Problema de aplicación del método de Bisección

Materia:

Seminario de Solución de Problemas de Métodos Matemáticos III

Sección:

D11

Integrantes:

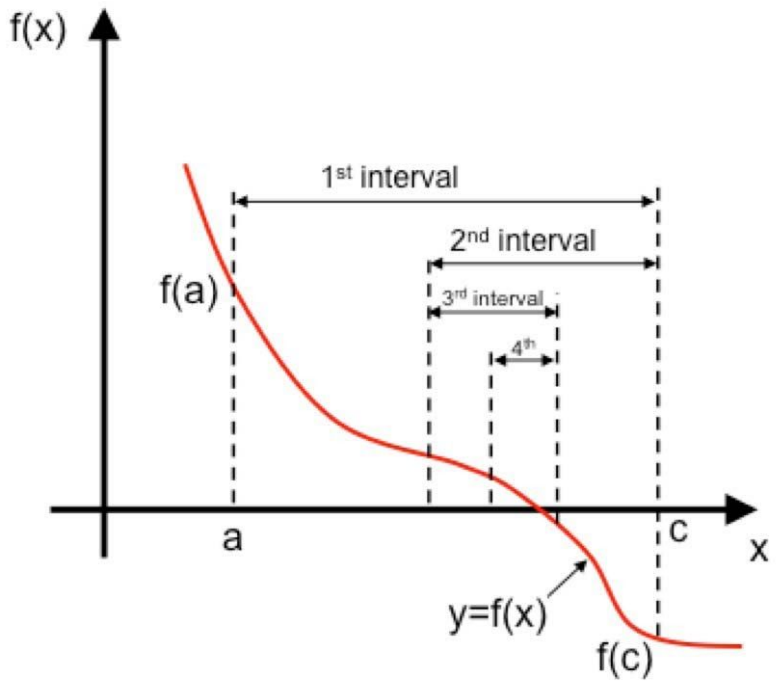
Ignacio David Vázquez Pérez 218292866

Joanna Nataly Araiza Abarca 214782907

Profesora:

María Elena Olivares Pérez

Fecha de entrega: 29 de Febrero 2019

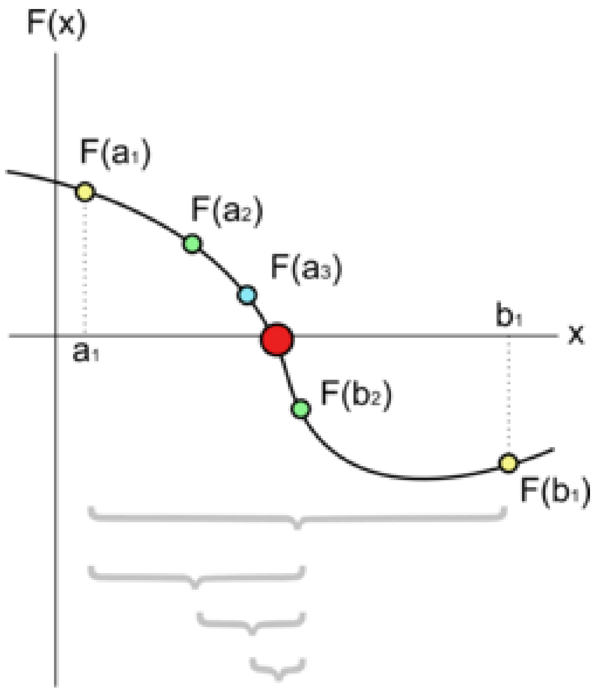


# Índice:

# Introducción

Este proyecto utilizaremos el metodo de Biseccion. Este es uno de los métodos más sencillos y de fácil intuición para resolver ecuaciones en una variable. Lo aplicaremos a desplazamientos estructurales ques es cuando una estructura está bajo la acción de fuerzas, los elementos que lo constituyen sufren “deformaciones” (o pequeños cambios en su forma) y, como consecuencia, cualquier punto de la estructura se desplazará hacia una nueva posición. En general, todos los puntos de la estructura, excepto los puntos de apoyo inmóviles, sufrirán dichos desplazamientos.

## Marco Teórico




El **método de bisección** es un método de búsqueda de raíz que se aplica a cualquier función continua para la que se conocen dos valores con signos opuestos.
Se basa en el teorema del valor intermedio(TVI), el cual establece que toda función continua f en un intervalo cerrado [a,b] toma todos los valores que se hallan entre f(a) y f(b). Esto es que todo valor entre f(a) y f(b) es la imagen de al menos un valor en el intervalo [a,b]. En caso de que f(a) y f(b) tengan signos opuestos, el valor cero sería un valor intermedio entre f(a) y f(b), por lo que con certeza existe un p en [a,b] que cumple f(p)=0. De esta forma, se asegura la existencia de al menos una solución de la ecuación f(a)=0

Suponiendo que se cumplen las condiciones iniciales para la puesta en práctica del algoritmo, definimos r como una raíz dentro del intervalo [a, b]. El intervalo de búsqueda en el n-ésimo paso tiene longitud:

$$l_n = \frac{b_n - a_n}{2} = \frac{|b-a|}{2^n}$$

Como rn que es la raíz n-ésima calculada, se encuentra siempre dentro del intervalo de búsqueda, tenemos entonces que:

$$|r - r_n| \leq \frac{|b-a|}{2^n}$$

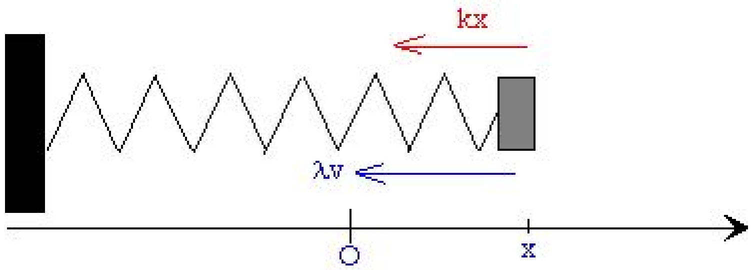
Tomando límites,

$$\lim_{n \to \infty} |r - r_n| = 0 \to \lim_{n \to \infty} r_n = r$$

Queda demostrado entonces, que si se cumplen las condiciones iniciales del problema, el método de bisección converge al menos, a una de las raíces que se encuentran en el intervalo señalado.

## Planteamiento del Problema

La experiencia nos muestra que la amplitud de un cuerpo vibrante tal como un resorte o un péndulo, decrece gradualmente hasta que se detiene**.**




Para explicar el amortiguamiento, podemos suponer que además de la fuerza elástica F=-kx, actúa otra fuerza opuesta a la velocidad Fr=-lv, donde l es una constante que depende del sistema físico particular. Todo cuerpo que se mueve en el seno de un fluido viscoso en régimen laminar experimenta una fuerza de rozamiento proporcional a la velocidad y de sentido contrario a ésta.

La ecuación del movimiento se escribe

ma=-kx-λv

La características esenciales de las oscilaciones amortiguadas:

- La amplitud de la oscilación disminuye con el tiempo.
- La energía del oscilador también disminuye, debido al trabajo de la fuerza Fr de rozamiento viscoso opuesta a la velocidad.
- En el espacio de las fases (v-x) el móvil describe una espiral que converge hacia el origen.

Si el amortiguamiento es grande, g  puede ser mayor que w0, y w puede llegar a ser cero (oscilaciones críticas) o imaginario (oscilaciones sobreamortiguadas). En ambos casos, no hay oscilaciones y la partícula se aproxima gradualmente a la posición de equilibrio. La energía que pierde la partícula que experimenta una oscilación amortiguada es absorbida por el medio que la rodea.

**Solución**

El desplazamiento de una estructura está definido por la ecuación siguiente para una oscilación amortiguada:

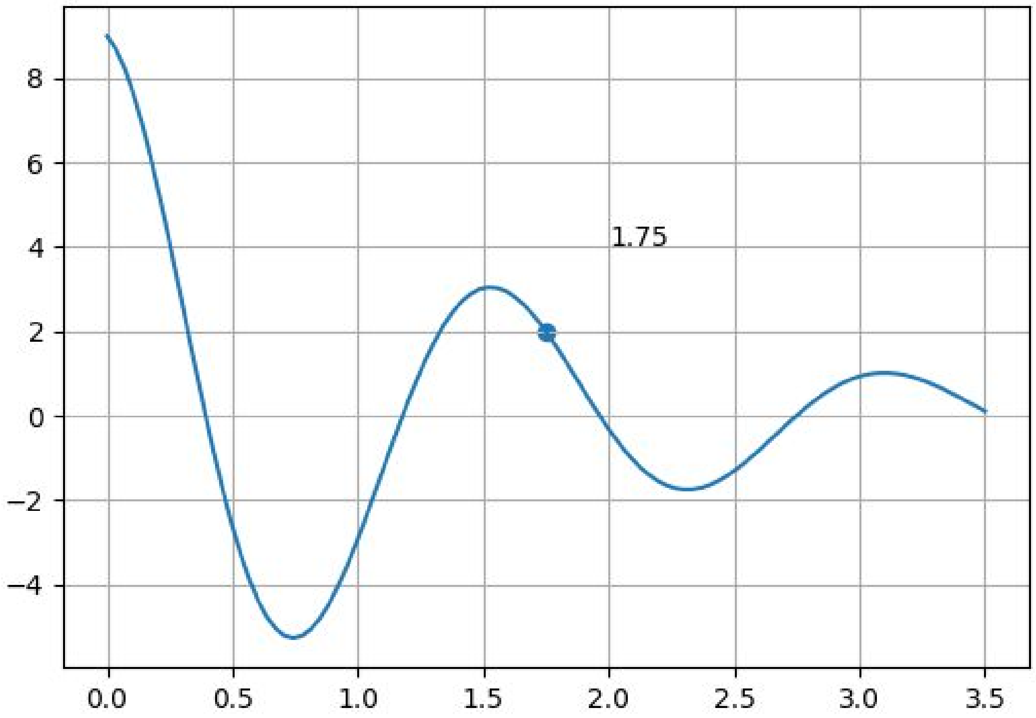
$$y = 9e^{-kt}cos(wt)$$

donde k = 0.7 y w = 4.

Determinar el tiempo que se requiere para que el desplazamiento disminuya a 3.5.

Emplee el método de bisección, para determinar la raíz con e = 0.01%

```python
import numpy as np
import matplotlib.pyplot as plt

def f(x):
    return (9*np.exp(-0.7*x))*(np.cos(4*x))

def signosIguales(x,y):
    if (x<0) and (y>=0):
        return False
    if (y<0) and (x>=0):
        return False
    return True


def biseccion(a,b,n):
    arreglo_c=[]
    i = 1
    while i<=n:
        c = (a+b)/2
        arreglo_c.append(c)
        print(a,b,c)
        if not signosIguales(f(a),f(c)):
            b = c
        if not signosIguales(f(b),f(c)):
            a = c
        i+=1
    return arreglo_c[n-1]

error=0.01
a=0
b=3.5

n= int(round((np.log(b-a/error))/np.log(2)))
print("n: ", n)

k=biseccion(a,b,n)
print("Error:", round(f(k), 3))

#grafica
eje_x = np.linspace(a,b, 100)
plt.plot(eje_x, f(eje_x))
plt.grid()
plt.scatter(k,f(k))
plt.text(2,4,k)
plt.show()
```




n: 2
0 3.5 1.75
0 3.5 1.75
Error: 1.993

El método consiste en lo siguiente:

- Debe existir seguridad sobre la continuidad de la función f(x) en el intervalo [a,b]
- Se calcula el punto medio m del intervalo [a,b] y se evalúa f(m) si ese valor es igual a cero, ya hemos encontrado la raíz buscada
- En caso de que no lo sea, verificamos si f(m) tiene signo opuesto con f(a) o con f(b)
- Se redefine el intervalo [a, b] como [a, m] ó [m, b] según se haya determinado en cuál de estos intervalos ocurre un cambio de signo
- Con este nuevo intervalo se continúa sucesivamente encerrando la solución en un intervalo cada vez más pequeño, hasta alcanzar la precisión deseada.

## Conclusiones

Para poder saber el comportamiento de una función, en este caso de sobre estructuras, podemos saber la raiz de la función para poder calcular y asegurar que la estructura satisfaga todos los criterios de diseño y que se deforme dentro de los límites de servicio y seguridad establecidas. Además de para ser utilizados al formular las relaciones de compatibilidad necesarias para resolver estructuras indeterminadas

## Referencias Bibliográficas


- **Richard L Burden, J. Douglas Faires (2000), "Numerical Analysis, (7th Ed)", Brooks/Cole. ISBN 0-534-38216-9.**
- **Corliss, George (1977), «Which root does the bisection algorithm find?», *SIAM Review* 19 (2): 325-327, ISSN 1095-7200, doi:10.1137/1019044**