FACULTAD
DE INGENIERIA
Universidad de Buenos Aires

Análisis numérico I

TRABAJO PRÁCTICO:

Vibraciones en una estructura excitada

Grupo 16

Integrado por:

Encinoza Vilela, Nathalia Lucia (106295)

Profesor(es):

Miguel Cavaliere

Fernando Sanchez Sarmiento

Introducción

El presente trabajo consiste en el análisis de un problema de vibraciones excesivas en una estructura que está siendo sometida a una carga cíclica y consta de dos partes.

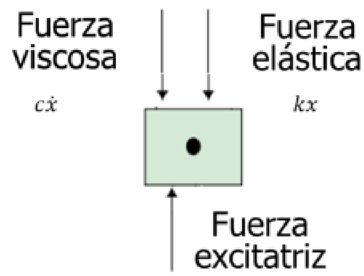
La primera parte consiste en resolver el problema mediante la segunda ley de Newton, donde obtenemos una ecuación diferencial de segundo grado que luego se convierte en un sistema de ecuaciones diferenciales de primer orden. Para resolver dicho sistema aplicamos el método de Runge Kutta de orden dos, en específico el método del punto medio, y se obtiene el valor de la velocidad y aceleración.

La segunda parte consiste en determinar el coeficiente de amortiguamiento para evitar vibraciones excesivas en la estructura, para esto definimos una función f(x), donde "f(x)" es la amplitud deseada y "x" el coeficiente de amortiguamiento. Luego para hallar el coeficiente se aplica el método de la secante, junto con el método del punto medio.

Desarrollo

1. **Primera parte**

Para analizar el problema tenemos el siguiente diagrama de cuerpo libre:



Aplicamos la segunda ley de Newton:

$$\sum F = ma \rightarrow f(t) - c\dot{x}(t) - kx(t) = m\ddot{x}$$

Con las condiciones iniciales $x(t) = 0$ y $\dot{x}(t) = 0$

Tenemos que reducir el orden de la ecuación diferencial de segundo orden, en un conjunto equivalente de dos ecuaciones diferenciales de primer orden. Para esto, introducimos las funciones:

$$x_1 = x(t) \qquad\qquad x_2 = \dot{x}(t) = \frac{dx(t)}{dt} \equiv \dot{x}_1(t)$$

Reescribimos la ecuación diferencial de segundo orden como:

$$f(t) - kx_1 - cx_2 = m\dot{x}_2$$

Con esto, obtenemos nuestro sistema de ecuaciones:

$$\dot{x}_1(t) = x_2(t)$$

$$\dot{x}_2(t) = f(t) - \frac{k}{m}x_1(t) - \frac{c}{m}x_2(t)$$

Con los siguientes datos:

k = 952 $\frac{N}{mm}$

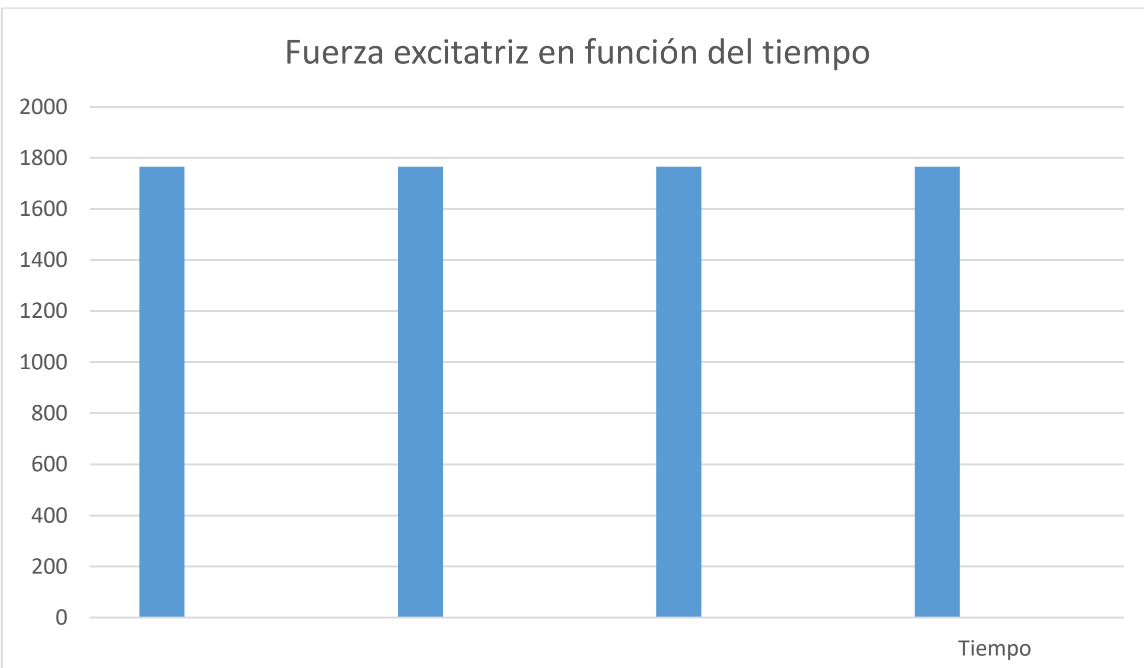
m = 10 ton

F_{MAX} = 1765

P (Periodo) = 0.6532945

C = 100

Se muestra a continuación la variación de la fuerza excitatriz:



La altura de las barras representa la fuerza máxima durante el periodo/10 y la distancia entre las barras corresponde al periodo P.

Para resolver el sistema de ecuaciones diferenciales aplicamos el método de Runge Kutta.

Los métodos de Runge-Kutta son una serie de métodos numéricos usados para encontrar aproximaciones de las soluciones de ecuaciones diferenciales y sistemas de ecuaciones diferenciales, lineales y no lineales.

Específicamente, trabajaremos con el método de punto medio, que consiste en hallar x_{n+1} de la forma:

$$x_{n+1} = x_n + h * f(x_n + \frac{h}{2} * f(x_n, t_n), t_n + \frac{h}{2})$$

Dicho método tiene un error de la forma $e \leq O(h^2)$, por lo tanto es de orden 2.

De resolver el sistema de ecuaciones, se obtuvieron las siguientes gráficas (para 10 ciclos):

- Grafica de la figura 1 que corresponde a la variación del desplazamiento respecto al tiempo

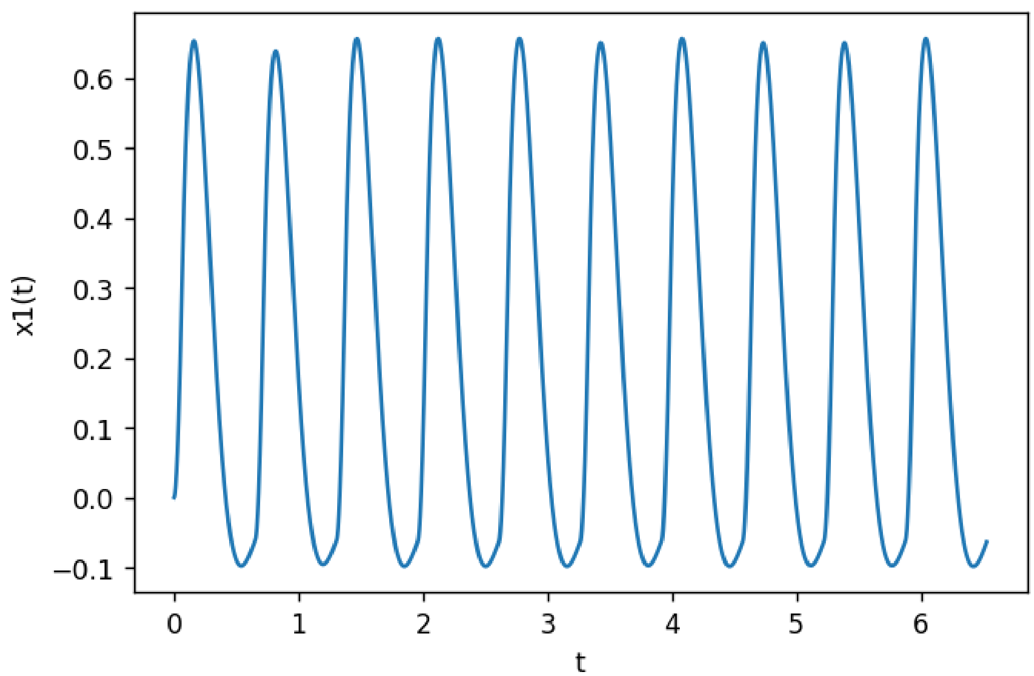


Fig.1: Gráfica de $x_1(t)$ vs t

Obteniéndose el valor máximo de desplazamiento: 0.6564912633163413

- Grafica de la figura 2 que corresponde a la variación de la velocidad respecto al tiempo

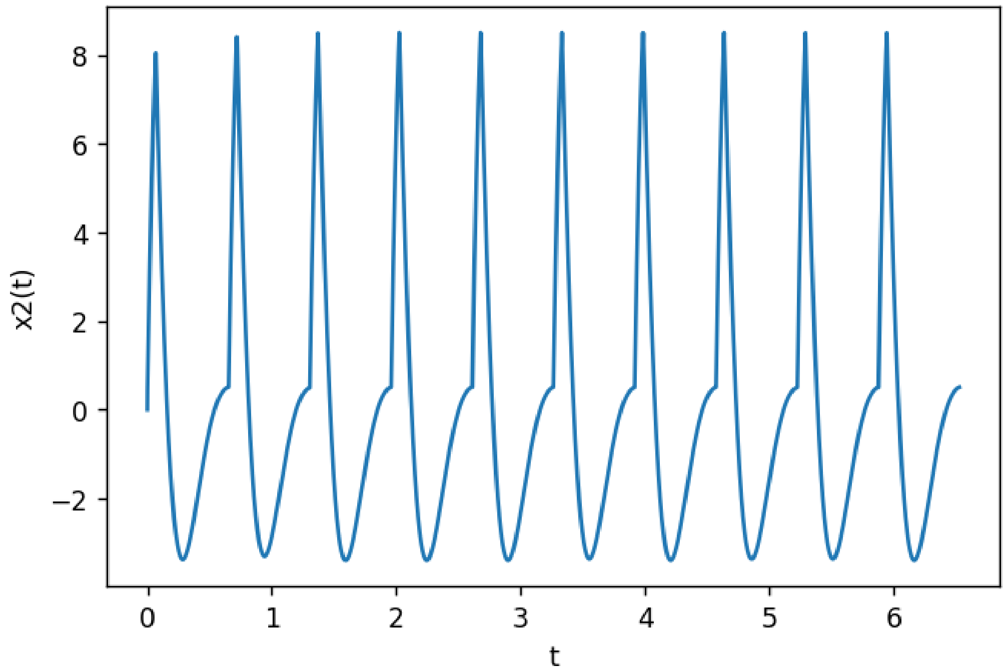


Fig.2: Gráfica de $x_2(t)$ vs t

Obteniéndose el valor máximo de la velocidad: 8.50293089329103

2. Segunda parte

Para la obtención del coeficiente de amortiguamiento tenemos como dato un valor de x admisible:

$$X_{ADM} = 0.61$$

Para determinar qué valor máximo(X_{MAX}) es igual a X_{ADM} tenemos que generar una ecuación no lineal, como no tenemos la ecuación analítica, por ende, no podemos derivarla, aplicaremos el método de la secante.

Llamamos a F(c), una función que depende del coeficiente de amortiguamiento a:

$$\vec{F}(\vec{X},t) = \begin{pmatrix} F_1(t) \\ F_2(t) \end{pmatrix} = \begin{pmatrix} x_2(t) \\ f(t) - \frac{k}{m}x_1(t) - \frac{c}{m}x_2(t) \end{pmatrix}$$

Decimos que:

$$x_{MAX} = F(c) \rightarrow x_{ADM} = F(c_B) \rightarrow F(c_B) - x_{ADM} = 0$$

Donde, X_{ADM} es el valor de x deseado, y C_B es el valor del coeficiente deseado.

Definimos la función:

$$g(c_B) = f(c_B) - x_{ADM}$$

Donde, g(c) será el residuo de la resta.

Para hallar el coeficiente aplicamos el método de la secante, donde:

$$c_{i+1} = \frac{g(c_i)(c_i - c_{i-1})}{g(c_i) - g(c_{i-1})}$$

Y obtenemos los siguientes datos:

C	F(c)	g(C)
100	0.6564912633163413	0.04649126331634135
120	0.5951006404240666	-0.014899359575933357
115.1460471081787	0.6082704827958799	-0.001729517204120068
114.50860629473658	0.6100417382613952	$4.173826139519665\text{x}10^{-5}$
114.52362709315521	0.6099999043198483	$-9.568015169048039 \text{ x}10^{-8}$
114.52359273846096	0.609999999994721	$-5.278999459790157 \text{ x}10^{-12}$
114.52359273656539	0.6100000000000001	$1.1102230246251565 \text{ x}10^{-16}$

Concluimos que el coeficiente de amortiguamiento que buscábamos es:

C = 114.52359273656539

Conclusión

El problema estudiado es un sistema de vibraciones forzadas con amortiguamiento, el cuerpo está sometido a una fuerza excitatriz y el movimiento es en una dirección por lo tanto se considera el sistema con un solo grado de libertad.

La fuerza viscosa está dada por:

$$F_v = c\dot{x}$$

donde c es el coeficiente de amortiguamiento, y $\dot{x}$ es la velocidad.

El coeficiente de amortiguamiento de acuerdo a su valor, se clasifica en tres casos:

- Subamortiguado: $c < \sqrt{4km}$
- Amortiguado crítico: $c = \sqrt{4km}$
- Sobreamortiguado: $c > \sqrt{4km}$

Como proporcionan un X_{ADM} para determinar el coeficiente de amortiguamiento los resultados obtenidos usando el método de la secante corresponden al caso de subamortiguado, cuyo valor es menor que:

$$\sqrt{4km} = 192.35$$

En la primera parte del trabajo se usó como coeficiente de amortiguamiento uno escogido al azar con un valor de 100 y se puede observar en la gráfica que los valores de x superan al X_{ADM} = 0.61.

Luego se procedió a usar el método de la secante para determinar el coeficiente de amortiguamiento con los datos proporcionados que corresponden al sistema, como lo son la masa, el coeficiente de rigidez del resorte, la fuerza excitatriz y el X_{ADM}, obteniéndose un coeficiente con un valor de 114.52359273656539, que es menor a la $\sqrt{4km}$.

ANEXO

Código escrito en Python:

```
#Estudiante: Nathalia Lucia Encinoza Vilela, padron: 106295

import numpy as np
import matplotlib.pyplot as plt

FUERZA_EXCITATRIZ_MAXIMA = 1765
PERIODO = 0.6532945
PUNTOS_POR_CICLO = 500
X_ADMISIBLE = 0.61
MASA = 10
K = 925
CANT_CICLOS = 5

def puntoMedio(funcion, c):
    #Inicializamos dos datos que se utilizaran
    h = PERIODO/PUNTOS_POR_CICLO
    x1 = 0
    x2 = 0
    x1_max = 0
    x2_max = 0
    t = 0
    ciclo = 1
    contador_puntos_por_ciclo = 0
    arreglo_x1 = []
    arreglo_x2 = []
    arreglo_t = []

    #Iteramos utilizando 500 puntos por ciclo, en este caso son 5 ciclos

    for i in range(CANT_CICLOS*PUNTOS_POR_CICLO):
        arreglo_x1.append(x1)
        arreglo_x2.append(x2)
        arreglo_t.append(t)

        q_1_x1 = x2
        q_1_x2 = funcion(x1, x2, t, ciclo, c)
        q_2_x1 = (x2 + q_1_x2*h/2)
        q_2_x2 = funcion(x1 + q_1_x1*h/2, x2 + q_1_x2*h/2, t + h/2, ciclo,
c)
        x1 += (q_1_x1 + q_2_x1)*h/2
        x2 += (q_1_x2 + q_2_x2)*h/2
        t = i*h
```

```
        #Actualizamos para buscar la amplitud maxima
        if (x1 > x1_max):
            x1_max = x1

        if (x2 > x2_max):
            x2_max = x2

        #Actualizamos por que ciclo van para usarlo en la funcion y
modificar el tiempo en funcion del periodo
        contador_puntos_por_ciclo += 1
        if (contador_puntos_por_ciclo == PUNTOS_POR_CICLO):
            ciclo += 1
            contador_puntos_por_ciclo = 0

    return arreglo_x1, arreglo_x2, arreglo_t, x1_max, x2_max

def funcion(x1, x2, t, ciclo, c):
    #Verificamos si el tiempo esta dentro de Periodo/10
    if (t >= PERIODO):
        tiempo = t - (PERIODO* (ciclo-1))
    else:
        tiempo = t
    if (tiempo <= PERIODO/10):
        a = FUERZA_EXCITATRIZ_MAXIMA/MASA
    else:
        a = 0
    b = K*x1/MASA
    c = c*x2/MASA
    return a-b-c

def g(c):
    arreglo_x1, arreglo_x2, arreglo_t, x1_max, x2_max= puntoMedio(funcion,
c)
    return x1_max - X_ADMISIBLE

def metodoSecante():
    c1 = 100
    c2 = 120
    c3 = 0
    valores_g = []
    valores_g.append(g(c1))
    valores_g.append(g(c2))

    valores_c = []
```

```python
    valores_c.append(c1)
    valores_c.append(c2)

    valores_f = []
    valores_f.append(g(c1) + X_ADMISIBLE)
    valores_f.append(g(c2) + X_ADMISIBLE)

    for i in range(5):
        c3 = c2 - (g(c2)*(c2-c1))/(g(c2)-g(c1))
        valores_g.append(g(c3))
        valores_c.append(c3)
        valores_f.append(g(c3) + X_ADMISIBLE)
        c1 = c2
        c2 = c3
    return valores_c, valores_f, valores_g

##PARA ENCONTRAR LOS VALORES OBTENIDOS DE LA TABLA DE C, F(C) Y G(C) CORRER:
c, f, f_g = metodoSecante()
print(c)
print(f)
print(f_g)

##PARA ENCONTRAR LOS GRAFICOS OBTENIDOS EN LA PRIMERA PARTE:
# x1, x2, t, x1_max, x2_max = puntoMedio(funcion, 100)
# plt.figure(dpi = 125)
# plt.plot(t, x1)
# plt.ylabel("x1(t)")
# plt.xlabel("t")
# plt.show()

# x1, x2, t, x1_max, x2_max = puntoMedio(funcion, 100)
# plt.figure(dpi = 125)
# plt.plot(t, x2)
# plt.ylabel("x2(t)")
# plt.xlabel("t")
# plt.show()
```

También se puede encontrar en el collab:
https://colab.research.google.com/drive/11jCmE4MF6_V5OWHYv_GMoRUfXUCVQJce?usp=sharing