Programa computacional para el análisis a la deflexión de vigas hiperestáticas

Valente Enrique Rodríguez Alzamora

vrodrigueza@unasam.edu.pe

José Delfín Pretel Rondan

jpretelr@unasam.edu.pe

Universidad Nacional Santiago Antúnez de Mayolo (Perú)

Resumen: Se implementó el método numérico de diferencias finitas con el fin de hallar la deflexión de una viga hiperestática y se comprobó que este resultado se asemeja a los resultados del cálculo exacto de doble integración. El estudio se realizó en base a la metodología descriptiva, la cual consistió primero escoger una viga hiperestática, para posteriormente calcular la deflexión de la misma mediante el método de doble integración y mediante método numérico de diferencias finitas que resuelve la ecuación diferencial de orden superior. El método de diferencias finitas implementado en *Python* para calcular la deflexión de una viga, es una forma muy eficiente de obtener los resultados muy precisos en el menor tiempo posible.

Palabras clave: Código computacional; viga Hiperestática; *Python*; deflexión; métodos numéricos.

Trabajo tutorado por el Dr. Maximiliano Epifanio Asís López
Recibido: 2 enero 2022/ Aceptado: 6 mayo 2022.

Computer program for the deflection analysis of hyperstatic beams

Abstract: The numerical method of finite differences was implemented in order to find the deflection of a hyperstatic beam and it was found that this result resembles the results of the exact calculation of double integration. The study was carried out based on the descriptive methodology, which consisted first of choosing a hyperstatic beam, to later calculate its deflection by means of the double integration method and by means of the finite difference numerical method that solves the higher order differential equation. It was verified that the finite difference method implemented in Python to calculate the deflection of a beam is a very efficient way to obtain very accurate results in the shortest possible time.

Keywords: Computational code; Hyperstatic Beam; Python; Deflection; Numerical methods.

Introducción

El uso de programas computacionales por parte de docentes, profesionales y estudiantes, ha sido una fuente de desarrollo cognitivo con el propósito de mejorar las prácticas en las diferentes asignaturas al igual que incrementar la competitividad y productividad de los mismos, consolidando conocimientos, habilidades gráficas computacionales e interés a las herramientas digitales.

Contreras, Muñoz, Contreras y López (2018) el uso de las nuevas Tecnologías de la Información y de la Comunicación (TIC), así como metodologías centradas en el trabajo autónomo de los estudiantes, son fundamentales en la mejora de la enseñanza y el aprendizaje; mejorando el rendimiento académico que reciben estos métodos de enseñanza innovadores.

Ayala (2021) aproximó la deflexión de una viga teniendo la ecuación diferencial, estableciendo la relación de causa efecto, mediante la prueba de la hipótesis: ley física -ecuación diferencial ordinaria– solución numérica. Obteniendo como resultado la aproximación de la deflexión de la viga con el método de disparo lineal y de diferencias finitas, ofreciendo una menor vulnerabilidad de error de redondeo para la aproximación de la deflexión de la viga por ambos métodos.

Collantes (2006) consideró el problema de valor de frontera unidimensional, aproximando la solución de la deflexión de la viga por el método de diferencias finitas, donde el método a su vez involucró la solución de sistema de ecuaciones lineales, obteniendo que el método de diferencias finitas ha permitido aproximar la solución de la ecuación en los nodos de una malla, con un error muy pequeño que tiende a cero.

Este trabajo propone analizar el cálculo de la deflexión de la viga hiperestática calculada esta misma mediante un código realizado en *Python* que logre resolver la ecuación diferencial mediante el método de diferencias finitas y comprobar el resultado con el método manual tradicional obtenido de la doble integración.

Materiales y métodos

El presente estudio es de tipo descriptivo esto debido a que se limita al análisis de una viga hiperestáticas y sus características tales como sus dimensiones, carga distribuida

o puntual, su respectivo módulo de elasticidad, grado de hiperestaticidad, principio de continuidad o de frontera y su ecuación de deflexión para cada tramo de la viga usando el método de doble integración. Y por otro lado usar el método numérico de diferencias finitas para el análisis de la deflexión a lo largo de la viga con la ayuda de un lenguaje de programación usando las ecuaciones de la deflexión de la viga encontradas usando el método de doble integración.

El procedimiento consta básicamente de dos partes diferenciadas: el modelamiento matemático y aplicación de métodos iterativos. Ambos procesos quedan resumidos en la figura 1 y 2.

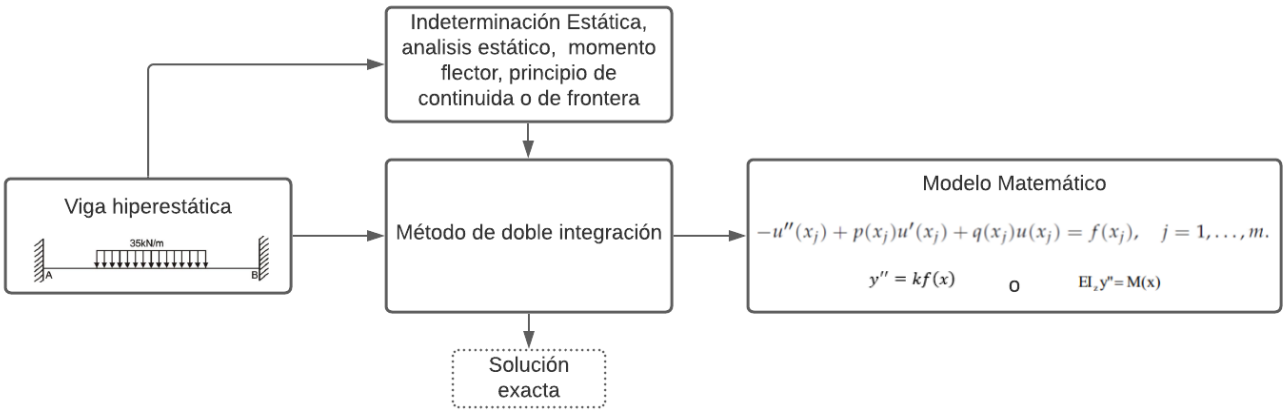

Figura 1. Pasos del modelamiento matemático.

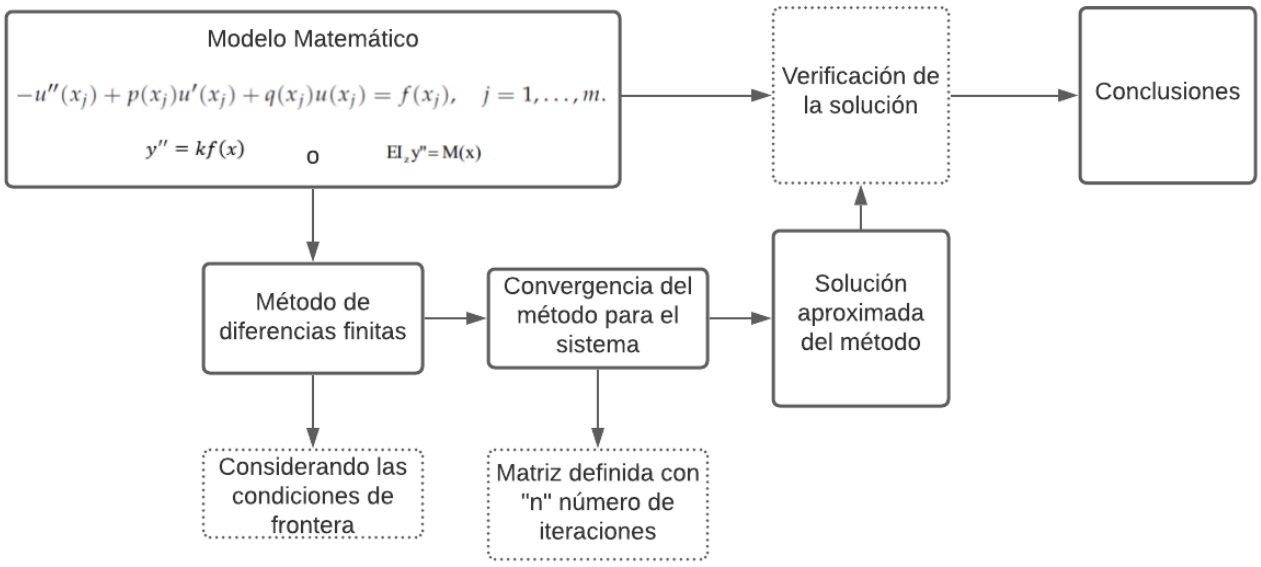

Figura 2. Procedimiento de solución numérica (*Python*).

El modelamiento matemático consiste principalmente en obtener la ecuación de la deflexión para cada tramo de la viga aplicando el método de integral doble; sin embargo, antes de aplicarlo debemos verificar las ecuaciones del momento flector para aplicarla en nuestra ecuación diferencial aproximada de la línea elástica de la viga en cada tramo de este mismo; verificando que estas ecuaciones sean diferentes debido a las cargas distribuidas o puntuales, salvo que esté presente una distribución simétrica. Teniendo en cuenta los principios de frontera y aplicando el método de integral doble; se obtendrá las ecuaciones de la deflexión de la viga para cada tramo. El método de integral doble, es uno de los métodos de cálculo para el análisis de la deflexión de vigas usado en la ingeniería, permitiendo conocer la deflexión presente en cada uno de los tramos de una estructura que se comprobará con el análisis matemático de diferencias finitas.

La aplicación de los métodos iterativos tuvo como fin determinar el valor de la deflexión de la viga para analizarla en cada nodo y encontrar una gráfica aproximada de la deformación que este mismo presenta a partir del modelo matemático obtenido; el método iterativo aplicado fue de diferencias finitas. Usando el programa *Phyton*, de tal manera que estas posean entradas de programación para el cálculo de la deflexión de la viga, siendo estos la ecuación diferencial, las condiciones de frontera, el número de iteraciones o particiones, una tolerancia de error, una matriz de coeficientes del sistema y una gráfica de deflexión. Esto visualizado en el programa en la figura 3.

```python
import numpy as np
import matplotlib.pyplot as plt
def exacta (x):
    """ Solucion exacta"""
    return (35*x**3/3-128.33*x**2/2-35*(x-2)**4/24)/(19*10**6*0.0016)

def funs(x):
    """ Funciones p(x), q(x) y f(x) """
    f = -(70*x-128.33-35*(x-2)**2/2)/(19*10**6*0.0016)
    p = 0*x
    q = 0*x
    return (p, q, f)

a = 2 # extremo inferior de la viga
b = 6 # extremo superior de la viga
alpha = -0.0054 # condicion de deflexión evaluado en a
beta = -0.0054 #condicion de la deflexión evaluado en b
N = 10 #numero de particiones
h = (b-a)/N #paso de malla
x = np.arange (a, b+h, h) #discretizacion del intervalo (a,b)
(p, q, f) = funs (x) #coeficientes de la ecuacion

A = np.zeros ([N -1, N -1]) #Matriz del sistema (A  9*9)
for i in range (N -2):
    A[i, i] = 2.+ q[i +1]* h **2 # diagonal
    A[i, i +1] = -1.+0.5* p[i +1]* h #superdiagonal
    A[i+1, i] = -1. -0.5*p[i +2]* h #subdiagonal
A[N -2, N -2] = 2.+ q[N -1]* h **2 # ultimo elemento de la diagonal
A = A/h **2

# modificacion del termino independiente (b)
f[1] += alpha *(1./h **2+0.5* p [1]/ h)
f[N -1] += beta *(1./ h **2 -0.5* p[N -1]/ h)

print("{} = {}".format("A",A))
print("{} = {}".format("b",f[1:N]))

v = np.linalg.inv(A).dot(f [1: N])
#condiciones de contorno
v = np.append (alpha , v)
v = np.append (v, beta)
for i in range (N-1):
    print("i: {:03d}\t x: {:.4f}\t v: {:.7f}\n".format(i+1, x[i+1], v[i+1]), end="")

ve = exacta (x) #solucion exacta
error = max ( abs (ve -v)) #error cometido

print ('paso de malla : ' + str (h))
xx = np.arange (a, b, 1.e-3)
ve = exacta (xx)
plt.figure(figsize = (10, 6))
plt.plot(x, v, 'b-o') #dibuja la solucion aproximada
plt.plot (xx, ve, 'r') #dibuja la solucion exacta
plt.title(' solucion aproximada')
plt.xlabel('x')
plt.ylabel('f(x)')
plt.legend (['aproximada','exacta'],loc ='lower left')
plt.grid()
```

Figura 3. Algoritmo del método de diferencias finitas en Phyton.

En el modelo matemático obtenido anteriormente proporciona la ecuación diferencial $y'' = kf(x)$ de tal manera que al aplicar el método de diferencias finitas en el Phyton, se debe definir nuestras funciones f(x), p(x) y q(x), los extremos de la viga definida por las condiciones de frontera (a, b) y la deflexión en dichos extremos (alpha, beta), los valores del sistema de ecuaciones para los nodos o cotas presentadas a las condiciones de $h = (b - a)/N$; así mismo se definió las matrices de nuestro sistema de ecuaciones $Av = b$ obtenidas del método te diferencias finitas, donde la matriz de coeficiente v

vienen a ser los nodos donde se presenta la deflexión. El Phyton arrojó los resultados para el método ejecutado como se verá en la siguiente sección.

Resultados

Se encontró que las dos vigas analizadas poseían un grado de hiperestaticidad diferente a cero, por lo tanto, se trataba de dos vigas hiperestáticas, y al aplicar el método de doble integral en los diferentes tramos especificados en la Figura 4.

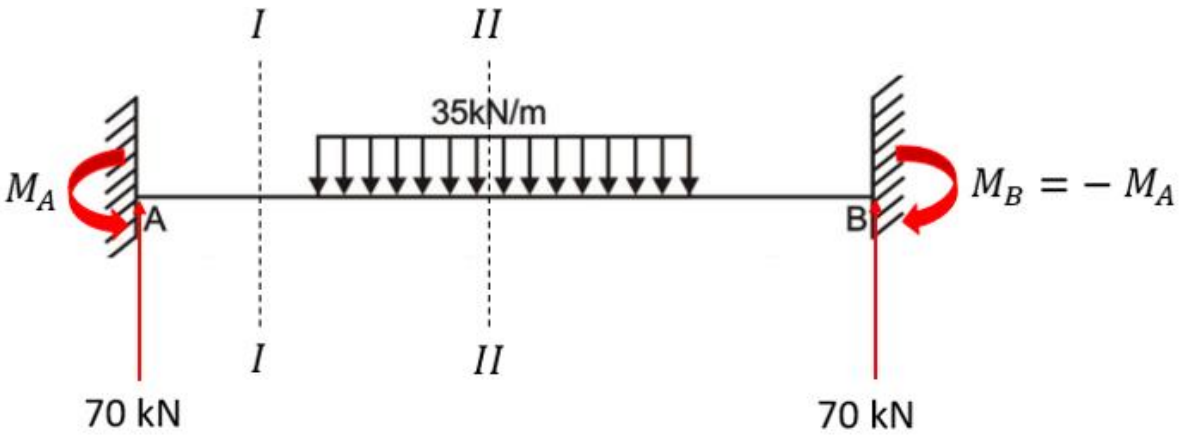

Figura 4. Viga analizada en 2 tramos.

En este proceso se consideró las siguientes dimensiones de b = 300mm, h = 400mm y el módulo de elasticidad de los elementos E = 19000 kN/mm^2.

Seguidamente se encontraron las ecuaciones para calcular la deflexión de la viga en cualquier punto que sea requerido.

Teniendo nuestras ecuaciones de pendiente y deflexión calculado por el método de integral doble:

- $Tramo\ I\ (0 \leq x \leq 2)$

$$EI\theta = 35x^2 - M_A x$$

$$EIy = \frac{35x^3}{3} - \frac{M_A x^2}{2}$$

- $Tramo\ II\ (2 \leq x \leq 6)$

$$EI\theta = 5x^2 - M_A x - \frac{35(x-2)^3}{6}$$

$$EIy = \frac{35x^3}{3} - \frac{M_A x^2}{2} - \frac{35(x-2)^4}{24}$$

Considerando que por ser simétrico los valores de la deflexión para $(\mathbf{6} \leq \boldsymbol{x} \leq \mathbf{8})$ va a tener los mismos valores que el tramo I

Después, se pasó a resolver la ecuación diferencial planteada mediante el método de diferencias finitas, tomando primero la ecuación diferencial del primer tramo $y'' = kf(x) = \frac{1}{EI}(70x - 128.33)$ y del segundo tramo en otra ecuación diferencial aparte, usaremos las conocidas formulas numéricas utilizadas en diferenciación numérica y tomar la ecuación diferencial en cada nodo. Resolviendo tendremos el siguiente sistema de ecuaciones**.**

- $Tramo\ I$

$$\begin{cases} 25y_2 - 50y_1 + 0 = -0.00376086 \\ 25y_3 - 2y_2 + 25y_1 = -0.00330033 \\ 25y_4 - 50y_3 + 25y_2 = -0.0028398 \\ 25y_5 - 50y_4 + 25y_3 = -0.00237928 \\ 25y_6 - 50y_5 + 25y_4 = -0.00191875 \\ 25y_7 - 50y_6 + 25y_5 = -0.00145822 \\ 25y_8 - 50y_7 + 25y_6 = -0.0009977 \\ 25y_9 - 50y_8 + 25y_7 = -0.00053717 \\ -50y_9 + 25y_8 = -0.13492356 \end{cases}$$

- $Tramo\ II$

$$\begin{cases} 6.75y_2 - 12.5y_1 = 0.03496283 \\ 6.75y_3 - 12.5y_2 + 6.75y_1 = 0.00185757 \\ 6.75y_4 - 12.5y_3 + 6.75y_2 = 0.00231809 \\ 6.75y_5 - 12.5y_4 + 6.75y_3 = 0.00259441 \\ 6.75y_6 - 12.5y_5 + 6.75y_4 = 0.00268651 \\ 6.75y_7 - 12.5y_6 + 6.75y_5 = 0.00259441 \\ 6.75y_8 - 12.5y_7 + 6.75y_6 = 0.00231809 \\ 6.75y_9 - 12.5y_8 + 6.75y_7 = 0.00185757 \\ -12.5y_9 + 6.75y_8 = 0.03496283 \end{cases}$$

Finalmente, para resolver el sistema de ecuaciones usamos la forma matricial $\boldsymbol{Av} = \boldsymbol{b}$, donde $\boldsymbol{v}$ es el vector de todas las deflexiones $\boldsymbol{y_i}$ a lo largo de la viga analizada la cual es:

- $Tramo\ I$

$$v = \begin{pmatrix} -0.0000841 \\ -0.0003186 \\ -0.0006852 \\ -0.0011653 \\ -0.0017406 \\ -0.0023927 \\ -0.0031031 \\ -0.0038534 \\ -0.0046252 \end{pmatrix}$$

- $Tramo\ II$

$$v = \begin{pmatrix} -0.0068922 \\ -0.0081903 \\ -0.0091912 \\ -0.0098213 \\ -0.0100362 \\ -0.0098213 \\ -0.0091912 \\ -0.0081903 \\ -0.0068922 \end{pmatrix}$$

Obteniendo de igual manera con el código de programación Phyton, la gráfica de la deflexión que de la solución exacta y de la solución aproximada por diferencias finitas, que se pueden visualizan en las figuras 5, 6, 7 y 8.

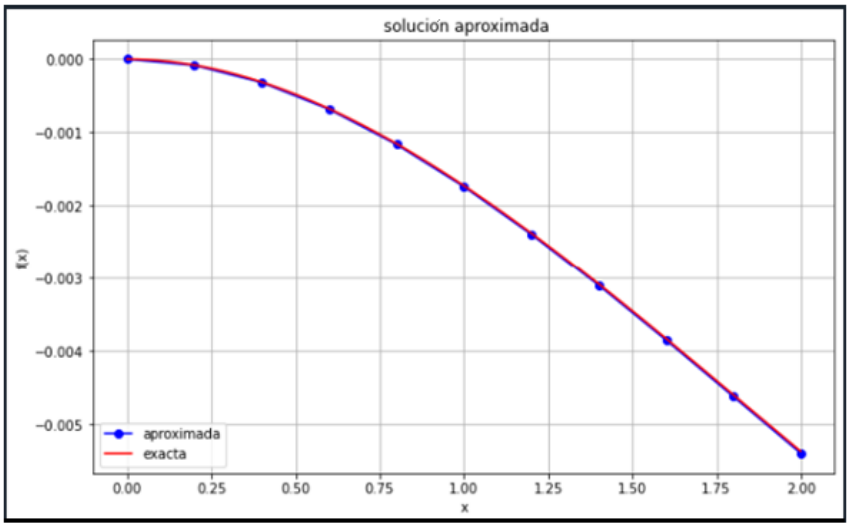

Figura 5. Resultados de la deflexión del tramo I aproximada y exacta obtenidos en *Python*.

```
In [44]: runfile('D:/Aplicaciones/Anaconda/Clases/Metodo de infinitos cuadrados.py', wdir='D:/
Aplicaciones/Anaconda/Clases')
A = [[ 50. -25.   0.   0.   0.   0.   0.   0.   0.]
 [-25.  50. -25.   0.   0.   0.   0.   0.   0.]
 [  0. -25.  50. -25.   0.   0.   0.   0.   0.]
 [  0.   0. -25.  50. -25.   0.   0.   0.   0.]
 [  0.   0.   0. -25.  50. -25.   0.   0.   0.]
 [  0.   0.   0.   0. -25.  50. -25.   0.   0.]
 [  0.   0.   0.   0.   0. -25.  50. -25.   0.]
 [  0.   0.   0.   0.   0.   0. -25.  50. -25.]
 [  0.   0.   0.   0.   0.   0.   0. -25.  50.]]
b = [ 0.00376086  0.00330033  0.0028398   0.00237928  0.00191875  0.00145822
  0.0009977   0.00053717 -0.13492336]
i: 001   x: 0.2000   v: -0.0000841
i: 002   x: 0.4000   v: -0.0003186
i: 003   x: 0.6000   v: -0.0006852
i: 004   x: 0.8000   v: -0.0011653
i: 005   x: 1.0000   v: -0.0017406
i: 006   x: 1.2000   v: -0.0023927
i: 007   x: 1.4000   v: -0.0031031
i: 008   x: 1.6000   v: -0.0038534
i: 009   x: 1.8000   v: -0.0046252
paso de malla : 0.2
```

Figura 6. Resultados de la deflexión del tramo I obtenidos en *Python.*

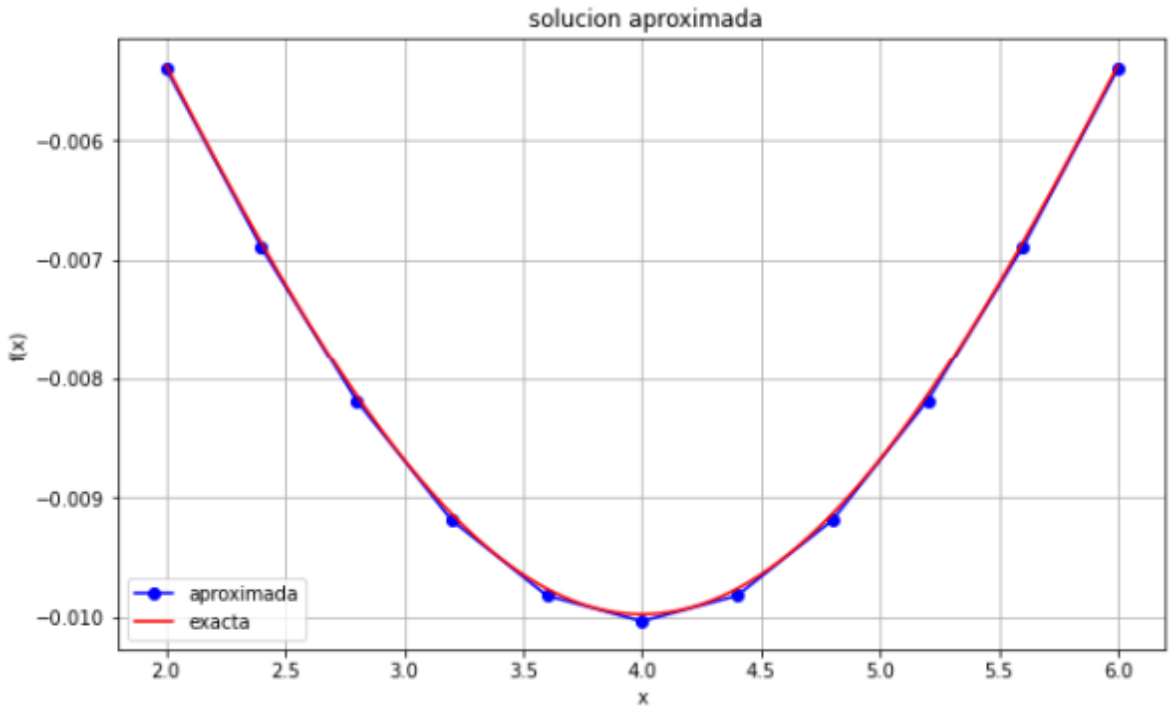

Figura 7. Resultados de la deflexión del tramo II aproximada y exacta obtenidos en *Python*.

```
In [45]: runfile('D:/Aplicaciones/Anaconda/Clases/Diferencias Finitas 2.py', wdir='D:/Aplicaciones/
Anaconda/Clases')
A = [[12.5  -6.25  0.    0.    0.    0.    0.    0.    0.  ]
 [-6.25 12.5  -6.25  0.    0.    0.    0.    0.    0.  ]
 [ 0.   -6.25 12.5  -6.25  0.    0.    0.    0.    0.  ]
 [ 0.    0.   -6.25 12.5  -6.25  0.    0.    0.    0.  ]
 [ 0.    0.    0.   -6.25 12.5  -6.25  0.    0.    0.  ]
 [ 0.    0.    0.    0.   -6.25 12.5  -6.25  0.    0.  ]
 [ 0.    0.    0.    0.    0.   -6.25 12.5  -6.25  0.  ]
 [ 0.    0.    0.    0.    0.    0.   -6.25 12.5  -6.25]
 [ 0.    0.    0.    0.    0.    0.    0.   -6.25 12.5 ]]
b = [-0.03496283 -0.00185757 -0.00231809 -0.00259441 -0.00268651 -0.00259441
 -0.00231809 -0.00185757 -0.03496283]
i: 001   x: 2.4000   v: -0.0068922
i: 002   x: 2.8000   v: -0.0081903
i: 003   x: 3.2000   v: -0.0091912
i: 004   x: 3.6000   v: -0.0098213
i: 005   x: 4.0000   v: -0.0100362
i: 006   x: 4.4000   v: -0.0098213
i: 007   x: 4.8000   v: -0.0091912
i: 008   x: 5.2000   v: -0.0081903
i: 009   x: 5.6000   v: -0.0068922
paso de malla : 0.4
```

Figura 8. Resultados de la deflexión del tramo II obtenidos en *Python*.

Discusión

El uso de programas, para el método numérico de diferencias finitas en *Python* para la solución de una ecuación diferencial es de suma ayuda, principalmente en el ahorro de tiempo y trabajo para obtener los resultados del mismo, a comparación de los métodos tradicionales, que si bien son muy aplicables, hoy en día se requiere herramientas que ahorren el mayor tiempo posible para aprovecharlo en otras partes de la investigación tales como el análisis de resultados e interpretarlos de la manera más acertada posible.

Se mostró que, con el método de diferencias finitas se puede determinar la deflexión de la viga a lo largo del tramo, obteniendo igualmente el punto máximo de deflexión a -0.0100362 m de distancia del punto inicial aproximadamente y que también cumple las condiciones de frontera $v(0) = 0\, u\, v(l) = 0$ para $0 < x < l\,.(l = 8\,m)$. Se es importante escoger el tamaño de paso o iteraciones adecuado para reducir el error, así el método de diferencias finitas tiene menor vulnerabilidad ante el error de aproximaciones de la deflexión de la viga.

La aplicación del programa *Python*, con el método iterativo de diferencias finitas para la resolución de ecuaciones diferenciales lineales, permite el ahorro en la inversión de tiempo empleado en cálculos que de otra manera hubiesen sido manuales; debido a esto, y según Ayala (2021), el método de diferencias finitas, para problemas lineales presentan mejores características de estabilidad, a costa de realizar más computo para

obtener la solución con la misma precisión. Estos métodos de diferencias finitas reemplazan las derivadas en la ecuación diferencial por un cociente de diferencias centradas, lo que obliga a escoger un parámetro h, tamaño del sub intervalo, no demasiado pequeño.

Conclusiones

Mediante el método de diferencias finitas se puede analizar la deflexión presente a lo largo de la viga, pues al aplicar dicho método se logró ver como se deformaba respecto al punto inicial; pudiendo obtener de esta manera la deflexión máxima que llegará a tener esta viga. Teniendo presente que un error relativamente pequeño, que es casi nulo.

El método de diferencias finitas aplicado en un lenguaje de programación como *Python* para calcular la deflexión de una viga, es una forma muy eficiente de lograrlo ya que obtendrás resultados muy precisos en el menor tiempo posible, esto dependerá de la cantidad de iteraciones, dependiendo del tamaño de la viga, en qué punto necesites la deflexión y por supuesto que es de mucha utilidad para los ingenieros civiles en la actualidad pero también ir dando estas nuevas herramientas a los estudiantes para acoplarse a las novedades que nacen en el mundo de la ingeniería, matemática y programación.

Referencias bibliográficas

AYALA, T. 2021. Análisis del método de disparo lineal y los métodos de diferencias finitas para problemas de valores en la frontera. Repositorio Institucional UNJFSC. Disponible en: http://repositorio.unjfsc.edu.pe/handle/UNJFSC/4242

COLLANTES SANTISTEBAN, L. 2006. Método de diferencias finitas para un problema de valor de Frontera unidimensional. *Revista ECIPeru* 3(2) 4-4. Disponible en: https://doi.org/10.33017/reveciperu2006.0011/

CONTRERAS, A., MUÑOZ, J., CONTRERAS, F., & LÓPEZ, P. 2018. Mejora de los resultados de una asignatura de Ingeniería Civil gracias a las TIC. Proyectos de innovación y mejora docentes. *indoc.uca*. 1-4. Disponible en: https://indoc.uca.es/articulos/sol-201800112637-tra.pdf.