi 2}) & \frac{dN_{w1}}{dx} & (-N_{\varphi 2}) (-N_{\varphi 1}) & (-N_{\varphi 2}) & \frac{dN_{w2}}{dx} & (-N_{\varphi 2}) (-N_{\varphi 2}) \end{bmatrix} dx$$

$$K_\tau^e = K_s GA \begin{bmatrix} \frac{1}{L} & \frac{1}{2} & -\frac{1}{L} & \frac{1}{2} \\ \frac{1}{2} & \frac{L}{3} & -\frac{1}{2} & \frac{L}{6} \\ -\frac{1}{L} & -\frac{1}{2} & \frac{1}{L} & -\frac{1}{2} \\ \frac{1}{2} & \frac{L}{6} & -\frac{1}{2} & \frac{L}{3} \end{bmatrix} \quad (2.75)$$

De acuerdo a lo expresado en (2.65) la matriz de rigidez total de la viga Timoshenko puede ser expresada como la suma de (2.74) y (2.75):

$$K^e = K_b^e + K_\tau^e = \begin{bmatrix} \frac{K_s GA}{L} & \frac{K_s GA}{2} & -\frac{K_s GA}{L} & \frac{K_s GA}{2} \\ \frac{K_s GA}{2} & \frac{K_s GA}{3} + \frac{EI}{L} & -\frac{K_s GA}{2} & \frac{K_s GA}{6} - \frac{EI}{L} \\ -\frac{K_s GA}{L} & -\frac{K_s GA}{2} & \frac{K_s GA}{L} & -\frac{K_s GA}{2} \\ \frac{K_s GA}{2} & \frac{K_s GA}{6} - \frac{EI}{L} & -\frac{K_s GA}{2} & \frac{K_s GA}{3} + \frac{EI}{L} \end{bmatrix}$$

Si se establece que $\rho = \frac{4EI}{K_s GA}$ se puede expresar K^e como (Piedra, 2019):

$$K^e = K_b^e + K_\tau^e = \frac{K_s GA}{4L} \begin{bmatrix} 4 & 2L & -4 & 2L \\ 2L & \frac{4}{3}L^2 + \rho & -2L & \frac{4}{6}L^2 - \rho \\ -4 & -2L & 4 & -2L \\ 2L & \frac{4}{6}L^2 - \rho & -2L & \frac{4}{3}L^2 + \rho \end{bmatrix} \quad (2.76)$$

o, a su vez, se puede establecer también que $\lambda = \frac{EI}{K_s GAL^2}$

$$K^e = K_b^e + K_\tau^e = \frac{EI}{6\lambda L^3} \begin{bmatrix} 6 & 3L & -6 & 3L \\ 3L & L^2(2 + 6\lambda) & -3L & L^2(1 - 6\lambda) \\ -6 & -3L & 6 & -3L \\ 3L & L^2(1 - 6\lambda) & -3L & L^2(2 + 6\lambda) \end{bmatrix} \quad (2.77)$$

Para el caso del vector frontera f_b se toma en cuenta las condiciones de las funciones de forma:

$$f_b = \begin{bmatrix} Q(L)N_{w1}(L) - Q(0)N_{w1}(0) \\ M(L)N_{\varphi1}(L) - M(0)N_{\varphi1}(0) \\ Q(L)N_{w2}(L) - Q(0)N_{w2}(0) \\ M(L)N_{\varphi2}(L) - M(0)N_{\varphi2}(0) \end{bmatrix}$$

$$f_b = \begin{bmatrix} -Q(0) \\ -M(0) \\ Q(L) \\ M(L) \end{bmatrix} \quad (2.78)$$

y para el caso del vector de carga f_q se tiene que:

$$f_q = \int_0^L q(x) \begin{bmatrix} N_{w1} \\ 0 \\ N_{w2} \\ 0 \end{bmatrix} dx \quad (2.79)$$

2.8 Integración Numérica.

El cálculo analítico de las integrales de los elementos de viga puede resultar laborioso, por lo que se recurre a la integración numérica. Para el desarrollo del presente trabajo se utiliza la integración numérica Gauss-Legendre, por ser el procedimiento más popular en relación al método de los elementos finitos, ya que utiliza el mínimo número de puntos de integración para conseguir un error determinado en el cálculo de la integral y por tanto minimiza el número de veces que hay que calcular el valor de la función a integrar (Oñate, 2013).



Se considera entonces que los puntos de integración están expresados en el espacio normalizado:

$$-1 \leq \xi \leq 1$$

donde:

$$\xi = -1 + 2 \frac{x}{L}$$

$$\frac{d\xi}{dx} = \frac{2}{L}$$

Las funciones de forma pueden ser expresadas como:

$$N_{w1}(\xi) = N_{\varphi1}(\xi) = \frac{1}{L}(1 - \xi) \quad (2.80)$$

$$N_{w2}(\xi) = N_{\varphi2}(\xi) = \frac{1}{L}(1 + \xi) \quad (2.81)$$

siendo $\frac{dN}{dx} = \frac{dN}{d\xi} \frac{d\xi}{dx}$

El cálculo de la matriz de rigidez de flexión K_b^e exige un solo punto de integración, esto debido a que todos los términos del integrando son constantes (Oñate, 2013). De esta manera realizando la integración numérica de -1 a 1 se obtiene que:

$$K_b^e = EI \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{1}{L} & 0 & -\frac{1}{L} \\ 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{L} & 0 & \frac{1}{L} \end{bmatrix} \quad (2.82)$$

La integración de la matriz de rigidez para el cortante precisa en cambio dos puntos de integración por aparecer en el integrando términos de segundo orden (Oñate, 2013). De esta manera realizando la integración se obtiene que:

$$K_\tau^e = K_s GA \begin{bmatrix} \frac{1}{L} & \frac{1}{2} & -\frac{1}{L} & \frac{1}{2} \\ \frac{1}{2} & \frac{L}{4} & -\frac{1}{2} & \frac{L}{4} \\ -\frac{1}{L} & -\frac{1}{2} & \frac{1}{L} & -\frac{1}{2} \\ \frac{1}{2} & \frac{L}{4} & -\frac{1}{2} & \frac{L}{4} \end{bmatrix} \quad (2.83)$$

Si se suman las ecuaciones (2.82) y (2.83) se obtiene la matriz de rigidez total y si además se asume que $\lambda = \frac{EI}{K_s GA L^2}$ resulta:



$$K^e = K_b^e + K_\tau^e = \frac{EI}{6\lambda L^3} \begin{bmatrix} 6 & 3L & -6 & 3L \\ 3L & L^2(\frac{3}{2} + 6\lambda) & -3L & L^2(\frac{3}{2} - 6\lambda) \\ -6 & -3L & 6 & -3L \\ 3L & L^2(\frac{3}{2} - 6\lambda) & -3L & L^2(2 + 6\lambda) \end{bmatrix} \quad (2.84)$$

De estos resultados se puede observar que (2.82) es igual a (2.74), esto nos indica que la ecuación de Gauss Legendre a un solo punto de integración es precisa, no siendo así para (2.84).

2.9 Uso del software Matlab.

En este apartado se explica la implementación de los códigos, la secuencia lógica, rutinas y subrutinas necesarias para el procesamiento de datos de entrada, interpretación de datos, cálculo de la matriz de rigidez, la implementación de las condiciones de borde y, finalmente, calcular la solución. Además, se define la parte grafica del programa que muestra los datos de entrada, posiciones y tipos de apoyos, magnitud y tipos de carga, cantidad y localización de los elementos y sus respectivos nodos, así como la gráfica de salida que mostrará las deflexiones en los puntos de interés de la viga.

2.9.1 Explicación de código y uso de sentencias.

Entendida la metodología y su correspondiente desarrollo utilizando cálculo variacional y los conceptos físicos aplicados a la mecánica de materiales, resta explicar la implementación del lenguaje de programación de MATLAB. Para esto, es necesario explicar algunos flujos propios del programa utilizados en el presente trabajo.

a) Sentencia if.

Sintaxis.

```
if expression
    statements
elseif expression
    statements
else
    statements
end
```

Descripción:

`if` expresión, `statements`, `end` evalúa una expresión y ejecuta un grupo de declaraciones cuando la expresión es verdadera. Una expresión es verdadera cuando su resultado no está vacío y contiene solo elementos distintos de cero (lógicos o numéricos reales). De lo contrario, la expresión es falsa.

Los bloques `elseif` y `else` son opcionales. Las declaraciones se ejecutan solo si las expresiones anteriores en el bloque `if...end` son falsas. Un bloque `if` puede incluir varios bloques `elseif` (MathWorks, 2021).



b) Sentencia for.

Sintaxis.

```
for index = values
    statements
end
```

Descripción:

`for index = values, statements, end` ejecuta un grupo de declaraciones en un bucle durante un número específico de veces (MathWorks, 2021).

c) Zeros.

Sintaxis.

```
X = zeros(n)
```

Descripción:

`X = zeros(n)` devuelve una matriz de ceros n-por- n (MathWorks, 2021).

d) Plot

Sintaxis

```
plot(X, Y)
```

Descripción:

`plot(X, Y)` crea una gráfica de línea 2-D de los datos en Y versus los valores correspondientes en X.

- Si X e Y son ambos vectores, entonces deben tener la misma longitud. La función `plot` traza Y versus X.
- Si X e Y son ambas matrices, deben tener el mismo tamaño. La función `plot` traza las columnas de Y en comparación con las columnas de X.
- Si X o Y es un vector y el otro es una matriz, la matriz debe tener dimensiones de forma que una de sus dimensiones sea igual a la longitud del vector. Si el número de filas de la matriz equivale a la longitud del vector, la función `plot` traza cada columna de matriz en comparación con el vector. Si el número de columnas matriciales equivale a la longitud del vector, la función trazará cada fila de matriz en comparación con el vector. Si la matriz es cuadrada, entonces la función traza cada columna frente al vector.



- Si x o y es un escalar y el otro es un escalar o un vector, entonces la función `plot` traza puntos discretos. Sin embargo, para ver los puntos debe especificar un símbolo de marcador, por ejemplo, `plot(x, y, 'o')`. (MathWorks, 2021)

e) Funciones propias del usuario.

El programa Matlab ofrece la facultad al usuario de realizar comandos personalizados mediante la función `function` descrita a continuación:

Sintaxis

```
function [y1,...,yN] = myfun(x1,...,xM)
```

Descripción:

`function [y1,...,yN] = myfun(x1,...,xM)` declara una función nombrada `myfun` que acepta entradas x_1, \dots, x_M y devuelve salidas y_1, \dots, y_N . Esta declaración de declaración debe ser la primera línea ejecutable de la función. Los nombres de funciones válidos comienzan con un carácter alfabético y pueden contener letras, números o guiones bajos.

Puede guardar su función:

- En un archivo de funciones que contiene solo definiciones de funciones. El nombre del archivo debe coincidir con el nombre de la primera función del archivo.
- En un archivo de secuencia de comandos que contiene comandos y definiciones de funciones. Las funciones deben estar al final del archivo. Los archivos de script no pueden tener el mismo nombre que una función en el archivo. Las funciones se admiten en scripts en R2016b o posterior.

Los archivos pueden incluir múltiples funciones locales o funciones anidadas. Para facilitar la lectura, use la palabra clave `end` para indicar el final de cada función en un archivo. La palabra clave `end` es necesaria cuando:

- ❖ Cualquier función del archivo contiene una función anidada.
- ❖ La función es una función local dentro de un archivo de función, y cualquier función local del archivo utiliza la palabra clave `end`.
- ❖ La función es una función local dentro de un archivo de secuencia de comandos. (MathWorks, 2021)

2.9.2 Ingreso de datos e interfaz del programa.

Para facilidad del usuario, los datos del problema se introducen mediante tablas en archivo `.xlsx`. Los datos necesarios son:

- Geométricos: Área, Inercia, Longitud, factor de corrección para cortante (Ks)
- Características mecánicas: Módulo de elasticidad (E), Módulo de Poisson (ν).
- Cargas y apoyos: Posición de cargas, posición de apoyos, tipos de cargas, grados de libertad del apoyo.
- Descripción de los elementos: cantidad y posición de nodos.



2.9.3 Descripción de Funciones propias.

a) Función para exportar datos (**Exportar_Datos_Entrada**).

Se extraen los datos de las tablas con información, generándose matrices para cada tipo de dato (nodos, elementos, apoyos, cargas nodales).

b) Función extraer los datos de partida (**Formar_Datos_Partida**).

Se extraen datos indirectos a partir de las matrices de datos, como son los números de elementos en la matriz (nodos, elementos, apoyos, cargas nodales), se crean vectores que caracterizan cada uno de los nodos con su respectiva condición de posición y carga. Además, se define las características geométricas y mecánicas de la viga y se la particiona para cada elemento. Se forman vectores para la longitud de los elementos, para representación de las cargas puntuales en los nodos y para las cargas distribuidas de los elementos.

c) Función para dibujar el modelo (**Dibujar_modelo**).

Se asigna una matriz con el nombre del elemento que contiene las coordenadas en x e y de los nodos que conforman cada elemento, se dibuja una línea definida entre dos nodos contiguos. Seguido, se grafica el apoyo caracterizado por una figura geométrica en su posición según los datos de entrada.

d) Función para dibujar cargas (**Dibujar_cargas**).

Se asignan el valor de cada fuerza puntual P a su coordenada de aplicación y se dibuja a una escala adecuada. Se asignan de igual manera la cantidad de cada carga distribuida q , en caso de existir, a su coordenada de aplicación y extensión de aplicación dibujando a lo largo de los elementos sometidos a dicha carga.

e) Función para formar la matriz **K** elemental (**Ensamblado_K_elem**).

Por comodidad de programación se crean variables como factores numéricos que afecta a elementos de la matriz de rigidez elemental K^e , con el fin de que se asemeje a la expresión 2.77 antes deducida en este documento. Se crea la matriz K^e .

f) Función para ensamblar la matriz **K** global (**Ensamblado_K_global**).

Se asigna la posición de cada matriz elemental dentro de la matriz global y se suma a su posición correspondiente, creando un proceso de ensamblaje interno a partir de las submatrices elementales.

g) Función vector fuerza elemental (**Ensamblado_F_elem**).



Se crea un vector cero de cuatro elementos y se evalúa la existencia de cargas puntuales o distribuida, cuando hay presencia de cargas se asigna su respectivo valor.

h) Función vector fuerza global (Ensamblado_F_glob).

Se crea un vector global de fuerzas añadiendo cada uno de los vectores elementales en su respectiva posición.

i) Función para la solución del sistema. (Solucion_Sistema).

Para encontrar las deflexiones y giros, se particiona la matriz K global en cuatro matrices: K_{pp} , K_{ps} , K_{sp} y K_{ss} . De la misma manera se particiona el vector de fuerzas en los nodos en un vector de fuerzas p y en un vector de fuerzas s . El vector de fuerzas p son las fuerzas aplicadas en los nodos que no tienen restricción y el vector de fuerzas s es desconocido ya que depende de las reacciones en los apoyos. De igual manera se particiona el vector de fuerzas de empotramiento en p y s . Se soluciona el sistema para U_p que contiene los desplazamientos desconocidos y posteriormente con estos se encuentran las fuerzas internas de los elementos multiplicando sus matrices de rigidez por los desplazamientos ya conocidos y sumándoles las fuerzas de empotramiento.

f) Función dibujar deformada analítica. (Dibujar_Deformada_Analitica).

Se selecciona el tipo de apoyos y el tipo de carga puntual o distribuida que afecta a la viga, en caso de ser puntual se indica la posición en la que se encuentra aplicada, la función plotea los nodos de entrada en su posición (x) y los resultados de las ecuaciones cargadas según el tipo de modelo seleccionado como (y).

g) Función dibujar deformada de Galerkin. (Dibujar_Deformada_Galerkin).

A partir del arreglo del sistema matricial resuelto, plotea los nodos cargados como datos de entrada en la posición (x) y los resultados de las deflexiones en su posición (y), además compara los resultados y remarca gráficamente el punto de mayor deflexión.

h) Función exportar datos solución.

Guarda en formato .xlsx las matrices de rigidez global (K_p), desplazamientos y giros de los nodos (U_p), vector de reacciones y fuerzas en los elementos (F_p).

2.9.4 Entorno gráfico del programa.

El programa implementado en Matlab cuenta con una interfaz gráfica que facilita al usuario la entrada de datos, la visualización del modelo implementado y las gráficas de las deflexiones, así como también la posterior exportación de los resultados. La ventana interactiva cuenta con botones, barras y menús que cumplen funciones específicas, las cuales se explican a continuación:



Permite cargar el archivo .xlsx que contiene los datos del modelo de viga que se desea calcular.



Permite guardar los cambios realizados en el modelo cargado, siendo las opciones presentadas *Guardar* y *Guardar como*.



Permiten visualizar en pantalla los datos cargados en el archivo .xlsx tales como: número de elementos, número de nodos, cargas, condiciones de apoyo, geometría de la viga.



Permite graficar el modelo cargado en el archivo .xlsx, mostrando las cargas y apoyos especificados.



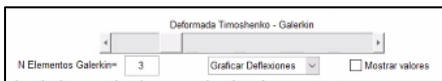
Permite ejecutar el programa para la obtención de los resultados.



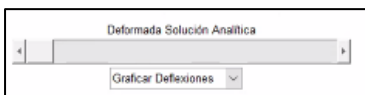
Permite crear un archivo .xlsx con los resultados obtenidos. El archivo contiene la matriz K global, desplazamientos, reacciones y fuerzas en los elementos.



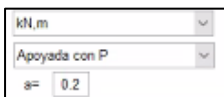
Permite visualizar las deflexiones de la viga, tanto la “exacta” como la obtenida por la teoría de Timoshenko.



Permite modificar la escala gráfica de la deflexión obtenida por la teoría de Timoshenko, mostrar los valores, elegir la gráfica de deflexión o giro y el número de elementos existentes entre nodos que se desea conocer su deflexión.



Permite modificar la escala gráfica de la deflexión obtenida por la teoría de clásica y elegir la gráfica de deflexión o giro.



Permiten elegir las unidades, el tipo de modelo cargado para las gráficas de deflexión: empotrada con carga puntual, simplemente apoyada con carga distribuida, simplemente apoyada con carga puntual a una distancia a del apoyo izquierdo. El valor de a representa la ubicación de la carga puntual (en porcentaje) en referencia a la longitud de la viga. Así, si la viga tiene 1m de longitud y $a=0,5$ (50%) la carga se ubicará en el centro del claro.

Para mayor información acerca del desarrollo del programa puede consultarse la sección de Anexos que contiene todo el código utilizado para la ejecución del programa.

El proceso general que se sigue para la obtención de los resultados se lo representa en el siguiente diagrama de flujo:

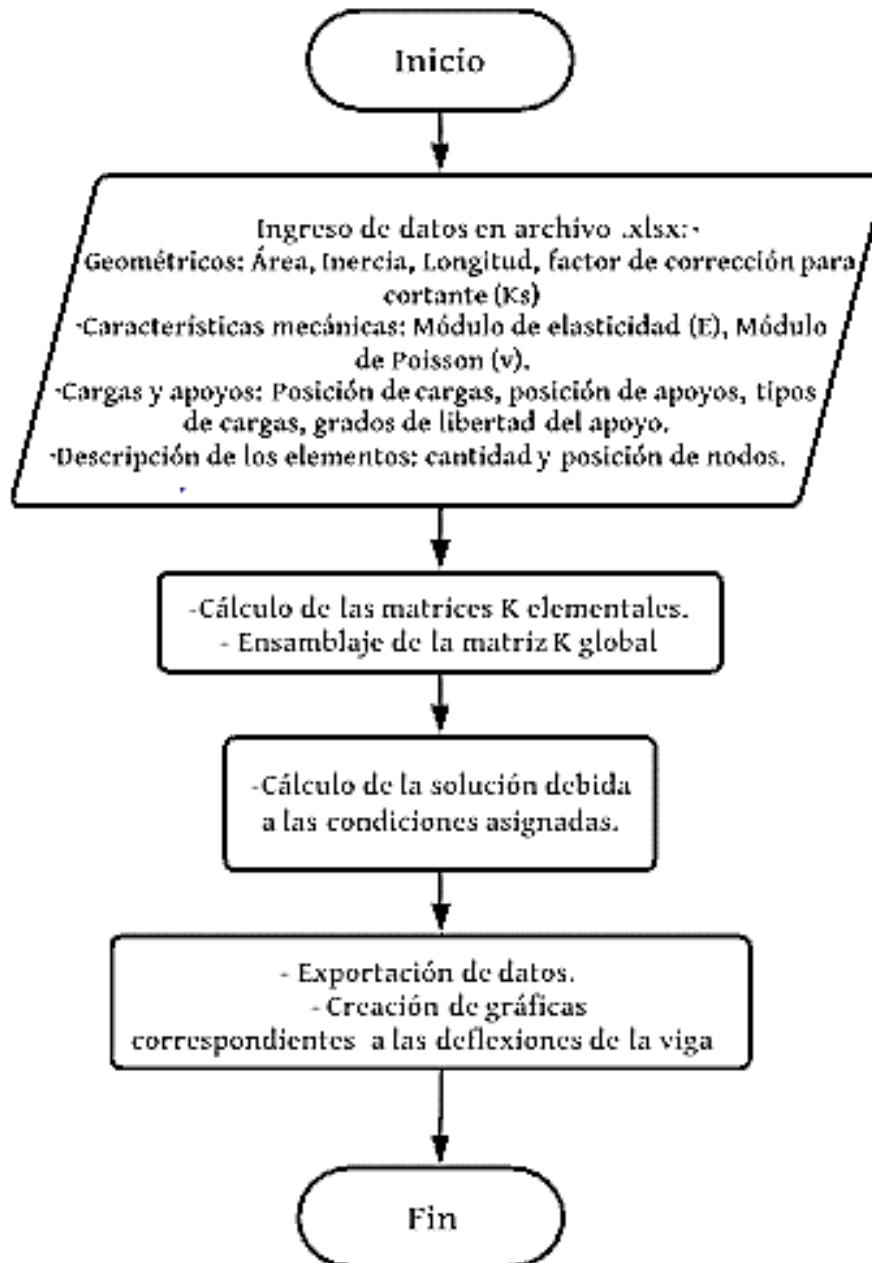


Diagrama 1. Diagrama de flujo para la solución del problema computacional.



CAPÍTULO 3

3. Ejemplos y análisis de resultados.

3.1 Solución de una ecuación diferencial por el método de Galerkin.

Se desea resolver la siguiente ecuación diferencial con sus respectivas condiciones de frontera:

$$\frac{dy^2}{dx^2} - \frac{4}{3}y = x^3 - 2$$

$$\text{condiciones de frontera} \begin{cases} y(0) = 1 \\ y(5) = 2 \end{cases}$$

Para la obtención de la solución exacta se establece:

- Homogénea:

$$\left(D^2 - \frac{4}{3}\right)y = 0$$

despejando:

$$D = \sqrt{\frac{4}{3}} \rightarrow D = \frac{2}{\sqrt{3}} ; D = -\frac{2}{\sqrt{3}}$$

por tanto:

$$y_h = c_1 e^{\frac{2}{\sqrt{3}}x} + c_2 e^{-\frac{2}{\sqrt{3}}x}$$

- Particular:

$$y_p = Ax^3 + Bx^2 + Cx + E \rightarrow y_p = -\frac{3}{4}x^3 - \frac{27}{8}x + \frac{3}{2}$$

La solución general es de la forma:

$$y_g = c_1 e^{\frac{2}{\sqrt{3}}x} + c_2 e^{-\frac{2}{\sqrt{3}}x} + \left(-\frac{3}{4}x^3 - \frac{27}{8}x + \frac{3}{2}\right)$$

Para $x = 0$:

$$c_1 e^{\frac{2}{\sqrt{3}}0} + c_2 e^{-\frac{2}{\sqrt{3}}0} + \left(-\frac{3}{4}0^3 - \frac{27}{8}0 + \frac{3}{2}\right) = 1$$

de donde:



$$c_1 + c_2 = -\frac{1}{2}$$

Para $x = 5$:

$$c_1 e^{\frac{2}{\sqrt{3}}5} + c_2 e^{-\frac{2}{\sqrt{3}}5} + \left(-\frac{3}{4}5^3 - \frac{27}{8}5 + \frac{3}{2}\right) = 2$$

de donde:

$$c_1 e^{\frac{10\sqrt{3}}{3}} + c_2 e^{-\frac{10\sqrt{3}}{3}} = \frac{889}{8}$$

Resolviendo el sistema:

$$\begin{bmatrix} 1 & 1 \\ e^{\frac{10\sqrt{3}}{3}} & e^{-\frac{10\sqrt{3}}{3}} \end{bmatrix} \cdot \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} \\ \frac{889}{8} \end{bmatrix}$$

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ e^{\frac{10\sqrt{3}}{3}} & e^{-\frac{10\sqrt{3}}{3}} \end{bmatrix}^{-1} \begin{bmatrix} -\frac{1}{2} \\ \frac{889}{8} \end{bmatrix}$$

se obtiene que:

$$c_1 = 0.34547903$$

$$c_2 = -0.84547903$$

De esta manera la solución de la EDO es:

$$y_{exacta} = 0.34547903 e^{\frac{2}{\sqrt{3}}x} - 0.84547903 e^{-\frac{2}{\sqrt{3}}x} + \left(-\frac{3}{4}x^3 - \frac{27}{8}x + \frac{3}{2}\right)$$

Se aplica el método de Galerkin y con el fin de aproximar una solución para la ecuación, que cumpla las condiciones de frontera indicadas, se establece que:

$$\check{y}(x) = a_0 + a_1x + a_2x^2$$

Reemplazando en la aproximación las condiciones de frontera:

$$\check{y}(0) = 1 \rightarrow a_0 = 1$$



$$\check{y}(5) = 2 \rightarrow 2 = 1 + a_1 5 + a_2 25$$

$$1 = +a_1 5 + a_2 25 \rightarrow a_2 = \frac{1 - 5a_1}{25}$$

$$\check{y}(x) = 1 + a_1 x + \frac{1 - 5a_1}{25} x^2$$

$$\check{y}(x) = 1 + \frac{x^2}{25} + a_1 x \left(1 - \frac{x}{5}\right)$$

Se establece la aproximación $\check{y}(x)$ en base a las funciones arbitrarias V_0 y V_1

$$\check{y}(x) = 1 + \frac{x^2}{25} + a_1 x \left(1 - \frac{x}{5}\right)$$

$$\check{y}(x) = V_0(x) + a_1 V_1(x)$$

siendo:

$$V_0(x) = 1 + \frac{x^2}{25}$$

$$V_1(x) = x \left(1 - \frac{x}{5}\right)$$

Reemplazando y derivando:

$$\check{y}(x) = 1 + \frac{x^2}{25} + a_1 x \left(1 - \frac{x}{5}\right)$$

$$\frac{d\check{y}}{dx} = \frac{2x}{25} + a_1 \left(-\frac{2x}{5} + 1\right)$$

$$\frac{d\check{y}^2}{dx^2} = \frac{2}{25} - a_1 \frac{2}{5}$$

La función residual queda expresada como:

$$R = \frac{2}{25} - a_1 \frac{2}{5} - \frac{4}{3} \left(1 + \frac{x^2}{25} + a_1 x \left(1 - \frac{x}{5}\right)\right) - x^3 + 2$$

Para determinar el coeficiente a_1 se usa la siguiente integral ponderada:



$$\int_0^5 V_1 R_x dx = 0$$
$$\frac{-(130a_1 + 1849)}{18} = 0$$
$$a_1 = -\frac{1849}{130}$$

De esta forma la solución aproximada queda de la siguiente forma:

$$\check{y}(x) = 1 + \frac{x^2}{25} - \frac{1849}{130}x\left(1 - \frac{x}{5}\right)$$

La solución aproximada ha sido desarrollada mediante una función de aproximación cuadrática, la cual no se ajusta todavía a la solución real de acuerdo con la gráfica (). Por lo tanto, ahora se establece una función de aproximación cúbica de la forma:

$$\check{y}(x) = a_0 + a_1x + a_2x^2 + a_3x^3$$

Reemplazando en la aproximación las condiciones de frontera

$$\check{y}(0) = 1 \rightarrow a_0 = 1$$

$$\check{y}(5) = 2 \rightarrow 2 = 1 + a_15 + a_225 + a_3125$$

$$1 = +a_15 + a_225 + a_3125 \rightarrow a_1 = \frac{1 - 25a_2 - 125a_3}{5}$$

$$\check{y}(x) = 1 + \frac{1 - 25a_2 - 125a_3}{5}x + a_2x^2 + a_3x^3$$

$$\check{y}(x) = 1 + \frac{x}{5} + a_2x(x - 5) + a_3x(x^2 - 25)$$

Se establece la aproximación $\check{y}(x)$ en base de las funciones arbitrarias V_0 y V_1 :

$$V_0(x) = x(x - 5)$$

$$V_1(x) = x(x^2 - 25)$$

Reemplazando y derivando:



$$\frac{d\tilde{y}}{dx} = \frac{1}{5} + a_2(2x - 5) + a_3(3x^2 - 25)$$

$$\frac{d\tilde{y}^2}{dx^2} = 2a_2 + 6a_3x$$

La función residual queda expresada como:

$$R = 2a_2 + 6a_3x - \frac{4}{3} \left(1 + \frac{x}{5} + a_2x(x - 5) + a_3x(x^2 - 25) \right) - x^3 + 2$$

Para determinar los coeficientes a_2 y a_3 se usa las siguientes integrales ponderadas:

$$I_1 = \int_0^5 V_0 R_x dx = 0$$

$$26a_2 + 195a_3 = 75 \quad (1)$$

$$I_2 = \int_0^5 V_1 R_x dx = 0$$

$$1365a_2 + 1052a_3 = 4507 \quad (2)$$

Resolviendo el sistema de ecuaciones (1 y 2) para a_2 y a_3 :

$$\begin{bmatrix} 26 & 195 \\ 1365 & 1052 \end{bmatrix} \cdot \begin{bmatrix} a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 75 \\ 4507 \end{bmatrix}$$

$$\begin{bmatrix} a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 26 & 195 \\ 1365 & 1052 \end{bmatrix}^{-1} \begin{bmatrix} 75 \\ 4507 \end{bmatrix}$$

se obtiene:

$$a_2 = \frac{-17973}{1469}$$

$$a_3 = \frac{1139}{565}$$

De esta forma la solución aproximada queda de la siguiente forma:

$$\tilde{y}(x) = 1 + \frac{x}{5} - \frac{17973}{1469}(x^2 - 5x) + \frac{1139}{565}(x^3 - 25x)$$

Repetiendo el proceso ejecutado para una función de aproximación de orden 2 y 3 se resuelve para una de orden 4, dando el siguiente resultado:



$$a_2 = \frac{27169569}{2975968}$$

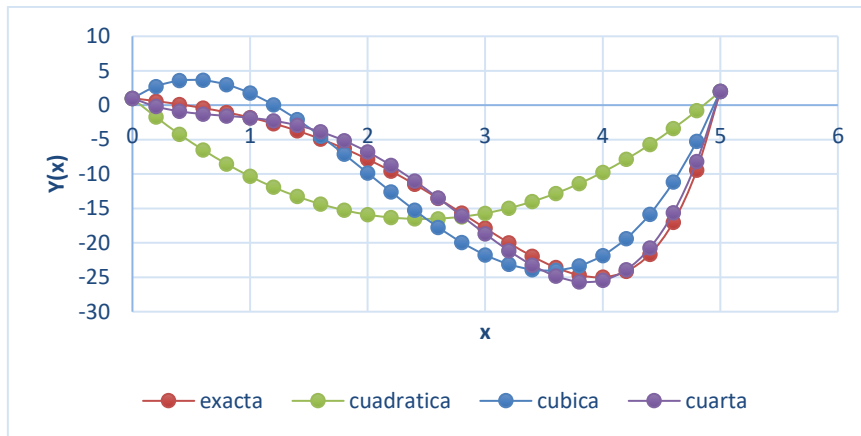
$$a_3 = \frac{-37504273}{743992}$$

$$a_4 = \frac{18585}{26336}$$

La solución aproximada queda de la siguiente forma:

$$\tilde{y}(x) = 1 + \frac{x}{5} + \frac{27169569}{2975968}(x^2 - 5x) - \frac{37504273}{743992}(x^3 - 25x) + \frac{18585}{26336}(x^4 - 125x)$$

Se puede observar en la gráfica () las soluciones aproximadas con respecto a la solución exacta; conforme el orden de la función de aproximación aumenta la solución tiende a ser la exacta.



Gráfica 1. Soluciones aproximadas y exacta de la EDO.

3.2 Aplicación del método del MEF mediante la variante de Galerkin para la viga Timoshenko.

Ejemplo 1: Viga en voladizo con carga puntual en el extremo libre.

La viga en voladizo con carga puntual en el extremo derecho tiene las siguientes características:

$$L = 1m, I = \frac{1}{120} m^4, E = 2.05 \cdot 10^8 \frac{KN}{m^2}, \nu = 0.3, A = 0.1m^2, K_s = \frac{5}{6}, P = 100KN$$

Se determinarán los desplazamientos verticales de la viga mostrada a continuación:

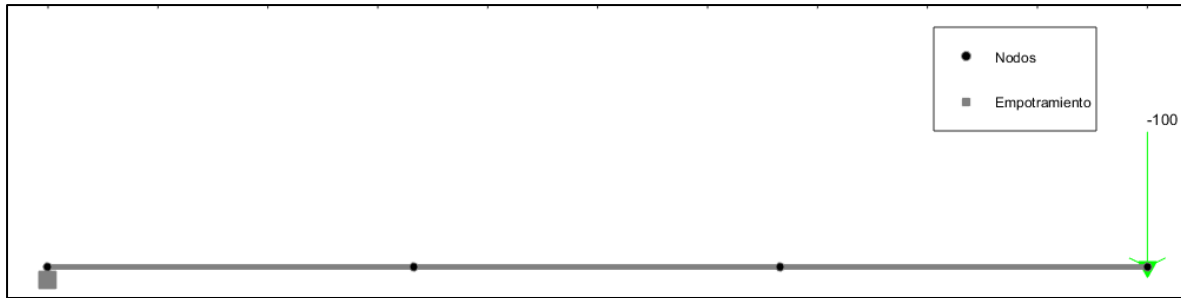


Figura 9. Gráfico en Matlab de la viga empotrada con carga puntual en el extremo libre. Fuente: propia.

1) *Discretización del dominio.*

- Se eligen 3 elementos de viga $e=1,2,3$
- Cada elemento de viga está definido con dos nodos, volviéndola un elemento lineal.
- Cada elemento de viga cuenta con 4 grados de libertad, 2 para la deflexión y dos para el giro.
- Cada elemento de viga tiene longitud $L_e = x_2^e - x_1^e$. Para este caso particular todos los elementos tienen la misma longitud igual a $L_e = L/e$.

2) *Matriz rigidez y vector fuerza para cada elemento.*

A partir de los datos del problema: $L = 1m$, $I = \frac{1}{120} m^4$, $E = 2.05 \cdot 10^8 \frac{KN}{m^2}$, $\nu = 0.3$, $A = 0.1m^2$, $K_s = \frac{5}{6}$, $P = 100KN$ y debido a que los elementos poseen la misma longitud se tiene que para los elementos 1,2,3, la matriz de rigidez es:

$$K^1 = K^2 = K^3 = \frac{EI}{6\lambda L^3} \begin{bmatrix} 6 & 3L & -6 & 3L \\ 3L & L^2(2 + 6\lambda) & -3L & L^2(1 - 6\lambda) \\ -6 & -3L & 6 & -3L \\ 3L & L^2(1 - 6\lambda) & -3L & L^2(2 + 6\lambda) \end{bmatrix} \text{ donde } L = \frac{1}{3}$$

siendo $\lambda = \frac{EI}{K_s GAL^2}$ y $G = \frac{E}{2(1+\nu)} = 78846153.85 \rightarrow \lambda = \frac{EI}{K_s GAL^2} = 2.34469$

Resolviendo la expresión se tiene que $\frac{EI}{6\lambda L^3} = 3287256.41$

Entonces:

$$K^1 = K^2 = K^3 = 3287256.41 \begin{bmatrix} 6 & 1 & -6 & 1 \\ 1 & 1.78535 & -1 & -1.45202 \\ -6 & -1 & 6 & -1 \\ 1 & -1.45202 & -1 & 1.78535 \end{bmatrix}$$

El vector de desplazamiento para cada elemento es:

$$a^e = \begin{bmatrix} \mu_1^e \\ \varphi_1^e \\ \mu_2^e \\ \varphi_2^e \end{bmatrix}$$

El vector frontera conformado por las fuerzas de reacción, para cada elemento queda expresado por:

$$f_b^e = \begin{bmatrix} V_1^e \\ M_1^e \\ V_2^e \\ M_2^e \end{bmatrix}$$

3) *Ensamblaje de las ecuaciones del elemento.*

Debido a los 4 desplazamientos globales, la matriz de rigidez ensamblada resulta ser de 8x8, y, además, hay que recordar que K es una matriz en banda y también simétrica. De lo analizado para la viga considerada con más de dos elementos y con cuatro desplazamientos por elemento, se puede deducir que el ensamble gráficamente es como se muestran en la Figura 10.

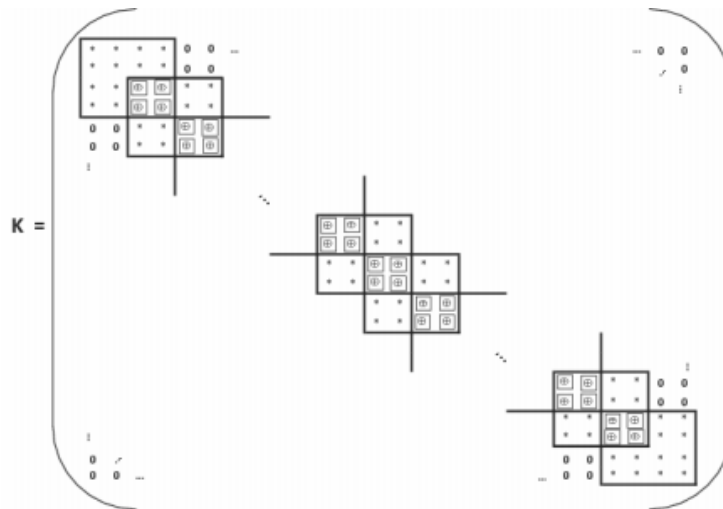


Figura 10. Ensamblado de la matriz de rigidez. Fuente: (Piedra, 2019)

$$K = 327875.41 \begin{bmatrix} 6 & 1 & -6 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1.78535 & -1 & -1.45202 & 0 & 0 & 0 & 0 \\ -6 & -1 & 6+6 & -1+1 & -6 & 1 & 0 & 0 \\ 1 & -1.45202 & -1+1 & 1.78535+1.78535 & -1 & -1.45202 & 0 & 0 \\ 0 & 0 & -6 & -1 & 6+6 & -1+1 & -6 & 1 \\ 0 & 0 & 1 & -1.45202 & -1+1 & 1.78535+1.78535 & -1 & -1.45202 \\ 0 & 0 & 0 & 0 & -6 & -1 & 6 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1.45202 & -1 & 1.78535 \end{bmatrix}$$

El vector frontera ensamblado para los tres elementos se resuelve como muestra en la Figura 11:

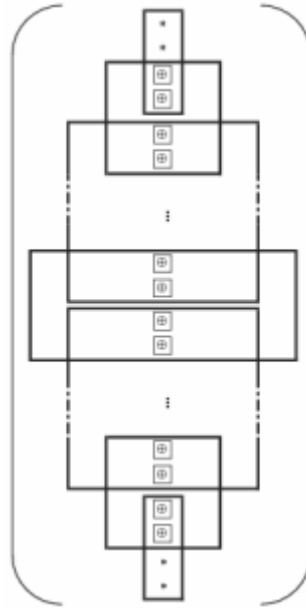


Figura 11. Ensamblado del vector frontera. Fuente: (Piedra, 2019).

$$f_b = \begin{bmatrix} V_1^1 \\ M_1^1 \\ V_2^1 + V_1^2 \\ M_2^1 + M_1^2 \\ V_2^2 + V_1^3 \\ M_2^2 + M_1^3 \\ V_2^3 \\ M_2^3 \end{bmatrix} = \begin{bmatrix} V_1 \\ M_1 \\ V_2 \\ M_2 \\ V_3 \\ M_3 \\ V_4 \\ M_4 \end{bmatrix} = \begin{bmatrix} V_1 \\ M_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -100 \end{bmatrix}$$

Debido a que el ejemplo no cuenta con cargas distribuidas el vector fuerza es igual al vector $f_b = f$.

El vector de desplazamiento global es:

$$a = [u_1 \ \varphi_1 \ u_2 \ \varphi_2 \ u_3 \ \varphi_3 \ u_4 \ \varphi_4]^T$$

Imponiendo las condiciones de frontera debido a que la viga se encuentra empotrada en el nodo 1 se tiene que para $a^1 \rightarrow u_1 = 0$ y $\varphi_1 = 0$, mientras para a^2, a^3, a^4 u y φ son diferentes a 0. Esto eliminaría las columnas y filas 1 y 2 de la matriz de rigidez ensamblada y las filas 1 y 2 del vector fuerza.

4) Solución del sistema discreto.

Para encontrar la solución del sistema reducido $\bar{k} \cdot \bar{a} = \bar{f}$, obteniendo así los giros en los nodos y sus desplazamientos verticales. Para esto se procede mediante eliminación Gaussiana.



$$\bar{k} \cdot \bar{a} = \bar{f}$$

$$3278756.41 \cdot \begin{bmatrix} (12)u_2 & 0 & -6u_3 & 1\varphi_3 & 0 & 0 \\ 0 & (3.5707)\varphi_2 & -1u_3 & -1.45202\varphi_3 & 0 & 0 \\ -6u_2 & -1\varphi_2 & (12)u_3 & 0 & -6u_4 & 1\varphi_4 \\ 1u_2 & -1.45202\varphi_2 & 0 & (3.5707)\varphi_3 & -1u_4 & -1.45202\varphi_4 \\ 0 & 0 & -6u_3 & -1\varphi_3 & 6u_4 & -1\varphi_4 \\ 0 & 0 & 1u_3 & -1.45202\varphi_3 & -1u_4 & 1.78535\varphi_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -100 \end{bmatrix}$$

Las deflexiones y los giros en los nodos son:

$$-0.0768 * 1.0e - 04 m$$

$$-0.1569 Rad$$

$$-0.1954 * 1.0e - 04 m$$

$$-0.2511 Rad$$

$$-0.3354 * 1.0e - 04 m$$

$$-0.2826 Rad$$

La deflexión máxima se da en el último nodo y resulta ser de $-0.3354 * 1.0e - 04 m$ o $0.33mm$.

5) Polinomio de aproximación.

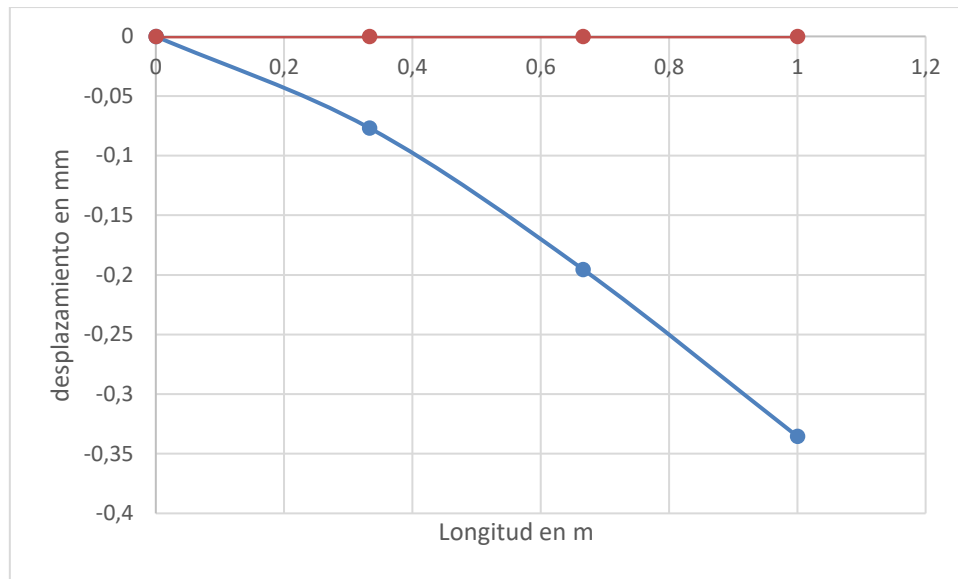
Ya que se ha considerado una relación lineal para todos elementos basada en las ecuaciones 2.69, 2.70 y 2.71 se tiene que las ecuaciones aproximadas de deflexión para los 3 elementos son:

$$\text{Elemento 1: } \left\{ \left(1 - \frac{x}{0.333} \right) \cdot 0 \right\} + \left\{ \left(\frac{x}{0.333} \right) \cdot -0.0768 \cdot 1.0e - 04 \right\} \text{ cuando } x \in [0,1/3]$$

$$\text{Elemento 2: } \left\{ \left(1 - \frac{x}{0.333} \right) \cdot -0.0768 \cdot 1.0e - 04 \right\} + \left\{ \left(\frac{x}{0.333} \right) \cdot -0.1954 \cdot 1.0e - 04 \right\} \text{ cuando } x \in [0,1/3]$$

$$\text{Elemento 3: } \left\{ \left(1 - \frac{x}{0.333} \right) \cdot -0.1954 \cdot 1.0e - 04 \right\} + \left\{ \left(\frac{x}{0.333} \right) \cdot -0.3354 \cdot 1.0e - 04 \right\} \text{ cuando } x \in [0,1/3]$$

Graficando las ecuaciones de aproximación se tiene:



Gráfica 2. Polinomios de aproximación para viga empotrada con carga puntual.

Debido a que la resolución por el método de los elementos finitos sigue un proceso iterativo, mediante el software Matlab se facilita el procedimiento para un mayor número de elementos. A continuación, se muestran las deflexiones máximas, vector desplazamiento y comparación de los resultados obtenidos con el programa para 5,10,100 y 1000 elementos.

- Para 5 elementos:

Vector desplazamiento:

$$U_p = 1.0e - 04 \cdot \begin{bmatrix} -0.0408 \\ -0.1040 \\ -0.1002 \\ \vdots \\ \vdots \\ \vdots \\ -0.3429 \\ -0.2890 \end{bmatrix} m$$

$$y_{m\acute{a}x} = 0.03429mm$$

$$\varphi_{m\acute{a}x} = -0.2890 \cdot 10^{-5} \text{ rad}$$

Se puede comparar gráficamente mediante el software Matlab la solución de Timoshenko (línea azul) con la analítica, explicada más adelante (línea roja):

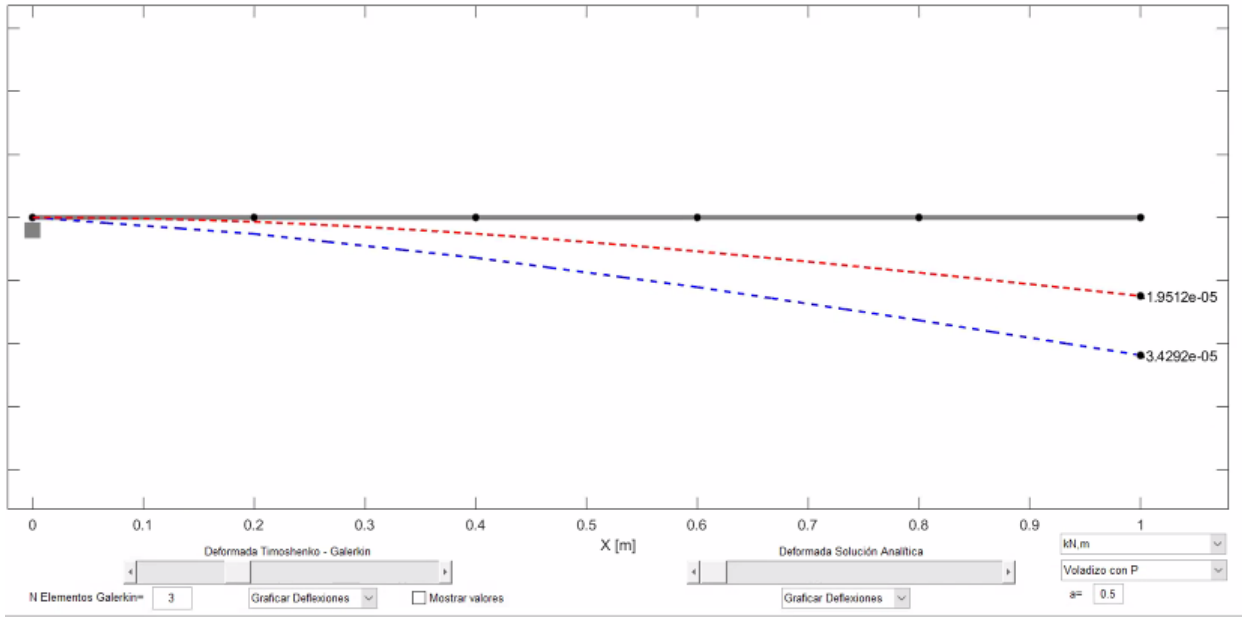


Figura 12. Deflexiones en la viga empotrada con carga puntual en el extremo libre para 5 elementos. Línea azul-solución de Timoshenko; línea roja-solución analítica. Fuente: propia

- Para 10 elementos:

Vector desplazamiento:

$$U_p = 1.0e - 04 \cdot \begin{bmatrix} -0.0180 \\ -0.0554 \\ -0.0412 \\ \vdots \\ \vdots \\ \vdots \\ -0.3462 \\ -0.2917 \end{bmatrix} m$$

$$y_{m\acute{a}x} = -0,03462mm$$

$$\varphi_{m\acute{a}x} = -2.926 \cdot 10^{-5} \text{ rad}$$

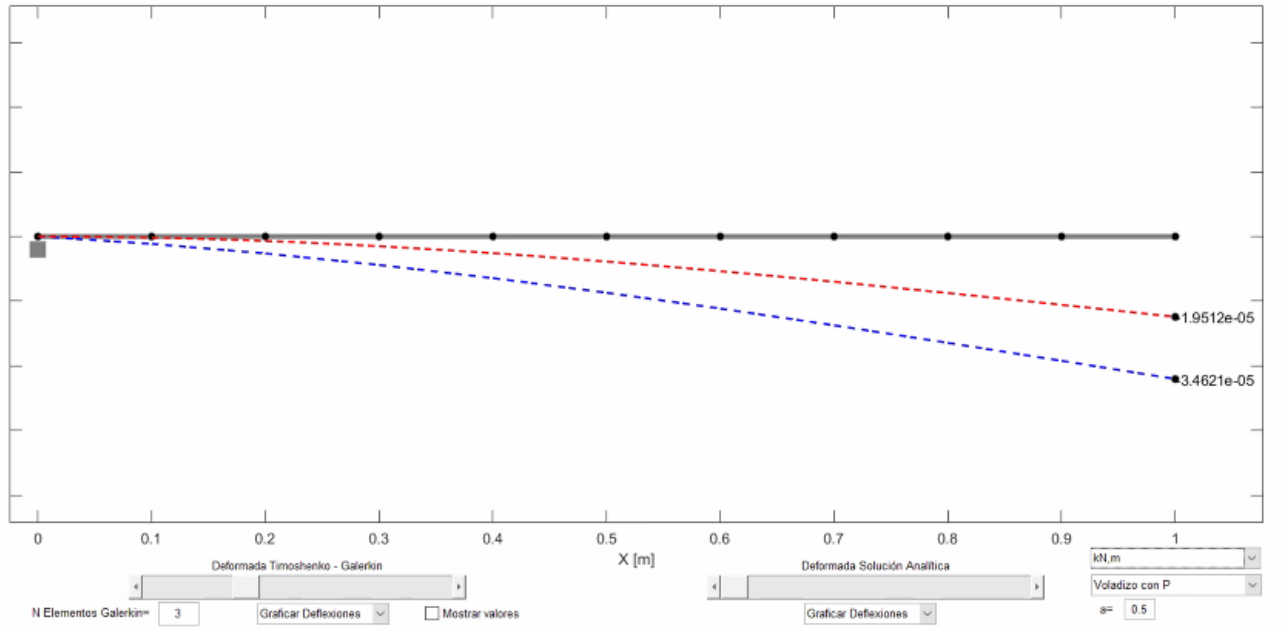


Figura 13. Deflexiones en la viga empotrada con carga puntual en el extremo libre para 10 elementos. Línea azul-solución de Timoshenko; línea roja-solución analítica. Fuente: propia

- Para 100 elementos:

Vector desplazamiento:

$$U_p = 1.0e - 04 \cdot \begin{bmatrix} -0.0016 \\ -0.0058 \\ -0.0032 \\ \vdots \\ \vdots \\ \vdots \\ -0.3473 \\ -0.2927 \end{bmatrix} m$$

$$y_{m\acute{a}x} = -0.03473mm$$

$$\varphi_{m\acute{a}x} = -0.2927 \cdot 10^{-5} \text{ rad}$$

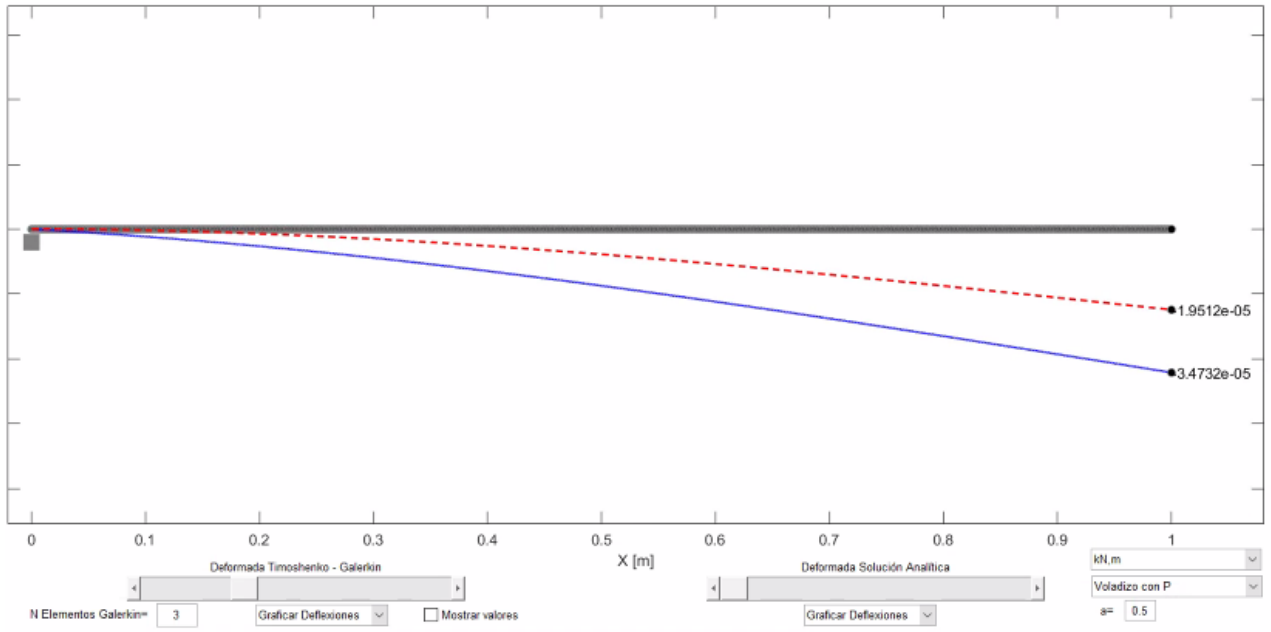


Figura 14. Deflexiones en la viga empotrada con carga puntual en el extremo libre para 100 elementos. Línea azul-solución de Timoshenko; línea roja-solución analítica. Fuente: propia

- Para 1000 elementos:

Vector desplazamiento:

$$U_p = 1.0e - 04 \cdot \begin{bmatrix} -0.0002 \\ -0.0006 \\ -0.0003 \\ \vdots \\ \vdots \\ \vdots \\ -0.3473 \\ -0.2927 \end{bmatrix} m$$

$$y_{m\acute{a}x} = -0.03473mm$$

$$\varphi_{m\acute{a}x} = -0.2927 \cdot 10^{-5} \text{ rad}$$

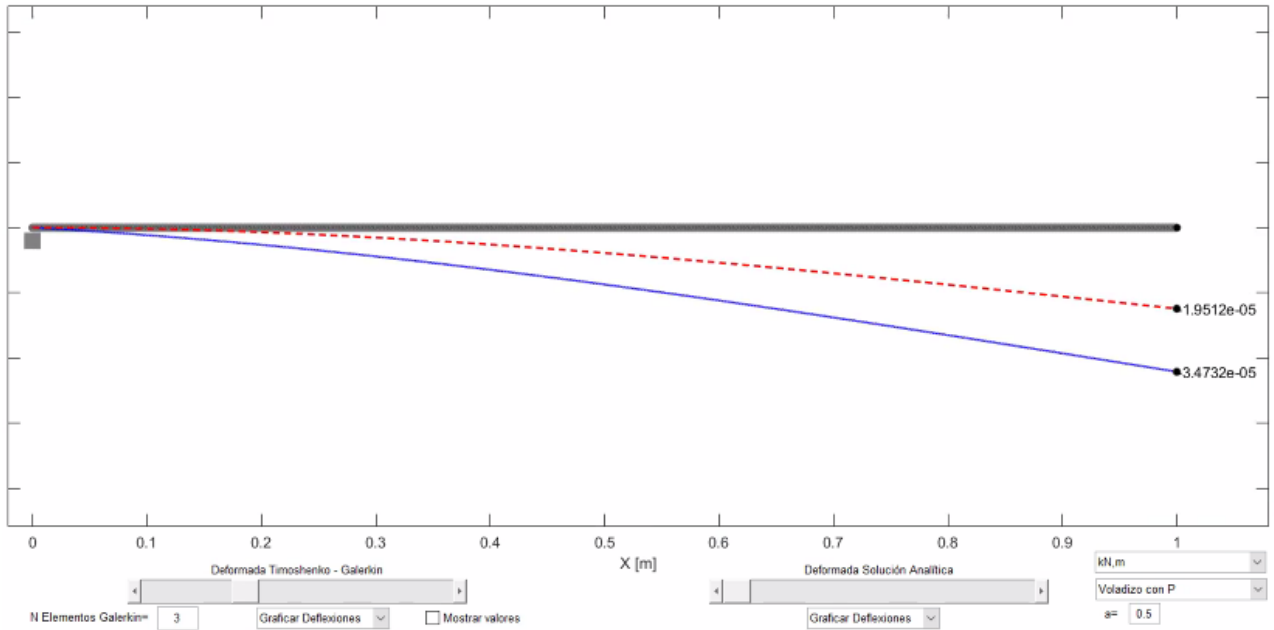


Figura 15. Deflexiones en la viga empotrada con carga puntual en el extremo libre para 1000 elementos. Línea azul-solución de Timoshenko; línea roja-solución analítica. Fuente: propia

De los resultados obtenidos se puede observar que a medida que el número de elementos crece la solución comienza a converger en -0.03473mm , que es la deflexión máxima en el nodo ubicado en el extremo libre de la viga.

También es de interés conocer la diferencia que existe entre los resultados obtenidos con otros métodos de la teoría clásica de deformación de vigas en la que se destacan por ejemplo la doble integración, viga conjugada, área de momentos entre otros. Utilizando cualquiera de los métodos antes mencionados la ecuación de deflexión para la viga del ejemplo es de la forma:

$$y = -\frac{P}{6EI} \cdot (3Lx^2 - x^3)$$

Para este caso se sabe que la deflexión máxima ocurre cuando $x = L$, entonces la expresión resulta ser:

$$y_{\text{máx}} = -\frac{PL^3}{3EI}$$

reemplazando los datos del problema se obtiene:

$$y_{\text{máx}} = -\frac{100 * 1^3}{3 * 2.05 * 10^8 * \frac{1}{120}}$$

$$y_{\text{máx}} = -1.951 * 10^{-5}m$$

$$y_{\text{máx}} = -0,0195mm$$

El giro se expresa como:



$$\varphi_{m\acute{a}x} = -\frac{PL^2}{2EI}$$

$$\varphi_{m\acute{a}x} = -2.926 * 10^{-5}$$

La deflexi3n m\acute{a}xima es de -0.0195 mm (l\acute{ı}nea roja de las gr\acute{a}ficas anteriores), siendo la discrepancia entre los resultados de 0.01523mm. La diferencia entre la deflexi3n obtenida por Timoshenko y la deflexi3n anal\acute{ı}tica obtenida por la teor\acute{ı}a de mec\acute{a}nica de materiales se debe a la consideraci3n del esfuerzo cortante que hace Timoshenko. Para observar de mejor manera esta diferencia se desarrollar\acute{a} el mismo caso para la discretizaci3n de un solo elemento de longitud L .

Partiendo de la ecuaci3n 2.67 para un elemento se tendr\acute{ı}a:

$$K^1 u^1 = F^1$$

$$\begin{bmatrix} \frac{K_s GA}{L} & \frac{K_s GA}{2} & -\frac{K_s GA}{L} & \frac{K_s GA}{2} \\ \frac{K_s GA}{2} & \frac{K_s GA}{3} + \frac{EI}{L} & -\frac{K_s GA}{2} & \frac{K_s GA}{6} - \frac{EI}{L} \\ -\frac{K_s GA}{L} & -\frac{K_s GA}{2} & \frac{K_s GA}{L} & -\frac{K_s GA}{2} \\ \frac{K_s GA}{2} & \frac{K_s GA}{6} - \frac{EI}{L} & -\frac{K_s GA}{2} & \frac{K_s GA}{3} + \frac{EI}{L} \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ \varphi_1 \\ u_2 \\ \varphi_2 \end{bmatrix} = \begin{bmatrix} V_1 \\ M_1 \\ P \\ 0 \end{bmatrix}$$

Como no existe desplazamiento ni giro en el empotramiento se tiene que $u_1 = 0$ y $\varphi_1 = 0$. Aplicando al vector desplazamiento y eliminando las filas que se vuelven cero se tiene:

$$\begin{bmatrix} \frac{K_s GA}{L} & -\frac{K_s GA}{2} \\ -\frac{K_s GA}{2} & \frac{K_s GA}{3} + \frac{EI}{L} \end{bmatrix} \cdot \begin{bmatrix} u_2 \\ \varphi_2 \end{bmatrix} = \begin{bmatrix} P \\ 0 \end{bmatrix}$$

Resolviendo la inversa de la matriz K (K^{-1}) y despejando el vector de desplazamiento se tiene

$$\begin{bmatrix} u_2 \\ \varphi_2 \end{bmatrix} = \begin{bmatrix} \frac{L}{K_s GA} + \frac{L^3}{3EI} & -\frac{L^2}{2EI} \\ \frac{L^2}{2EI} & \frac{L}{EI} \end{bmatrix} \begin{bmatrix} P \\ 0 \end{bmatrix}$$

Resolviendo para u_2 :

$$u_2 = \left(\frac{L}{K_s GA} + \frac{L^3}{3EI} \right) \cdot P$$

Se puede observar que la diferencia con la expresión obtenida por el método clásico, es el termino $\frac{L}{K_s GA}$, por lo tanto, la deflexión en el extremo de la viga es:

- sin esfuerzo cortante $\frac{L^3}{3EI} \cdot P$
- con esfuerzo cortante $(\frac{L}{K_s GA} + \frac{L^3}{3EI}) \cdot P$

Ejemplo 2: Viga simplemente apoyada con carga distribuida q .

La viga simplemente apoyada con carga distribuida q a lo largo de su longitud tiene las siguientes características: $L = 2m, E = 2.05 \times 10^8, \nu = 0.2, A = 0.03m^2, I = 0.00045m^4, K_s = \frac{5}{6}$ y $q = 10 \frac{KN}{m}$.

Siguiendo el mismo procedimiento del ejemplo 1 y mediante la programación realizada en el software Matlab se obtienen los siguientes resultados para 2, 6, 50 y 100 elementos:

- Para 2 elementos:

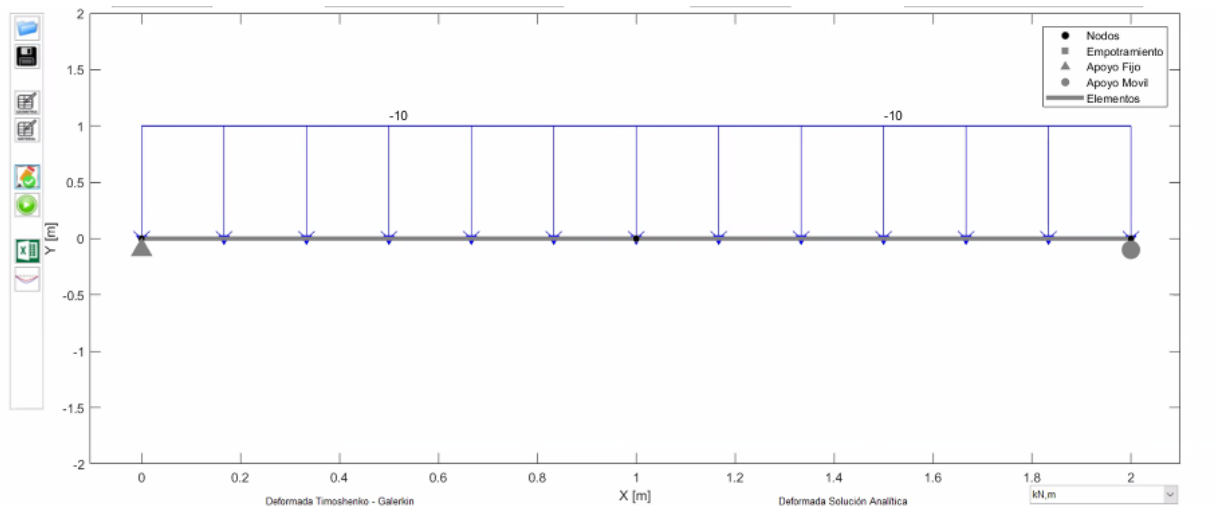


Figura 16. Modelo en Matlab para 2 elementos de la viga simplemente apoyada con carga distribuida q a lo largo de su longitud. Fuente: propia.

Vector desplazamiento:

$$U_p = 1.0e - 05 \cdot \begin{bmatrix} -0.9252 \\ -0.6968 \\ 0 \\ 0.9252 \end{bmatrix} m$$

La deflexión máxima ocurre en el centro de la viga, en el nodo 2, es decir $y_{m\acute{a}x} \rightarrow u_2 = -0.006968mm$. De igual manera se puede comparar gráficamente mediante el software Matlab la solución de Timoshenko (línea azul) con la analítica, explicada más adelante (línea roja):

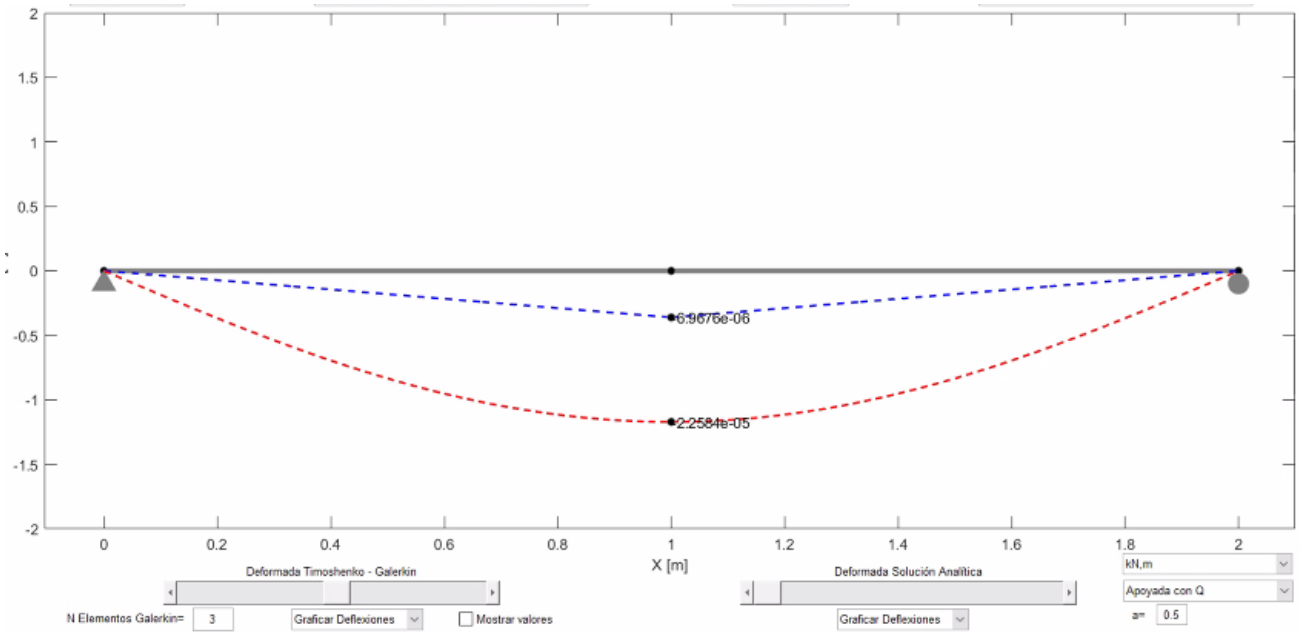


Figura 17. Deflexiones en la viga simplemente apoyada con carga distribuida para 2 elementos. Línea azul-solución de Timoshenko; línea roja-solución analítica. Fuente: propia.

■ Para 6 elementos:

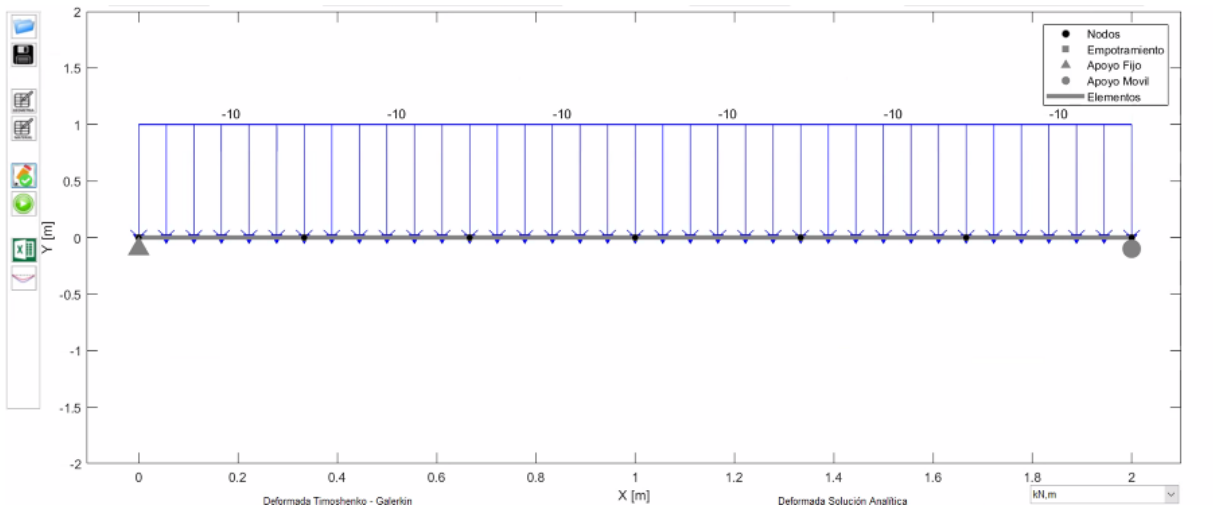


Figura 18. Modelo en Matlab para 6 elementos de la viga simplemente apoyada con carga distribuida q a lo largo de su longitud. Fuente: propia.

Vector desplazamiento:

$$U_p = 1.0e - 04 \cdot \begin{bmatrix} -0.2893 \\ -0.1026 \\ -0.0003 \\ \vdots \\ -0.2011 \\ \vdots \\ -0.1026 \\ 0.2893 \end{bmatrix} m$$

La deflexión máxima ocurre en el centro de la viga, en el nodo 4, es decir $y_{m\acute{a}x} \rightarrow u_4 = -0.02011mm$

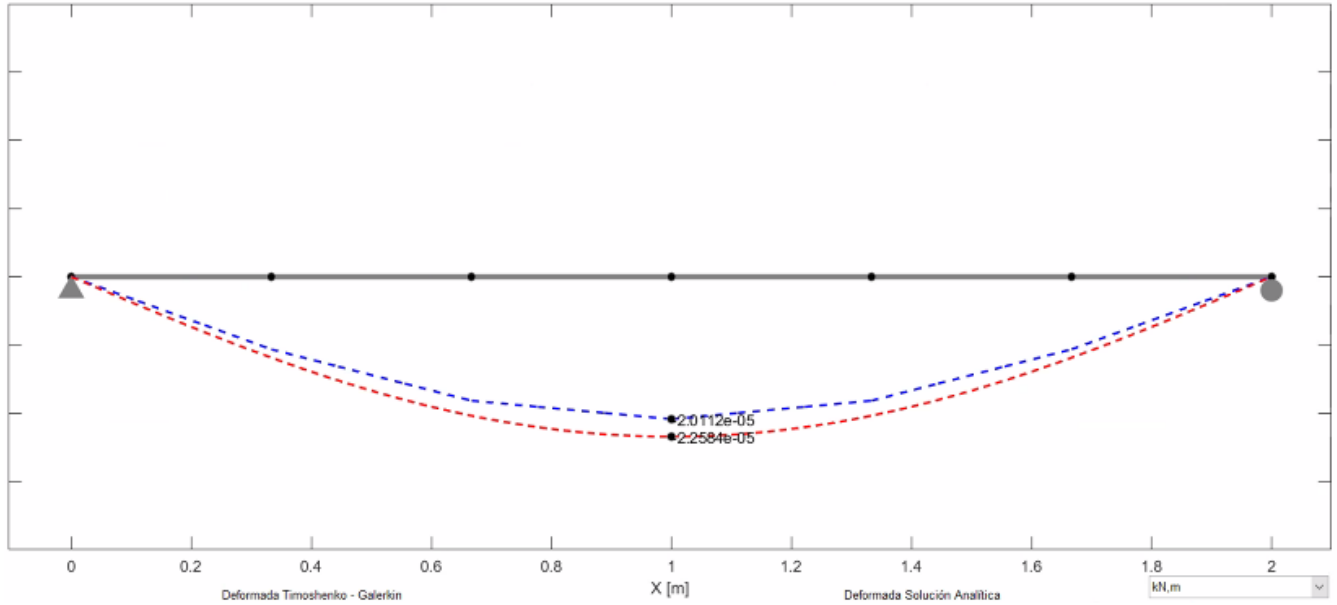


Figura 19. Deflexiones en la viga simplemente apoyada con carga distribuida para 6 elementos. Línea azul-solución de Timoshenko; línea roja-solución analítica. Fuente: propia.

- Para 50 elementos:

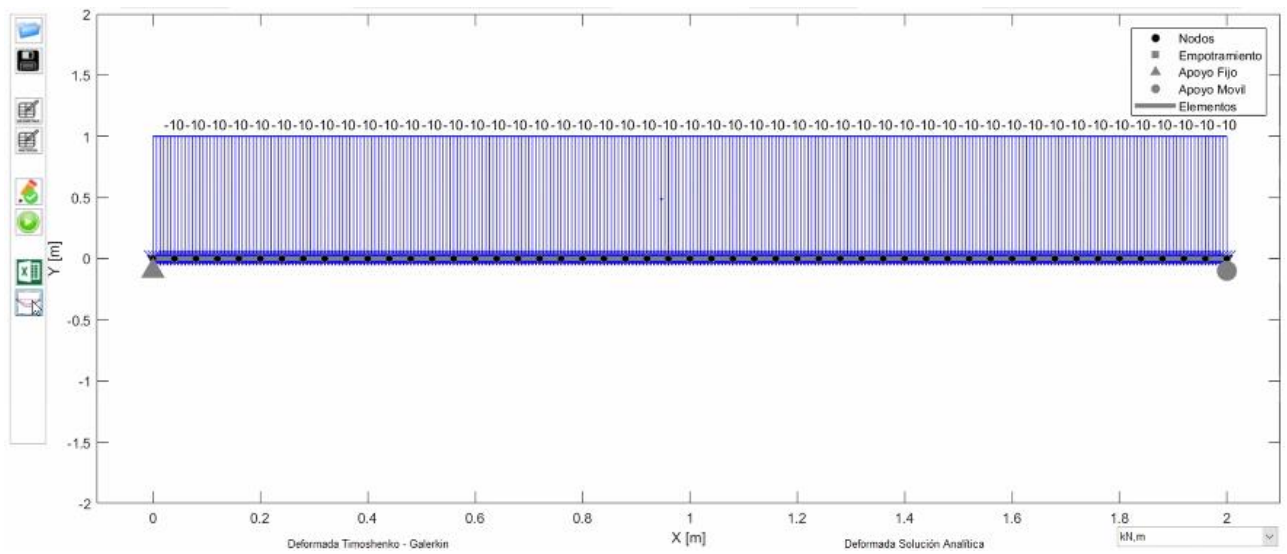


Figura 20. Modelo en Matlab para 50 elementos de la viga simplemente apoyada con carga distribuida q a lo largo de su longitud. Fuente: propia.

Vector desplazamiento:

$$U_p = 1.0e - 04 \cdot \begin{bmatrix} -0.3601 \\ -0.0162 \\ -0.3592 \\ \vdots \\ -\mathbf{0.2484} \\ \vdots \\ -0.0162 \\ 0.3601 \end{bmatrix} m$$

La deflexión máxima ocurre en el centro de la viga, en el nodo 26, es decir $y_{m\acute{a}x} \rightarrow u_{26} = -0.02484mm$.

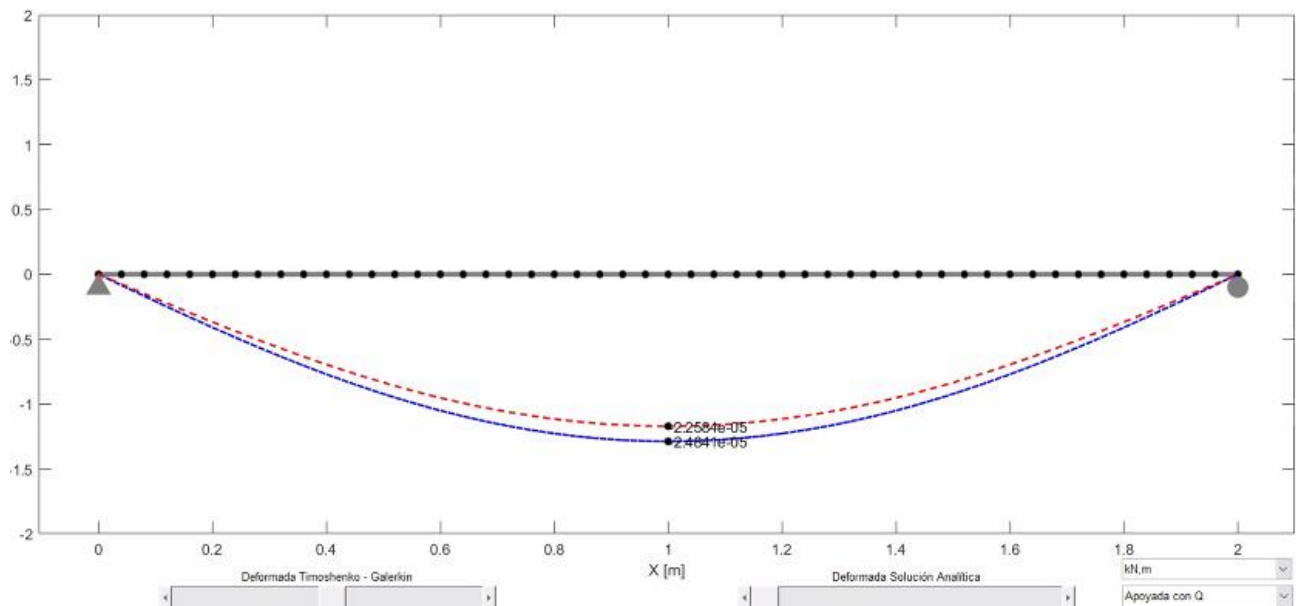


Figura 21. Deflexiones en la viga simplemente apoyada con carga distribuida para 50 elementos. Línea azul-solución de Timoshenko; línea roja-solución analítica. Fuente: propia.

- Para 100 elementos:

Vector desplazamiento:

$$U_p = 1.0e - 04 \cdot \begin{bmatrix} -0.3610 \\ -0.0081 \\ -0.3592 \\ \vdots \\ -\mathbf{0.2490} \\ \vdots \\ -0.0081 \\ 0.3610 \end{bmatrix} m$$

La deflexión máxima ocurre en el centro de la viga, en el nodo 51, es decir $y_{m\acute{a}x} \rightarrow u_{51} = -0.02490mm$.

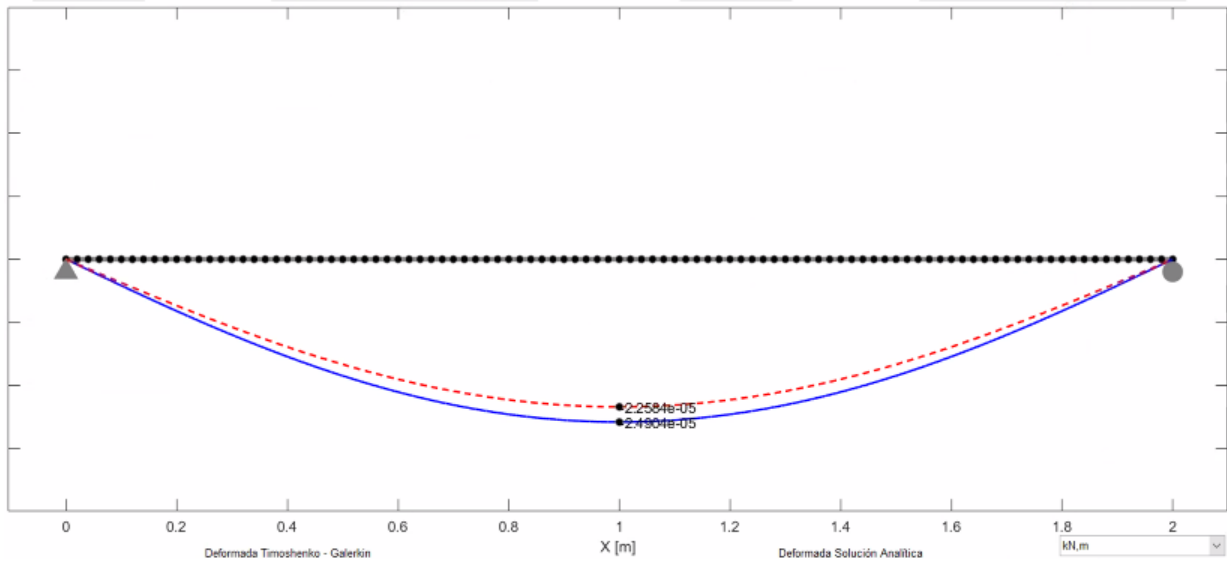


Figura 22. Deflexiones en la viga simplemente apoyada con carga distribuida para 100 elementos. Línea azul-solución de Timoshenko; línea roja-solución analítica. Fuente: propia.

La ecuación “exacta” de la teoría de la deflexión de vigas para el caso de una viga simplemente apoyada (línea roja de las anteriores graficas) con carga distribuida es de la forma:

$$y = \frac{-q \cdot x \cdot (L^3 - 2Lx^2 + x^3)}{24 \cdot E \cdot I}$$

Se conoce que la deflexión máxima ocurre en el centro del claro de la viga es decir para $x = \frac{L}{2}$, quedando la expresión anterior como:

$$y_{m\acute{a}x} = \frac{-5 \cdot q \cdot L^4}{384 \cdot E \cdot I}$$

Reemplazando los datos del problema:

$$y_{m\acute{a}x} = \frac{-5 \cdot 10 \cdot 2^4}{384 \cdot 2.05 \cdot 10^8 \cdot 0.00045}$$

$$y_{m\acute{a}x} = 0.02258mm$$

La diferencia entre la ecuación “exacta” y la de Timoshenko radica en la consideración del cortante tal y como se explicó en el ejemplo anterior.

Ejemplo 3. Viga simplemente apoyada con carga puntual a una distancia x de uno de los apoyos.

Se tiene una viga de longitud $L = 1m$ con carga puntual a $0.2m$ desde su apoyo izquierdo con los siguientes datos: $I = \frac{1}{120} m^4$, $E = 2.05 \cdot 10^8 \frac{KN}{m^2}$, $\nu = 0.3$, $A = 0.1m^2$, $K_s = \frac{5}{6}$, $P = 100KN$. Siguiendo el mismo procedimiento del ejemplo 1 y 2 mediante la programación

realizada en el software Matlab se obtienen los siguientes resultados para 5, 55 y 100 elementos:

- Para 5 elementos.

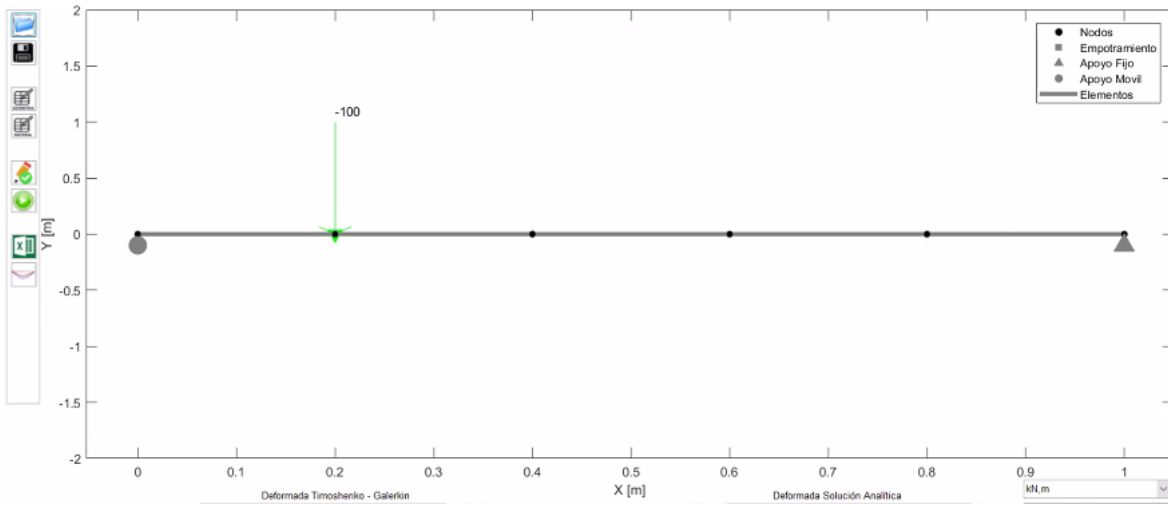


Figura 23. Modelo en Matlab para 5 elementos de la viga simplemente apoyada con carga puntual a una distancia a del apoyo izquierdo. Fuente: propia.

Vector desplazamiento:

$$U_p = 1.0e - 05 \cdot \begin{bmatrix} -0.277 \\ -0.2710 \\ -0.1848 \\ -\mathbf{0.2897} \\ \vdots \\ 0.1617 \\ 0.1848 \end{bmatrix} m$$

La deflexión máxima ocurre en el nodo 2, es decir donde se ubica la carga puntual P $y_{m\acute{a}x} \rightarrow u_2 = -0.002897mm$.

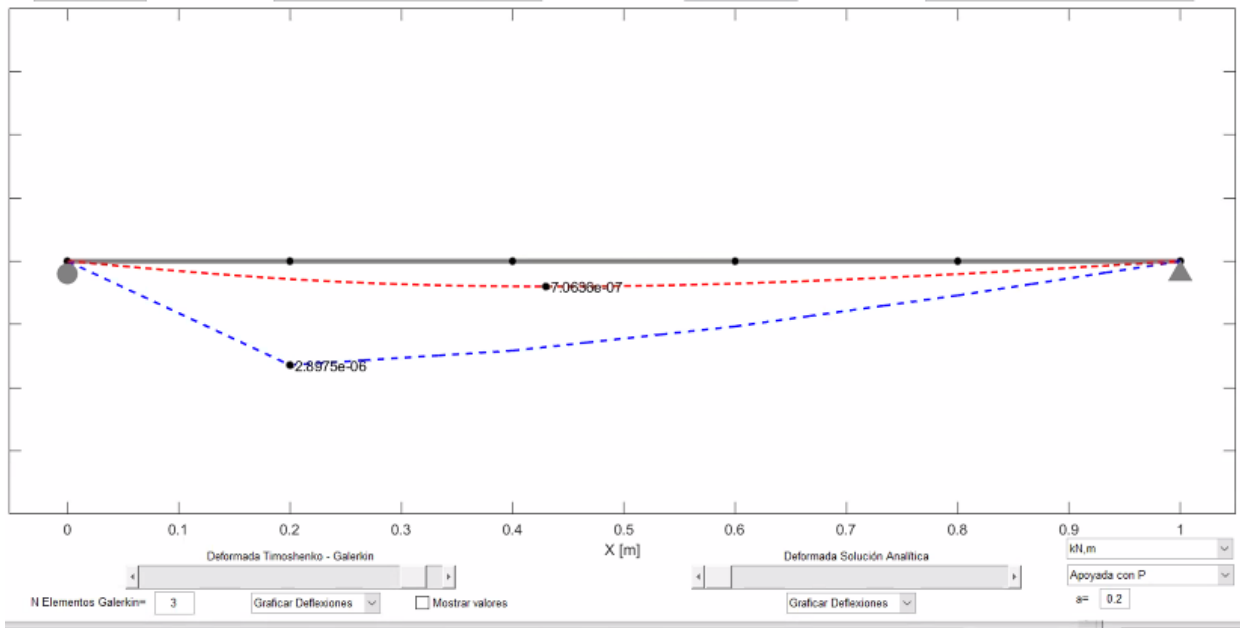


Figura 24. Deflexiones en la viga simplemente apoyada con carga puntual para 5elementos. Línea azul-solución de Timoshenko; línea roja-solución analítica. Fuente: propia.

■ Para 55 elementos.

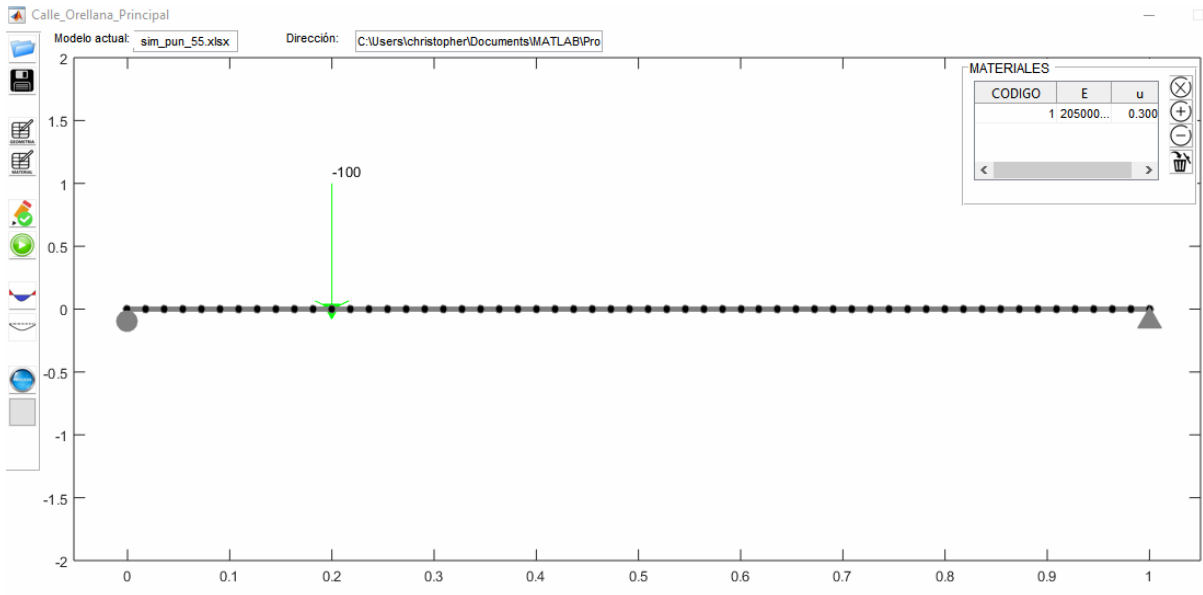


Figura 25. Modelo en Matlab para 55 elementos de la viga simplemente apoyada con carga puntual a una distancia a del apoyo izquierdo. Fuente: propia.



Vector desplazamiento:

$$U_p = 1.0e - 05 \cdot \begin{bmatrix} -0.2809 \\ -0.0272 \\ -0.2802 \\ \vdots \\ -0.2934 \\ \vdots \\ 0.1871 \\ 0.1873 \end{bmatrix} m$$

La deflexión máxima ocurre en el nodo 12, es decir donde se ubica la carga puntual P
 $y_{m\acute{a}x} \rightarrow u_{12} = -0.002934mm$.

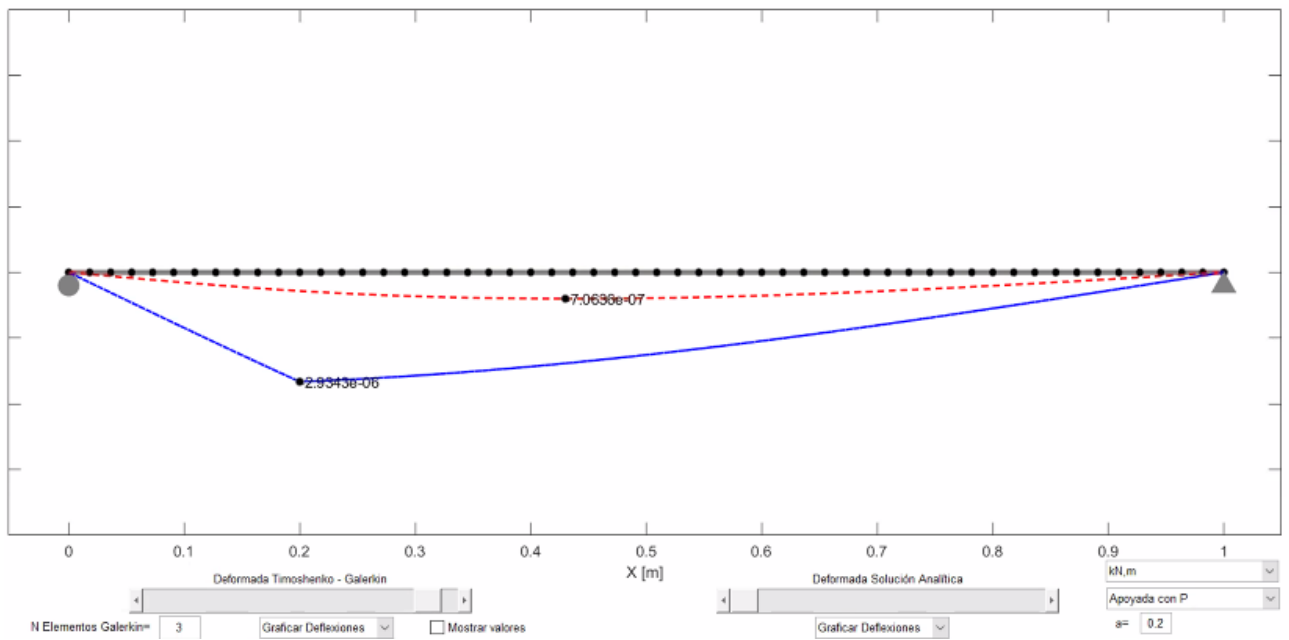


Figura 26. Deflexiones en la viga simplemente apoyada con carga puntual para 55 elementos. Línea azul-solución de Timoshenko; línea roja-solución analítica. Fuente: propia.

- Para 100 elementos.

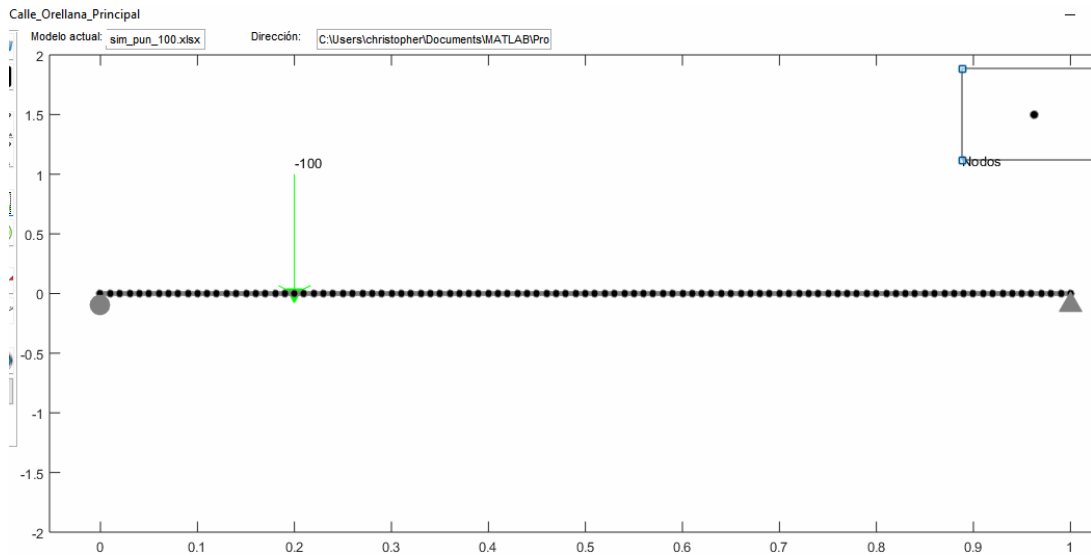


Figura 27. Modelo en Matlab para 100 elementos de la viga simplemente apoyada con carga puntual a una distancia a del apoyo izquierdo. Fuente: propia.

Vector desplazamiento:

$$U_p = 1.0e - 05 \cdot \begin{bmatrix} -0.2810 \\ -0.0150 \\ -0.2807 \\ \vdots \\ -0.2935 \\ \vdots \\ 0.1873 \\ 0.1873 \end{bmatrix} m$$

La deflexión máxima ocurre en el nodo 21, es decir donde se ubica la carga puntual P $y_{m\acute{a}x} \rightarrow u_{21} = -0.002935mm$

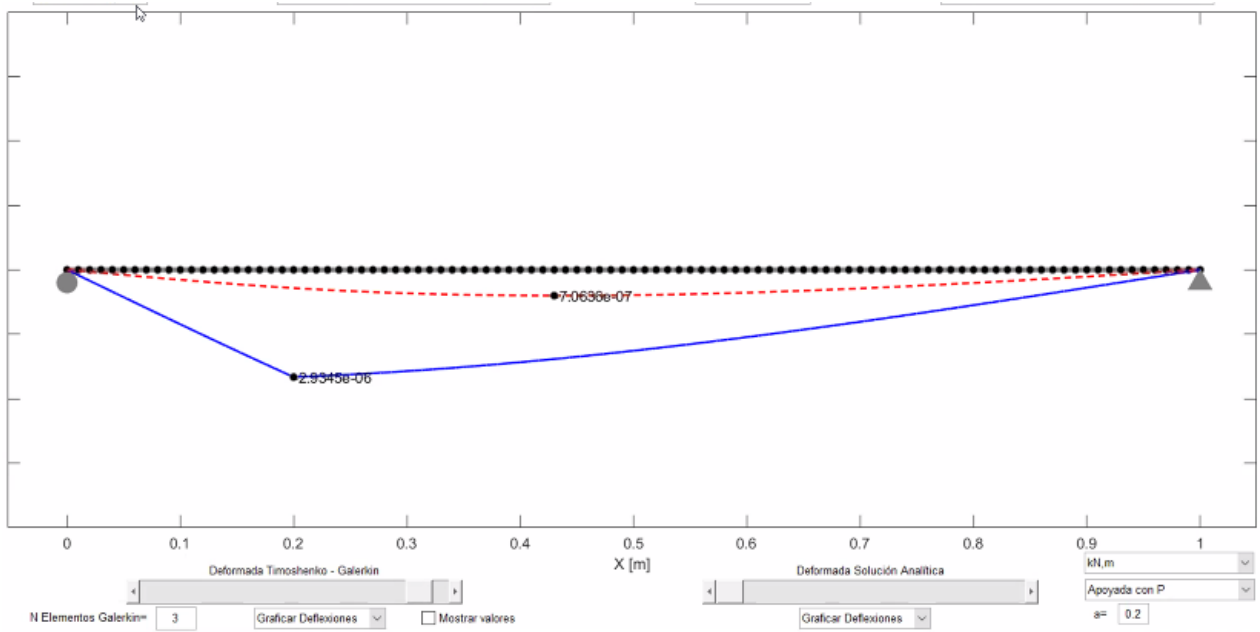


Figura 28. Deflexiones en la viga simplemente apoyada con carga puntual para 100 elementos. Línea azul-solución de Timoshenko; línea roja-solución analítica. Fuente: propia.

Para este mismo caso, se obtienen las deflexiones para diferentes longitudes de la viga:

- $L=1.5m$

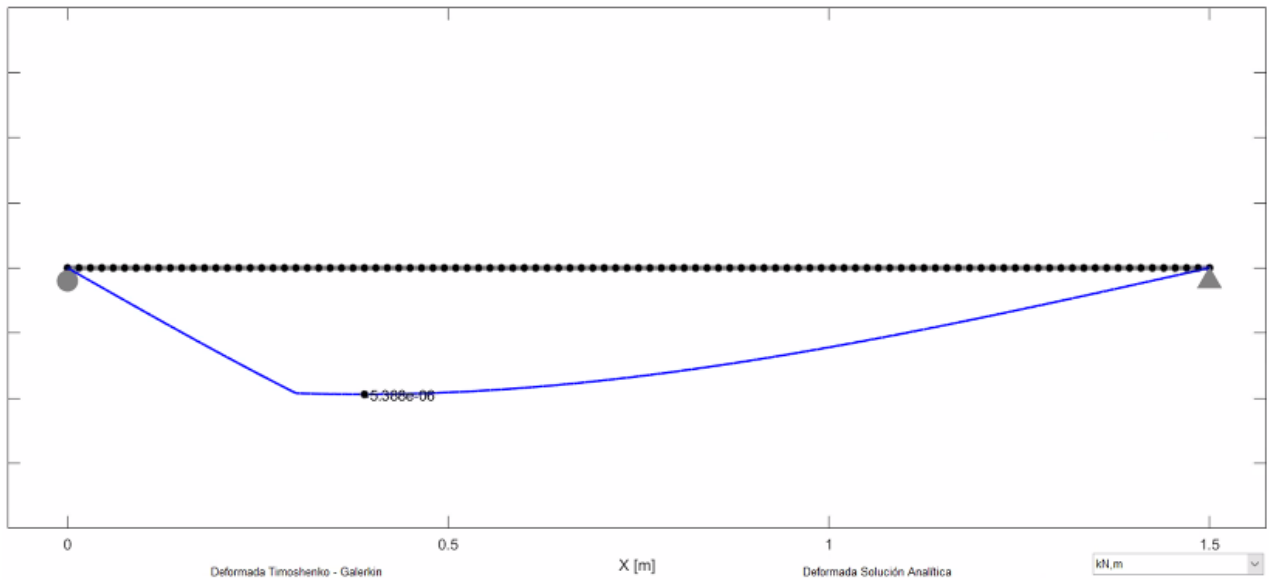


Figura 29. Deflexión en la viga simplemente apoyada $L= 1,5m$ con carga puntual para 100 elementos. Línea azul-solución de Timoshenko.

- $L=2\text{m}$

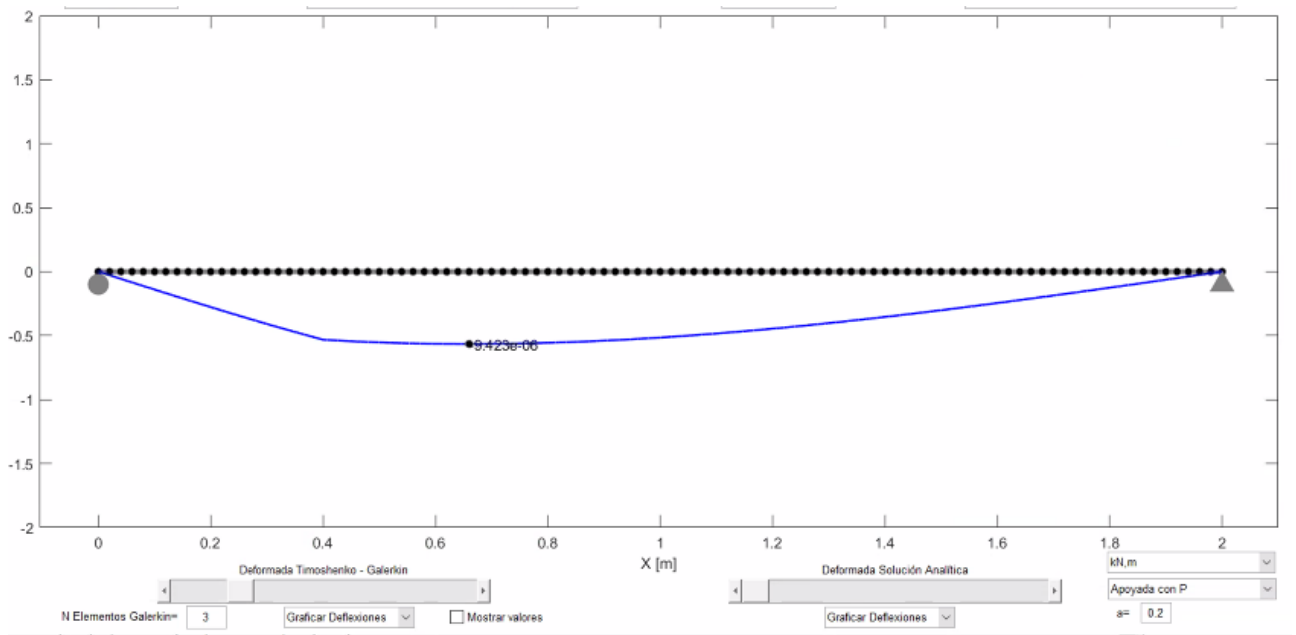


Figura 30. Deflexión en la viga simplemente apoyada $L= 2\text{m}$ con carga puntual para 100 elementos. Línea azul-solución de Timoshenko.

De las figuras 28, 29 y 30 se puede observar que, cuando se eligen 100 elementos lineales y diferentes longitudes de viga para aplicar la teoría de Timoshenko mediante el método de Galerkin, al aumentar la relación de esbeltez $\lambda = L/h$ la deflexión máxima comienza a alejarse de la dirección donde se aplica la carga puntual.



CAPÍTULO 4.

4. Conclusiones.

- El método de Galerkin permite el uso de funciones de prueba en la solución de EDOs; en el método las condiciones de frontera ayudan cuando no hay una relación física clara en un mismo punto de la función, pero sí cuando se tiene el valor en los extremos del dominio de la función. En el ejemplo desarrollado se puede apreciar que mientras el orden de la ecuación de aproximación es mayor la función-solución aproximada tiende a ser la exacta.
- La utilidad de los métodos variacionales desde el punto de vista del cálculo numérico es que proporcionan una forma sencilla de obtener soluciones aproximadas, lo cual ha quedado comprobado mediante la implementación de la variante de Galerkin para la formulación de elementos finitos de la teoría de Timoshenko, utilizando funciones de aproximación y en donde la solución para el problema discreto se puede encontrar explícitamente como una combinación lineal de las funciones base con coeficientes desconocidos, es decir, se obtiene un sistema de n ecuaciones lineales con n incógnitas que permite determinar los coeficientes.
- Las aproximaciones para la deflexión y giro son de clase C_0 , lo que quiere decir que garantizan que el desplazamiento axial es continuo dentro del elemento y entre elementos, además que las funciones de forma N_{wi} y $N_{\phi i}$ que acompañan las aproximaciones interpolan dentro del elemento únicamente los desplazamientos y giros correspondientes al nodo en el que se encuentran. De ahí que el aporte del MEF (Método de los elementos finitos) al método de Galerkin consiste en una forma sistemática y de fácil automatización para la construcción de las funciones de forma a partir de funciones definidas de manera local.
- La utilización de la herramienta computacional MATLAB ha facilitado la resolución de los diferentes modelos de vigas y cargas aplicadas de una manera eficaz y rápida, debido a la naturaleza reiterativa del proceso de solución, convierte la programación computacional en una poderosa alternativa para el ensamble de sistemas matriciales de gran tamaño y la obtención de resultados.
- Los resultados obtenidos de los modelos implementados debido a la teoría de Timoshenko con la formulación de Galerkin varían con respecto a la teoría clásica de deflexión de vigas. Como se explicó en el ejemplo 1 de la sección 3.2 correspondiente a la viga empotrada con carga puntual en el extremo libre, la discrepancia radica en la acción considerable del cortante siendo la diferencia con la expresión obtenida por el método clásico el término $\frac{L}{K_S GA}$ para este caso. Se puede observar entonces que depende en parte de la geometría de la viga, siendo de interés la esbeltez ($\lambda = L/h$), ya que es



conocido que valores de λ altos (vigas esbeltas) provocan que el efecto del cortante sea despreciable haciendo que la solución coincida con la de la teoría clásica. Esta misma característica mecánica interviene en los resultados obtenidos para la deflexión de una viga simplemente apoyada con carga puntual para diferentes valores de longitud L (1, 1.5 y 2m), en donde se puede observar que al aumentar la esbeltez la deflexión máxima de la viga comienza a alejarse del punto de aplicación de la carga puntual.



5. Referencias.

- Cheng, X., Han, W., & Huang, H. (1997). Finite element methods for Timoshenko beam, circular arch and Reissner-Mindlin plate problems. *Journal of Computational and Applied Mathematics*.
- Edem, I. (2006). The Exact Two-Node Timoshenko Beam Finite Element Using Analytical Bending and Shear Rotation Interdependent Shape Functions. *International Journal for Computational Methods in Engineering Science and Mechanics*, 7, 425-431.
- Eisenberger, M. (1994). Derivation of shape functions for an exact 4-d.o.f. Timoshenko beam element. *Communications in numerical methods in engineering*, 10, 673-681.
- Friendman, Z., & Kosmatka, J. (1993). An improved two-node timoshenko beam finite element. *Computers and Structures*, 47(3), 473-481.
- Lee, S., Koo, J., & Choi, J. (1994). Variational formulation for timoshenko beam element by separation of deformation mode. *Communications in numerical methods in engineering*, 10, 599-610.
- MathWorks. (1 de Agosto de 2021). *MathWorks*. Obtenido de https://la.mathworks.com/help/matlab/ref/if.html?s_tid=srchtitle
- MathWorks. (1 de Agosto de 2021). *MathWorks*. Obtenido de https://la.mathworks.com/help/matlab/ref/for.html?searchHighlight=for&s_tid=srchtitle
- MathWorks. (1 de Agosto de 2021). *MathWorks*. Obtenido de https://la.mathworks.com/help/matlab/ref/zeros.html?searchHighlight=zeros&s_tid=srchtitle
- MathWorks. (1 de Agosto de 2021). *MathWorks*. Obtenido de <https://la.mathworks.com/help/matlab/ref/plot.html>
- MathWorks. (1 de Agosto de 2021). *MathWorks*. Obtenido de https://la.mathworks.com/help/matlab/ref/function.html?searchHighlight=function&s_tid=srchtitle
- Mukherjee, S., & Prathap, G. (2002). Analysis of delayed convergence in the three noded Timoshenko beam element using the function space approach.
- Oñate, E. (2013). *CALCULO DE ESTRUCTURAS POR EL METODO DE ELEMENTOS FINITOS* (Vol. 2). Barcelona.
- Ottosen, N., & Petersson, H. (1992). *Introductions to the Finite Element Method*. (Primera ed.). Suecia: Printice Hall International.
- Paiva, C., & Guimarães, G. (2018). Análise comparativa das teorias de Euler-Bernoulli e de Timoshenko via método das diferenças finitas com implementação computacional. *Revista Tecnologia. Fortaleza, Brasil*, 39(1), 1-12. Obtenido de <https://periodicos.unifor.br/tec/article/view/7916>
- Piedra, C. (2019). Determinación de la máxima deflexión en vigas de Euler-Bernoulli y Timoshenko aplicando el Método de los Elementos Finitos. *Tesis para maestro en Ingeniería Matemática*.



Universidad Nacional de Trujillo, Trujillo, Perú. Obtenido de
<https://dspace.unitru.edu.pe/handle/UNITRU/15187>

Reddy, J. N. (1999). On the dynamic behaviour of the Timoshenko beam finite Elements. *24*(3), 175-198.

Turner, C. M. (1956). Stiffness and deflection analysis of complex structures. *Journal of the Aeronautical Sciences*, *23*(9).



6. Anexos.

Vector resultado de fuerzas de los ejercicios uno dos y tres

Ejemplo 1)

Remplazando los desplazamientos obtenidos en el sistema de ecuaciones original se obtiene el siguiente vector

$F_s =$

100.0000

100.0000

Que indica una reacción en el sentido 'y' positivo de 100 KN y un momento de 100 KN m.

Ejemplo 2)

Remplazando los desplazamientos obtenidos en el sistema de ecuaciones original se obtiene el siguiente vector

$F_s =$

10.0000

10.0000

El vector obtenido indica una reacción en el sentido 'y' positivo de 10 KN en el apoyo izquierdo, y una reacción en el sentido 'y' positivo de 10 KN en el apoyo derecho y momentos 0 en ambos apoyos.

Ejemplo 1)

Remplazando los desplazamientos obtenidos en el sistema de ecuaciones original se obtiene el siguiente vector

$F_s =$

100.0000

100.0000

Que indica una reacción en el sentido 'y' positivo de 100 KN y un momento de 100 KN m.

$F_s =$

80.0000



20.0000

El vector obtenido indica una reacción en el sentido ‘y’ positivo de 80 KN en el apoyo izquierdo, y una reacción en el sentido ‘y’ positivo de 20 KN en el apoyo derecho y momentos 0 en ambos apoyos

Programa en Matlab

La codificación del programa Calle_Orellana_Galerkin, Extrae los datos cargados de una hoja de Excel para representar la viga sus cargas y apoyos de forma gráfica, obtiene la matriz de rigidez de cada elemento, el ensamble de las matrices rigidez, la imposición de las condiciones de frontera y las soluciones del sistema reducido, grafica las deflexiones de cada nodo obtenidas del sistema reducido y la gráfica de deflexiones del método clásico.

A) Extracción de datos cargados en Excel y organización de datos en vectores

```
function
[N_Nodos,N_Elementos,N_Apoyos,N_F_Nodal,N_F_Elementos,Nx,Ny,NI,NJ,Pos_ID_i,
Pos_ID_j,ID_Nodos,ID_Nodos_Apoyo,ID_Elementos,...

A,I,Ks,L,E,u,GDL,GDLP,GDLS,GDLi,GDLj,GDLi_Q,GDLj_Q,Pos_ID_Apoyo_Simple,Pos_
ID_Apoyo_Fijo,Pos_ID_Apoyo_Empotrado,...
    Cord_X_P_i, Cord_X_P_j,Pos_ID_ElemQ,Cord_Y_P_i,
Cord_Y_P_j,Cord_X_Q_i, Cord_X_Q_j,Cord_Y_Q_i, Cord_Y_Q_j,...
    Cargas_Nodales, Cargas_Distribuidas, Q, P] =
Formar_Datos_Partida(handles)

%% Se extraen las matrices de las Tablas con los datos del Modelo
M_Nodos = get(handles.tab_Nodos,'data'); %Se extrae los
datos a la matriz nodos
M_Elementos = get(handles.tab_Elementos,'data'); %Se extrae los
datos a la matriz elementos
M_Apoyos = get(handles.tab_Apoyos,'data'); %Se extrae los
datos a la matriz apoyos
M_Cargas_Nodos = get(handles.tab_Carga_Nodal,'data'); %Se extrae los
datos a la matriz de cargas nodales
M_Cargas_Elementos = get(handles.tab_Carga_Elementos,'data'); %Se extrae
los datos a la matriz de cargas en los elementos
M_Materiales_Data = cell2mat(get(handles.tab_Materiales,'data')); %Se
extrae los datos a la matriz de cargas en los elementos

%% Se extrae el número de elementos y fuerzas
[N_Nodos,~] = size(M_Nodos); % # de Nodos
[N_Elementos,~]= size(M_Elementos); % # de Elementos
[N_Apoyos,~] = size(M_Apoyos); % # de Apoyos
[N_F_Nodal,~] = size(M_Cargas_Nodos); % # de Fuerzas Nodales
[N_F_Elementos,~] = size(M_Cargas_Elementos); % # de Fuerzas en los
Elementos

%% Informacion de los Nodos
ID_Nodos = M_Nodos(:,1); %Vector con todos los ID de los
Nodos
Nx = M_Nodos(:,2); %Vector con la Posicion X de los
Nodos
```



```
Ny = M_Nodos(:,3); %Vector con la Posicion Y de los Nodos

%% Informacion Elementos
NI = M_Elementos(:,2); %Vector con nodos inicial de las Elementos
NJ = M_Elementos(:,3); %Vector con nodos final de las Elementos
A = M_Elementos(:,5); %Vector con las Areas de cada Elemento
I = M_Elementos(:,6); %Vector con las Inercias de cada Elemento
Ks = M_Elementos(:,7); %Vector con los Factores de forma de las secciones
E = zeros(N_Elementos,1); %Vector con los Modulos de elasticidad
u = zeros(N_Elementos,1); %Vector con los Modulos de poisson
Pos_ID_i = [];
Pos_ID_j = [];

% Se forma la matriz con los datos del material de cada elemento
Cod_M = M_Elementos(:,4); %Vector con los codigos del Material de cada Elemento
Codigo_Material = M_Materiales_Data(:,1); %Codigo de los materiales disponibles

for i = 1:N_Elementos
    aux = find(Codigo_Material==Cod_M(i));
    E(i) = M_Materiales_Data(aux,2); % Se extrae el modulo de elasticidad para cada elemento
    u(i) = M_Materiales_Data(aux,3); % Se extrae el modulo de poisson para cada elemento
    aux_i = find(ID_Nodos == NI(i));
    Pos_ID_i(end+1) = aux_i;
    aux_j = find(ID_Nodos == NJ(i));
    Pos_ID_j(end+1) = aux_j;
end

%% Informacion de los Apoyos
ID_Nodos_Apoyo = M_Apoyos(:,2); % En que nodod estael apoyo
N_GDL = N_Nodos*2; % Calcula el numero de GDL
GDL = (1:N_GDL); % Vector con todos los GDL del Modelo
GDL_aux = GDL;
GDLS_X = M_Apoyos(:,3);
GDLS_Y = M_Apoyos(:,4);
GDLS_Z = M_Apoyos(:,5);
GDLS = [];

Pos_ID_Apoyo_Simple = [];
Pos_ID_Apoyo_Fijo = [];
Pos_ID_Apoyo_Empotrado = [];

for i = 1: N_Apoyos
    if GDLS_X(i) == 0 && GDLS_Y(i) == 1 && GDLS_Z(i) == 0 %
        Condicion de apoyo simple
            aux_sim = find(ID_Nodos_Apoyo(i)==ID_Nodos); % Se encuentra la posicion del nodo
            Pos_ID_Apoyo_Simple(end+1) = aux_sim; % Se almacenan para graficar las posiciones
```



```
        GDLS(end+1) = aux_sim*2-1;
    elseif GDLS_X(i) == 1 && GDLS_Y(i) == 1 && GDLS_Z(i) == 0 %
Condición de apoyo fijo
        aux_fij = find(ID_Nodos_Apoyo(i)==ID_Nodos); % Se
encuentra la posición del nodo
        Pos_ID_Apoyo_Fijo(end+1) = aux_fij; % Se almacenan
para graficar las posiciones
        GDLS(end+1) = aux_fij*2-1;
    elseif GDLS_X(i) == 1 && GDLS_Y(i) == 1 && GDLS_Z(i) == 1 %
Condición de empotramiento
        aux_emp = find(ID_Nodos_Apoyo(i)==ID_Nodos);
        Pos_ID_Apoyo_Empotrado(end+1) = aux_emp;
        GDLS(end+1) = aux_emp*2-1;
        GDLS(end+1) = aux_emp*2;
    end
end

GDLS = sort(GDLS); % Vector con los grados de libertad S
cont = 0;
for i = GDLS
    cont = cont + 1;
    aux(cont) = find(i==GDL);
end
GDL(GDLS) = [];
GDL_P = GDL; % Vector con Grados de Libertad P
GDL = GDL_aux;
GDLi = zeros(N_Elementos,2); % Matriz con los GDL iniciales de los
Elementos
GDLj = zeros(N_Elementos,2); % Matriz con los GDL finales de los
Elementos

%% Se forman vectores de Longitud de los elementos
L = zeros(N_Elementos,1);
for i=1:N_Elementos
    aux_i = find(NI(i)==ID_Nodos);
    aux_j = find(NJ(i)==ID_Nodos);
    L(i) = abs(Nx(aux_j(1))-Nx(aux_i(1)));
    GDLi(i,:) = [aux_i*2-1 , aux_i*2];
    GDLj(i,:) = [aux_j*2-1 , aux_j*2];
end

%% Se forman los vectores para representación de las cargas puntuales en
los nodos
ID_Nodos_Cargas = M_Cargas_Nodos(:,1);
Cargas_P = M_Cargas_Nodos(:,2);
Cord_X_P_i = [];
Cord_X_P_j = [];

Cord_Y_P_i = [];
Cord_Y_P_j = [];
Cargas_Nodales = []; % Matriz con el GDL y el valor de la carga P [GDL P]

for i = 1:N_F_Nodal
    if Cargas_P(i) ~= 0 % Se comprueba que la carga sea diferente de cero
para tomarla en cuenta
        aux = find(ID_Nodos_Cargas(i) == ID_Nodos);
        Cord_X_P_i(end+1) = Nx(aux);
        Cord_X_P_j(end+1) = Nx(aux);
```



```
Cargas_Nodales(end+1,1) = aux*2-1;
Cargas_Nodales(end,2) = Cargas_P(i);

% Se toma en cuenta si es positiva o negativa
if Cargas_P(i) < 0
    Cord_Y_P_i(end+1) = Ny(i)+1;
    Cord_Y_P_j(end+1) = Ny(i);
else
    Cord_Y_P_i(end+1) = Ny(i);
    Cord_Y_P_j(end+1) = Ny(i)+1;
end
end
end

%% Se forman los vectores de las cargas distribuidas de los elementos
ID_Elementos = M_Elementos(:,1); %Vector con todos los ID de
los Nodos
ID_Elementos_Cargas = M_Cargas_Elementos(:,1);
Cargas_Q = M_Cargas_Elementos(:,2);
Pos_ID_ElemQ = [];

Cord_X_Q_i = [];
Cord_X_Q_j = [];

Cord_Y_Q_i = [];
Cord_Y_Q_j = [];

GDLi_Q = [];
GDLj_Q = [];

Cargas_Distribuidas = []; % Matriz con informacion de las cargas
distribuidas [GDL_i GDL_j Valor]

for i = 1:N_F_Elementos
    if Cargas_Q(i) ~= 0 % Se comprueba que la carga sea diferente de cero
para tomarla en cuenta
        aux = find(ID_Elementos_Cargas(i) == ID_Elementos);
        Pos_ID_ElemQ(end+1) = aux;
        pos_nodo_i = find(NI(aux) == ID_Nodos); % Nodo Inicial del
elemento
        pos_nodo_j = find(NJ(aux) == ID_Nodos); % Nodo Final del
elemento
        Cord_X_Q_i(end+1) = Nx(pos_nodo_i);
        Cord_X_Q_j(end+1) = Nx(pos_nodo_j);

        GDLi_Q(end+1,1:2) = [pos_nodo_i*2-1 , pos_nodo_i*2];
        GDLj_Q(end+1,1:2) = [pos_nodo_j*2-1 , pos_nodo_j*2];

        Cargas_Distribuidas(end+1,1) = pos_nodo_i*2-1; % Matriz [GDLi
GDLj Q] Es lo mismo de lo anterior pero en Matriz
        Cargas_Distribuidas(end,2) = pos_nodo_j*2;
        Cargas_Distribuidas(end,3) = Cargas_Q(i);

% Se toma en cuenta si es positiva o negativa
```



```
if Cargas_Q(i) < 0
    Cord_Y_Q_i(end+1) = Ny(i)+1;
    Cord_Y_Q_j(end+1) = Ny(i);
else
    Cord_Y_Q_i(end+1) = Ny(i);
    Cord_Y_Q_j(end+1) = Ny(i)+1;
end
end
end
end

Q = Cargas_Q(:);
P = Cargas_P(:);
end
```

B) Ensamblado de la matriz elemental k

```
function [Ke] = Ensamblado_K_elem(A, I, L, Ks, E, G)
ld = E*I/(Ks*G*A*L^2); % Factor Lamda
ft = E*I/(6*ld*L^3);
fp = L^2*(2+6*ld);
fn = L^2*(1-6*ld);

Ke = ft*[6      3*L      -6      3*L
          3*L    fp      -3*L    fn
          -6     -3*L    6      -3*L
          3*L    fn     -3*L    L^2*(2+6*ld)];

end
```

c) Ensamblado de la matriz global k

```
function [K_glob] = Ensamblado_K_global(K_glob,Ke,GDLi_elem,GDLj_elem)
% FILAS = [NI,NI,NJ,NJ];
% COLUM = [NI,NJ,NI,NJ];
% K(FILAS,COLUM) = Ke;

Filas = [GDLi_elem GDLj_elem];
Columnas = [GDLi_elem GDLj_elem];
K_glob(Filas,Columnas) = K_glob(Filas,Columnas) + Ke;
end
```

D) Ensamblado del vector elemental F

```
function [Fe] = Ensamblado_F_elem(Q, L, Xi, Xj)
R = Q*L/2;
Fe = [-R; 0; -R; 0];

end
```

E) Ensamblado del vector Global f

```
function [Fglob] = Ensamblado_F_glob(Fglob,Fe,GDLi,GDLj)
```



```
Fglob([GDLi,GDLj]) = Fglob([GDLi,GDLj]) + Fe;  
end
```

F) Llamamiento de funciones y solución a la ecuación $ka=f$

```
Arreglo_Ke = cell(N_Elementos,1);  
K_glob = zeros(N_Nodos*2,N_Nodos*2);  
for i=1:N_Elementos  
    G = E(i)/(2*(1+u(i))); % Modulo de cortante  
    GDLi_elem = GDLi(i,:);  
    GDLj_elem = GDLj(i,:);  
    [Ke] = Ensamblado_K_elem(A(i),I(i),L(i),Ks(i),E(i),G); %Obtencion de  
    los coeficientes ke y de las matrices elementales [Kele]  
    Arreglo_Ke{i} = Ke; % Almacena las matrices elementales en este  
    arreglo  
    [K_glob] = Ensamblado_K_global(K_glob,Ke,GDLi_elem,GDLj_elem);  
end  
  
%% Se obtiene los vectores Fuerza  
F_glob = zeros(N_Nodos*2,1);  
Femp_glob = zeros(N_Nodos*2,1);  
Arreglo_Fe_emp = cell(N_Elementos,1); % Arreglo para almacenar las  
fuerzas de empotramiento  
  
for i = 1:N_Elementos  
    Arreglo_Fe_emp{i} = [0;0;0;0]; % Se llenan de ceros los vectores de  
fuerzas de cada elemento  
end  
  
Xi = 0;  
Xj = 1;  
  
if isempty(Pos_ID_ElemQ)  
    N_F_Elementos = 0;  
end  
  
for i=1:N_F_Elementos  
    ID_Element= Pos_ID_ElemQ(i); % Se obtien el ID del elemento al que se  
le aplica la carga distribuida  
    GDL_i = GDLi_Q(i,:);  
    GDL_j = GDLj_Q(i,:);  
    [Fe] = Ensamblado_F_elem(Q(i),L(ID_Element),Xi,Xj); %Obtencion de los  
coeficientes ke y de las matrices elementales [Kele]  
    Arreglo_Fe_emp{ID_Element} = Fe; % Almacena  
los vectores elementales de fuerzas de cada elemento  
    %[F_glob] = Ensamblado_F_glob(F_glob,Fe,GDL_i,GDL_j);  
    [Femp_glob] = Ensamblado_F_glob_emp(Femp_glob,Fe,GDL_i,GDL_j);  
end  
  
for i=1:N_F_Nodal  
    GDL_i = Cargas_Nodales(i,1);  
    P = Cargas_Nodales(i,2);  
    if GDL_i == 0  
        % No hace nada  
    else  
        F_glob(GDL_i) = P;
```



```
end
end

% Se resuelve el Sistema por Particion de Matrices
[Up, Fs, Kpp]=Solucion_Sistema(K_glob,F_glob,Femp_glob,GDLS,GDLP);
a = 0;
N_GDL = N_Nodos*2;
U = zeros(N_GDL,1);
U(GDLP) = Up;
```

G) grafica del modelo de viga, cargas y apoyos

```
function [Xmin, Xmax, Ymin, Ymax] =
Dibujar_Modelo(Pos_ID_i,Pos_ID_j,Nx,Ny,N_Elementos,Pos_ID_Apoyo_Simple,Pos_
ID_Apoyo_Fijo,Pos_ID_Apoyo_Empotrado)

hold on
axis on
box on
%% SE GRAFICAN LOS ELEMENTOS
for k = 1:N_Elementos
    Elemento = [Nx(Pos_ID_i(k)) , Nx(Pos_ID_j(k))];
    ELEMENTOS =
line(Elemento,[0,0], 'LineWidth',3.5, 'Color',[0.5,0.5,0.5]);
end

%% SE GRAFICAN LOS NODOS
NODOS =
scatter(Nx,Ny,'o','SizeData',30,'MarkerFaceColor','black','MarkerEdgeColor'
,[.5 .5 .5]);

%% SE GRAFICAN LOS APOYOS
plot(0,2);
plot(0,-2);

Apoyo_x = Nx(Pos_ID_Apoyo_Empotrado);
Apoyo_y = Ny(Pos_ID_Apoyo_Empotrado);
APOYO_EMPOTRADO = scatter(Apoyo_x,Apoyo_y-
0.1,'s','SizeData',200,'MarkerFaceColor',[.5 .5 .5],'MarkerEdgeColor',[.5
.5 .5]);

Apoyo_x = Nx(Pos_ID_Apoyo_Fijo);
Apoyo_y = Ny(Pos_ID_Apoyo_Fijo);
APOYO_FIJO = scatter(Apoyo_x,Apoyo_y-
0.1,'^','SizeData',200,'MarkerFaceColor',[.5 .5 .5],'MarkerEdgeColor',[.5
.5 .5]);

Apoyo_x = Nx(Pos_ID_Apoyo_Simple);
Apoyo_y = Ny(Pos_ID_Apoyo_Simple);
APOYO_SIMPLE = scatter(Apoyo_x,Apoyo_y-
0.1,'o','SizeData',200,'MarkerFaceColor',[.5 .5 .5],'MarkerEdgeColor',[.5
.5 .5]);

legend([NODOS,APOYO_EMPOTRADO,APOYO_FIJO,APOYO_SIMPLE,ELEMENTOS], 'Nodos', 'E
mpotramiento', 'Apoyo Fijo', 'Apoyo Movil', 'Elementos');
Xmin = min(Nx)- min(Nx)*0.1;
```




```
Xmax = max(Nx)*1.05;
if Xmin == 0
    Xmin = Xmin - 0.05*Xmax;
end
Ymin = min(Ny)-2;
Ymax = max(Ny)+2;

axis([Xmin Xmax Ymin Ymax])
end

function []=Dibujar_Cargas(Cord_X_P_i, Cord_X_P_j,Cord_Y_P_i,
Cord_Y_P_j,Cord_X_Q_i,...
    Cord_X_Q_j,Cord_Y_Q_i, Cord_Y_Q_j,Cargas_Nodales,
Cargas_Distribuidas)
N_Cargas_P = length(Cord_X_P_i);
N_Cargas_Q = length(Cord_X_Q_i);

cla % Limpia el espacio de dibujo
hold on
%% Se grafican las cargas P
for i = 1:N_Cargas_P
    Pi = [Cord_X_P_i(i),Cord_Y_P_i(i)]; % Coordenadas del punto inicial
    Pj = [Cord_X_P_j(i),Cord_Y_P_j(i)]; % Coordenadas del punto inicial
    dp = Pj - Pi; % Se calculala diferencia del punto
inicial con el final
    quiver(Pi(1),Pi(2),dp(1),dp(2),0,'color',[0 1 0]); % Se dibuja la
carga inicial
    if dp(2) > 0
        scatter(Pj(1),Pj(2),'^','SizeData',100,'MarkerFaceColor',[0 1
0],'MarkerEdgeColor',[0 1 0]); % Fuerza Positiva
        text(Pj(1),Pj(2)*1.1, sprintf('%.0f',Cargas_Nodales(i)))
    else
        scatter(Pi(1),Pj(2),'v','SizeData',100,'MarkerFaceColor',[0 1
0],'MarkerEdgeColor',[0 1 0]); % Fuerza Negativa
        text(Pi(1),Pi(2)*1.1, sprintf('%.0f',Cargas_Nodales(i)))
    end
end

%% Se Grafican las Cargas Q
for i = 1:N_Cargas_Q
    salto = (Cord_X_Q_j(i) - Cord_X_Q_i(i))/6; % Para poder dibujar la
carga distribuida

    Px = (Cord_X_Q_i(i):salto:Cord_X_Q_j(i)); % Se crea un vector de
cordenadas en xi

    for j=1:7
        Pi = [Px(j),Cord_Y_Q_i(i)]; % Coordenadas del punto inicial
        Pj = [Px(j),Cord_Y_Q_j(i)]; % Coordenadas del punto inicial
        dp = Pj - Pi; % Se calculala diferencia del
punto inicial con el final
        quiver(Pi(1),Pi(2),dp(1),dp(2),0,'color',[0 0 1]); % Se dibuja la
carga inicial
        if dp(2) > 0
            scatter(Pj(1),Pj(2),'^','SizeData',50,'MarkerFaceColor',[0 0
1],'MarkerEdgeColor',[0 0 1]); % Fuerza Positiva
        else
```



```
scatter(Pj(1),Pj(2), 'v', 'SizeData',50, 'MarkerFaceColor',[0 0
1], 'MarkerEdgeColor',[0 0 1]); % Fuerza Negativa
end

if j==4
if dp(2) > 0
text(Pj(1),Pj(2)*1.1, sprintf('%.0f',Cargas_Distribuidas(i)))
else
text(Pi(1),Pi(2)*1.1, sprintf('%.0f',Cargas_Distribuidas(i)))
end
end

end

plot([Cord_X_Q_i(i) Cord_X_Q_j(i)], [Cord_Y_Q_i(i)
Cord_Y_Q_i(i)], 'Color', 'b', 'LineWidth',1)

end
end
```

H) Grafica comparativa de deflexiones resueltas por Galerkin y método clásico

```
function [] = Dibujar_Deformada_Analitica(L,E,I,P,W,Xmin, Xmax, Ymin,
Ymax,bandera_condicion_viga,Escala,a)

hold on
b = 1- a;
%% SE GRAFICAN LOS ELEMENTOS
dx = L/100;
x = (0:dx:L); % Discretizo en 100 tramos
N_Elementos = length(x)-1;
Y_max = [];
X_max = [];
for i = 1:N_Elementos
X = [x(i) , x(i+1)];
if bandera_condicion_viga == 1 % Voladizo P
Y = P/(6*E*I)*(3*L*X.^2-X.^3);
elseif bandera_condicion_viga == 2 % Voladizo W
Y = W/(24*E*I).*(X.^4-4*L*X.^3+6*L^2*X.^2);
elseif bandera_condicion_viga == 3 % Apoyada P
if X(1) <= a*L % Primera mitad de la Viga
Y = P*b*X/(6*L*E*I).*(L^2-b^2-X.^2);
else % Segunda mitad de la Viga
Y = P*a*(L-X)/(6*L*E*I).*(L^2-a^2-(L-X).^2);
end
elseif bandera_condicion_viga == 4 % Apoyada W
Y = W*X/(24*E*I).*(L^3-2*L*X.^2+X.^3);
end
Y_max(end+1:end+2,1) = Y';
X_max(end+1:end+2,1) = X';

Y = Y*Escala;
line(X,Y, 'LineWidth',1.5, 'Color',[1,0.,0.], 'Linestyle','--');
end
Y_maximo = min(Y_max);
aux = find(Y_max == Y_maximo); % Posicion
X_maximo = X_max(aux);
```



```
scatter(X_maximo(1),Y_maximo(1)*Escala,'o','SizeData',30,'MarkerFaceColor',
'black','MarkerEdgeColor',[.5 .5 .5]);
text(X_maximo(1),Y_maximo(1)*Escala,string(Y_maximo));

legend off
axis([Xmin Xmax Ymin Ymax])

end
function [] =
Dibujar_Deformada(Pos_ID_i,Pos_ID_j,Nx,Ny,N_Elementos,Pos_ID_Apoyo_Simple,P
os_ID_Apoyo_Fijo,Pos_ID_Apoyo_Empotrado,Xmin, Xmax, Ymin, Ymax)

hold on
%% SE GRAFICAN LOS ELEMENTOS
for k = 1:N_Elementos
    ElementoX = [Nx(Pos_ID_i(k)) , Nx(Pos_ID_j(k))];
    ElementoY = [Ny(Pos_ID_i(k)) , Ny(Pos_ID_j(k))];
    ELEMENTOS =
line(ElementoX,ElementoY,'LineWidth',1.5,'Color',[0.5,0.5,0.5],'LineStyle',
'--');
end

%% SE GRAFICAN LOS NODOS
NODOS =
scatter(Nx,Ny,'o','SizeData',30,'MarkerFaceColor','black','MarkerEdgeColor'
,[.5 .5 .5]);

%% SE GRAFICAN LOS APOYOS
plot(0,2);
plot(0,-2);

Apoyo_x = Nx(Pos_ID_Apoyo_Empotrado);
Apoyo_y = Ny(Pos_ID_Apoyo_Empotrado);
APOYO_EMPOTRADO = scatter(Apoyo_x,Apoyo_y-
0.1,'s','SizeData',200,'MarkerFaceColor',[.5 .5 .5],'MarkerEdgeColor',[.5
.5 .5]);

Apoyo_x = Nx(Pos_ID_Apoyo_Fijo);
Apoyo_y = Ny(Pos_ID_Apoyo_Fijo);
APOYO_FIJO = scatter(Apoyo_x,Apoyo_y-
0.1,'^','SizeData',200,'MarkerFaceColor',[.5 .5 .5],'MarkerEdgeColor',[.5
.5 .5]);

Apoyo_x = Nx(Pos_ID_Apoyo_Simple);
Apoyo_y = Ny(Pos_ID_Apoyo_Simple);
APOYO_SIMPLE = scatter(Apoyo_x,Apoyo_y-
0.1,'o','SizeData',200,'MarkerFaceColor',[.5 .5 .5],'MarkerEdgeColor',[.5
.5 .5]);

legend([NODOS,APOYO_EMPOTRADO,APOYO_FIJO,APOYO_SIMPLE,ELEMENTOS],'Nodos','E
mpotramiento','Apoyo Fijo','Apoyo Movil','Elementos');

axis([Xmin Xmax Ymin Ymax])

end
```