

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA MECÁNICA

DESARROLLO DE UNA HERRAMIENTA COMPUTACIONAL DE DISEÑO DE ESTRUCTURAS METÁLICAS

(COMPONENTE I)

ANÁLISIS ESTRUCTURAL

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO
MECÁNICO**

MOREIRA NOGUERA IVÁN ANDRES

ivan.moreira@epn.edu.ec

DIRECTOR: ING. LUIS FELIPE FERNÁNDEZ AYALA, PhD.

luis.fernandez@epn.edu.ec

Quito, septiembre 2024

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Iván Andres Moreira Noguera, bajo mi supervisión.

Ing. LUIS FELIPE FERNÁNDEZ AYALA, PhD.
DIRECTOR DE PROYECTO

DECLARACIÓN

Yo, Iván Andres Moreira Noguera, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectuales correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

MOREIRA NOGUERA IVÁN ANDRES

DEDICATORIA

Dedico este proyecto a todas las personas que, desde el principio hasta el final, aportaron con su grano de arena para que pudiera completar este objetivo.

AGRADECIMIENTOS

A Dios y mis padres, quienes han sido los pilares fundamentales brindándome su apoyo incondicional y guía en cada momento difícil de mi carrera.

A toda mi familia y amigos, cuyo cariño y presencia durante este proceso han sido invaluable; les agradezco profundamente desde el fondo de mi corazón.

Al Ing. Luis Felipe Fernández, director del tema de integración curricular, por su constante apoyo y seguimiento a lo largo del desarrollo de este proyecto.

Al Ing. Miguel Ortega, mi tutor de carrera, por estar siempre atento a mi desempeño universitario y ofrecerme su orientación a lo largo de toda mi formación.

Al Ing. Luis Lozada, tutor de mis prácticas profesionales, por compartir conmigo su vasta experiencia laboral y ayudarme en mi camino hacia la profesionalización.

Y a mis angelitos, quienes sé que también me han ayudado desde el cielo; que Dios los tenga en su santa gloria.

CONTENIDO

Resumen	1
Abstract	2
1 INTRODUCCIÓN	3
1.1 Objetivo general	7
1.2 Objetivos específicos	7
1.3 Alcance	8
2 MARCO TEÓRICO	10
2.1 Elemento y nodo	10
2.2 Cordenadas Globales y del elemento	11
2.3 Interpretación cinemática del método	12
2.4 Matriz de rigidez del marco-elemento	14
2.4.1 Matriz de rigidez del elemento análisis axial	16
2.4.2 Matriz de rigidez del elemento análisis cortante y flexión	17
2.4.3 Matriz de rigidez del elemento	19
2.5 Matriz de transformación de los desplazamientos y de las fuerzas	20
2.6 Matriz de rigidez global del marco-elemento	23
2.7 Aplicación del método	24
3 METODOLOGÍA	27
3.1 Diseño de Ingreso de datos	27
3.1.1 Diseño de ingreso de datos característicos del elemento y la armadura	28
3.1.2 Diseño de ingreso de coordenadas	29
3.1.3 Diseño de ingreso de grados de libertad	30
3.1.4 Diseño de ingreso de cargas y desplazamientos de la armadura	32
3.1.5 Diseño de ingreso de cargas distribuidas	33
3.2 Diseño de la función metodo de rigidez	36
3.2.1 Diseño de la matriz de rigidez global	36
3.2.2 Diseño de solución del sistema de rigidez	37
3.2.3 Diseño del código para la obtención de cargas internas de los elementos	38

3.2.4	Diseño del código para la impresión de resultados	40
3.2.5	Análisis de las cargas cortantes y los momentos internos	41
4	RESULTADOS Y DISCUSIÓN	44
4.1	Ejemplo 1. - Armadura de 8 barras con cargas puntuales	44
4.2	Ejemplo 2. - Armadura de 11 barras con cargas distribuidas	51
4.3	Ejemplo 3. - Armadura con 61 barras con cargas puntuales y distribuidas . .	60
4.4	Tiempo de procesamiento	64
5	CONCLUSIONES	65
6	RECOMENDACIONES	67
7	REFERENCIAS BIBLIOGRÁFICAS	68
8	ANEXOS	I

RESUMEN

El presente proyecto tiene como objetivo desarrollar una herramienta computacional para analizar, evaluar y optimizar estructuras metálicas en dos dimensiones, considerando cargas axiales, cortantes y momentos, mediante el método de la rigidez. Este escrito, que forma parte de un proyecto dividido en tres componentes, se centrará exclusivamente en el análisis de marco-armaduras.

Se proporciona una introducción sobre la historia y los fundamentos del método de la rigidez. Posteriormente, se explica la teoría subyacente, incluidas las ecuaciones y matrices utilizadas por el método para llevar a cabo el análisis. A continuación, se describe el diseño del código empleado para analizar las marco-armaduras. Se destacan las técnicas de programación y los algoritmos implementados, así como la optimización del código para mejorar la eficiencia y precisión del análisis. Finalmente, se presentan y detallan ejemplos que validan y verifican el funcionamiento del código.

Además de este escrito, se desarrolla un manual de usuario para la herramienta computacional, con el fin de que pueda ser utilizado por otras personas que necesiten realizar análisis de marco-armaduras. Este manual incluye instrucciones detalladas sobre el uso de la herramienta computacional, facilitando su aplicación en diversos contextos de ingeniería estructural. De la misma manera, este código se emplea en los otros dos componentes de este proyecto de integración curricular que son la evaluación y optimización de estructuras metálicas, las cuales buscan la mejora continua y la seguridad de las mismas.

Palabras Clave: MEF, Método, Rigidez, Algoritmo, Optimizar.

ABSTRACT

The objective of this project is to develop a computational tool to analyze, evaluate and optimize metallic structures in two dimensions, considering axial loads, shears and moments, using the rigidity method. This writing, which is part of a project divided into three components, will focus exclusively on the analysis of frame-trusses.

An introduction to the history and foundations of the rigidity method is provided. The underlying theory is then explained, including the equations and matrices used by the method to carry out the analysis. The code design used to analyze the frame-trusses is described below. The programming techniques and algorithms implemented are highlighted, as well as the optimization of the code to improve the efficiency and accuracy of the analysis. Finally, examples that validate and verify the operation of the code are presented and detailed.

In addition to this writing, a user manual is developed for the computational tool, so that it can be used by other people who need to perform frame-truss analysis. This manual includes detailed instructions on the use of the computational tool, facilitating its application in various structural engineering contexts. In the same way, this code is used in the other two components of this curricular integration project, which are the evaluation and optimization of metal structures, which seek continuous improvement and safety.

Keywords: FEM, Method, Rigidity, Algorithm, Optimize.

1 INTRODUCCIÓN

El método que se aplicará en este proyecto es el método de la rigidez, con el cual se puede realizar el análisis de cualquier estructura mecánica, incluidas estructuras en dos dimensiones y estructuras estáticamente indeterminadas. Este método ha demostrado ser altamente efectivo en la resolución de problemas estructurales debido a su precisión y versatilidad. A continuación, se presenta una breve historia del método de la rigidez, su descripción, sus funcionalidades, y los casos en los que se utiliza.

El método de la rigidez tiene sus orígenes hace más de un siglo, cuando el matemático italiano Carlo Alberto Castigliano (Figura 1.1) dió los primeros pasos con sus estudios sobre análisis de estructuras deformables en 1879. Unos años después, en 1914, Alfred Bendixen desarrolló lo que se conoció en ese tiempo como el método giro-deformación, que eventualmente se convertiría en el método de la rigidez [1]. Sin embargo, este método no se empleó en la práctica debido a que implicaba resolver difíciles sistemas de ecuaciones lineales de gran tamaño sin contar con herramientas computacionales. En los años 1930 y 1932, Hardy Cross introdujo un método iterativo e ingenioso para resolver los complicados sistemas de ecuaciones de rigidez de la época, sin abandonar el concepto de rigidez. Finalmente, en la década de los cincuenta, hubo una gran revolución en el método debido a la aparición de los computadores, lo que permitió revivir y perfeccionar la solución de los arduos sistemas de ecuaciones del método de rigidez. Hoy en día, este método se utiliza ampliamente en gran cantidad de programas de computación dedicados al análisis estructural [2].



Figura 1.1: Carlo Alberto Castigliano [3].

Existen dos enfoques principales para analizar estructuras utilizando métodos discretos: el método de desplazamientos y el método de fuerzas. En este proyecto, se empleará el método de análisis de desplazamientos, dejando de lado el método de análisis de fuerzas. La elección del método de desplazamientos se basa en su capacidad para abordar tanto estructuras estáticamente determinadas como indeterminadas de manera más eficiente y unificada. A diferencia del método de fuerzas, que requiere metodologías distintas para cada tipo de estructura, el método de desplazamientos simplifica el proceso. El método de desplazamientos ofrece ventajas significativas, ya que permite obtener directamente las fuerzas y desplazamientos de toda la estructura. Además, la formulación de las matrices necesarias para operar mediante algoritmos es más sencilla, facilitando la solución de las matrices del método de rigidez [4].

El método de la rigidez es una aplicación del Método de Elementos Finitos (MEF). El MEF es una herramienta general que se utiliza para calcular desplazamientos y, a partir de estos, deformaciones y tensiones en problemas de mecánica de sólidos deformables. Además, es aplicable a otros problemas de mecánica de medios continuos, como la dinámica de fluidos y la transferencia de calor (Figura 1.2). La idea principal del MEF es transformar un problema físico, originalmente formulado mediante ecuaciones diferenciales parciales y sus condiciones de frontera, en un problema discreto representado por un sistema de ecuaciones algebraicas. Este sistema de ecuaciones puede resolverse de manera eficiente utilizando una computadora.

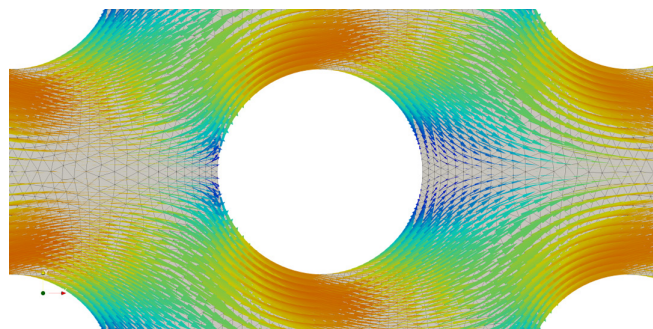


Figura 1.2: Simulación de mecánica de fluidos usando el MEF [5].

Mediante este enfoque, el MEF permite abordar problemas complejos en ingeniería y física, como la simulación de efectos de nanopartículas en la mecánica cuántica y sistemas de baja escala como nanotubos (Figura 1.3). El MEF ofrece soluciones numéricas precisas y prácticas que serían difíciles de obtener por métodos analíticos tradicionales. Este método es versátil y popular debido a su adaptabilidad a dominios complejos. Al discretizar el do-

minio continuo en una malla de “elementos finitos” y aplicar técnicas numéricas avanzadas, el MEF convierte estos problemas en sistemas manejables que pueden resolverse con la capacidad computacional moderna [6].

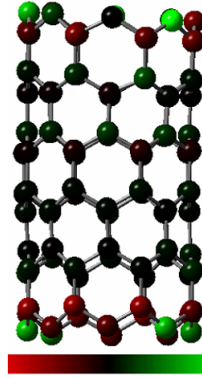


Figura 1.3: Simulación de nanotubos de carbono usando el MEF [7].

La clave para el uso efectivo del MEF es la convergencia, que significa que al realizar particiones más finas del sistema, las respuestas del cálculo se aproximan a la solución exacta. Este proceso, conocido como discretización, incluye puntos representativos denominados "nodos". Al conectar estos nodos se forma la malla, que es la base para el procesamiento y cálculo. Los cálculos realizados sobre esta malla implican la inserción de variables desconocidas definidas para cada nodo, conocidas como grados de libertad del sistema. Estas variables se relacionan a través de un sistema de ecuaciones lineales, cuya matriz se denomina matriz de rigidez, donde el número de ecuaciones es proporcional al número de nodos [6].

En el caso del método de la rigidez, se emplea el MEF de manera macro, donde cada elemento estructural, como vigas y columnas, se considera un elemento finito. Se realiza el mallado, en el cual cada nodo representa la unión de los elementos en la estructura. Los cálculos se efectúan individualmente para cada elemento y se adicionan estos cálculos sucesivamente a los elementos siguientes, lo que permite obtener la matriz de rigidez global y el vector de fuerza global de la estructura. Finalmente, se resuelve un sistema de ecuaciones, y mediante el postprocesamiento, se determinan las fuerzas internas y los esfuerzos de cada miembro estructural. En la sección del marco teórico, abordaremos estos detalles con mayor precisión.

Una de las aplicaciones de este método son los programas avanzados de análisis estructural que existen en el mercado, como Ansys, Inventor, SAP2000, Revit, entre otros. Estos programas requieren que se ingresen la configuración de la estructura, sus dimensiones y

las propiedades del material utilizado, así como las cargas externas a las que está sometida. La mayoría de estos programas utilizan el MEF para determinar las fuerzas internas, las reacciones en los apoyos, los desplazamientos, las deformaciones y los esfuerzos de las estructuras analizadas. Estas aplicaciones analizan y evalúan la estructura según las normativas de construcción internacionales. Sin embargo, es responsabilidad del diseñador seleccionar varios perfiles para asegurar que la estructura cumpla con las normas de construcción. El objetivo de este proyecto es desarrollar una herramienta computacional integral que no solo analice y evalúe estructuras, sino que también diseñe automáticamente perfiles, optimizando así el rendimiento de la estructura metálica.

En este caso, este escrito se centrará únicamente en la realización de la parte del análisis estructural. Los otros componentes de esta herramienta computacional serán abordados en documentos separados, pero enlazados a este, con el fin de crear una herramienta computacional integral, innovadora y excelente para el diseño global de estructuras metálicas.

1.1 OBJETIVO GENERAL

Desarrollar una herramienta computacional que aplique el método de la rigidez para analizar estáticamente estructuras metálicas en 2D, permitiendo una evaluación precisa de su comportamiento axial, cortante, y a flexión.

1.2 OBJETIVOS ESPECÍFICOS

1. Revisar la literatura sobre el método de la rigidez.
2. Implementar el método de la rigidez para el análisis estructural estático de armaduras en dos dimensiones.
3. Desarrollar un código en python para calcular las cargas axiales y cortantes internas, además de los momentos internos de estructuras metálicas.
4. Facilitar los resultados del análisis para que sean utilizados tanto en la evaluación como en la selección de perfiles que son los otros componentes de este proyecto.
5. Elaborar un manual de usuario y documentación de la herramienta computacional para facilitar su utilización y comprensión.

1.3 ALCANCE

Este proyecto tiene como objetivo desarrollar un código abierto en el lenguaje de programación Python para analizar las cargas que soporta una estructura metálica mediante tres etapas: ingreso de datos, procesamiento y salida de resultados. Se espera que el usuario ingrese datos característicos de una estructura, como su configuración, dimensiones, secciones, restricciones a las que está sujeta, además de la magnitud y tipo de carga externa aplicada en los elementos estructurales como barras o vigas. Después de realizar el procesamiento de estos datos, se obtendrá en detalle de las cargas internas que soporta cada miembro de toda la estructura.

El código desarrollado tiene limitaciones, tales como su enfoque en estructuras bidimensionales, la resolución exclusiva de problemas estáticos y la restricción a tipos específicos de cargas aplicadas. Estas limitaciones han influido en el diseño y funcionalidad del programa, y se describirán en detalle para proporcionar una comprensión completa de los desafíos enfrentados y las soluciones implementadas.

Respecto al dimensionamiento, este código solo podrá resolver estructuras en dos dimensiones. Aunque muchas estructuras metálicas son tridimensionales, es posible reducirlas y analizarlas en dos dimensiones. Este enfoque ofrece importantes ventajas, como la reducción del tamaño de las matrices involucradas y la capacidad de descomponer partes de estructuras más complejas. Además, dado el tiempo limitado de este proyecto de Trabajo de Integración Curricular (TIC), es razonable mantener nuestra herramienta en un análisis en 2D. En el futuro, podemos extender el análisis a 3D fuera del marco del proyecto TIC.

Otra consideración importante es que, en esta primera versión, se realizará un análisis estático de la estructura. Sin embargo, es fundamental reconocer que en el diseño de estructuras modernas no se puede ignorar las cargas dinámicas, especialmente en edificaciones ubicadas en zonas sísmicas. A pesar de esto, si una estructura presenta fallas en un análisis estático, también lo hará en un régimen dinámico. Además, el análisis estático es computacionalmente económico y proporciona resultados rápidos. En esta versión inicial por el poco tiempo para desarrollo del TIC, se limitará el análisis a cargas estáticas, ya que esto representa un buen punto de partida para desarrollar un análisis sólido. Una vez que se haya establecido una base confiable, se planea incorporar análisis dinámicos en futuras versiones del código, mejorando así la precisión y aplicabilidad del código en el diseño de

estructuras en diversas condiciones.

Respecto a las cargas que se permiten ingresar, el sistema admite únicamente: cargas puntuales en los nodos del plano de la estructura, cargas distribuidas en dirección a la gravedad que estén uniformemente distribuidas a lo largo del elemento, y momentos externos aplicados en los nodos. El método de la rigidez considera exclusivamente cargas y desplazamientos en los nodos. Por lo tanto, más adelante en el desarrollo del análisis de las estructuras, se explicará cómo se manejará la combinación de estas cargas y momentos para asegurar resultados confiables del análisis.

2 MARCO TEÓRICO

El código que se espera desarrollar utiliza el método de la rigidez para realizar el análisis de las cargas en la estructura y obtener las cargas internas de los elementos de la armadura. A continuación, se detallarán los principales aspectos del método de la rigidez, incluyendo algunos conceptos, simbología, convenciones, ecuaciones, y matrices pertinentes para llevar a cabo el análisis.

2.1 ELEMENTO Y NODO

En todas nuestras estructuras es esencial identificar claramente sus elementos y nodos para el respectivo análisis y desarrollo de las ecuaciones. Por ello los elementos o miembros de la estructura se identificarán con un número encerrado en un cuadrado, mientras que los nodos se numerarán dentro de un círculo. Además, es crucial determinar un nodo denominado “cercano” y un nodo denominado “lejano” para cada elemento. Esto se representará mediante una flecha en el elemento, donde la punta de la flecha indica el nodo lejano.

Por ejemplo, en la Figura 2.1 se muestra un ejemplo de armadura triangular compuesta por tres elementos y tres nodos. En la configuración, el elemento “uno” se ubica en el lado inferior con el nodo “uno” como cercano y el nodo “dos” como lejano. El elemento “dos” se posiciona en el lado derecho, con el nodo “dos” como cercano y el nodo “tres” como lejano. Finalmente, el elemento “tres” está en el lado izquierdo, con el nodo “uno” como cercano y el nodo “tres” como lejano. Es importante destacar que el analista tiene libertad para asignar la numeración de los elementos y nodos, así como la dirección de las flechas, de acuerdo a su preferencia. Sin embargo, esta elección debe mantenerse consistente a lo largo de todo el proceso de análisis.

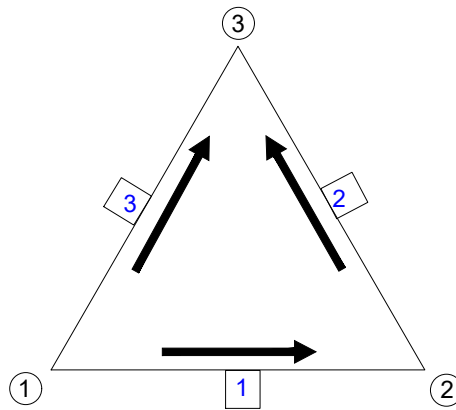


Figura 2.1: Armadura triangular de ejemplo (Fuente propia)

2.2 CORDENADAS GLOBALES Y DEL ELEMENTO

Para trabajar con fuerzas y desplazamientos vectoriales, es necesario emplear un sistema de coordenadas. En el análisis mediante el método de la rigidez, se utilizarán dos sistemas de coordenadas: el sistema global (X, Y, Z) y un sistema local por cada elemento estructural (X', Y', Z') . En este proyecto, la coordenada Z permanece constante debido al análisis en 2D. Es crucial destacar que, aunque los desplazamientos se restringen al plano XY , también se consideran magnitudes en la dirección Z , como los ángulos de rotación de las vigas y los momentos flectores. Cada nodo tiene tres grados de libertad: dos desplazamientos en el plano XY (U_X y U_Y) y un grado de libertad rotacional en el eje Z , que indica el ángulo de flexión θ_Z del elemento estructural en ese nodo. Es importante aclarar que estos grados de libertad no implican un sistema tridimensional, ya que el código desarrollado está diseñado específicamente para resolver armaduras en dos dimensiones, como se explicó en el alcance del proyecto.

El sistema de coordenadas global (X, Y, Z) se utiliza para identificar: coordenadas, fuerzas, momentos, desplazamientos, rotaciones externas de la estructura o armadura. Este sistema se puede ubicar en cualquier nodo de la estructura. Por otro lado, el sistema de coordenadas local (X', Y', Z') se emplea para cada elemento de la armadura con el fin de determinar las fuerzas o momentos internos del elemento definido localmente. Este sistema siempre se coloca en el nodo denominado “cercano” de un elemento, con el eje X' orientado hacia el nodo “lejano” del elemento. A continuación, en la Figura 2.2, se ilustra la correcta ubicación de los dos sistemas de coordenadas, facilitando así el estudio de las fuerzas y desplazamientos tanto en el sistema global como en el local.

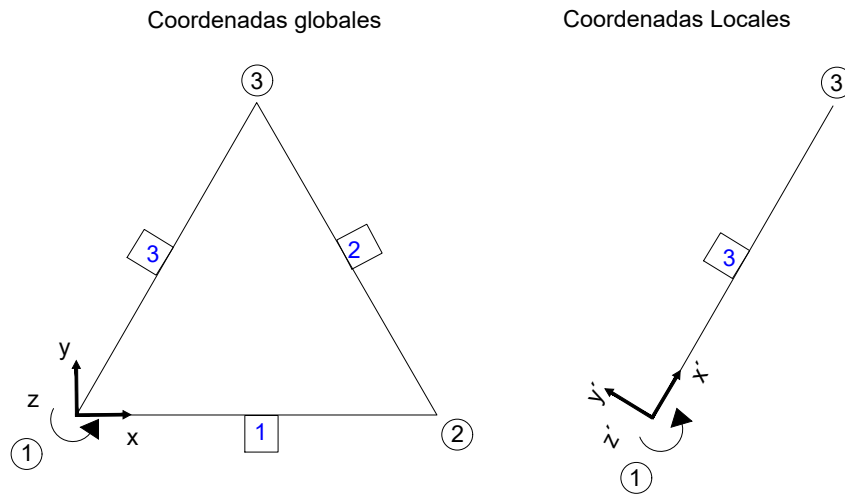


Figura 2.2: (Izquierda) Coordenadas globales para el sistema. (Derecha) Coordenadas locales para el elemento 3 (Fuente: propia).

2.3 INTERPRETACIÓN CINEMÁTICA DEL MÉTODO

En la parte cinemática del método de la rigidez, tradicionalmente se considera que cada junta o nodo tiene dos grados de libertad, es decir, dos posibles desplazamientos. Sin embargo, en este proyecto se aplica el método de rigidez con análisis de flexión, lo que permite considerar tres grados de libertad o tres posibles movimientos por nodo. Este enfoque amplía la capacidad del método para analizar una mayor variedad de casos.

Para implementar esto, cada grado de libertad se identificará con un número y una flecha, alineados con las direcciones de las coordenadas globales en su nodo correspondiente. En la Figura 2.3 se muestra cómo se asocian los grados de libertad en cada nodo.

Por lo tanto, si analizamos el nodo uno, observamos que puede tener tres tipos de movimiento; el movimiento uno a lo largo del eje X de las coordenadas globales, el movimiento dos a lo largo del eje Y de las coordenadas globales, y el movimiento tres, que es la rotación en radianes que puede experimentar el elemento en ese extremo. Es importante señalar que la numeración de los grados de libertad puede comenzar en cualquier nodo elegido por el analista; no es necesario que comience en el nodo uno. Además, una vez seleccionado el nodo inicial, no es obligatorio iniciar la numeración por el eje X; puede hacerse por el eje Y o Z. Sin embargo, es fundamental mantener esta numeración consistente a lo largo de todo el análisis. Aunque se recomienda mantener un orden coherente, por ejemplo, asignando al

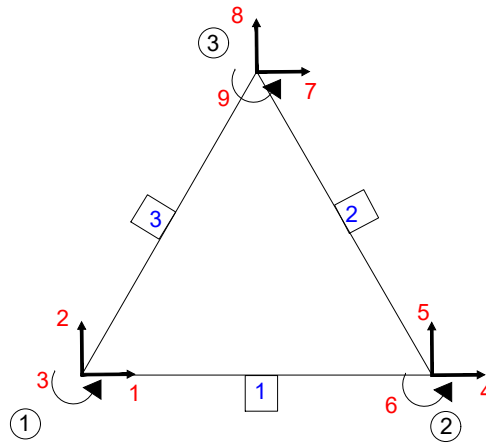


Figura 2.3: Grados de libertad por nodo (Fuente propia).

nodo i los grados de libertad $3i - 2, 3i - 1, 3i$, que corresponden a los desplazamientos en X, Y y la rotación en Z .

De todos estos grados de libertad, es importante determinar cuáles son libres y cuáles están restringidos. Para ello, se aplicarán soportes a nuestra armadura triangular utilizada como ejemplo, los cuales podemos ver en la Figura 2.4

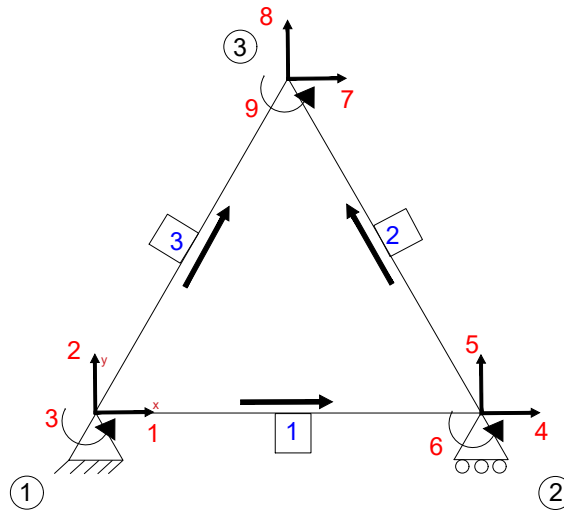


Figura 2.4: Armadura ejemplo con soportes en los nodos 1 y 2 (Fuente: propia).

Como se puede ver en la Figura 2.4, en el nodo 1 se ha agregado un soporte fijo y en el nodo dos se ha agregado un soporte con rodillos, por lo cual ahora si se puede determinar cuál de los grados de libertad son libres y cuales son restringidos.

1. Analizando el nodo 1, al estar anclado, no puede tener el movimiento del grado de libertad 1 ni el movimiento del grado de libertad 2. Para el caso del grado de libertad

3 este no va a estar restringido ya que este si puede rotar.

2. Analizando el nodo 2, al estar anclado a un soporte con ruedas, permite que si tenga el movimiento del grado de libertad 4 por lo cual este será libre, pero no puede tener el movimiento del grado de libertad 5 ya que en este eje si está anclado. Para el movimiento del grado de libertad 6 de la misma manera permite rotar por lo cual este será libre.
3. Para el nodo 3 podemos ver que el movimiento 7 y 8 no están restringidos por lo cual estos son libres. Y en el caso del grado de libertad 9, nada impide que esto rote por lo cual también estará libre.

Como conclusión decimos que los movimientos de los grados de libertad: 3, 4, 6, 7, 8 y 9 son libres y por otra parte los movimientos 1, 2 y 5 están restringidos.

Una vez codificados y clasificados los grados de libertad de la estructura, se procede a encontrar la matriz de rigidez del sistema K . Antes de ello, es necesario determinar la matriz de rigidez del elemento k' , la cual nos va ayudar a encontrar la relación entre carga y desplazamiento en función de las coordenadas locales de los elementos. En las secciones a continuación se explica cómo se determinan estas matrices.

2.4 MATRIZ DE RIGIDEZ DEL MARCO-ELEMENTO

En esta sección desarrollaremos la matriz de rigidez para un elemento en el sistema de coordenadas locales (X', Y', Z') . Se tomará como ejemplo el Elemento 3 de la Figura 2.4, el cual presenta cargas internas y desplazamientos locales, y todo esto se detalla en la Figura 2.5. Es importante aclarar que las letras de los subíndices “N” se refieren a “cerca” por su inicial en inglés (near) y “F” se refieren a “lejos” por su inicial en inglés (far).

A continuación detallamos cada una de las variables que usaremos para definir las cargas de nuestros elementos estructurales:

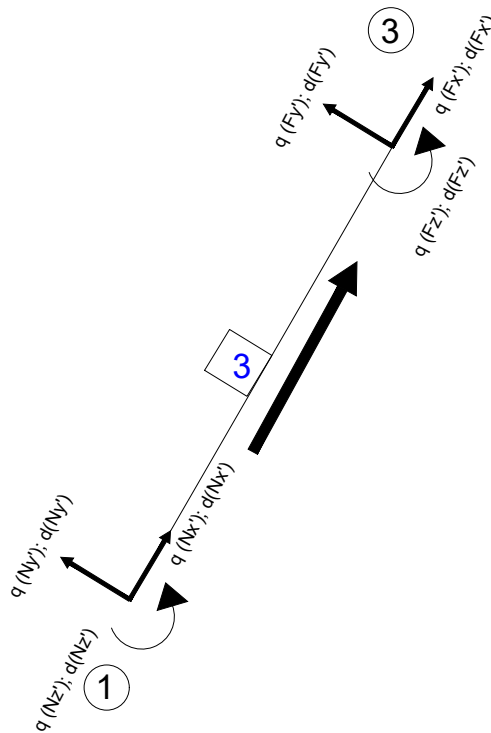


Figura 2.5: Cargas internas y desplazamientos locales del Elemento 3 (Fuente: propia).

- q_{Nx} es la carga axial en dirección X en el nodo cercano
 d_{Nx} es el desplazamiento en dirección X en el nodo cercano
- q_{Ny} es la carga cortante en dirección Y en el nodo cercano
 d_{Ny} es el desplazamiento en dirección Y en el nodo cercano
- q_{Nz} es el momento flector en dirección Z en el nodo cercano
 d_{Nz} es el desplazamiento angular en dirección Z en el nodo cercano
- q_{Fx} es la carga axial en dirección X en el nodo lejano
 d_{Fx} es el desplazamiento en dirección X en el nodo lejano
- q_{Fy} es la carga cortante en dirección Y en el nodo lejano
 d_{Fy} es el desplazamiento en dirección Y en el nodo lejano
- q_{Fz} es el momento flector en dirección Z en el nodo lejano
 d_{Fz} es el desplazamiento angular en dirección Z en el nodo lejano

Después de realizar el análisis, se espera obtener los desplazamientos locales y las cargas internas de cada elemento estructural. A partir de estos resultados, se podrán calcular los esfuerzos internos. Para este proyecto, se adoptará la convención de sentidos positivos de los ejes locales, tal como se ilustra en la Figura 2.5.

Después de haber explicado cómo se definen las cargas internas y los desplazamientos locales que se obtendrán para cada elemento, procederemos a explicar en las siguientes secciones la matriz de rigidez correspondiente a nuestros elementos estructurales. Esta matriz establece la relación entre las cargas internas y los desplazamientos. Para ello, se separará el análisis en dos partes: la primera, que obtiene la matriz de rigidez para las cargas y desplazamientos axiales, y la segunda, que obtiene la matriz de rigidez para los momentos flectores, considerando desplazamientos y rotaciones por flexión. Utilizaremos el método de superposición, que consiste en sumar estos dos efectos, con el objetivo de obtener la matriz de rigidez completa para el análisis de un marco, permitiendo calcular fuerzas axiales, fuerzas cortantes y los momentos flectores relacionados a los desplazamientos y rotaciones de los nodos del elemento.

2.4.1 Matriz de rigidez del elemento análisis axial

De la Figura 2.5 solo nos vamos a concentrar en las cargas y desplazamientos axiales, o sea aquellas que se encuentran alineadas al eje X' . Por equilibrio, tenemos que $q'_{Nx} = -q'_{Fx}$ donde las fuerzas axiales en los extremos son iguales, pero de signo contrario. Aplicando la ley de Hooke y la definición del módulo de Young podemos obtener la relación de la fuerza interna como:

$$\sigma'_x = E \epsilon'_x \quad (2.1)$$

$$\frac{q'_{Fx}}{A} = E \frac{\Delta L'_x}{L}, \quad (2.2)$$

donde σ'_x es el esfuerzo axial interno, ϵ'_x es la deformación normal unitaria en dirección X' , E es el módulo de Young, A es el área de la sección transversal del elemento, L la longitud del elemento, y $\Delta L'_x$ el desplazamiento axial. Reemplazamos que el desplazamiento axial es igual a la diferencia de desplazamientos nodales, i.e., $\Delta L'_x = d'_{Fx} - d'_{Nx}$. Con todo esto, obtenemos la siguiente ecuación para la fuerza interna en los nodos "lejos" y "cerca":

$$q'_{Fx} = \frac{AE}{L} (d'_{Fx} - d'_{Nx}) \quad (2.3)$$

$$q'_{Nx} = -q'_{Fx} = \frac{AE}{L} (d'_{Nx} - d'_{Fx}) . \quad (2.4)$$

Así de esta manera de estas dos ecuaciones se obtiene el sistema de rigidez para el análisis de cargas y desplazamientos axiales:

$$\begin{bmatrix} q_{Nx'} \\ q_{Fx'} \end{bmatrix} = \frac{AE}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} d_{Nx'} \\ d_{Fx'} \end{bmatrix} . \quad (2.5)$$

La Ecuación 2.5 también se la puede representar de la siguiente manera:

$$\mathbf{q}' = \mathbf{k}' \mathbf{d}' , \quad (2.6)$$

donde la matriz \mathbf{k}' , es la denominada matriz de rigidez local para un análisis de cargas axiales y está definida por:

$$\mathbf{k}' = \frac{AE}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} .$$

2.4.2 Matriz de rigidez del elemento análisis cortante y flexión

Ahora se realizará el análisis para las direcciones de los grados de libertad Y' y Z' . En la Figura 2.6 se muestra un elemento con sus desplazamientos y cargas internas. Existen dos reacciones en los extremos del elemento que consisten en las fuerzas cortantes q'_{Ny} y q'_{Fy} y los momentos flectores q'_{Nz} y q'_{Fz} . A la vez se aprecian los desplazamientos en el eje Y' y las rotaciones en el eje Z' , las cuales son d'_{Ny} , d'_{Fy} y d'_{Nz} d'_{Fz} , respectivamente. Estas cargas tomarán sentido positivo en las direcciones que muestra la Figura 2.6.

Las dos consideraciones de movimiento cuando se tiene un $\Delta d'_{Ny}$ o $\Delta d'_{Fy}$ se las muestra en la Figura 2.7, junto a las ecuaciones que relaciona la carga con su desplazamiento. De la misma manera se realiza las suposiciones de movimiento cuando se tiene un $\Delta d'_{Nz}$ o $\Delta d'_{Fz}$ las cuales se las aprecia en la Figura 2.8 junto a las ecuaciones que relacionan su momento con su rotación.

Las ecuaciones de las Figuras 2.7 y 2.8, se obtienen del análisis sumatoria de fuerzas

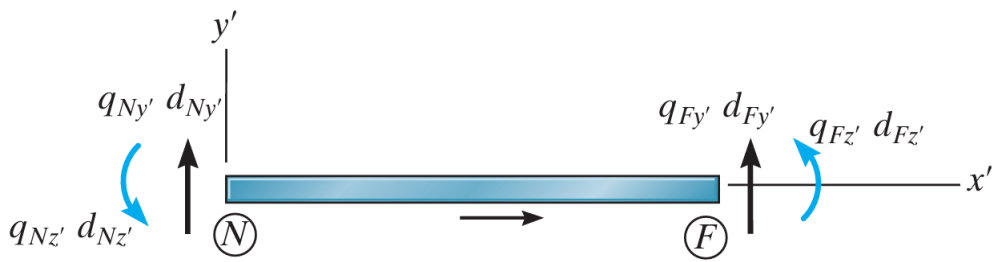


Figura 2.6: Elemento con desplazamientos y cargas internas en dirección de grados de libertad de Y' y Z' [4].

y momentos relacionado con las suposiciones de desplazamientos y restricciones que se aprecian en las mismas Figuras. En el capítulo 11 del libro [4], se explica en detalle la obtención de los coeficientes que relacionan las cargas y momentos con los desplazamientos y giros. A continuación, presentaremos la matriz de rigidez para el caso de flexión, uniendo estos casos, pero el lector interesado en la deducción de esta matriz puede encontrar los detalles en [4].

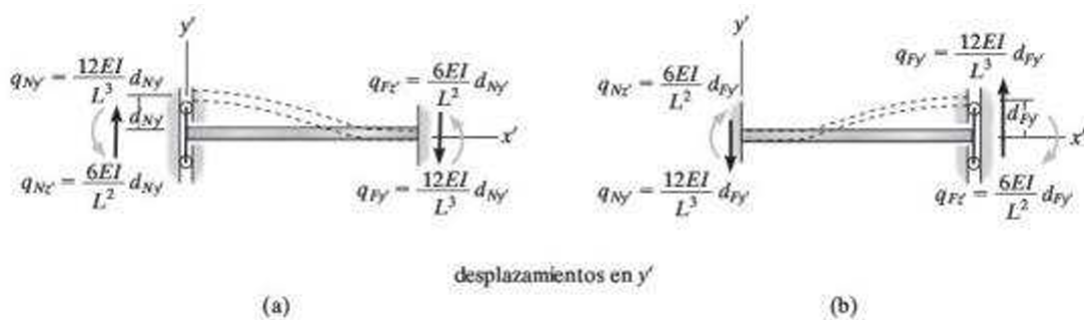


Figura 2.7: Elemento con ecuaciones de desplazamientos y cargas internas en dirección de grados de libertad de Y' [4].

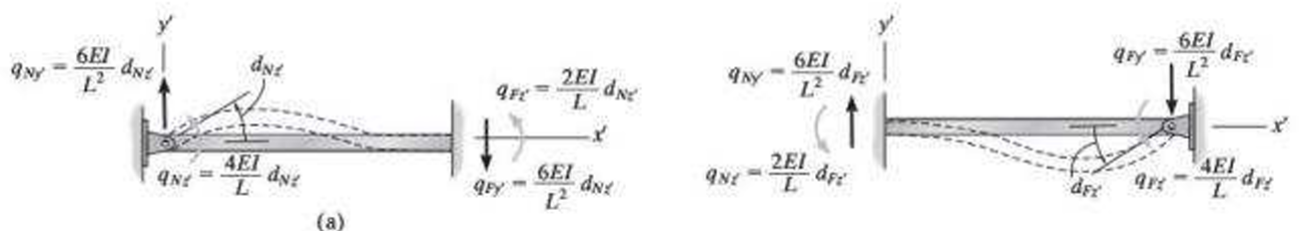


Figura 2.8: Elemento con ecuaciones de rotaciones y momentos flectores en dirección de grados de libertad de Z' [4].

Por superposición de las cuatro ecuaciones que relacionan la carga con el desplazamiento de las Figuras 2.7 y 2.8, se obtiene que:

$$\begin{bmatrix} q_N y' \\ q_N z' \\ q_F y' \\ q_F z' \end{bmatrix} = \begin{bmatrix} \frac{12EI}{L^3} & \frac{6EI}{L^2} & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ \frac{6EI}{L^2} & \frac{4EI}{L} & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ \frac{6EI}{L^2} & \frac{2EI}{L} & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \begin{bmatrix} d_N y' \\ d_N z' \\ d_F y' \\ d_F z' \end{bmatrix}. \quad (2.7)$$

La Ecuación 2.7 también se la puede representar de la siguiente manera:

$$\mathbf{q}' = \mathbf{k}' \mathbf{d}',$$

donde la matriz \mathbf{k}' , es la denominada matriz de rigidez local para un análisis de cargas cortantes y momentos flectores y está definida por:

$$\mathbf{k}' = \begin{bmatrix} \frac{12EI}{L^3} & \frac{6EI}{L^2} & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ \frac{6EI}{L^2} & \frac{4EI}{L} & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ \frac{6EI}{L^2} & \frac{2EI}{L} & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix}. \quad (2.8)$$

2.4.3 Matriz de rigidez del elemento

Como se mencionó anteriormente, utilizaremos el método de superposición para combinar estos dos análisis antes descritos (axial y flexión) y obtener así la matriz de rigidez del elemento en nuestro estudio. Este enfoque es posible porque ambos fenómenos son linealmente independientes y estamos trabajando dentro del marco de la mecánica elástica y lineal de pequeñas deformaciones de los elementos estructurales, sin considerar la plasticidad. Así que, de manera general, en cada elemento tenemos, en cada nodo una fuerza axial, una fuerza cortante, y un momento flector.

Entonces las cargas mostradas en la Figura 2.5 se las puede obtener del siguiente sistema de rigidez:

$$\begin{bmatrix} q_{Nx'} \\ q_{Ny'} \\ q_{Nz'} \\ q_{Fx'} \\ q_{Fy'} \\ q_{Fz'} \end{bmatrix} = \begin{bmatrix} \frac{AE}{L} & 0 & 0 & -\frac{AE}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{AE}{L} & 0 & 0 & \frac{AE}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \begin{bmatrix} d_{Nx'} \\ d_{Ny'} \\ d_{Nz'} \\ d_{Fx'} \\ d_{Fy'} \\ d_{Fz'} \end{bmatrix}. \quad (2.9)$$

La Ecuación 2.9 también se la puede representar como:

$$\mathbf{q}' = \mathbf{k}' \mathbf{d}',$$

donde la matriz \mathbf{k}' , es la denominada matriz de rigidez local para un análisis de cargas: axiales, cortantes y de momentos flectores y está definida por:

$$\mathbf{k}' = \begin{bmatrix} \frac{AE}{L} & 0 & 0 & -\frac{AE}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{AE}{L} & 0 & 0 & \frac{AE}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix}. \quad (2.10)$$

Esta matriz se conforma de treinta y seis coeficientes los cuales cumplen con la relación de equilibrio entre el desplazamiento y carga a la que está sometido un elemento. Cada columna y cada fila desde la primera a la sexta se relaciona directamente con sus grados de libertad, ya sean desplazamientos o rotaciones, d_{Nx} , d_{Ny} , d_{Nz} , d_{Fx} , d_{Fy} , d_{Fz} respectivamente.

2.5 MATRIZ DE TRANSFORMACIÓN DE LOS DESPLAZAMIENTOS Y DE LAS FUERZAS

Al analizar armaduras, es crucial tratar el sistema como un conjunto de elementos integrados. Por lo tanto, debemos ser capaces de transformar las cargas y desplazamientos del sistema de coordenadas locales (X' , Y' , Z') al sistema de coordenadas global (X , Y , Z). En

esta sección, desarrollaremos esta transformación para comprender cómo se comportan las cargas y desplazamientos en la armadura completa.

Se obtiene las Ecuaciones (2.14) y (2.15) del análisis de los desplazamientos de las Figuras 2.9 y 2.10. Estas Ecuaciones son las que transforman vectores en sistemas de coordenadas rotados.

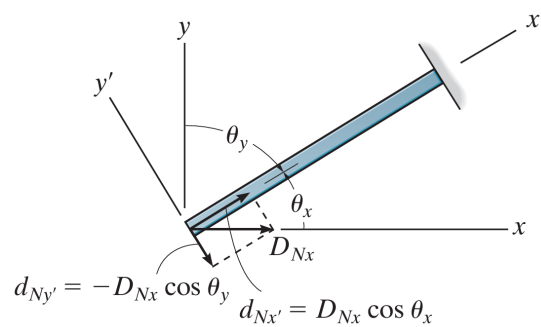


Figura 2.9: Análisis de desplazamientos locales para desplazamiento D_{Nx} [4].

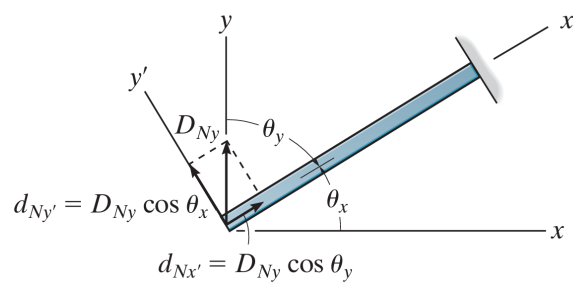


Figura 2.10: Análisis de desplazamientos locales para desplazamiento D_{Ny} [4]

Realizando una superposición de estos dos casos se llegan a las Ecuaciones (2.11) y (2.12), las cuales nos sirven para nuestro sistema de transformación de los desplazamientos.

$$d_{Nx}' = D_{Nx} \cos \theta_x + D_{Ny} \cos \theta_x, \quad (2.11)$$

$$d_{Ny}' = -D_{Nx} \cos \theta_y + D_{Ny} \cos \theta_x, \quad (2.12)$$

Los ejes Z y Z' se encuentran el mismo eje por lo cual cuando se genera una rotación D_{Nz} , se genera una rotación local d'_{Nz} donde:

$$d'_{Nz} = D_{Nz}. \quad (2.13)$$

De la misma manera, si se imponen desplazamientos en el extremo lejano D_{Fx} y D_{Fy} , y rotación D_{Fz} , se van a obtener los desplazamientos locales con las siguientes ecuaciones:

$$d_{Fx}' = D_{Fx} \cos\theta_x + D_{Fy} \cos\theta_y, \quad (2.14)$$

$$d_{Fy}' = -D_{Fx} \cos\theta_y - D_{Fy} \cos\theta_x, \quad (2.15)$$

y en el caso de tener D_{Fz} ,

$$d_{Fz}' = D_{Fz} \quad (2.16)$$

Realizando la superposición de los movimientos en el extremo lejano en los ejes locales X' y Y' llegamos a las Ecuaciones (2.14) y (2.15) las cuales nos sirven para nuestro sistema de transformación de los desplazamientos.

Por simplicidad, definimos las variables lambda como $\lambda_x = \cos\theta_x$ y que $\lambda_y = \cos\theta_y$, los cuales representan los cosenos directores de los elementos. A continuación, se presenta la transformación de un sistema de coordenadas global a un sistema de coordenadas local para los desplazamientos obtenidos mediante:

$$\begin{bmatrix} d_{Nx}' \\ d_{Ny}' \\ d_{Nz}' \\ d_{Fx}' \\ d_{Fy}' \\ d_{Fz}' \end{bmatrix} = \begin{bmatrix} \lambda_x & \lambda_y & 0 & 0 & 0 & 0 \\ -\lambda_y & \lambda_x & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_x & \lambda_y & 0 \\ 0 & 0 & 0 & -\lambda_y & \lambda_x & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} D_{Nx} \\ D_{Ny} \\ D_{Nz} \\ D_{Fx} \\ D_{Fy} \\ D_{Fz} \end{bmatrix}. \quad (2.17)$$

El siguiente sistema también se lo puede expresar de la siguiente manera:

$$\mathbf{d} = \mathbf{T} \mathbf{D},$$

donde T es la matriz la cual transforma los 6 desplazamientos globales en locales, por lo tanto, también es conocida como la matriz de transformación de desplazamientos.

De la misma manera que los desplazamientos, se puede hacer el análisis para el caso de las cargas. Las cargas locales se transforman en globales usando:

$$\begin{bmatrix} Q_{Nx} \\ Q_{Ny} \\ Q_{Nz} \\ Q_{Fx} \\ Q_{Fy} \\ Q_{Fz} \end{bmatrix} = \begin{bmatrix} \lambda_x & -\lambda_y & 0 & 0 & 0 & 0 \\ \lambda_y & \lambda_x & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_x & -\lambda_y & 0 \\ 0 & 0 & 0 & \lambda_y & \lambda_x & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} q_{Nx'} \\ q_{Ny'} \\ q_{Nz'} \\ q_{Fx'} \\ q_{Fy'} \\ q_{Fz'} \end{bmatrix}, \quad (2.18)$$

la cual también se la puede expresar de la siguiente manera:

$$\mathbf{Q} = \mathbf{T}^T \mathbf{q}, \quad (2.19)$$

donde T^T es la matriz la cual transforma las fuerzas internas de un elemento a las fuerzas internas en el sistema global de la armadura.

2.6 MATRIZ DE RIGIDEZ GLOBAL DEL MARCO-ELEMENTO

Para la obtención de la matriz de rigidez global de un elemento de la armadura, utilizaremos las transformaciones antes descritas y explicaremos su obtención a continuación. Utilizaremos la ecuación $\mathbf{d} = \mathbf{T} \mathbf{D}$ y se la reemplazará en la ecuación $\mathbf{q} = \mathbf{k}' \mathbf{d}$, dándonos la siguiente ecuación:

$$\mathbf{q} = \mathbf{k}' \mathbf{T} \mathbf{D}. \quad (2.20)$$

Por lo que, si analizamos la ecuación obtenida, vemos que se relaciona directamente con los desplazamientos globales. Por lo cual este resultado se va a sustituir en la Ecuación 2.19, i.e., $\mathbf{Q} = \mathbf{T}^T \mathbf{q}$, dándonos el siguiente resultado:

$$\mathbf{Q} = \mathbf{T}^T \mathbf{k}' \mathbf{T} \mathbf{D}. \quad (2.21)$$

Que también podemos definir de la siguiente manera:

$$\mathbf{Q} = \mathbf{K} \mathbf{D},$$

lo que resulta en:

$$\mathbf{K} = \mathbf{T}^T \mathbf{k}' \mathbf{T},$$

donde K representa la matriz de rigidez global de la armadura. Este procedimiento será de gran utilidad para determinar las cargas y desplazamientos globales de la armadura y analizar su comportamiento. Por lo que esta matriz global resulta en:

$$K = \begin{bmatrix} \left(\frac{AE}{L}\lambda_x^2 + \frac{12EI}{L^3}\lambda_y^2\right) & \left(\frac{AE}{L} - \frac{12EI}{L^3}\right)\lambda_x\lambda_y & -\frac{6EI}{L^2}\lambda_y & -\left(\frac{AE}{L}\lambda_x^2 + \frac{12EI}{L^3}\lambda_y^2\right) & -\left(\frac{AE}{L} - \frac{12EI}{L^3}\right)\lambda_x\lambda_y & -\frac{6EI}{L^2}\lambda_y \\ \left(\frac{AE}{L} - \frac{12EI}{L^3}\right)\lambda_x\lambda_y & \left(\frac{AE}{L}\lambda_y^2 + \frac{12EI}{L^3}\lambda_x^2\right) & \frac{6EI}{L^2}\lambda_x & -\left(\frac{AE}{L} - \frac{12EI}{L^3}\right)\lambda_x\lambda_y & -\left(\frac{AE}{L}\lambda_y^2 + \frac{12EI}{L^3}\lambda_x^2\right) & \frac{6EI}{L^2}\lambda_x \\ -\frac{6EI}{L^2}\lambda_y & \frac{6EI}{L^2}\lambda_x & \frac{4EI}{L} & \frac{6EI}{L^2}\lambda_y & -\frac{6EI}{L^2}\lambda_x & \frac{2EI}{L} \\ -\left(\frac{AE}{L}\lambda_x^2 + \frac{12EI}{L^3}\lambda_y^2\right) & -\left(\frac{AE}{L} - \frac{12EI}{L^3}\right)\lambda_x\lambda_y & \frac{6EI}{L^2}\lambda_y & \left(\frac{AE}{L}\lambda_x^2 + \frac{12EI}{L^3}\lambda_y^2\right) & \left(\frac{AE}{L} - \frac{12EI}{L^3}\right)\lambda_x\lambda_y & \frac{6EI}{L^2}\lambda_y \\ -\left(\frac{AE}{L} - \frac{12EI}{L^3}\right)\lambda_x\lambda_y & -\left(\frac{AE}{L}\lambda_y^2 + \frac{12EI}{L^3}\lambda_x^2\right) & -\frac{6EI}{L^2}\lambda_x & \left(\frac{AE}{L} - \frac{12EI}{L^3}\right)\lambda_x\lambda_y & \left(\frac{AE}{L}\lambda_y^2 + \frac{12EI}{L^3}\lambda_x^2\right) & -\frac{6EI}{L^2}\lambda_x \\ -\frac{6EI}{L^2}\lambda_y & \frac{6EI}{L^2}\lambda_x & \frac{2EI}{L} & \frac{6EI}{L^2}\lambda_y & -\frac{6EI}{L^2}\lambda_x & \frac{4EI}{L} \end{bmatrix} \quad (2.22)$$

La matriz presentada en la Ecuación 2.22 es una matriz simétrica de 6x6. Es importante destacar que esta matriz establece la relación entre los desplazamientos de los nodos cercanos y lejanos de un elemento estructural, expresados en coordenadas globales, y las fuerzas internas correspondientes en ambos nodos, también en coordenadas globales.

2.7 APLICACIÓN DEL MÉTODO

Una vez comprendido todos los parámetros y matrices del método de la rigidez, es necesario aplicarlos de una manera lógica. Para el análisis de marco-elementos se ha tomado los siguientes pasos recomendados:

1. Dividir la estructura en elementos finitos, y arbitrariamente numerar sus nodos y elementos.
2. Establecer el sistema de coordenadas global. Como se mencionó anteriormente, este sistema de coordenadas puede ubicarse en cualquier nodo de la armadura. Sin embargo, se recomienda colocarlo en un nodo estratégico donde la mayoría de sus coordenadas tengan valores positivos.
3. En este paso, enumeramos los grados de libertad. Recordar que son 3 grados de libertad por cada nodo y el analista puede empezar en el nodo que prefiera, aunque se recomienda que los números bajos estén en los nodos libres y los números altos estén en los nodos restringidos.
4. Definir el vector de cargas externas global Q y el vector de desplazamientos global D . Esto dependerá de las cargas que estará sometida la armadura y las restricciones de movimientos que esta tenga.

5. Determinar los cosenos directores λ_x y λ_y para cada elemento y empezar a calcular la matriz global del marco-armadura. Tener en cuenta todas las constantes características del elemento como área, rigidez o inercia para poder calcular todos los coeficientes de la matriz de rigidez global. A continuación se muestra cómo calcular los cosenos directores para cada elemento si conocemos las coordenadas de sus nodos (x_N, y_N) y (x_F, y_F) :

$$\lambda_x = \frac{x_F - x_N}{L}, \quad (2.23)$$

$$\lambda_y = \frac{y_F - y_N}{L}. \quad (2.24)$$

6. Obtener la matriz de rigidez en el marco de referencia global \mathbf{K} para cada elemento y una vez obtenidas, estas se suman acorde a los grados de libertad para así poder encontrar la matriz global del marco-armadura.
7. Una vez que se tenga el sistema de rigidez, hacer una partición de la matriz de la siguiente manera:

$$Q_k = K_{11}D_u + K_{12}D_k, \quad (2.25)$$

$$Q_u = K_{21}D_u + K_{22}D_k, \quad (2.26)$$

donde:

- Q_k : Vector de cargas conocidas.
- Q_u : Vector de cargas desconocidas.
- D_k : Vector de desplazamientos conocidos.
- D_u : Vector de desplazamientos desconocidos.

Los desplazamientos desconocidos D_u se los puede encontrar con la Ecuación (2.25), luego que se encuentren estos desplazamientos se pueden encontrar las reacciones Q_u con la Ecuación (2.26).

8. Finalmente para poder encontrar las cargas internas q se puede utilizar la siguiente ecuación para cada elemento:

$$q = k' T D. \quad (2.27)$$

En los siguientes capítulos, se presentarán ejemplos donde se aplicarán todas estas definiciones, ecuaciones y matrices. Estos ejemplos serán escritos en un código abierto en Python que implementará todas estas ecuaciones y el algoritmo del método.

3 METODOLOGÍA

En este capítulo, detallamos el desarrollo del código encargado de realizar el análisis de las estructuras. Los resultados obtenidos se presentarán de manera organizada y estarán disponibles para la integración con otros componentes de este proyecto, destinado a la evaluación de los elementos estructurales bajo la norma AISC y el otro a la selección de perfiles.

Se han creado dos archivos en Python: uno donde se ingresan los datos característicos tanto de la armadura como de los elementos, y otro que contiene las funciones necesarias para implementar el método de la rigidez.

3.1 DISEÑO DE INGRESO DE DATOS

A continuación, se explicará en detalle cada parte del código del primer archivo de Python, denominado "Ingreso de datos". Este código constituye la primera etapa del proyecto, encargada de la entrada de todos los datos característicos de los elementos de la armadura, tales como: resistencia a la fluencia y a la ruptura, rigidez de los elementos, momento de inercia y área transversal de los elementos.

Además, se procederá a la entrada de las coordenadas de los nodos de la estructura, seguida de la especificación de los grados de libertad, determinando si cada uno de estos está restringido o no. Finalmente, se ingresarán las cargas puntuales y distribuidas, así como los desplazamientos de la armadura.

Todos estos datos se ingresarán en forma de números flotantes, enteros, y vectores, permitiendo que posteriormente puedan ser llamados y utilizados en el archivo de Python "Metodo Rigidez" el cual contiene las funciones correspondientes del análisis estructural utilizando el método de la rigidez. Este meticuloso proceso de ingreso de datos garantiza que toda la información necesaria esté disponible para el análisis mediante el método de rigidez,

proporcionando así una base sólida para obtener resultados precisos y confiables.

3.1.1 Diseño de ingreso de datos característicos del elemento y la armadura

Como se mencionó anteriormente, la primera parte del código consiste en el ingreso de los datos característicos de los elementos y la armadura. Para el elemento se ingresa la resistencia a la fluencia y a la ruptura, la rigidez de los elementos, el momento de inercia y el área transversal de los elementos. En el caso de la armadura se ingresa el número de elementos que la conforma y el número de grados de libertad de toda la armadura. En la Tabla 3.1 se muestra cómo se denominó a cada entrada de datos característicos del elemento y la armadura.

Tabla 3.1: Lista de variables para ingreso de datos en el código de Python.

Variable	Nombre de variable en el código	Anotación
Resistencia a la fluencia	Fy	Resistencia a la fluencia de los elementos en [KSI]
Resistencia a la ruptura	Fu	Resistencia a la ruptura de los elementos en [KSI]
Número de elementos	Num_eslabones	Ingresar cuantos elementos tiene la armadura
Número de grados de libertad	mov	Ingresar cuantos grados de libertad tiene la armadura
Área transversal de los elementos	area_elemento	Ingresar el área transversal para todos los elementos en in^2
Rigidez de los elementos	rigidez_elemento	Ingresar la rigidez para todos los elementos en [KSI]
Inercia de los elementos	inercia	Ingresar la inercia para todos los elementos en in^4

Se puede apreciar que cada valor a ingresar está acompañado de un comentario detallado que indica exactamente qué dato se debe introducir. Después del ingreso de datos, se crean los vectores de Áreas A , Rigidez E e Inercia I . Estos valores ingresados pueden ser los mismos para todos los elementos de la armadura. En caso de que se desee asignar diferentes características a otros elementos y no repetir las mismas propiedades para todos, será necesario crear los vectores correspondientes e ingresar cada valor manualmente para cada elemento.

3.1.2 Diseño de ingreso de coordenadas

Después del ingreso de los datos característicos de los elementos y de la armadura, se encuentran cuatro vectores en los que se deben cargar las coordenadas de todos los elementos de la estructura. A continuación, se explicará cómo se diseñó la forma de estos cuatro vectores utilizando como ejemplo la armadura triangular presentada en el marco teórico de este escrito.

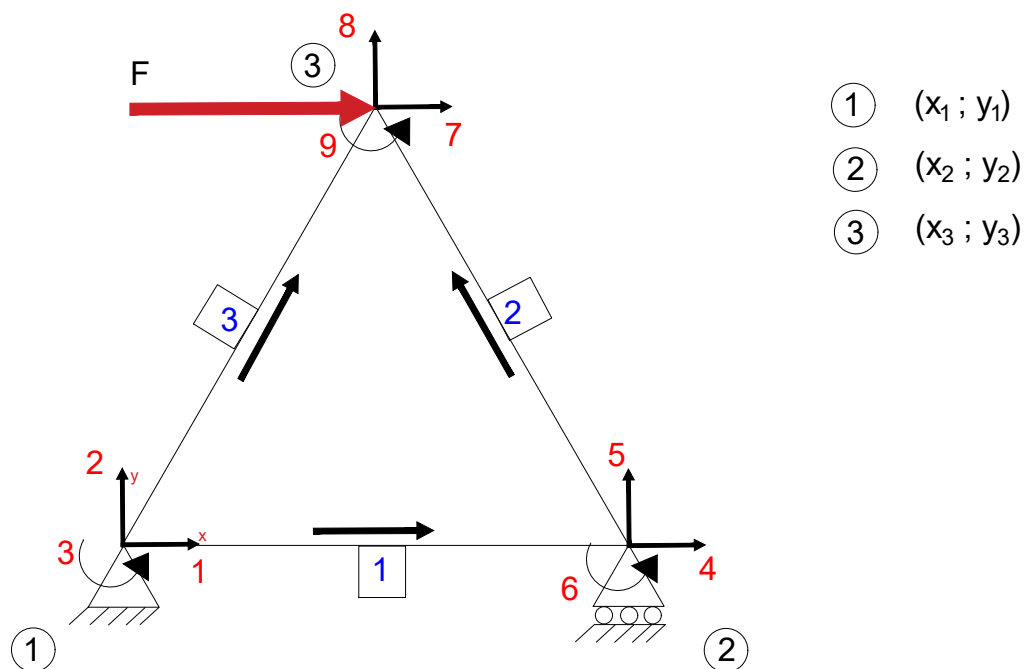


Figura 3.1: Armadura triangular de ejemplo para ingreso de datos (Fuente propia).

En la Figura 3.1 se observa que la armadura ejemplo tiene tres elementos, junto a las coordenadas de sus nodos. Como ejemplo, describiremos al elemento 1 y las coordenadas de sus dos nodos. El elemento 1 está en la parte inferior de la armadura, se puede notar que por la flecha apegada al elemento tenemos como extremo “cercano” el nodo uno y como extremo “lejano” el nodo dos. El nodo cercano corresponde al nodo uno con coordenadas $(x_1; y_1)$ las cuales van a corresponder para el primer valor de nuestros vectores x_n y y_n respectivamente. Por otra parte el nodo dos tiene las coordenadas $(x_2; y_2)$ las cuales van a corresponder al primer valor de nuestros vectores x_f y y_f respectivamente.

De la misma manera se tiene que realizar el análisis antes explicado, pero con el resto de elementos. En la Figura 3.2 se puede apreciar cómo quedan los vectores de coordenadas de la armadura.

①	(x ₁ ; y ₁)	$x_n = (x_1, x_2, x_1)$
②	(x ₂ ; y ₂)	$y_n = (y_1, y_2, y_1)$
③	(x ₃ ; y ₃)	$x_f = (x_2, x_3, x_3)$ $y_f = (y_2, y_3, y_3)$

Figura 3.2: Ingreso de coordenadas de armadura triangular de ejemplo (Fuente propia).

3.1.3 Diseño de ingreso de grados de libertad

Después de ingresar las coordenadas, se diseña el ingreso de los grados de libertad. En la Figura 3.1 se puede apreciar que la armadura tiene tres nodos. Dado que el análisis considera tres grados de libertad por nodo, se tiene un total de nueve grados de libertad para toda la armadura. A continuación, como ejemplo, describimos los grados de libertad para el elemento uno. El elemento uno tiene dos nodos, y cada nodo tiene 3 grados de libertad. Para el extremo cercano “nodo uno” se tienen los grados de libertad 1, 2 y 3 en los ejes x, y y z respectivamente, y estos grados de libertad pertenecen al primer coeficiente de los vectores $Near_x$, $Near_y$ y $Near_z$ respectivamente. Por otra parte en el extremo lejano “nodo dos” se tiene de igual manera los grados de libertad 4, 5 y 6 en el eje x, y y z y estos pertenecen al primer coeficiente de los vectores Far_x , Far_y y Far_z respectivamente.

Del mismo modo, se debe realizar el análisis de los demás elementos. El número de coeficientes de estos factores depende del número de elementos de la armadura que se esté analizando. Para este ejemplo tenemos tres coeficientes por vector. En la Figura 3.3 se detalla cómo quedan los vectores de grados de libertad respecto al análisis de la armadura triangular de ejemplo.

Elemento 1
Elemento 2
Elemento 3

$$\text{Near}_x = (1, 4, 1)$$

$$\text{Near}_y = (2, 5, 2)$$

$$\text{Near}_z = (3, 6, 3)$$

$$\text{Far}_x = (4, 7, 7)$$

$$\text{Far}_y = (5, 8, 8)$$

$$\text{Far}_z = (6, 9, 9)$$

Figura 3.3: Diseño de ingreso de grados de libertad de armadura triangular de ejemplo (Fuente propia).

Adicionalmente en este proceso, debemos determinar cuáles grados de libertad son libres o restringidos. Para este análisis, es necesario evaluar qué grados de libertad son posibles de moverse y cuáles no lo son. Los grados de libertad restringidos se encuentran en los soportes de una armadura. A continuación, se presenta una descripción de movimientos por nodos de la armadura de la Figura 3.1.

- Para el nodo 1, se tiene los grados de libertad 1, 2 y 3. Este nodo al estar con un soporte articulado se tiene que los grados de libertad 1 y 2 no se pueden mover por lo cual son restringidos, mientras tanto el grado de libertad 3 si lo puede hacer, por lo cual no tiene restricción y es libre.
- Para el nodo 2 notamos también un soporte, pero en este notamos la diferencia que el grado de libertad 4 se puede mover, por lo cual este no tiene restricción. El grado de libertad 5 no se puede mover ya que está anclado a la articulación, por lo cual este es restringido. Finalmente, el grado de libertad 6 es libre de rotar por lo cual este no es restringido.

- Para el nodo 3, notamos que no hay soportes, por lo cual los grados de libertad 7, 8 y 9 no son restringidos.

Estos grados de libertad se van a ingresar en dos vectores: “gdl no restringidos” y “gdl restringidos” por lo cual después del análisis, los vectores quedan de la siguiente manera:

- $gdl\ no\ restringidos = (3, 4, 6, 7, 8, 9)$

- $gdl\ restringidos = (1, 2, 5)$

3.1.4 Diseño de ingreso de cargas y desplazamientos de la armadura

El método de la rigidez admite cargas en los nodos, por lo que es fundamental ingresar los vectores de carga y desplazamientos para que puedan interactuar con la matriz de rigidez. En la Figura 3.1 se observa una única carga en los nodos, denominada F. Por ello, se ha creado el vector Q para ingresar las cargas y el vector D para el ingreso de desplazamientos. A continuación, se presenta el análisis de cada movimiento por nodo:

- Como se analizó previamente, para los desplazamientos de los grados de libertad 1 y 2 tienen un valor de cero por estar restringidos. El desplazamiento del grado de libertad 3 es incógnita del problema por estar libre de rotar.
- En contraste, para las cargas en el nodo 1, tenemos reacciones que son incógnitas para el grado de libertad 1 y 2. Mientras que para el grado de libertad 3, el momento externo es cero.
- Para el desplazamiento del grado de libertad 5 tiene un valor de cero por estar restringidos. El desplazamiento de los grados de libertad 4 y 6 son incógnitas por estar libres.
- Para las cargas en el nodo 2, se tiene una reacción que es una incógnita para el grado de libertad 5. Mientras que para los grados de libertad 4 y 6 son cero.
- En el nodo 3, para los desplazamientos de los grados de libertad 7, 8 y 9 son incógnitas por el hecho de estar libres.

- Al analizar las cargas, en la dirección del grado de libertad 7 se encuentra una fuerza que debe registrarse en el vector Q . mientras que los grados de libertad 8 y 9 tienen el valor de cero.

Como conclusión notamos que, al analizar cargas, cuando se tiene un grado de libertad restringido las cargas externas toman el valor de las reacciones, y al analizar desplazamientos estos toman el valor de cero. Por otra parte, al analizar grados de libertad no restringidos, las cargas pueden tener el valor de cero o una fuerza externa en el caso de haber, y para desplazamientos toma el valor de una incógnita d . De la armadura triangular de ejemplo, los vectores Q y D quedan de la siguiente manera:

$$Q = (R_1, R_2, 0, 0, R_5, 0, F, 0, 0),$$

$$D = (0, 0, d_3, d_4, 0, d_6, d_7, d_8, d_9),$$

donde Q es el vector de cargas externas, D es el vector de desplazamientos global, R_i es la reacción o momento en los soportes y finalmente d_i es el desplazamiento o rotación de los nodos de la armadura.

Para la parte de código, únicamente se va a ingresar cargas y desplazamientos conocidos que en el marco teórico de este escrito se los denominó Q_k y D_k , por lo cual se van a obviar las R y los d de los vectores y quedan de la siguiente manera:

$$Q_k = (0, 0, 0, F, 0, 0),$$

$$D_k = (0, 0, 0).$$

Al haber creado los vectores que determinan que grados de libertad son restringidos y cuales no, el código ya reconoce con que carga o desplazamiento trabajar.

3.1.5 Diseño de ingreso de cargas distribuidas

Como se describió en el alcance, el código desarrollado admite cargas distribuidas sobre los elementos, siempre y cuando la carga esté en dirección de la gravedad y se distribuya a lo largo de la totalidad del elemento. Sin embargo, el método de la rigidez solo acepta cargas en los nodos. Para resolver esto, se utiliza una descomposición de la carga distribuida que

transfiere toda la carga a los extremos del elemento.

En la Figura 3.4 se muestra cómo se descompone la carga distribuida en un elemento a sus nodos, en la cual mostramos como la carga distribuida se la descompone en la suma de dos partes: en las carga sobre las juntas de un elemento con extremo fijos; y las cargas y reacciones reales sobre un elemento fijamente apoyado.

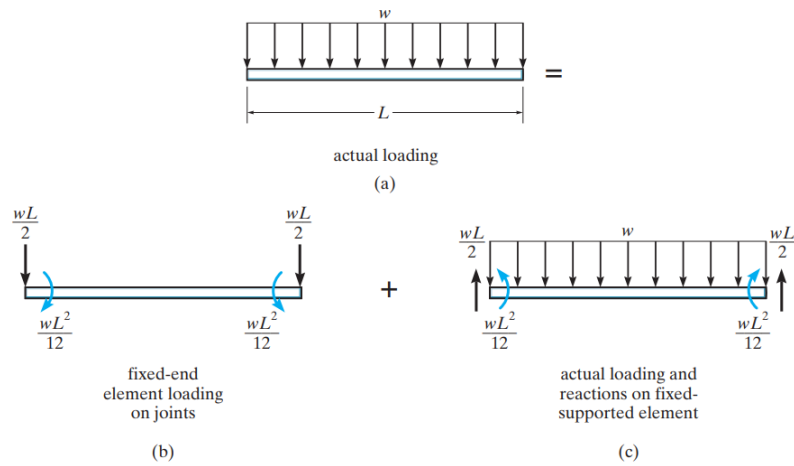


Figura 3.4: Descomposición carga distribuida [4].

Por lo cual las cargas sobre las juntas de un elemento con extremos fijos, se van a ingresar en el vector de cargas ya antes visto Q_k . Y las cargas y reacciones reales sobre un elemento fijamente apoyado serán ingresadas en una matriz, donde cada vector (Elemento N) representa estas cargas por elemento que contenga carga distribuida. En el caso que el elemento no tenga carga distribuida, se ingresará el vector del elemento con coeficientes igual a cero.

El vector por elemento que se tiene que ingresar es de la siguiente forma:

$$\text{Elemento } N = \left(0, \frac{wL}{2}, \frac{wL^2}{12}, 0, \frac{wL}{2}, -\frac{wL^2}{12} \right).$$

Observamos las siguientes características de este vector:

- El primer coeficiente del vector representa la fuerza axial en el eje x local del extremo cercano del elemento, al estar trabajando únicamente con cargas distribuidas en dirección de la gravedad esta siempre va a ser cero.
- El segundo coeficiente del vector representa la fuerza cortante en el eje y local del extremo cercano del elemento. Aquí w representa la carga aplicada y L la longitud del

elemento.

- El tercer coeficiente representa el momento ejercido por la carga distribuida en el eje z en el extremo cercano del elemento. Aquí w representa la carga distribuida y L la longitud del elemento.
- El cuarto coeficiente del vector representa la fuerza en el eje x local del extremo lejano del elemento, al estar trabajando únicamente con cargas distribuidas en dirección de la gravedad esta siempre va a ser cero.
- El quinto coeficiente del vector representa la fuerza cortante en el eje y local del extremo lejano del elemento. Aquí w representa la carga aplicada y L la longitud del elemento.
- El sexto coeficiente representa el momento ejercido por la carga distribuida en el eje z en el extremo lejano del elemento. Aquí w representa la carga distribuida y L la longitud del elemento.

Se tiene que considerar la convención de las direcciones de las cargas que se muestra en la Figura 3.4 para ingresar con el signo correspondiente respecto a la dirección de los grados de libertad. Se realizará un ejercicio con la Figura 3.1 de nuestra armadura triangular de ejemplo. En esta únicamente el elemento 1 puede contener una carga distribuida. Por lo cual se va a suponer que este elemento contiene una carga distribuida w .

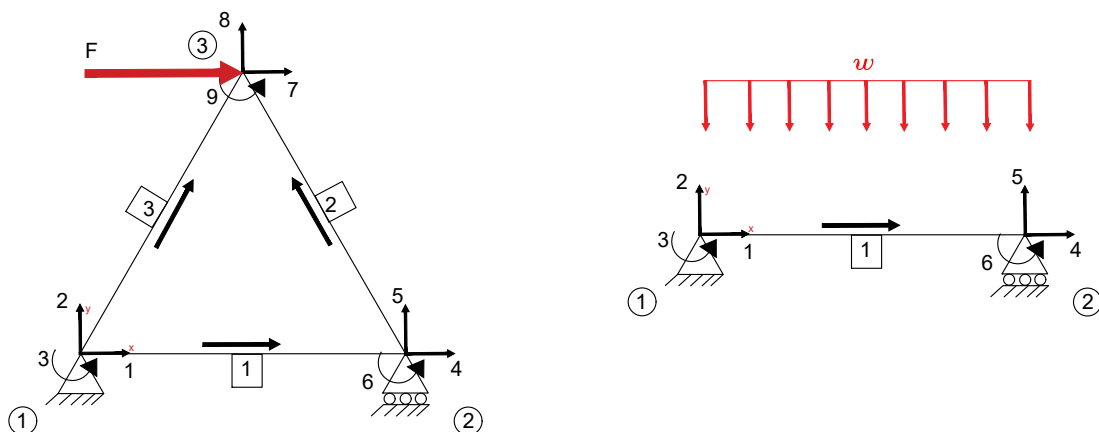


Figura 3.5: Carga distribuida en el elemento 1 (Fuente propia).

En la Figura 3.5 se aprecia la carga w sobre el elemento 1, la cual se va descomponer para realizar la matriz que contiene los vectores de las cargas y reacciones reales sobre un elemento fijamente apoyado. En la Ecuación (3.1) notamos que cada fila de la matriz, representa la carga distribuida por elemento, por lo cual en la primera fila tenemos la descomposición de la carga w sobre el elemento 1, mientras que en las demás filas toman el valor de cero por lo que no cuentan con carga distribuidas.

$$CARGAS_{EX} = \begin{bmatrix} 0 & \frac{wL}{2} & \frac{wL^2}{12} & 0 & \frac{wL}{2} & -\frac{wL^2}{12} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (3.1)$$

Al realizar el código se necesita saber si los elementos de la armadura están con carga distribuida, por lo cual el siguiente vector determina el tipo de elemento que representa lo siguiente: Valor de w si contiene carga distribuida, o valor de cero si no contiene carga distribuida. Para el ejemplo de la Figura 3.5 el vector que caracteriza los elementos queda de la siguiente manera:

$$tipo = [w \quad 0 \quad 0].$$

En el capítulo de análisis de resultados se apreciará de mejor manera el ingreso de datos ya que se realizará ejemplos de armaduras con y sin la presencia de cargas distribuidas con datos numéricos y explicados a detalle.

3.2 DISEÑO DE LA FUNCIÓN METODO DE RIGIDEZ

3.2.1 Diseño de la matriz de rigidez global

Lo primero que hace la función del método de rigidez, es crear la matriz de rigidez global de la armadura. Esto se logra con el condicional mostrado en el algoritmo de la Figura 3.6. De esta figura se observa que dentro del condicional, cada iteración trabaja exclusivamente con los datos característicos de cada elemento para construir su respectiva matriz de rigidez. Por lo tanto, en la primera iteración se obtiene la matriz de rigidez del elemento 1, en la segunda iteración la del elemento 2, y así sucesivamente. Cada matriz de rigidez del

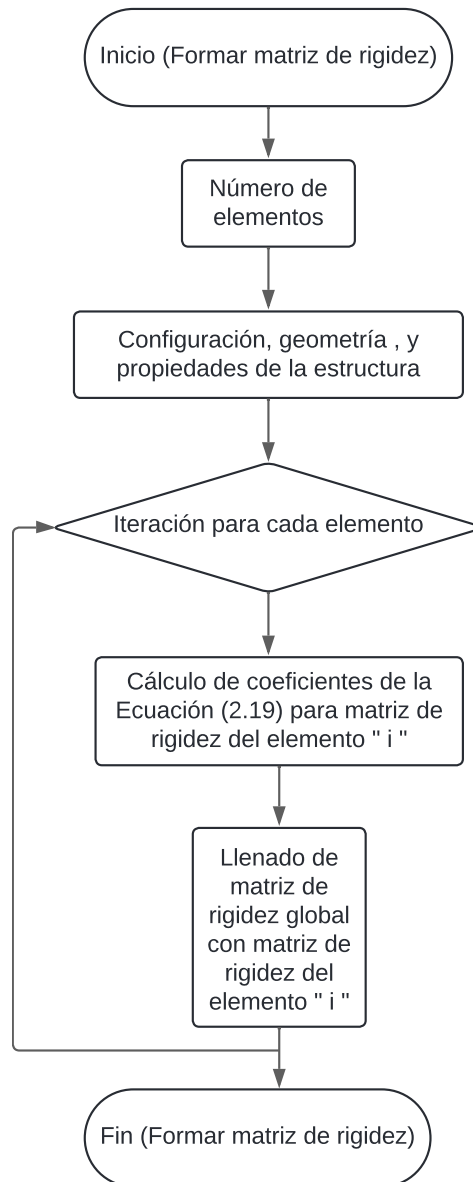


Figura 3.6: Algoritmo para construir y llenar la matriz de rigidez (Fuente: propia).

elemento se suma a la matriz de rigidez global de la armadura, acumulando así la matriz de rigidez global total al final del proceso.

3.2.2 Diseño de solución del sistema de rigidez

Después de obtener la matriz de rigidez ya se puede obtener el sistema de rigidez $Q = K D$ donde Q es el vector de cargas ingresado por el analista, K es la matriz de rigidez obtenida y D es el vector de desplazamientos ingresado por el analista.

Es fundamental organizar el sistema de rigidez para su posterior partición y resolución, tal

cual se explicó en la parte de aplicación del método. En el código desarrollado se emplea la biblioteca libre de Python conocida como Numpy, que permite fragmentar una matriz según las filas y columnas deseadas y así de esta forma obtener el sistema de las Ecuaciones (2.25) y (2.26). A continuación en la Figura 3.7, se muestra el código utilizado para la partición del sistema de rigidez.

```

1 k_11 = k[np.ix_(gdl_no_restringidos , gdl_no_restringidos)]
- k_21 = k[np.ix_(gdl_no_restringidos , gdl_restringidos)]
- k_12 = k[np.ix_(gdl_restringidos , gdl_no_restringidos)]
- k_22 = k[np.ix_(gdl_restringidos , gdl_restringidos)]

```

Figura 3.7: código para partición del sistema de ecuaciones (Fuente propia).

Para obtener las matrices Q_k y D_k de las Ecuaciones (2.25) y (2.26), no es necesario particionar las matrices ingresadas por el usuario. Como se explicó en la parte de diseño de ingreso de cargas y desplazamientos de la armadura, el usuario tiene que ingresar las cargas y desplazamientos sin las incógnitas, y el código realiza la partición necesaria. Utilizando las Ecuaciones (2.25) y (2.26) se las programa y se utiliza el código de la Figura 3.8 para la obtención de las incógnitas en los vectores Q y D .

```

D_u = np.linalg.solve(k_11,Q_k)
Q_u = np.dot(D_u,k_21)

```

Figura 3.8: Código para solucionar el sistema de rigidez (Fuente propia).

Una vez obtenidas todas las matrices y vectores del sistema de rigidez, es crucial ordenar los vectores de cargas Q_k con las cargas obtenidas Q_u y desplazamientos D_k con los desplazamientos obtenidos D_u . Estos vectores proporcionan al analista una visión clara del comportamiento esperado de la armadura frente a las cargas externas. En la parte de anexos se encuentra la parte del código que realiza esta operación descrita.

3.2.3 Diseño del código para la obtención de cargas internas de los elementos

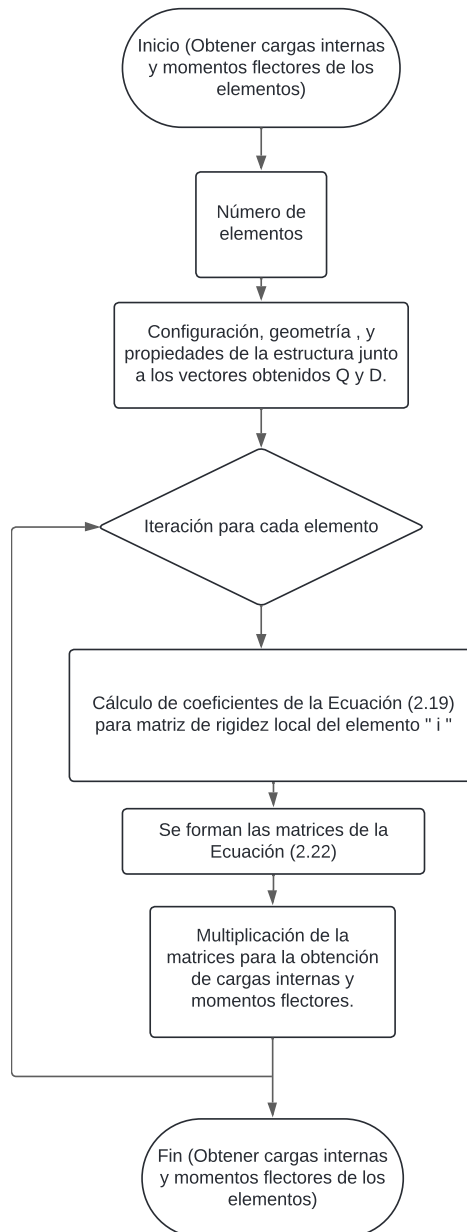


Figura 3.9: Algoritmo para obtener cargas internas (Fuente propia).

Al resolver el sistema de rigidez, se obtienen cargas y desplazamientos externos. Luego, se procede al desarrollo del código para la obtención de cargas y desplazamientos analizados por elemento. En este proceso, se utilizan las ecuaciones y matrices estudiadas en la primera parte del marco teórico. Es fundamental calcular la matriz de rigidez local para cada elemento y luego multiplicarla por una matriz de transformación para analizar el comportamiento de cada elemento de la armadura.

Este análisis es crucial para evaluar si los perfiles seleccionados cumplen con las normativas. Se obtiene la distribución de carga axial, cortante y los momentos en los extremos de los elementos provocados por las cargas externas.

Para obtener las cargas internas utilizamos la Ecuación (2.27) explicada en la parte de "Aplicación del método" del Marco teórico, donde se obtendrán las matrices relacionadas en la ecuación para poder encontrar la matriz de rigidez local de los elementos y así de esta manera conocer las cargas internas y momentos flectores. En la Figura 3.9 se muestra el algoritmo utilizado para esta parte del método.

En la parte de Anexos de este escrito se detalla la parte del código el cual consiste en el seguimiento del algoritmo de la Figura 3.9 para la obtención de las cargas internas.

3.2.4 Diseño del código para la impresión de resultados

Una vez resuelto el sistema global de rigidez y a la vez encontrado las cargas internas de los elementos, se puede terminar con el análisis de método de la rigidez. Antes de terminar, se realizará una impresión de las cargas internas máximas de los elementos y los momentos flectores para cada elemento.

Para ello, se realizará la impresión de las cargas de la siguiente manera (Tabla 3.2):

Tabla 3.2: Impresión de cargas máximas

Valor	Característica o longitud de ubicación
N	Cero o Uno
V_{max}	Longitud de ubicación
M_1	Longitud Near
M_2	Longitud de ubicación
M_3	Longitud Far

En la impresión de resultados, se obtendrá una lista la cual contendrá tantas matrices de la forma de la Tabla (3.2) como elementos la armadura tenga. A continuación, se analizará la impresión de los resultados máximos para comprender los resultados obtenidos:

- N será la carga axial que soporta el elemento y, a la derecha de esta carga, se encontrará el valor cero o uno: si refleja cero, indica que el elemento está en compresión; por otro lado, si muestra uno, significa que el elemento está en tensión.
- V_{max} representará la fuerza cortante máxima a la que está sometido el elemento. A la derecha se encontrará la distancia a lo largo del eje X' local en la que se ubica desde el nodo near (osea en su sistema de coordenadas local).
- M_1 , M_2 y M_3 representarán los momentos más representativos a los que está sometido el elemento. A la derecha de cada momento se encontrará la distancia en el eje

X' del sistema de coordenadas local del elemento en la cual se presenta dicho momento. En la siguiente sección que es de análisis de momentos, se explicará el tipo de momento generado, el cual depende de si el elemento está sometido a una carga distribuida o no.

3.2.5 Análisis de las cargas cortantes y los momentos internos

Para los análisis de carga cortante que soportarán los elementos a analizar existen dos casos: cuando el elemento esté sometido a una carga distribuida y en el caso que no exista presencia de esta.

En la Figura 3.10 se aprecia el modelo de diagrama de fuerza cortante que se obtendría en el primer caso, mientras que en la Figura 3.11 se aprecia el modelo de diagrama de fuerza cortante que se obtendría para el segundo caso. Estos valores son obtenidos del análisis del método de la rigidez.

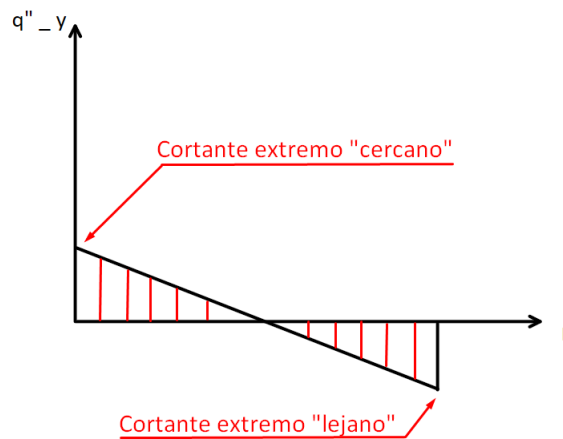


Figura 3.10: Diagrama del cortante sobre elemento con carga distribuida (Fuente Propia).

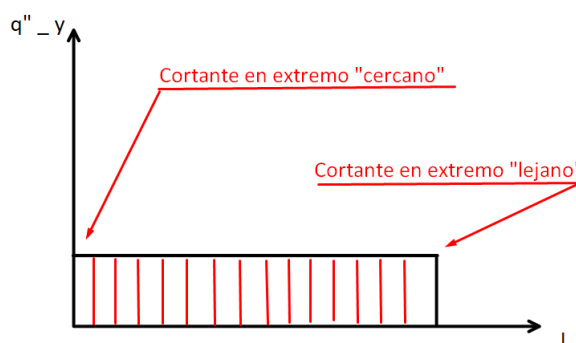


Figura 3.11: Diagrama del cortante sobre elemento sin carga distribuida (Fuente Propia).

Para obtener los tres momentos en las respuestas descritas anteriormente, el elemento debe estar sometido a una carga distribuida; de lo contrario, solo se obtendrán dos momentos de los extremos cercano y lejano. A continuación, se muestran los dos tipos de momentos que se pueden generar al analizar armaduras con este código.

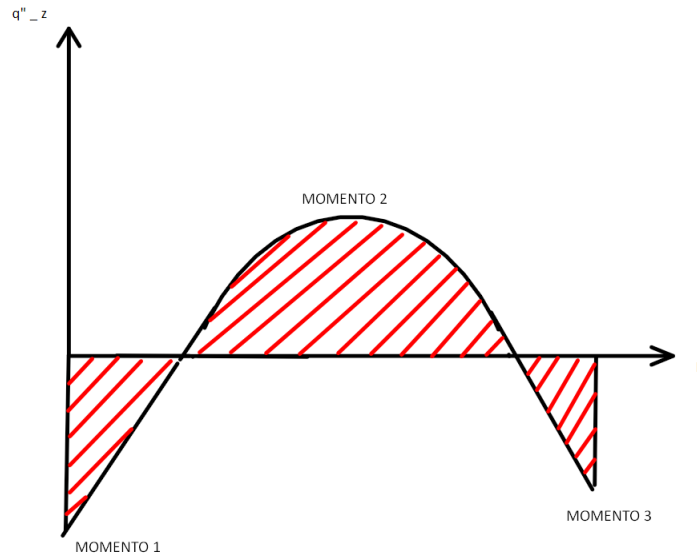


Figura 3.12: Diagrama del momento sobre elemento con carga distribuida (Fuente Propia).

De la Figura 3.12, el momento 1 y 3 son obtenidos del método de rigidez, pero con estos, después de realizar un análisis estático desde el extremo cercano del elemento, se puede obtener el momento 2. La ecuación a la que se llegó para obtener el momento 2 es la siguiente:

$$M_2 = -\frac{wL^2}{2} + q_{N_y}L - q_{N_z} \quad (3.2)$$

donde w es la carga distribuida sobre el elemento, L es la longitud del elemento, q_{N_y} es la carga cortante en el extremo cercano del elemento y q_{N_z} es el momento en el extremo cercano del elemento. El objetivo de determinar estos tres momentos es identificar cuál de estos puntos contiene el momento máximo, ya que esto es necesario para su posterior evaluación.

En la Figura 3.13 se observa el tipo de momentos presentes en el caso de que el elemento no tenga carga distribuida. En este caso, únicamente se considerarán los momentos 1 y 3, ya que son los momentos máximos a los que estará sometido el elemento, y estos son necesarios para realizar una evaluación posterior.

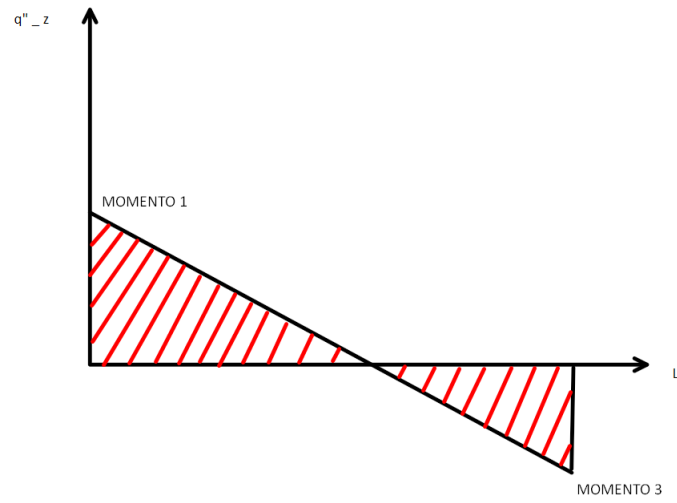


Figura 3.13: Diagrama del momento sobre elemento sin carga distribuida (Fuente propia).

4 RESULTADOS Y DISCUSIÓN

Para demostrar el funcionamiento del código desarrollado, se presentarán tres ejemplos de estructuras con sus análisis utilizando el método de la rigidez. En el primer ejemplo, se estudia las cargas en los elementos de una armadura sin la presencia de cargas distribuidas. En el segundo ejemplo, en cambio estudiaremos una estructura con la presencia de cargas distribuidas. Estos ejemplos serán explicados a detalle, abarcando tanto el ingreso de datos como la interpretación de resultados. Finalmente, se desarrollará un tercer ejemplo con un número considerable de elementos. Los resultados de estos ejemplos son reutilizados en los otros componentes del proyecto.

4.1 EJEMPLO 1. - ARMADURA DE 8 BARRAS CON CARGAS PUNTUALES

Empleando el método de la rigidez matricial, se calcula las reacciones en los soportes y la fuerza en cada uno de los elementos de la armadura mostrada en la Figura 4.1. El área transversal de los elementos es de 1.18 in^2 , los elementos tienen una inercia de 1.93 in^4 y una rigidez de 29000 ksi. Se utilizará acero estructural con una características de factores de resistencia a la fluencia y a la ruptura de $F_y = 30 \text{ ksi}$ y $F_u = 55 \text{ ksi}$ respectivamente.

Para resolver, se seguirán todos los pasos y recomendaciones previamente establecidos en el marco teórico y en la metodología de este escrito. Se realizará un análisis detallado, paso a paso, de cómo obtener los datos necesarios para ingresar al código y obtener los resultados requeridos.

Como primer paso, se enumerarán tanto los elementos como los nodos de la armadura. A continuación, se asignarán los números de grados de libertad a cada nodo y, finalmente, se determinarán los extremos cercano y lejano de cada elemento. El orden en el que se realizan estas tareas no está predeterminado, permitiendo al analista la libertad de escoger

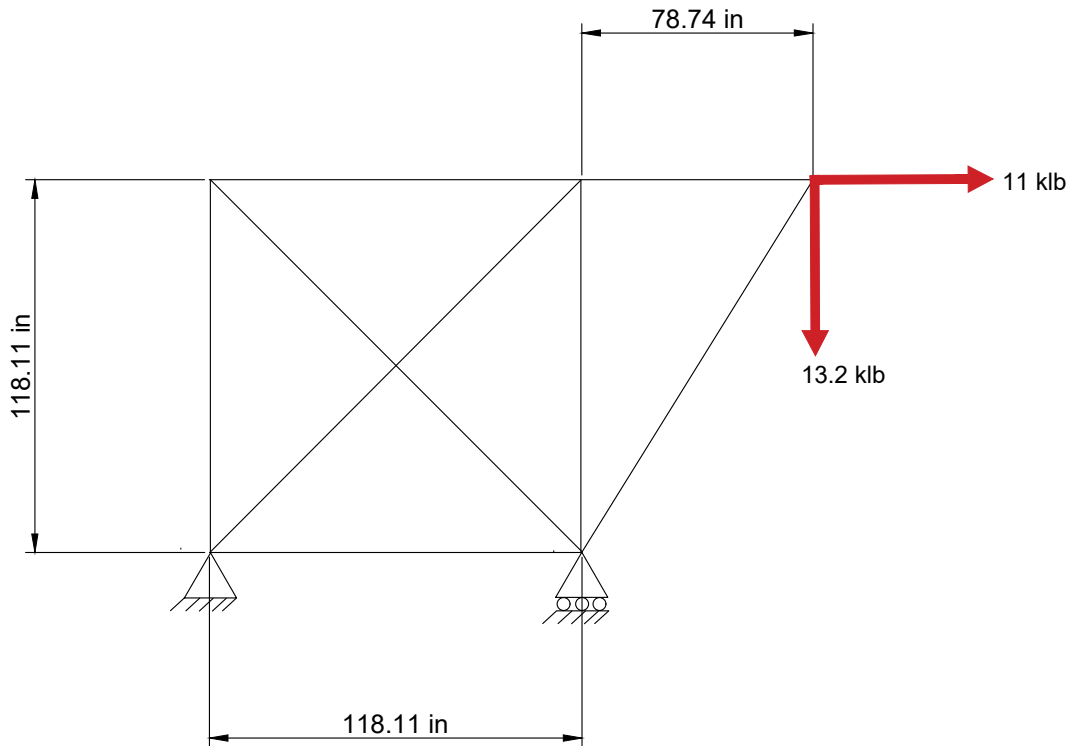


Figura 4.1: Configuración de la armadura para el Ejemplo 1 con sus dimensiones, fuerzas externas, y restricciones (Fuente propia).

el orden en que desea enumerar los elementos, nodos o grados de libertad.

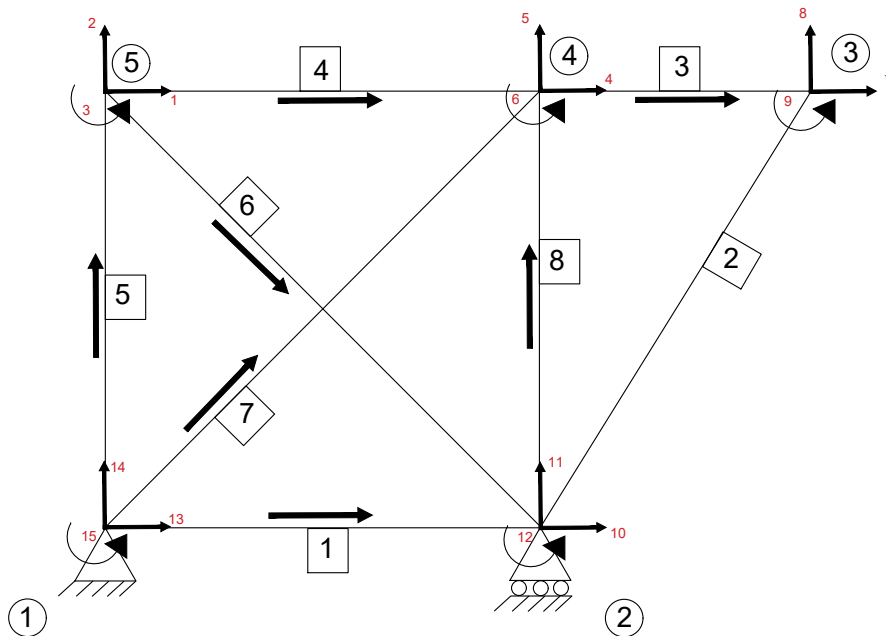


Figura 4.2: Enumeración de elementos, nodos, y grados de libertad para el ejemplo 1 (Fuente: propia).

En la Figura 4.2 se presenta la enumeración que se le a dado a todos estos elementos de la

armadura. Por lo tanto, nuestra armadura tiene 8 elementos y estos 8 elementos se juntan por medio de 5 nodos y se tiene un total de 15 grados de libertad.

Por lo cual se va a ir extrayendo información conforme pide el código. En primer lugar, pide datos característicos de la armadura y de los elementos. En la Tabla 4.1 se adjuntan los primeros datos que se tienen que ingresar.

Tabla 4.1: Datos característicos de la armadura y de los elementos

Tipo de coeficiente	Coeficiente en el código	Valor
Esfuerzo de fluencia	Fy	30 <i>KSI</i>
Esfuerzo de ruptura	Fu	55 <i>KSI</i>
Numero de elementos	Num_eslabones	8
Numero de grados de libertad	mov	15
Area de los elementos	area_elemento	1.1842 <i>in</i> ²
Rigidez de los elementos	rigidez_elemento	29000 <i>KSI</i>
Inercia de los elementos	inercia	1.93 <i>in</i> ⁴

Luego el código nos pide ingreso de coordenadas los cuales detallamos en la Tabla 4.2.

Tabla 4.2: Cordenadas ejemplo 1

Nodo	Cordenadas [<i>in</i>]
1	(0; 0)
2	(118.11; 0)
3	(196.85; 118.11)
4	(118.11; 118.11)
5	(0; 118.11)

Además en la Tabla 4.3 mostramos cómo se tiene que llenar los vectores de coordenadas del código:

Tabla 4.3: Cordenadas ejemplo 1

Vector	Datos [<i>in</i>]
X_n	(0, 118.11, 118.11, 0, 0, 0, 0, 118.11)
Y_n	(0, 0, 118.11, 118.11, 0, 118.11, 0, 0)
X_f	(0, 196.85, 196.85, 118.11, 0, 118.11, 118.11, 118.11)
Y_f	(0, 118.11, 118.11, 118.11, 118.11, 0, 118.11, 118.11)

Adicionalmente el código nos pide el ingreso de los grados de libertad, esto lo detallamos en la Tabla 4.4.

Tabla 4.4: Grados de libertad ejemplo 1

Vector	GDL
Near_x	(13, 10, 4, 1, 13, 1, 13, 10)
Near_y	(14, 11, 5, 2, 14, 2, 14, 11)
Near_z	(15, 12, 6, 3, 15, 3, 15, 12)
Far_x	(10, 7, 7, 4, 1, 10, 4, 4)
Far_y	(11, 8, 8, 5, 2, 11, 5, 5)
Far_z	(12, 9, 9, 6, 3, 12, 6, 6)

Luego se va a caracterizar estos grados de libertad como: restringidos y no restringidos como mostramos en los siguientes vectores de la Tabla 4.5:

Tabla 4.5: Grados de libertad libres y restringidos ejemplo 1

Vector	GDL
$\text{GDL}_{\text{NO RESTRINGIDO}}$	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 15)
$\text{GDL}_{\text{RESTRINGIDO}}$	(11, 14, 15)

También, se tiene que ingresar las matrices de carga y desplazamiento. En la Figura 4.1 notamos que en el nodo 3 se tiene dos cargas y están en los grados de libertad 7 y 8. Por lo cual hay que considerar estas posiciones para poner el valor de las cargas en la posición correspondiente a su grado de libertad. En la Tabla 4.7 se muestra cómo quedan estos vectores de carga y desplazamientos.

Tabla 4.6: Grados de libertad libres y restringidos ejemplo 1

Vector	Carga y desplazamiento
\mathbf{Q}	(0, 0, 0, 0, 0, 0, 11, -13.2, 0, 0, \mathbf{R}_{11} , 0, \mathbf{R}_{13} , \mathbf{R}_{14} , 0)
\mathbf{D}	(\mathbf{d}_1 , \mathbf{d}_2 , \mathbf{d}_3 , \mathbf{d}_4 , \mathbf{d}_5 , \mathbf{d}_6 , \mathbf{d}_7 , \mathbf{d}_8 , \mathbf{d}_9 , \mathbf{d}_{10} , 0, \mathbf{d}_{12} , 0, 0, \mathbf{d}_{15})

Pero se tiene que recordar que los vectores que se van a ingresar al código son sin las reacciones o los desplazamientos libres. Los vectores Q_k y D_k quedan de la siguiente manera:

Tabla 4.7: Grados de libertad libres y restringidos ejemplo 1

Vector	Carga y desplazamiento
\mathbf{Q}_k	(0, 0, 0, 0, 0, 0, 11, -13.6, 0, 0, 0, 0)
\mathbf{D}_k	(0, 0, 0)

Seguido de llenar estos datos, se tiene que ingresar la información de acuerdo con las cargas distribuidas de la armadura. En este ejemplo como no se tiene cargas distribuidas las matrices que abordan esta información tendrán valores de ceros. De la misma manera el vector "tipo" que registra la carga distribuida en los elementos, al ser inexistente en estos se procede a llenar con ceros, como mostramos a continuación en la Tabla 4.8.

En la parte de anexos se puede visualizar cómo queda el código de la parte de ingreso de

Tabla 4.8: Grados de libertad libres y restringidos ejemplo 1

Matriz o vector	Ingreso de valores
$CARGAS_{EX}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
<i>tipo</i>	[0 0 0 0 0 0 0 0]

datos del Ejemplo 1. Se procede a ejecutar el código y obtener los resultados (Tablas 4.9 y 4.10). Más adelante se realizará un análisis gráfico para mejorar la comprensión de la salida de datos.

Tabla 4.9: Resultado de reacciones y desplazamientos para el Ejemplo 1.

Vector	Vector de carga [<i>klb</i>] y desplazamiento [<i>in</i>]
Q	(0, 0, 0, 0, 0, 0, 11, -13.6, 0, 0, 33.66, 0, -11, -20.06, 0)
D	(0.1163, 0.031, -0.00068, 0.14, -0.038, -0.0013, 0.19, -0.21, -0.0005, 0, -0.0007, 0, 0, -0.0004)

En la Tabla 4.10 se muestran los valores de cargas internas y momentos flectores máximos, los cuales están en el formato de impresión del código. Se imprimen valores máximos de carga y los momentos. Esto será útil para el componente de evaluación.

Tabla 4.10: Resultado de máximas cargas internas y momentos flectores del ejemplo 1

Valor	Característica o longitud de ubicación
Elemento 1	
N = 0.2 klb	0 = Compresión
$V_{max} = 0$ klb	0 in
$M_1 = 1.5$ klb in	0 in
$M_2 = 0$ klb in	0 in
$M_3 = -1.9$ klb in	118.11 in
Elemento 2	
N = 16.3 klb	0 = Compresión
$V_{max} = 0$ klb	0 in
$M_1 = 1.4$ klb in	0 in
$M_2 = 0$ klb in	0 in
$M_3 = -0.1$ klb in	142 in
Elemento 3	
N = 20 klb	1 = Tensión
$V_{max} = 0$ klb	0 in
$M_1 = 1.8$ klb in	0 in
$M_2 = 0$ klb in	0 in
$M_3 = 0.1$ klb in	78.70 in
Elemento 4	
N = 8.9 klb	1 = Tensión
$V_{max} = 0$ klb	0 in
$M_1 = 0.9$ klb in	0 in
$M_2 = 0$ klb in	0 in
$M_3 = -1.5$ klb in	118.11 in
Elemento 5	
N = 8.9 klb	1 = Tensión
$V_{max} = 0$ klb	0 in
$M_1 = 1.4$ klb in	0 in
$M_2 = 0$ klb in	0 in
$M_3 = 1.1$ klb in	118.11 in
Elemento 6	
N = 12.5 klb	0 = Compresión
$V_{max} = 0$ klb	0 in
$M_1 = 0.2$ klb in	0 in
$M_2 = 0$ klb in	0 in
$M_3 = -0.3$ klb in	167 in
Elemento 7	
N = 15.8 klb	1 = Tensión
$V_{max} = 0$ klb	0 in
$M_1 = 0.1$ klb in	0 in
$M_2 = 0$ klb in	0 in
$M_3 = -0.5$ klb in	167 in
Elemento 8	
N = 11.2 klb	0 = Compresión
$V_{max} = 0$ klb	0 in
$M_1 = -0.8$ klb in	0 in
$M_2 = 0$ klb in	0 in
$M_3 = -0.2$ klb in	118.11 in

Se tiene valores de cortante y de momentos bajos relativos a las cargas de compresión y tensión que están sometidos los elementos, lo cual se puede VER en la Figura 4.3. Por otra parte, en la Figura 4.4 se puede observar que, mediante el método de la rigidez cuando se obtiene el vector Q del sistema de solución ya se puede conocer cuáles son las reacciones de la armadura. En la Figura 4.5 mostramos de una manera gráfica los elementos junto a sus cargas axiales indicando la dirección si estos se encuentran en tensión o compresión.

Elemento 2

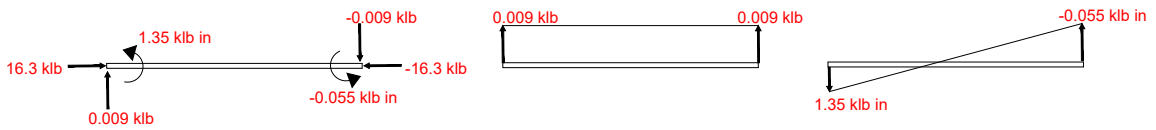


Figura 4.3: Diagramas cortante y momento del elemento 2 del Ejemplo 1 (Fuente propia).

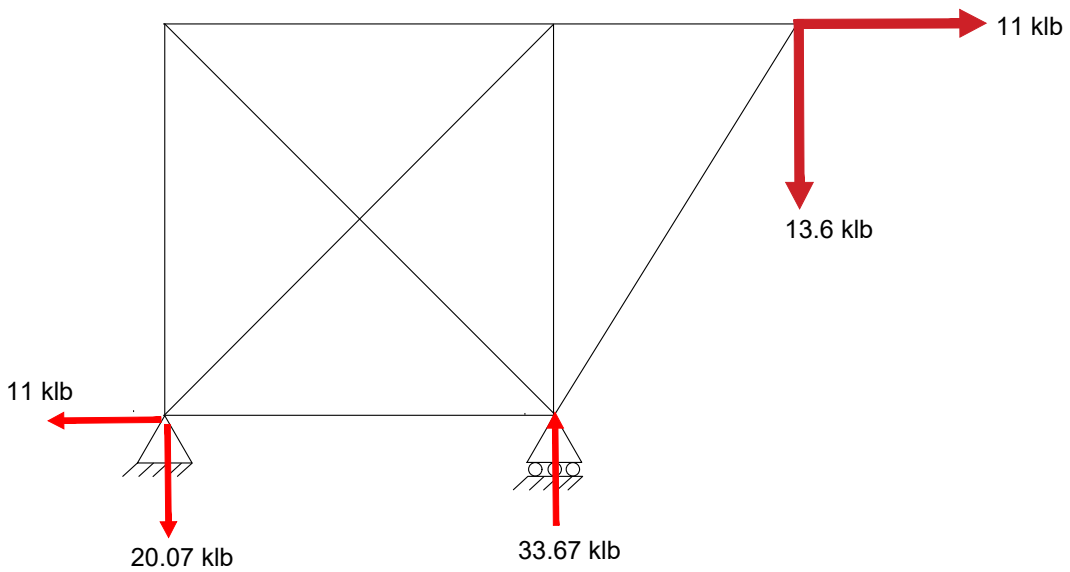


Figura 4.4: Resultado gráfico reacciones del análisis del ejemplo 1 (Fuente propia).

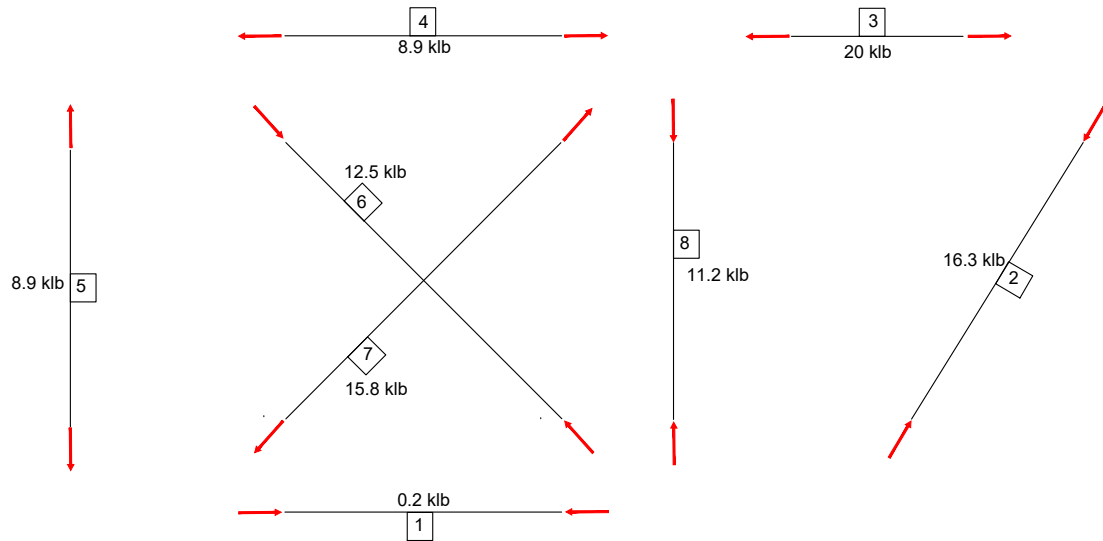


Figura 4.5: Resultado gráfico del análisis del ejemplo 1 (Fuente propia).

4.2 EJEMPLO 2. - ARMADURA DE 11 BARRAS CON CARGAS DISTRIBUIDAS

A diferencia del ejemplo analizado anteriormente, en este ejemplo la armadura considera cargas distribuidas uniformemente en algunos elementos, gracias a este tipo de cargas de este ejemplo se completará el alcance del análisis de marco elementos de esta herramienta computacional.

En este ejemplo buscamos determinar los momentos flectores y cargas internas de todos los elementos de la armadura de la Figura 4.6. La armadura se forma de elementos armando triángulos equiláteros con una longitud de 118.11 in, tienen un área transversal de 1.18 in² y una inercia de 1.93 in⁴. Los elementos son de acero y tienen las siguientes características: $F_y = 30$ ksi, $F_u = 55$ ksi y una rigidez de $E = 29000$ ksi.

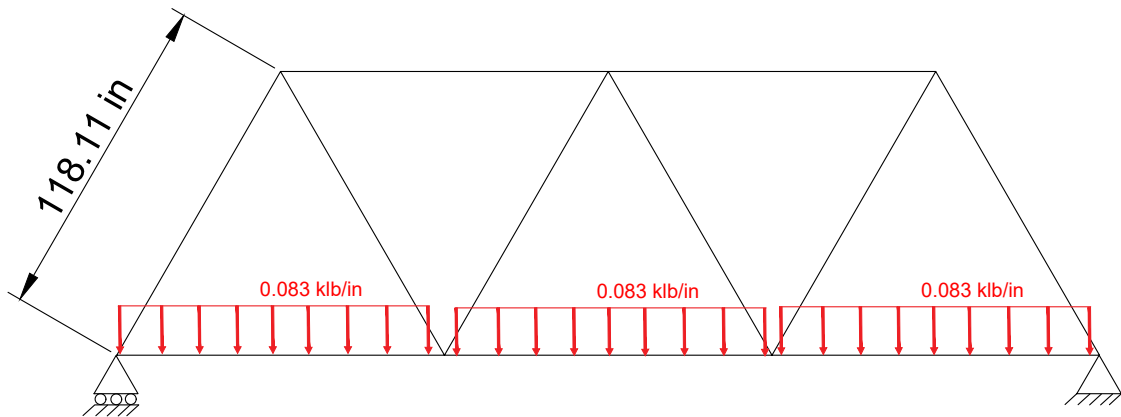


Figura 4.6: Configuración de la armadura para el Ejemplo 2 con sus dimensiones, fuerzas externas distribuidas, y restricciones (Fuente propia).

Como primer paso, se enumerarán tanto los elementos como los nodos de la armadura. A continuación, se asignará la numeración de grados de libertad a cada nodo, y finalmente, se determinarán los extremos cercano y lejano de cada elemento. A continuación en la Figura 4.7 se detalla la enumeración asignada para los elementos, nodos, y grados de libertad en la armadura.

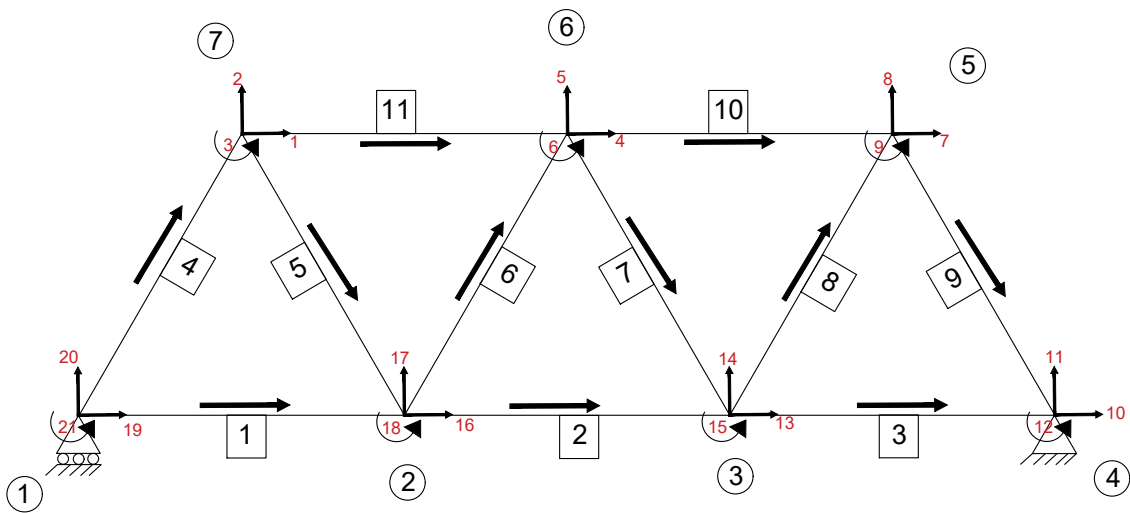


Figura 4.7: Enumeración de elementos, nodos, y grados de libertad para el ejemplo 2 (Fuente propia).

Por lo tanto, se puede decir que la siguiente armadura tiene 11 elementos, estos elementos se juntan por medio de 7 nodos y se contabilizaron 21 grados de libertad. Del enunciado del Ejemplo 2 y el análisis de la armadura se tiene en la Tabla 4.11 los datos característicos de la armadura y de los elementos, los cuales serán ingresados al código.

Tabla 4.11: Datos característicos de la armadura y de los elementos Ejemplo 2

Tipo de coeficiente	Coeficiente en el código	Valor
Esfuerzo de fluencia	Fy	30 <i>KSI</i>
Esfuerzo de ruptura	Fu	55 <i>KSI</i>
Numero de elementos	Num_eslabones	11
Numero de grados de libertad	mov	21
Area de los elementos	area_elemento	1.1842 <i>in</i> ²
Rigidéz de los elementos	rigidez_elemento	29000 <i>KSI</i>
Inercia de los elementos	inercia	1.93 <i>in</i> ⁴

Luego se realiza los cálculos geométricos correspondientes para saber las coordenadas de cada nodo. El sistema de referencia global está ubicado en el nodo uno. Y al tener triángulos equiláteros con una longitud de 118.11 *in*, las coordenadas de los nodos queda de la siguiente manera (Tabla 4.12) :

Tabla 4.12: Cordenadas ejemplo 2

Nodo	Cordenadas [<i>in</i>]
1	(0; 0)
2	(118.11; 0)
3	(236.22; 0)
4	(354.33; 0)
5	(295.275; 102.28)
6	(177.165; 102.28)
7	(59.055; 102.28)

De la convención de entrada de datos en el código se va a ingresar los datos respecto a nuestros vectores de extremos cercanos y lejanos según corresponda. En la Tabla 4.13 se aprecia cómo quedan los vectores de las coordenadas que se tiene que ingresar en el código en la parte de ingreso de datos:

Tabla 4.13: Cordenadas ejemplo 2

Vector	Datos [<i>in</i>]
X_n	(0, 118.11, 236.22, 0, 59.055, 118.11, 177.165, 236.22, 295.275, 177.165, 59.055)
Y_n	(0, 0, 0, 0, 102.28, 0, 102.28, 0, 102.28, 102.28, 102.28)
X_f	(118.11, 236.22, 354.33, 59.055, 118.11, 177.165, 236.22, 295.275, 354.33, 295.275, 177.165)
Y_f	(0, 0, 0, 102.28, 0, 102.28, 0, 102.28, 0, 102.28, 102.28)

Luego el código nos pide el ingreso de los grados de libertad. De la misma forma que se ingresó las coordenadas se va a realizar el ingreso de los grados de libertad por cada

elemento. En la Tabla 4.14 mostramos cómo se tienen que ingresar los vectores de grados de libertad:

Tabla 4.14: Grados de libertad ejemplo 2

Vector	GDL
Near_x	(19, 16, 13, 19, 1, 16, 4, 13, 7, 4, 1)
Near_y	(20, 17, 14, 20, 2, 17, 5, 14, 8, 5, 2)
Near_z	(21, 18, 15, 21, 3, 18, 6, 15, 9, 6, 3)
Far_x	(16, 13, 10, 1, 16, 4, 13, 7, 10, 7, 4)
Far_y	(17, 14, 11, 2, 17, 5, 14, 8, 11, 8, 5)
Far_z	(18, 15, 12, 3, 18, 6, 15, 9, 12, 9, 6)

De los grados de libertad ingresados hay que llenar los vectores de grados de libertad restringidos y no restringidos. Para ello se analiza los soportes de la armadura donde: se tiene un soporte articulado con ruedas en el nodo 1 y un soporte articulado fijo en el nodo 4. Para el nodo 1 involucra los grados de libertad 19, 20 y 21, por lo cual al tener ruedas permite al nodo desplazarse en el eje x dejando el grado de libertad 19 libre, por otra parte al estar fijo no permite desplazarse en el eje y por lo cual el grado de libertad 20 es restringido y finalmente al ser articulado es libre de rotar, por lo cual el grado de libertad 21 es libre.

Para el nodo 4 al ser un soporte fijo articulado, no permite desplazarse en el eje x y y , por lo que los grados de libertad 10 y 11 están restringidos y al ser articulado es libre de rotar por lo que el grado de libertad 12 es libre. Todos los otros grados de libertad del sistema serán considerados no restringidos debido a que no existe presencia de soportes. En la Tabla 4.15 se detallan los vectores de grados de libertad restringidos y grados de libertad libres de la armadura.

Tabla 4.15: Grados de libertad libres y restringidos ejemplo 2

Vector	GDL
GDL_{NO RESTRINGIDO}	(1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 13, 14, 15, 16, 17, 18, 19, 21)
GDL_{RESTRINGIDO}	(10, 11, 20)

A continuación, se tiene que realizar el análisis de las cargas distribuidas en los elementos 1, 2 y 3. Como se explicó en la parte de diseño de ingreso de cargas distribuidas refiriéndonos a la Figura 3.4 se procede a realizar el mismo análisis para los elementos antes mencionados:

- Se procede a realizar el cálculo para las cargas sobre las juntas de un elemento con

extremos fijos:

$$\frac{W L}{2} = \frac{0.083[klb/in] * 118.11[in]}{2} = 4.9[klb]$$

- El momento sobre las juntas de un elemento con extremos fijos:

$$\frac{W L^2}{12} = \frac{0.083[klb/in] * (118.11[in])^2}{12} = 96.48[klb in]$$

De la misma manera se tiene para la parte de cargas y reacciones reales sobre un elemento fijamente apoyado. En la Figura 4.8 mostramos cómo queda la descomposición de esta carga distribuida:

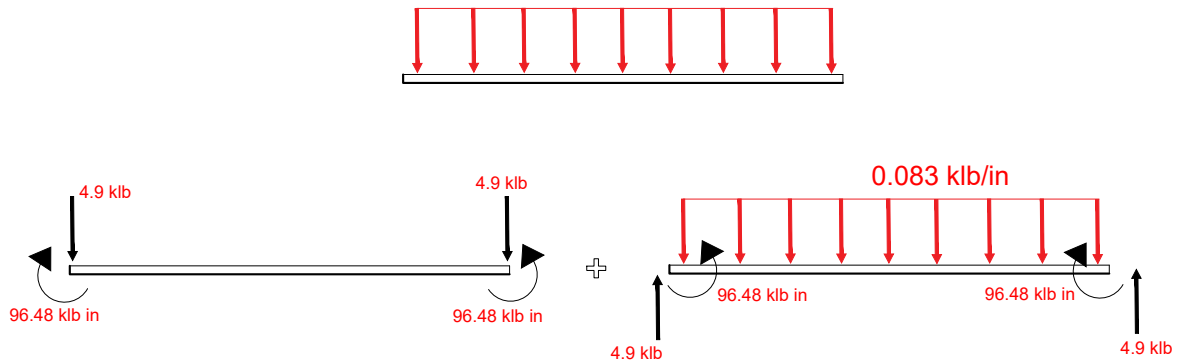


Figura 4.8: Descomposición carga distribuida (Fuente propia)

Por lo cual agregando las cargas descompuestas hacia los nodos se proceden a ingresar en la armadura y quedan de la siguiente manera como se muestra en la Figura 4.9:

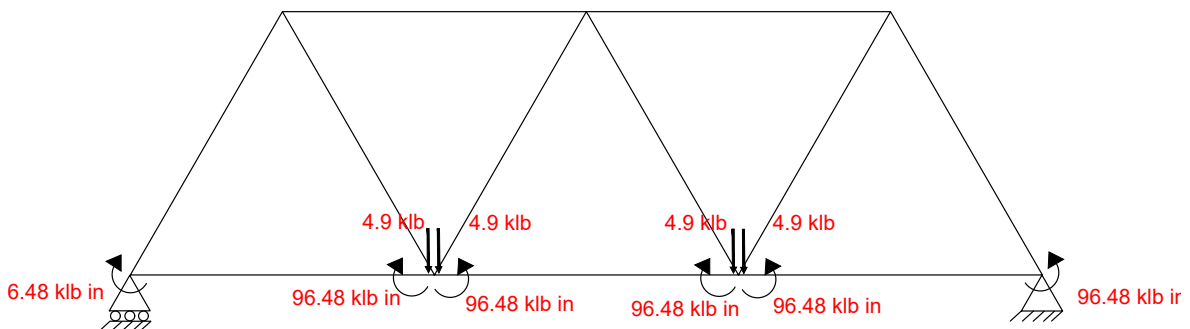


Figura 4.9: Descomposición carga distribuida en la armadura (Fuente propia)

Del análisis realizado se tiene que relacionar la carga obtenida con el grado de libertad correspondiente de cada elemento para poder ir ingresando los valores de carga en el vector Q . En la Tabla 4.17 se detalla cómo quedan los vectores Q y D para el análisis de nuestra armadura:

Tabla 4.16: Grados de libertad libres y restringidos ejemplo 2

Vector	Carga y desplazamiento
Q	$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, R_{10}, R_{11}, 96.48, 0, -9.8, 0, 0, -9.8, 0, 0, R_{20}, -96.48)$
D	$(d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9, 0, 0, d_{12}, d_{13}, d_{14}, d_{15}, d_{16}, d_{17}, d_{18}, d_{19}, 0, d_{21})$

Se tiene que obviar reacciones de los nodos prescritos y los desplazamientos de los nodos libres porque esas son las incógnitas de problema. De tal manera que nos queden coeficientes numéricos en nuestros vectores Q_k y D_k . En la Tabla 4.17 mostramos cómo quedan los vectores para nuestro ingreso de datos.

Tabla 4.17: Grados de libertad libres y restringidos ejemplo 2

Vector	Carga y desplazamiento
Q_k	$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 96.48, 0, -9.8, 0, 0, -9.8, 0, 0, -96.48)$
D_k	$(0, 0, 0)$

Luego se tiene que ingresar la matriz que contiene los vectores de las cargas y reacciones reales de un elemento fijamente apoyado. En el caso de nuestro ejemplo analizado colocamos carga distribuida en los elementos 1, 2 y 3. Por lo que los vectores de estos elementos tendrán las cargas de la parte derecha de la Figura 4.8. Con esto se tendrán las cargas reales de la armadura. A continuación, en la matriz $CARGAS_{EX}$ se aprecia como quedan los vectores a ingresar al código:

$$CARGAS_{EX} = \begin{bmatrix} 4.9 & 0 & 96.48 & 4.9 & 0 & -96.48 \\ 4.9 & 0 & 96.48 & 4.9 & 0 & -96.48 \\ 4.9 & 0 & 96.48 & 4.9 & 0 & -96.48 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Y para realizar la parte de análisis de momentos se llena el vector **tipo**, en la cual solo los primeros tres elementos tienen que ir con el valor de la carga distribuida debido a que solo los tres primeros elementos tienen este tipo de carga, por otra parte, los otros valores se los llena con ceros de la siguiente manera.

$$tipo = [0.083 \quad 0.083 \quad 0.083 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

En la parte de anexos se encuentra el código del ingreso de datos del Ejemplo 2. Una vez ingresado los vectores y matrices y coeficientes característicos de la armadura y de los elementos, se procede a correr el código y se obtiene directamente las cargas internas y momentos flectores de cada elemento. En la Tabla 4.18 se detallan los resultados obtenidos.

Tabla 4.18: Resultado de máximas cargas internas y momentos flectores del ejemplo 1

Valor	Característica o longitud de ubicación
Elemento 1	
N = 6.3 klb	1 = Tensión
$V_{max} = 5.5$ klb	0 in
$M_1 = 47.2$ klb in	0 in
$M_2 = 65.1$ klb in	0 in
$M_3 = -116$ klb in	118.11 in
Elemento 2	
N = 11.3 klb	1 = Tensión
$V_{max} = 5$ klb	0 in
$M_1 = 103.6$ klb in	0 in
$M_2 = 48.9$ klb in	0 in
$M_3 = -88.1$ klb in	118.11 in
Elemento 3	
N = 5 klb	1 = Tensión
$V_{max} = 5.5$ klb	0 in
$M_1 = 74.5$ klb in	0 in
$M_2 = 36.5$ klb in	0 in
$M_3 = -146.3$ klb in	118.11 in
Elemento 4	
N = 11.7 klb	0 = Compresión
$V_{max} = 0.6$ klb	0 in
$M_1 = 49.2$ klb in	0 in
$M_2 = 0$ klb in	0 in
$M_3 = -18.1$ klb in	118.11 in
Elemento 5	
N = 11.2 klb	1 = Tensión
$V_{max} = 0.2$ klb	0 in
$M_1 = 11$ klb in	0 in
$M_2 = 0$ klb in	0 in
$M_3 = 9.6$ klb in	118.11 in
Elemento 6	
N = 0.9 klb	1 = Tensión
$V_{max} = 0$ klb	0 in
$M_1 = 2.8$ klb in	0 in
$M_2 = 0$ klb in	0 in
$M_3 = -0.8$ klb in	118.11 in
Elemento 7	
N = 0.9 klb	0 = Compresión
$V_{max} = 0$ klb	0 in
$M_1 = 0.7$ klb in	0 in
$M_2 = 0$ klb in	0 in
$M_3 = 4.3$ klb in	118.11 in
Elemento 8	
N = 11.4 klb	1 = Tensión
$V_{max} = 0.2$ klb	0 in
$M_1 = 9.4$ klb in	0 in
$M_2 = 0$ klb in	0 in
$M_3 = 11.1$ klb in	118.11 in

Elemento 9	
N = 10.9 klb	0 = Compresión
$V_{max} = 5.9$ klb	0 in
$M_1 = 17.2$ klb in	0 in
$M_2 = 0$ klb in	0 in
$M_3 = -46.7$ klb in	118.11 in
Elemento 10	
N = 10.9 klb	0 = Compresión
$V_{max} = 0$ klb	0 in
$M_1 = 0.6$ klb in	0 in
$M_2 = 0$ klb in	0 in
$M_3 = 6.1$ klb in	118.11 in
Elemento 11	
N = 11.8 klb	0 = Compresión
$V_{max} = 0.1$ klb	0 in
$M_1 = 7.2$ klb in	0 in
$M_2 = 0$ klb in	0 in
$M_3 = -2.1$ klb in	118.11 in

Además, en la Figura 4.10, presentamos los resultados gráficamente de superior a inferior para los elementos 1, 2, 3 y 4 respectivamente. Los demás elementos son muy parecidos al elemento 4 debido a la inexistencia de carga distribuida en los elementos. De esta manera se puede observar las cargas internas y los momentos flectores a los que están sometidos los elementos. La figura muestra, de izquierda a derecha: Las cargas internas en cada elemento, calculadas mediante el método de rigidez, el diagrama de fuerza cortante a lo largo del elemento y el diagrama de momentos internos para el mismo elemento. Es importante destacar que las cargas internas representadas, obtenidas con el método de rigidez, corresponden a cargas externas en relación con otros elementos. Los diagramas de fuerza cortante y momentos internos reflejan estas cargas internas, proporcionando una visión integral del comportamiento estructural de los elementos analizados.

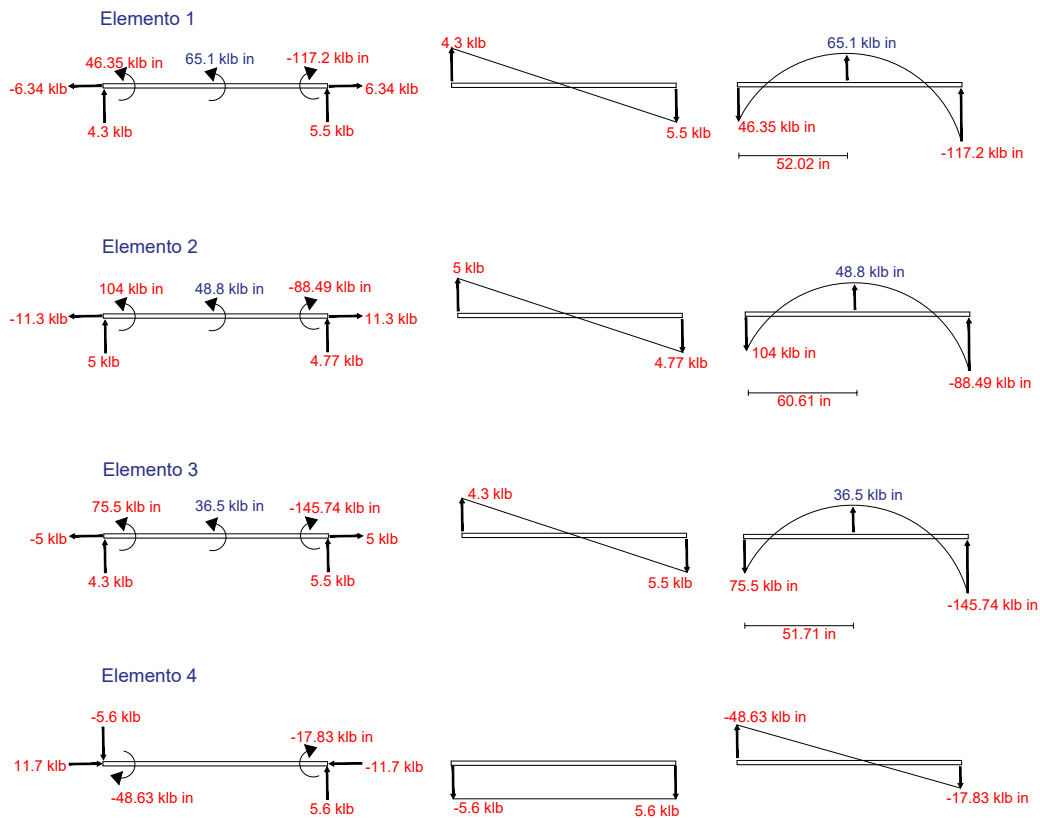


Figura 4.10: Diagramas Ejemplo 2 (Fuente propia).

4.3 EJEMPLO 3. - ARMADURA CON 61 BARRAS CON CARGAS PUNTUALES Y DISTRIBUIDAS

Se realiza el análisis de una estructura compuesta por 61 elementos como se muestra en la Figura 4.11. Esta estructura está soportada por 4 nodos los cuales tienen restringidos los grados de libertad en dirección x y y , además se colocan 7 fuerzas puntuales en los nodos superiores y dos cargas distribuidas en los elementos 2 y 15. Las características mecánicas de los elementos son las mismas que en los ejemplos anteriores. Debido a la gran cantidad de datos que se necesita ingresar, por conveniencia mostraremos los resultados directamente de los 61 elementos. El procedimiento de análisis es similar al del ejemplo 1 y 2, con la diferencia de que ahora se incluye un mayor número de elementos y más cargas externas. En la sección de anexos se puede evidenciar el ingreso de datos al código.

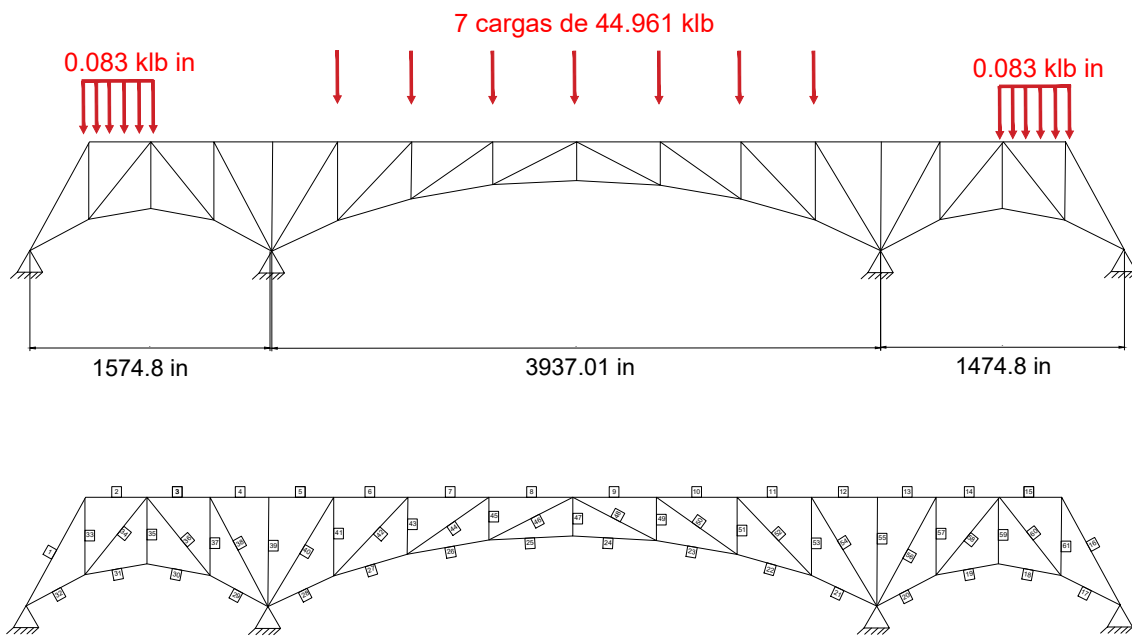


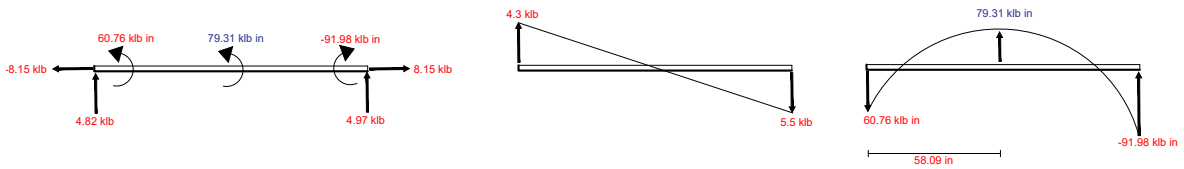
Figura 4.11: Configuración de la armadura para el Ejemplo 2 con sus dimensiones, fuerzas externas distribuidas, y restricciones (Fuente propia).

En la Tabla 4.19 se presentan las cargas axiales obtenidas del análisis mediante el método de rigidez. La tabla muestra el número de elemento junto con su carga axial y su estado. Según la metodología descrita, el código asigna un valor de cero o uno, donde cero indica compresión y uno indica tensión. Por otra parte, en la Figura 4.12 se aprecian los diagramas cortantes y momentos internos de los elementos 2 y 15, los cuales contienen la carga distribuida. La figura muestra, de izquierda a derecha: Las cargas internas en cada elemento, calculadas mediante el método de rigidez, el diagrama de fuerza cortante a lo largo del elemento y el diagrama de momentos internos para el mismo elemento.

Tabla 4.19: Resultado de cargas axiales del Ejemplo 3

Elemento	Carga Axial N [klb]	Estado	Elemento	Carga Axial N [klb]	Estado
1	17.35	1=Tensión	32	15.04	1=Tensión
2	8.15	1=Tensión	33	20.08	0=Compresión
3	44.33	1=Tensión	34	25.58	1=Tensión
4	53.41	1=Tensión	35	1.71	1=Tensión
5	53.41	1=Tensión	36	33.24	0=Compresión
6	13.37	1=Tensión	37	16.80	1=Tensión
7	56.49	0=Compresión	38	19	0=Compresión
8	142.08	0=Compresión	39	0.03	0=Compresión
9	142.26	0=Compresión	40	78.27	0=Compresión
10	56.71	0=Compresión	41	22.57	1=Tensión
11	13.09	1=Tensión	42	102.21	0=Compresión
12	53.21	1=Tensión	43	29.93	1=Tensión
13	53.21	1=Tensión	44	104.85	0=Compresión
14	44.42	1=Tensión	45	15.9	1=Tensión
15	4.23	1=Tensión	46	49.48	0=Compresión
16	9.12	1=Tensión	47	0.65	1=Tensión
17	20.98	1=Tensión	48	49.28	0=Compresión
18	0.75	1=Tensión	49	15.84	1=Tensión
19	0.66	1=Tensión	50	104.78	0=Compresión
20	24.34	0=Compresión	51	29.9	1=Tensión
21	212.72	0=Compresión	52	102.16	0=Compresión
22	128.8	0=Compresión	53	22.65	1=Tensión
23	38.39	0=Compresión	54	78.38	0=Compresión
24	5.95	1=Tensión	55	0.03	0=Compresión
25	5.95	1=Tensión	56	18.40	0=Compresión
26	38.57	0=Compresión	57	16.27	1=Tensión
27	129.07	0=Compresión	58	35.53	0=Compresión
28	213.04	0=Compresión	59	0.22	1=Tensión
29	17.81	0=Compresión	60	28.75	1=Tensión
30	5.05	1=Tensión	61	12.84	0=Compresión
31	5.15	1=Tensión			

Elemento 2



Elemento 15

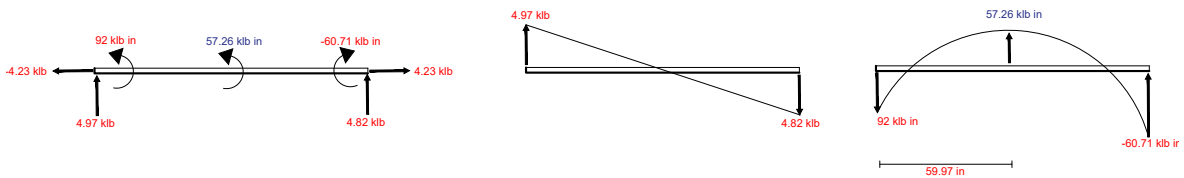


Figura 4.12: Configuración de la armadura para el Ejemplo 2 con sus dimensiones, fuerzas externas distribuidas, y restricciones (Fuente propia).

4.4 TIEMPO DE PROCESAMIENTO

Se analiza el tiempo de procesamiento o recurso computacional de la herramienta computacional desarrollada, para ello utilizamos el tiempo computacional que tomó obtener los resultados de los Ejemplos 1, 2 y 3. Estos ejemplos van de menor a mayor número de elementos, con el cual realizamos una gráfica de tiempo de procesamiento vs número de elementos.

Se utiliza la librería libre *time* con la cual, calcula el tiempo en segundos en procesar un código. El cálculo se lo realizó en un computador con un procesador I7 de sexta generación con 32Gb de memoria Ram. En la Figura 4.13 mostramos el tiempo de procesamiento en milisegundos vs los casos analizados.

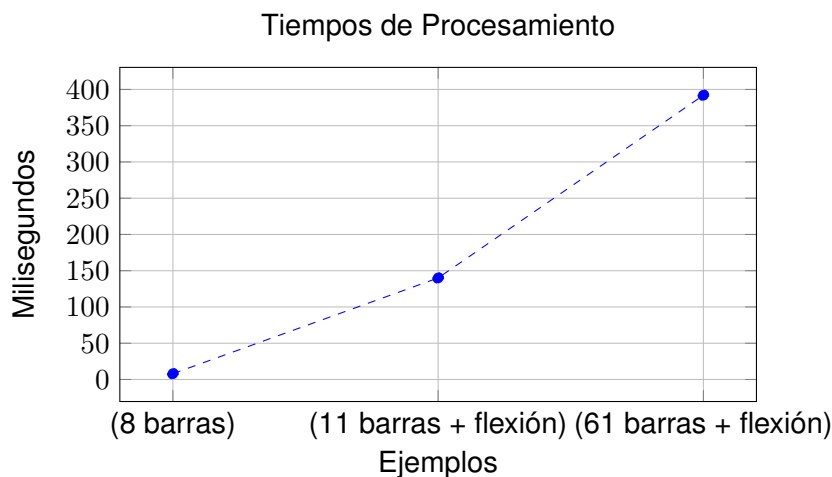


Figura 4.13: Tiempo de procesamiento en milisegundos para diferentes ejemplos (Fuente propia).

De la Figura 4.13 se nota claramente que a mayor número de nodos, mayor es el tiempo computacional usado por nuestra herramienta computacional. Esto se debe a que el tamaño de la matriz que resolvemos crece en orden proporcional a N^2 , siendo N el número de elementos. A pesar de añadir más elementos a nuestros ejemplos, observamos que los tiempos de procesamiento se mantienen notablemente eficientes. Esto confirma que nuestra herramienta computacional es altamente efectiva para el análisis de estructuras. Su escalabilidad también garantiza su aplicabilidad para los otros componentes de este proyecto, asegurando un rendimiento óptimo incluso con un mayor número de elementos.

5 CONCLUSIONES

- El método de la rigidez ha evolucionado significativamente desde sus inicios a finales del siglo XIX hasta convertirse en una herramienta esencial en el análisis estructural moderno. Gracias a los avances en la computación, ha sido posible superar las limitaciones iniciales asociadas con la resolución de grandes sistemas de ecuaciones lineales.
- En conclusión, la fusión del análisis del método de rigidez para armaduras sometidas a carga axial con el análisis del método de rigidez para vigas sometidas a flexión ha permitido desarrollar un enfoque estructural robusto. Este enfoque integrado facilita el cálculo de armaduras complejas en 2D, permitiendo obtener de manera precisa las cargas axiales, cortantes y momentos flectores de los elementos. La combinación de estos métodos amplía significativamente las capacidades de análisis estructural, ofreciendo una solución integral y eficiente para una amplia gama de aplicaciones en ingeniería estructural.
- El método de la rigidez es un método extenso que involucra matrices de gran tamaño, lo cual complica su realización manual. En estructuras con un número considerable de elementos, aplicar el método de la rigidez manualmente se vuelve inviable. Sin embargo, el método de la rigidez utilizando esta herramienta computacional, ha demostrado ser óptimo para el análisis. De esta manera, se puede obtener resultados precisos de manera eficiente de cargas externas e internas y momentos flectores de toda una estructura, sin importar el número de elementos.
- Una de las ventajas del código desarrollado es que, como su primera versión y al ser de código abierto, ofrece la opción de agregar más funciones y utilizar resultados de cargas y desplazamientos para otros análisis y evaluaciones.
- Finalmente se concluye que el código exhibe excelentes tiempos de procesamiento, lo que indica que es capaz de analizar estructuras con un gran número de elementos

fácilmente y de manera inmediata. Esto sugiere que el código es eficiente y escalable, permitiendo análisis de estructuras más complejas sin un impacto notable en el rendimiento.

6 RECOMENDACIONES

- Se recomienda el desarrollo de una interfaz gráfica de usuario para la herramienta computacional. Esto permitiría a los usuarios interactuar con la herramienta de manera más intuitiva y amigable, eliminando la necesidad de manipular el código directamente. Esto no solo facilitaría su uso por parte de ingenieros y técnicos que no estén familiarizados con la programación, sino que también podría mejorar la eficiencia y reducir el margen de error humano en el proceso de entrada de datos y configuración de análisis.
- Se recomienda mejorar el sistema de ingreso de datos para la herramienta computacional. Actualmente, el tiempo requerido para ingresar los datos puede ser considerable. Se estima que para el Ejemplo 1 se invirtieron aproximadamente 15 minutos, para el Ejemplo 2 se invirtieron 30 minutos y para el Ejemplo 3 se invirtieron 60 minutos.

7 REFERENCIAS BIBLIOGRÁFICAS

- [1] T. Celigueta, *Método de rigidez history*, 2016.
- [2] C. A. Felippa y C. A. Felippa, «A Historical Outline of Matrix Structural Analysis,» *University of Colorado*, págs. 1-18, 2000.
- [3] Torinoscienza, *Carlo Alberto Castigliano (1847 - 1884)*, 2016. dirección: <https://www.torinoscienza.it/personaggi/carlo-alberto-castigliano>.
- [4] H. R. C., *Structural-Analysis-Eighth-Edition-R.-C.-Hibbeler*. 2012, vol. 8, págs. 539-608.
- [5] A. : Adrià, G. Vivas, R. Javier y P. Rubio, *Simulación mdiante dinámica de fluidos computacional (CFD) de un intercambiador de flujo cruzado*, oct. de 2017.
- [6] F. Darío y D. Felipe, *Método del elemento finito Y Auto FEM*, ene. de 2018.
- [7] G. Iván y A. López, *Producción y simulación de nanotubos de carbono crecidos por la técnica de descomposición química de vapores optimizada por plasma PE CVD*, 2010.

8 ANEXOS

CARGA DE DATOS DEL EJEMPLO 1

```
1 #Este código soluciona un ejercicio de cálculo de estructura metálica
- #por el método de la rigidez.
- #Realizado EL 29 de marzo 2024 por Ivan Moreira
-
5 import numpy as np
- from funcion_def import metodo_rigidez
- import pandas as pd
- from libreria_funciones import analisis_solo_tension , seleccion_analisis
-
10 Fy = 30 #ksi
- Fu = 55 #ksi
- Num_eslabones = 8 #INGRESAR CUANTOS ELEMENTOS TIENE LA ARMADURA
- mov = 15 #INGRESAR CUANTOS MOVIMIENTOS TIENE LA ARMADURA
- area_elemento=1.1842 #[in^2] #INGRESAR EL AREA PARA TODOS LOS ELEMENTOS DE LA
  ARMADURA
15 rigidez_elemento=29000 #[ksi] #INGRESAR LA RIGIDEZ PARA TODOS LOS ELEMENTOS DE
  LA ARMADURA
- inercia=1.93 #in^4 #INGRESAR LA INERCIA PARA TODOS LOS ELEMENTOS DE
  LA ARMADURA
-
- A = np.full((Num_eslabones) , area_elemento)
- E = np.full((Num_eslabones) , rigidez_elemento)
20 I = np.full((Num_eslabones) , inercia)
-
- #A continuación ingresar las distancias en Pulgadas
- xn = np.array([0 , 118.11 , 118.11 , 0 , 0 , 0 , 0 , 118.11])
- yn = np.array([0 , 0 , 118.11 , 118.11 , 0 , 118.11 , 0 , 0])
25 xf = np.array([118.11 , 196.85 , 196.85 , 118.11 , 0 , 118.11 , 118.11 , 118.11])
- yf = np.array([0 , 118.11 , 118.11 , 118.11 , 118.11 , 0 , 118.11 , 118.11])
```

```

- #Ingresar los números de movimientos correspondientes.
- Near_x = np.array([13, 10, 4, 1, 13, 1, 13, 10])
- Near_y = np.array([14, 11, 5, 2, 14, 2, 14, 11])
30 Near_z = np.array([15, 12, 6, 3, 15, 3, 15, 12])
- Far_x = np.array([10, 7, 7, 4, 1, 10, 4, 4])
- Far_y = np.array([11, 8, 8, 5, 2, 11, 5, 5])
- Far_z = np.array([12, 9, 9, 6, 3, 12, 6, 6])
- #Ingresar numero de movimientos restringidos y no restringidos.
35 gdl_no_restringidos=np.array([1,2,3,4,5,6,7,8,9,10,12,15])-1
- gdl_restringidos= np.array([11,13,14])-1
- #Ingresar las cargas y desplazamientos de los nodos
- Q_k= np.array([0, 0, 0, 0, 0, 0, 11, -13.6, 0, 0, 0, 0]) #[klb o Klb*in]
- D_k= np.array([0, 0, 0]) #[in]
40 #Ingresar matriz de respuesta inicial en el caso de haber cargas distribuidas.
- CAR_EX=np.array([[0, 0, 0, 0, 0, 0],[0, 0, 0, 0, 0, 0],[0, 0, 0, 0, 0, 0],[0, 0,
    0, 0, 0, 0],[0, 0, 0, 0, 0, 0],[0, 0, 0, 0, 0, 0],[0, 0, 0, 0, 0, 0],[0, 0,
    0, 0, 0, 0]])
- #ingresar el valor de carga distribuida.
- tipo=np.array([0,0,0,0,0,0,0,0])
-
45 k,Respuestas_maximas,Cargas_vigas, Longitudes= metodo_rigidez (Num_eslabones,mov,
    A,E,I,xn,yn,xf,yf,Near_x,Near_y,Near_z, Far_x, Far_y, Far_z,Q_k,D_k,CAR_EX, tipo ,
    gdl_no_restringidos ,gdl_restringidos)

```

CARGA DE DATOS DEL EJEMPLO 2

```

1 #Este código soluciona un ejercicio de cálculo de estructura metálica
- #por el método de la rigidez.
- #Realizado EL 29 de marzo 2024 por Ivan Moreira
-
5 import numpy as np
- from funcion_def import metodo_rigidez
- import pandas as pd
- from libreria_funciones import analisis_solo_tension , seleccion_analisis
-
10 Fy = 30 #ksi
- Fu = 55 #ksi
- Num_eslabones = 11 #INGRESAR CUANTOS ELEMENTOS TIENE LA ARMADURA
- mov = 21 #INGRESAR CUANTOS MOVIMIENTOS TIENE LA ARMADURA
- area_eslabones=1.1842 #[in^2] #INGRESAR EL AREA PARA TODOS LOS ELEMENTOS DE
    LA ARMADURA

```

```

15 rigidez_eslabones=29000 #[ksi]      #INGRESAR LA RIGIDEZ PARA TODOS LOS ELEMENTOS
    DE LA ARMADURA
- inercia=1.93 #in^4                  #INGRESAR LA INERCIA PARA TODOS LOS ELEMENTOS
    DE LA ARMADURA
-
- A = np.full((Num_eslabones), area_eslabones)
- E = np.full((Num_eslabones), rigidez_eslabones)
20 I = np.full((Num_eslabones), inercia)
-
- #A continuación ingresar las distancias en Pulgadas
- xn = np.array([0, 118.11, 236.22, 0, 59.055, 118.11, 177.165, 236.22, 295.275,
    177.165, 59.055])
- yn = np.array([0, 0, 0, 0, 102.28, 0, 102.28, 0, 102.28, 102.28, 102.28])
25 xf = np.array([118.11, 236.22, 354.33, 59.055, 118.11, 177.165, 236.22, 295.275,
    354.33, 295.275, 177.165])
- yf = np.array([0, 0, 0, 102.28, 0, 102.28, 0, 102.28, 0, 102.28, 102.28])
- #Ingresar los números de movimientos correspondientes.
- Near_x = np.array([19, 16, 13, 19, 1, 16, 4, 13, 7, 4, 1])
- Near_y = np.array([20, 17, 14, 20, 2, 17, 5, 14, 8, 5, 2])
30 Near_z = np.array([21, 18, 15, 21, 3, 18, 6, 15, 9, 6, 3])
- Far_x = np.array([16, 13, 10, 1, 16, 4, 13, 7, 10, 7, 4])
- Far_y = np.array([17, 14, 11, 2, 17, 5, 14, 8, 11, 8, 5])
- Far_z = np.array([18, 15, 12, 3, 18, 6, 15, 9, 12, 9, 6])
- #Ingresar las cargas y desplazamientos de los nodos
35 Q_k= np.array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -96.48, 0, -9.8, 0, 0, -9.8, 0, 0,
    -98.48]) #[klb o Klb*in]
- D_k= np.array([0, 0, 0]) #[in]
- #Ingresar numero de movimientos restringidos y no restringidos.
- gdl_no_restringidos=np.array([1,2,3,4,5,6,7,8, 9, 12, 13, 14, 15, 16, 17, 18, 19,
    21])-1
- gdl_restringidos= np.array([10, 11, 20])-1
40 #Ingresar matriz de respuesta inicial en el caso de haber cargas distribuidas.
- CAR_EX=np.array ([[0, 4.9, 96.48, 0, 4.9, -96.48],[0, 4.9, 96.48, 0, 4.9,
    -96.48],[0, 4.9, 96.48, 0, 4.9, -96.48],[0, 0, 0, 0, 0, 0],[0, 0, 0, 0, 0,
    0],[0, 0, 0, 0, 0, 0],[0, 0, 0, 0, 0, 0],[0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0,
    0], [0, 0, 0, 0, 0, 0],[0, 0, 0, 0, 0, 0]])
- #ingresar el valor de carga distribuida.
- tipo=np.array([0.083,0.083,0.083,0,0,0,0,0,0,0,0,0,0])
-
45 k,Respuestas_maximas,Cargas_vigas, Longitudes= metodo_rigidez (Num_eslabones,mov,
    A,E,I,xn,yn,xf,yf,Near_x,Near_y,Near_z, Far_x, Far_y, Far_z,Q_k,D_k,CAR_EX, tipo,
    gdl_no_restringidos,gdl_restringidos)

```

CARGA DE DATOS DEL EJEMPLO 3

```
1 #Este código soluciona un ejercicio de cálculo de estructura metálica
- #por el método de la rigidez.
- #Realizado EL 29 de marzo 2024 por Ivan Moreira
-
5 import numpy as np
- from funcion_def import metodo_rigidez
- import pandas as pd
- from libreria_funciones import analisis_solo_tension , seleccion_analisis
-
10 Fy = 36 #ksi
- Fu = 55 #ksi
- Num_eslabones = 61 #INGRESAR CUANTOS ELEMENTOS TIENE LA ARMADURA
- mov = 96 #INGRESAR CUANTOS MOVIMIENTOS TIENE LA ARMADURA
- area_elemento=1.1842 #[in^2] #INGRESAR EL AREA PARA TODOS LOS ELEMENTOS DE LA
  ARMADURA
15 rigidez_elemento=29000 #[ksi] #INGRESAR LA RIGIDEZ PARA TODOS LOS ELEMENTOS DE
  LA ARMADURA
- inercia=1.93 #in^4 #INGRESAR LA INERCIA PARA TODOS LOS ELEMENTOS
  DE LA ARMADURA
-
- A = np.full((Num_eslabones), area_elemento)
- E = np.full((Num_eslabones), rigidez_elemento)
20 I = np.full((Num_eslabones), inercia)
-
- #A continuación ingresar las distancias en Pulgadas
- xn = np.array([0, 385.9844, 787.402, 1188.78, 1574.804, 1996.65, 2477.914,
  3000.159, 3543.309, 4086.222, 4608.66, 5089.766, 5511.814, 5897.64, 6299.216,
  6700.39, 7086.618, 6699.21, 6299.216, 5897.64, 5511.814, 5089.76, 4608.66,
  4086.22, 3543.309, 3000.15, 2477.9147, 1996.654, 1574.804, 1188.780, 787.402,
  385.9844, 385.82698 , 385.82698 , 787.402 , 1188.7801695,
  1188.7801695, 1574.804 , 1574.804 , 1574.804 , 1996.6546215,
  1996.6546215, 2477.9147239, 2477.9147239, 3000.1591004, 3000.1591004,
  3543.309, 4086.222679 , 4086.222679 , 4608.663906 , 4608.663906 , 5089.766528
  , 5089.766528 , 5511.814 , 5511.814 , 5511.814 , 5897.64098 ,
  5897.64098 , 6299.216 , 6700.003618 , 6700.003618])
- yn = np.array([0, 708.66, 708.66, 708.66, 708.66, 708.66, 708.66, 708.66, 708.66,
  708.66, 708.66, 708.66, 708.66, 708.66, 708.66, 0, 204.488, 273.54,
```

```

204.488, 0, 194.448, 338.976, 427.952, 457.9923, 427.9529, 338.976, 194.4489,
0, 204.4882, 273.543, 204.488, 204.4882994, 204.4882994, 273.5434548,
204.4882994, 204.4882994, 0.          , 0.          , 0.          , 194.4489239,
194.4489239, 338.976561 , 338.976561 , 427.952987 , 427.952987 , 457.9923733,
427.952987 , 427.952987 , 338.976561 , 338.976561 , 194.4489239,
194.4489239, 0.          , 0.          , 0.          , 204.4882994, 204.4882994,
273.5434548, 204.4882994, 204.4882994])
25 xf = np.array([385.984, 787.402, 1188.78, 1574.804, 1996.65, 2477.914, 3000.159,
3543.309, 4086.222, 4608.66, 5089.766, 5511.814, 5897.64, 6299.216, 6700.39,
7086.618, 6699.216, 6299.216, 5897.64, 5511.814, 5089.766, 4608.6639,
4086.222, 3543.309, 3000.159, 2477.9147, 1996.654, 1574.804, 1188.780,
787.402, 385.9844, 0, 385.82698 , 787.402 , 787.402 , 787.402
, 1188.7801695, 1188.7801695, 1574.804 , 1996.6546215, 1996.6546215,
2477.9147239, 2477.9147239, 3000.1591004, 3000.1591004, 3543.309 ,
3543.309 , 3543.309 , 4086.222679 , 4086.222679 , 4608.663906 ,
4608.663906 , 5089.766528 , 5089.766528 , 5511.814 , 5897.64098 ,
5897.64098 , 6299.216 , 6299.216 , 6299.216 , 6299.216 , 6700.397319])
- yf = np.array([708.66, 708.66, 708.66, 708.66, 708.66, 708.66, 708.66, 708.66,
708.66, 708.66, 708.66, 708.66, 708.66, 708.66, 0, 204.488, 273.543,
204.488, 0, 194.448, 338.976, 427.9529, 457.992, 427.9529, 338.976, 194.4489,
0, 204.4882, 273.543, 204.488, 0, 708.6618, 708.6618, 708.6618, 708.6618,
708.6618, 708.6618, 708.6618, 708.6618, 708.6618, 708.6618, 708.6618,
708.6618, 708.6618, 708.6618, 708.6618, 708.6618, 708.6618, 708.6618,
708.6618, 708.6618, 708.6618, 708.6618, 708.6618, 708.6618, 708.6618,
708.6618, 708.6618, 708.6618, 708.6618])
- #Ingresar los números de movimientos correspondientes.
- Near_x = np.array([94, 1, 4, 7, 10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, 43,
46, 49, 52, 55, 58, 61, 64, 67, 70, 73, 76, 79, 82, 85, 88, 91, 91, 91, 88,
85, 85, 82, 82, 82, 79, 79, 76, 76, 73, 73, 70, 67, 67, 64, 64, 61, 61, 58,
58, 58, 55, 55, 52, 49, 49 ])
- Near_y = np.array([95, 2, 5, 8, 11, 14, 17, 20, 23, 26, 29, 32, 35, 38, 41, 44,
47, 50, 53, 56, 59, 62, 65, 68, 71, 74, 77, 80, 83, 86, 89, 91, 92, 92, 89,
86, 86, 83, 83, 83, 80, 80, 77, 77, 74, 74, 71, 68, 68, 65, 65, 62, 62, 59,
59, 59, 56, 56, 53, 50, 50 ])
30 Near_z = np.array([96, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45,
48, 51, 54, 57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 93, 93, 90,
87, 87, 84, 84, 84, 81, 81, 78, 78, 75, 75, 72, 69, 69, 66, 66, 63, 63, 60,
60, 60, 57, 57, 54, 51, 51 ])
- Far_x = np.array([1, 4, 7, 10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, 43, 46,
49, 52, 55, 58, 61, 64, 67, 70, 73, 76, 79, 82, 85, 88, 91, 94, 1, 4, 4, 4,
7, 7, 10, 13, 13, 16, 16, 19, 19, 22, 22, 22, 25, 25, 28, 28, 31, 31, 34, 37,
37, 40, 40, 40, 43 ])

```



```

45 k,Respuestas_maximas,Cargas_vigas, Longitudes= metodo_rigidez (Num_eslabones,mov,
    A,E,I,xn,yn,xf,yf,Near_x,Near_y,Near_z, Far_x, Far_y, Far_z,Q_k,D_k,CAR_EX,tipo,
    gdl_no_restringidos,gdl_restringidos)

```

FUNCIÓN MÉTODO DE RIGIDEZ

```

1 #Este código analiza un ejercicio de cálculo de estructura metálica
- #por el método de la rigidez.
- #Realizado EL 9 de junio 2024 por Ivan Moreira
-
5 import numpy as np
- import math as mt
- import sympy as sp
-
- '''
10 La siguiente función "metodo_rigidez" toma los valores ingresados del archivo "
    flexión_def" para realizar un analisis
- de la estrucutra, en la cual analizará cargas cortantes, axiales, y momentos.
- El objetivo es tener estas cargas internas, llevarlas a esfuerzos y verificarlos
    por norma AISC y NEC.
- Esta parte será evaluada con funciones creadas a continuación por Nicolas
    Montalvo.
- '''
15
-
- def metodo_rigidez (Num_eslabones,mov,A,E,I,xn,yn,xf,yf,Near_x,Near_y,Near_z,
    Far_x, Far_y, Far_z,Q_k,D_k,CAR_EX,tipo,gdl_no_restringidos,gdl_restringidos):
-     lamx=np.zeros(Num_eslabones)
20     lamy=np.zeros(Num_eslabones)
-     L=np.zeros(Num_eslabones)
-     k= np.zeros((mov,mov))
-
-
25     for i in range (Num_eslabones):
-
-         L[i]=((xf[i]-xn[i])**2+(yf[i]-yn[i])**2)**(1/2)
-
-         First=(A[i]*E[i])/L[i]
30         Second=(12*E[i]*I[i])/(L[i])**3
-         Third= (6*E[i]*I[i])/(L[i])**2

```

```

- Fourth= (4*E[i]*I[i])/(L[i])
- Fifth = (2*E[i]*I[i])/(L[i])
-
35 lamx[i]=(xf[i]-xn[i])/(L[i])
- lamy[i]=(yf[i]-yn[i])/(L[i])
- #A continuación se muestra todos los coeficientes de mi matriz de rigidez
-
- k[Near_x[i]-1,Near_x[i]-1]= ((First*(lamx[i]**2)+(Second*(lamy[i]**2))))
+ k[Near_x[i]-1,Near_x[i]-1]
- k[Near_x[i]-1,Near_y[i]-1]= (First-Second)*lamx[i]*lamy[i] + k[Near_x[i]
-1,Near_y[i]-1]
40 k[Near_x[i]-1,Near_z[i]-1]= ((-1)*(Third)*lamy[i]) + k[Near_x[i]-1,Near_z
[i]-1]
- k[Near_x[i]-1,Far_x[i]-1]= ((First*(lamx[i]**2)+(Second*(lamy[i]**2))))
*(-1) + k[Near_x[i]-1,Far_x[i]-1]
- k[Near_x[i]-1,Far_y[i]-1]= ((First-Second)*lamx[i]*lamy[i])*(-1) + k[
Near_x[i]-1,Far_y[i]-1]
- k[Near_x[i]-1,Far_z[i]-1]= ((-1)*(Third)*lamy[i]) + (k[Near_x[i]-1,Far_z[
i]-1])
- k[Near_y[i]-1,Near_x[i]-1]= (First-Second)*lamx[i]*lamy[i] + k[Near_y[i]
-1,Near_x[i]-1]
45 k[Near_y[i]-1,Near_y[i]-1]= ((First*(lamy[i]**2)+(Second*(lamx[i]**2))))
+ k[Near_y[i]-1,Near_y[i]-1]
- k[Near_y[i]-1,Near_z[i]-1]= (Third)*lamx[i] + k[Near_y[i]-1,Near_z[i]-1]
- k[Near_y[i]-1,Far_x[i]-1]= (-1)*((First-Second)*lamx[i]*lamy[i]) + k[
Near_y[i]-1,Far_x[i]-1]
- k[Near_y[i]-1,Far_y[i]-1]= (-1)*(((First*(lamy[i]**2)+(Second*(lamx[i]
**2)))) + k[Near_y[i]-1,Far_y[i]-1]
- k[Near_y[i]-1,Far_z[i]-1]= (Third)*lamx[i] + k[Near_y[i]-1,Far_z[i]-1]
50 k[Near_z[i]-1,Near_x[i]-1]= (-1)*(Third)*lamy[i] + k[Near_z[i]-1,Near_x[i]
-1]
- k[Near_z[i]-1,Near_y[i]-1]= (Third)*lamx[i] + k[Near_z[i]-1,Near_y[i]-1]
- k[Near_z[i]-1,Near_z[i]-1]= Fourth + k[Near_z[i]-1,Near_z[i]-1]
- k[Near_z[i]-1,Far_x[i]-1]= (Third)*lamy[i] + k[Near_z[i]-1,Far_x[i]-1]
- k[Near_z[i]-1,Far_y[i]-1]= (-1)*(Third)*lamx[i] + k[Near_z[i]-1,Far_y[i]
-1]
55 k[Near_z[i]-1,Far_z[i]-1]= Fifth + k[Near_z[i]-1,Far_z[i]-1]
- k[Far_x[i]-1,Near_x[i]-1]= ((First*(lamx[i]**2)+(Second*(lamy[i]**2))))
*(-1) + k[Far_x[i]-1,Near_x[i]-1]
- k[Far_x[i]-1,Near_y[i]-1]= (First-Second)*lamx[i]*lamy[i]*(-1) + k[Far_x[
i]-1,Near_y[i]-1]
- k[Far_x[i]-1,Near_z[i]-1]= ((Third)*lamy[i]) + k[Far_x[i]-1,Near_z[i]-1]

```



```

-
-
90  for i in range (len(Q_k)):
-      cargas[gdl_no_restringidos[i]]=Q_k[i]
-
-      for i in range (len(D_k)):
-          cargas[gdl_restringidos[i]]=Q_u[i]
-
-
95  for i in range (len(D_k)):
-      desplazamientos[gdl_restringidos[i]]=D_k[i]
-
-      for i in range (len(Q_k)):
-          desplazamientos[gdl_no_restringidos[i]]=D_u[i]
-
-
100 #print (cargas)
-     #print (desplazamientos)
-     #
-         ////////////////////////////////////////
-
-
105 Cargas_vigas=[]
-     for i in range (Num_eslabones):
-         L[i]=((xf[i]-xn[i])**2+(yf[i]-yn[i])**2)**(1/2)
-
-         First=(A[i]*E[i])/L[i]
110        Second=(12*E[i]*I[i])/(L[i])**3
-         Third= (6*E[i]*I[i])/(L[i])**2
-         Fourth= (4*E[i]*I[i])/(L[i])
-         Fifth = (2*E[i]*I[i])/(L[i])
-
-
115        lamx[i]=(xf[i]-xn[i])/(L[i])
-         lamy[i]=(yf[i]-yn[i])/(L[i])
-
-         k_esl= np.zeros((6,6))          #Estas son las tres matrices que se van a
-             multiplicar para obtener los esfuerzos internos
-         matriz_lam= np.zeros((6,6))
120        desplazamientos_new=np.zeros(6)
-
-         k_esl[0,0]= First
-         k_esl[1,0]= 0
-         k_esl[2,0]= 0
125        k_esl[3,0]= -First
-         k_esl[4,0]= 0
-         k_esl[5,0]= 0

```

```

-
-      k_esl[0,1]= 0
130      k_esl[1,1]= Second
-      k_esl[2,1]= Third
-      k_esl[3,1]= 0
-      k_esl[4,1]= -Second
-      k_esl[5,1]= Third
135
-      k_esl[0,2]= 0
-      k_esl[1,2]= Third
-      k_esl[2,2]= Fourth
-      k_esl[3,2]= 0
140      k_esl[4,2]= -Third
-      k_esl[5,2]= Fifth
-
-      k_esl[0,3]= -First
-      k_esl[1,3]= 0
145      k_esl[2,3]= 0
-      k_esl[3,3]= First
-      k_esl[4,3]= 0
-      k_esl[5,3]= 0
-
150      k_esl[0,4]= 0
-      k_esl[1,4]= -Second
-      k_esl[2,4]= -Third
-      k_esl[3,4]= 0
-      k_esl[4,4]= Second
155      k_esl[5,4]= -Third
-
-      k_esl[0,5]= 0
-      k_esl[1,5]= Third
-      k_esl[2,5]= Fifth
160      k_esl[3,5]= 0
-      k_esl[4,5]= -Third
-      k_esl[5,5]= Fourth
-
-
165      matriz_lam[0,0]= lamx[ i ]
-      matriz_lam[1,0]= -lamy[ i ]
-      matriz_lam[0,1]= lamy[ i ]
-      matriz_lam[1,1]= lamx[ i ]
-      matriz_lam[2,2]= 1

```

```

170     matriz_lam[3,3]=lamx[i]
-     matriz_lam[4,3]=-lamy[i]
-     matriz_lam[3,4]=lamy[i]
-     matriz_lam[4,4]=lamx[i]
-     matriz_lam[5,5]=1
175
-     desplazamientos_new[0]=desplazamientos[Near_x[i]-1]
-     desplazamientos_new[1]=desplazamientos[Near_y[i]-1]
-     desplazamientos_new[2]=desplazamientos[Near_z[i]-1]
-     desplazamientos_new[3]=desplazamientos[Far_x[i]-1]
180     desplazamientos_new[4]=desplazamientos[Far_y[i]-1]
-     desplazamientos_new[5]=desplazamientos[Far_z[i]-1]
-
-     q1= np.dot(k_esl, matriz_lam)
-     q=np.dot(q1, desplazamientos_new)
185     Cargas_vigas.append(q)
-
-     Cargas_vigas=Cargas_vigas+CAR_EX
-
-
190     Respuestas_maximas=[]
-
-     for i in range (Num_eslabones):
-         tipos=tipo[i]
-         maximos=np.zeros((5, 2))
195     if tipos==0:
-         maximos[0,0]=max(abs(Cargas_vigas[i][0]),abs(Cargas_vigas[i][3]))
-         if (Cargas_vigas[i][0])>0:
-             maximos[0,1]=0
-         else:
200             maximos[0,1]=1
-         maximos[1,0]=max(abs(Cargas_vigas[i][1]),abs(Cargas_vigas[i][4]))
-         maximos[2,0]=abs(Cargas_vigas[i][2])
-         maximos[2,1]=0
-         maximos[3,0]=0
205     maximos[3,1]=0
-         maximos[4,0]=Cargas_vigas[i][5]
-         maximos[4,1]=L[i]
-     else:
-         maximos[0,0]=max(abs(Cargas_vigas[i][0]),abs(Cargas_vigas[i][3]))
210     if (Cargas_vigas[i][0])>0:
-         maximos[0,1]=0

```

```

-         else:
-             maximos[0,1]=1
-             maximos[1,0]=max((Cargas_vigas[i][1]),(Cargas_vigas[i][4]))
215         if maximos[1,0]==Cargas_vigas[i][1]:
-             maximos[1,1]=0
-         else:
-             maximos[1,1]=L[i]
-             maximos[2,0]=abs(Cargas_vigas[i][2])
220         maximos[2,1]=0
-
-         x=sp.symbols("x")
-         ecuacion=((-tipo[i])/2)*x**2 + (Cargas_vigas[i][1])*x - (Cargas_vigas
-             [i][2])
-         derivada = sp.diff(ecuacion, x)
225         soluciones_x = sp.solve(derivada, x)
-         soluciones_y = [ecuacion.subs(x, sol) for sol in soluciones_x]
-         maximos[3,0]=soluciones_y[0]
-         maximos[3,1]=soluciones_x[0]
-         maximos[4,0]=Cargas_vigas[i][5]
230         maximos[4,1]=L[i]
-
-         #maximos= np.round(maximos, decimals=2)
-         Respuestas_maximas.append(maximos)
-         #Respuestas_maximas = np.round(Respuestas_maximas, decimals=1)
-         return (k, Respuestas_maximas, Cargas_vigas, L)

```

[4])

MANUAL DE CARGA DE DATOS CON EJEMPLO DE 2 BARRAS (EJEMPLO TOMADO DEL CAPÍTULO 16 DEL LIBRO [4])

Determine las cargas en los extremos de cada elemento de la estructura que se muestra en la figura 16-5a. Considere que $I = 600 \text{ pulg}^4$, $A = 12 \text{ pulg}^2$ y $E = 29(10^3) \text{ ksi}$ para cada elemento.

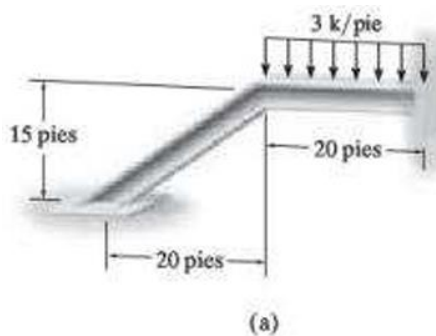
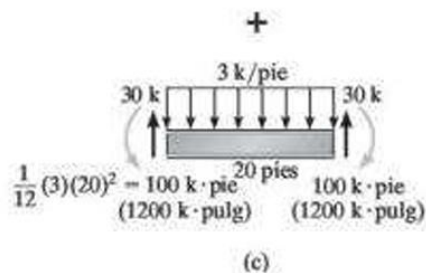
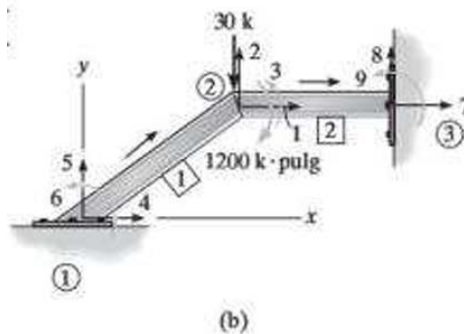


Figura 16-5



PASO 1

Se realiza el análisis de numeración de elementos, numeración de nodos y grados de libertad. A la vez se analiza la carga distribuida para poder determinar las cargas puntuales en los nodos como se muestra a continuación:

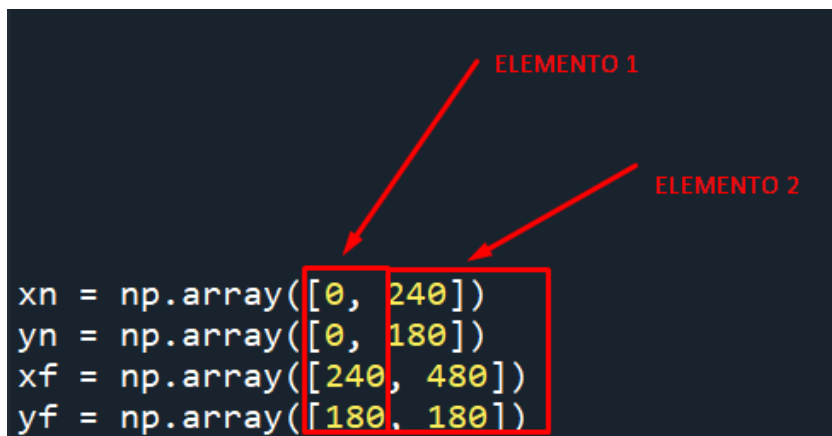
PASO 2.- Datos de armadura y elementos

Del enunciado y de la Figura 16-5 se procede a llenar la información de la armadura y los elementos como se muestra a continuación (Se supone que $F_y = 30$ ksi y $F_u = 55$ ksi):

```
Fy = 30 #ksi
Fu = 55 #ksi
Num_eslabones = 2 #INGRESAR CUANTOS ELEMENTOS TIENE LA ARMADURA
mov = 9 #INGRESAR CUANTOS MOVIMIENTOS TIENE LA ARMADURA
area_elemento=12 #[in^2] #INGRESAR EL AREA PARA TODOS LOS ELEMENTOS DE LA ARMADURA
rigidez_elemento=29000 #[ksi] #INGRESAR LA RIGIDEZ PARA TODOS LOS ELEMENTOS DE LA ARMADURA
inerzia=600 #in^4 #INGRESAR LA INERCIA PARA TODOS LOS ELEMENTOS DE LA ARMADURA
```

PASO 3.- Coordenadas

Luego se procede a llenar las coordenadas de los elementos de la siguiente manera. Las medidas se las observa en la Figura 16-5 (a) y se realiza la conversión de pies a pulgadas.



```
xn = np.array([0, 240])
yn = np.array([0, 180])
xf = np.array([240, 480])
yf = np.array([180, 180])
```

PASO 4.- Grados de libertad (GDL)

Luego se procede a ingresar los grados de libertad que se determinó en la Figura 16-5 (b) de la siguiente manera:

```

Near_x = np.array([4, 1])
Near_y = np.array([5, 2])
Near_z = np.array([6, 3])
Far_x = np.array([1, 7])
Far_y = np.array([2, 8])
Far_z = np.array([3, 9])

```

ELEMENTO 1
ELEMENTO 2

PASO 5.- GDL restringidos y libres

Luego se realiza el análisis de grados de libertad restringidos y libres:

- El nodo 1 se encuentra empotrado, por lo cual los grados de libertad 4, 5, 6 son restringidos.
- El nodo 2 no tiene soporte de la armadura por lo cual los grados de libertad 1, 2 y 3 son libres.
- El nodo 3 se encuentra empotrado, por lo cual los grados de libertad 7, 8 y 9 son restringidos.

Se procede a llenar la información descrita de la siguiente manera:

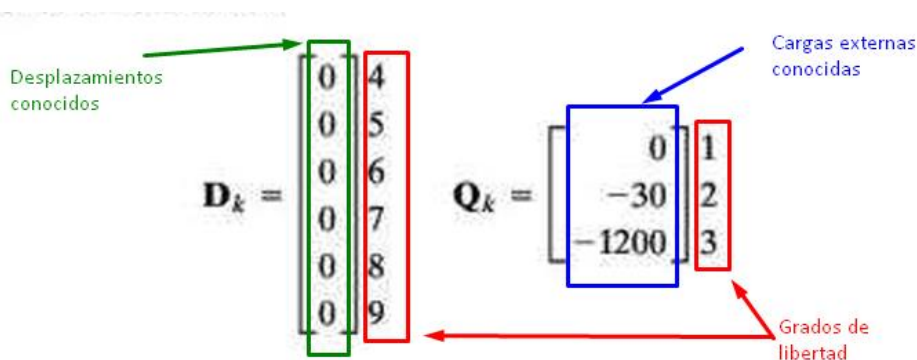
```

gdl_no_restringidos=np.array([1,2,3])-1
gdl_restringidos= np.array([4, 5, 6, 7, 8, 9])-1

```

PASO 6.- Cargas y desplazamientos

Luego se procede a llenar cargas y desplazamientos. En la Figura 16-5 (b) se observa las cargas externas aplicadas en los nodos, y se descomponen de la siguiente manera:



Y se registran en el código de la siguiente manera:

```
Q_k= np.array([0, -30, -1200]) #[klb o Klb*in]
D_k= np.array([0, 0, 0, 0, 0, 0]) #[in]
```

PASO 7.- Cargas de elemento fijamente apoyado

Luego se registra las cargas y reacciones reales sobre un elemento fijamente apoyado. Estas cargas se observan en la figura 16-5 (c) y se ingresan de la siguiente manera:

```
Vector para Elemento 1
Vector para Elemento 2
CAR_EX=np.array([[0, 0, 0, 0, 0, 0], [0, 30, 1200, 0, 30, -1200]])
```

PASO 8.- Carga distribuida

Y finalmente se registra el valor de la carga distribuida de la Figura 16-5 con valor de 3k/pie

```
Carga distribuida del Elemento 1
Carga distribuida del Elemento 2
tipo=np.array([0,3])
```

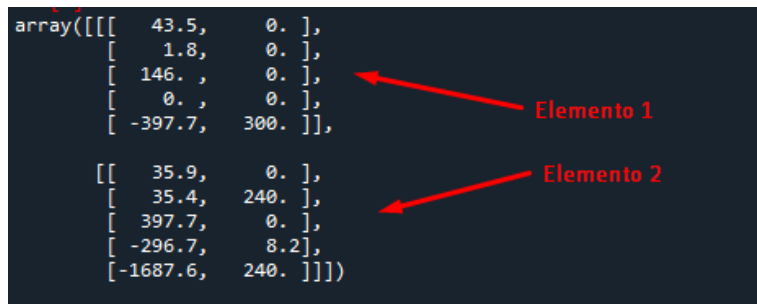
PASO 9.- Ejecutar el código

Al correr el código se obtiene las cargas internas de los elementos de la siguiente manera:

```
Out[2]:
array([[ 43.5,  -1.8, -146. , -43.5,   1.8, -397.7],
       [ 35.9,  24.6,  397.7, -35.9,  35.4, -1687.6]])
Elemento 1
Elemento 2
```

Y a la vez se puede obtener las respuestas máximas. La interpretación de esta se describe en la metodología del escrito del proyecto.

```
array([[ 43.5,  0. ],
       [  1.8,  0. ],
       [ 146. ,  0. ],
       [  0. ,  0. ],
       [-397.7, 300. ]],
       [[ 35.9,  0. ],
       [ 35.4, 240. ],
       [ 397.7,  0. ],
       [-296.7,  8.2],
       [-1687.6, 240. ]])
```



Elemento 1

Elemento 2

