

Actividad 10

José Daniel Gaytán Villarreal
Grupo 3

28 de mayo de 2019

Resumen

Se resolvió la ecuación de Duffling a través del comando *ode* con diferentes valores para β y ω en cada iteración, y se estudió el fenómeno de histéresis presente en los resultados.

1. Introducción

El presente trabajo busca resolver la ecuación de Duffling a través del lenguaje Python, con el fin de visualizar en una gráfica el fenómeno de histeresia que se presenta en este tipo de ecuaciones.

El objetivo de la actividad fue el resolver un sistema de ecuaciones diferenciales de segundo orden, manejándolo como un sistema de primer orden, a través de un código en el cookbook de SciPy.

En la sección 2, se habla brevemente sobre cómo se modificó la ecuación diferencial para introducirla en el código; a su vez, se mostrarán las gráficas obtenidas dadas ciertas condiciones iniciales al sistema. Por último, en la sección 3, se emite una breve conclusión referente a la metodología utilizada y sus subsecuentes resultados.

2. Desarrollo

2.1. Marco Teórico

2.1.1. Histéresis

Antes de proseguir a resolver la ecuación diferencial, es necesario conocer primero qué es la *histéresis*. Podemos referirnos a este fenómeno como "*la tendencia de un material a conservar sus propiedades en ausencia del estímulo que la ha generado*"[1]. Es, básicamente, un fenómeno que depende no solo de sus circunstancias actuales, sino también de cómo ha llegado a esas circunstancias. Si toma cierto camino de ida, el camino de vuelta no será el mismo, pues se verá afectado por sus condiciones anteriores.

La histéresis está presente en varios fenómenos químicos, como eventos de transición de fase, y físicos, como en el magnetismo. Ahondando en éste último, la histéresis magnética sucede cuando se magnetiza en una dirección un material ferromagnético y éste no vuelve de nuevo a magnetización cero cuando cesa el campo magnético impulsor [2], restando un magnetismo residual. Si aplicasemos un campo magnético alterno, su magnetización trazará un lazo o bucle llamado *ciclo de histéresis*.

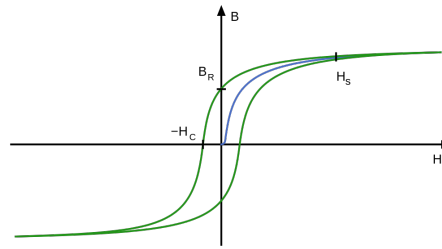


Figura 1: Ciclo de histéresis.

Es importante remarcar que cada material ferromagnético tiene su propio lazo de histéresis característico [3], el cual puede conocerse a través del llamado magnetómetro de Köpsel, que se encarga de proporcionarle al material ferromagnético los cambios senoidales de la corriente eléctrica para modificar el sentido de los imanes.

2.1.2. Ecuación de Duffing

La llamada *ecuación de Duffing* fue establecida a principios del siglo XX por el ingeniero **George Duffing** con el objetivo de modelar las vibraciones forzadas de una máquina industrial [4]. Se trata de una ecuación no lineal de segunda forma y se escribe de la siguiente manera:

$$\ddot{x} + \delta\dot{x} + \alpha x + \beta x^3 = \gamma \cos(\omega t)$$

donde la función desconocida $x = x(t)$ es el desplazamiento en un tiempo t , \dot{x} es la primera derivada de la posición con respecto al tiempo (velocidad), \ddot{x} es la segunda derivada de la posición con respecto al tiempo (aceleración), δ es el coeficiente de amortiguamiento, α es la rigidez linear del sistema, β es la fuerza restauradora no lineal, γ es la amplitud de la fuerza periódica y ω es su frecuencia angular [5]. Nótese que lo único que la diferencia de un oscilador forzado es la presencia de un término no lineal βx^3 , al cual se le conoce como *término de Duffing* [4].

Para ciertos parámetros, la ecuación de Duffing presenta "saltos" en su frecuencia, lo que genera una curvatura para altas frecuencias que pueden apreciarse en la siguiente gráfica:

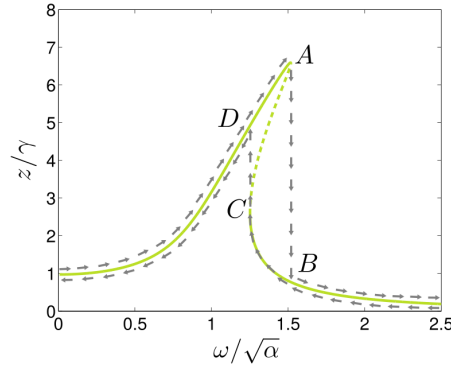


Figura 2: "Saltos" en la frecuencia de un oscilador de Duffing.

El oscilador de Duffing es verdaderamente interesante de estudiar, pues presenta ciertas características que lo vuelven un sistema complejo e impredecible. Si bien es utilizado algunas veces como aproximación a sistemas de péndulos, es estudiado por su casi inevitable tendencia a ser catastrófico [6], hecho que es evidente dada la presencia de histéresis [7].

2.2. Metodología

Se buscó modelar un oscilador de Duffing con los siguientes parámetros constantes:

$$\alpha = \gamma = 1, \delta = 0,1$$

Los parámetros ω y β fueron modificados para cada solución del sistema: el primero tomaba valores ascendentes, o descendentes, del intervalo $[0,2.5]$, mientras que el segundo tomó cuatro valores distintos: -0.003, 0.00, 0.01 y 0.04.

La ecuación se resolvió a partir del comando *ode* de la biblioteca *scipy*, cuyo desarrollo se muestra a continuación.

Importamos las bibliotecas a utilizar:

```
from math import cos
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import ode
```

Definimos la función a integrar; en este caso, definimos la ecuación de Duffing:

```
def f(t, z, p):

    delta, alpha, beta, gamma, omega = p
    x, y = z
    f = [y, - delta*y - alpha*x - beta*x**3 + gamma*cos(omega*t)]

    return f
```

Establecimos el método de integración con el que se resolverá la ecuación diferencial; en este caso, utilizamos el método *dopri5*, el cual se trata del método de Runge-Kutta de 4to orden.

```
solver = ode(f)
solver.set_integrator('dopri5')
```

A partir de aquí, se resolvió el sistema de dos maneras diferentes: una, actualizando en cada iteración las condiciones iniciales a las condiciones finales de la última iteración y realizando una trayectoria de ida y otra de vuelta; la segunda, manteniendo de manera constante nuestras condiciones iniciales en una única trayectoria de ida, modificando el valor de β en cada iteración. Este detalle fue crucial al momento de resolver las ecuaciones, pues los resultados, aunque similares, presentan conclusiones diferentes.

Para la primera forma, introducimos el solucionador en un loop de la siguiente manera:

```
gamma = 1.0
alpha = 1.0
beta = 0.01
delta = 0.1
```

```

omega = np.linspace(0, 2.5, pts)

t0 = 0.0
z0 = [1.0, 0.0]
z = 0

for i in range(0,pts):

     $\omega$  = omega[i]
    val =  $\delta$ ,  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\omega$ 
    solver.set_f_params(val)

    solver.set_initial_value(z0, t0)

    t1 = 20
    n = 400
    t = np.linspace(t0, t1, n)
    sol = np.empty((n, 2))
    sol[0] = z0

    k = 1
    while solver.successful() and solver.t < t1:
        solver.integrate(t[k])
        sol[k] = solver.y
        k += 1

    x1 = sol[:,0]
    y1 = sol[:,1]

    t0 = 0.0
    z0 = [max(x1),0]
    w1.append( $\omega$ )
    graf1.append(z0[0])

```

El algoritmo para la vuelta es prácticamente igual, siendo la única diferencia la manera en que establecemos los valores de ω :

```

if omega[-j]==0.0:
     $\omega$  = 2.5
    val =  $\delta$ ,  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\omega$ 
    solver.set_f_params(val)
else:
     $\omega$  = omega[-j]
    val =  $\delta$ ,  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\omega$ 
    solver.set_f_params(val)

```

El código anterior es la manera en que indicamos que se trata de una vuelta; es decir, lee los valores de ω al revés. Los resultados de ambas iteraciones se mostrarán como una sola gráfica en la sección 2.3.

Por otro lado, si mantenemos las condiciones iniciales constantes y modificamos el valor de β , el algoritmo de solución es casi igual:

```

     $\beta_j$  = [-0.003,0.0000000001,0.01,0.04]

for j in range (0,len( $\beta_j$ )):

    wi=[]
    grafi=[]

    for i in range(0,251):

         $\gamma$  = 1.0
         $\alpha$  = 1.0
         $\delta$  = 0.1
         $\omega$  = i*0.01
         $\beta$  =  $\beta_j[j]$ 
        # $\beta$ =0.04
        val =  $\delta$ ,  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\omega$ 
        solver.set_f_params(val)

        t0 = 0.0
        z0 = [1.0, 0]
        solver.set_initial_value(z0, t0)

        t1 = 40
        n = 400
        t = np.linspace(t0, t1, n)
        sol = np.empty((n, 2))
        sol[0] = z0

        k = 1
        while solver.successful() and solver.t < t1:
            solver.integrate(t[k])
            sol[k] = solver.y
            k += 1

        x = sol[:,0]

        wi.append( $\omega$ )
        grafi.append(max(abs(x)))

    w.append(wi)
    graf.append(grafi)

```

Es importante remarcar que en esta iteración **no** hubo un camino de vuelta, pues se resolvió solamente una vez para valores de ω ascendentes. Los resultados para cada valor de β se muestran en una sola gráfica en la sección 2.3.

2.3. Resultados

Para la solución donde se actualizaban las condiciones iniciales y se realizó una ida y una vuelta, se obtuvo la siguiente gráfica:

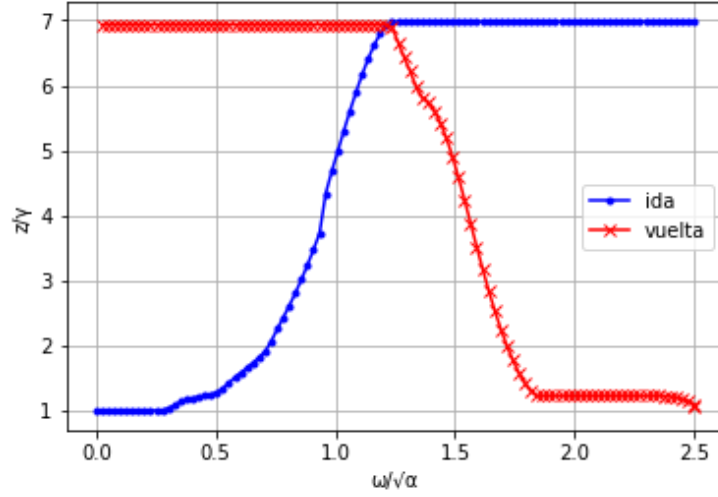


Figura 3: Solución para la ecuación de Duffing con condiciones iniciales actualizándose cada iteración.

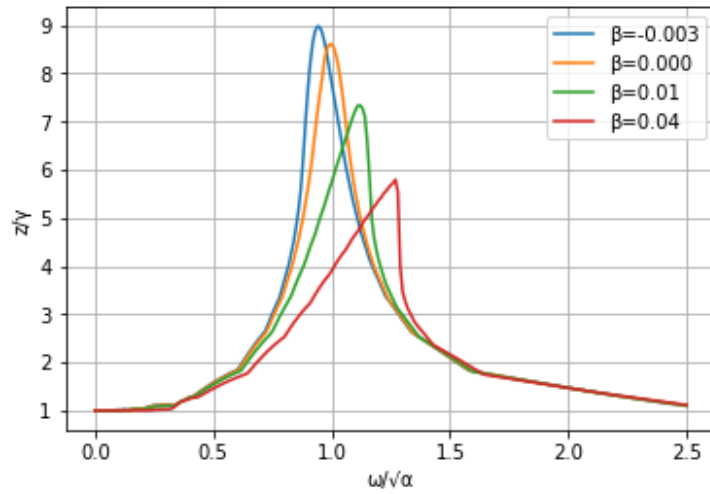


Figura 4: Solución para la ecuación de Duffing con condiciones iniciales constantes, beta variable.

3. Conclusión

Podemos eximir varias conclusiones para cada gráfica resultante.

En la primera gráfica, se observa de manera clara que a partir de cierto punto crítico, el algoritmo se "trunca" el resultado se vuelve constante. Esto sucede debido a que, al actualizar en cada iteración las condiciones iniciales, se llega a un valor de velocidad crítico que hace que la solución tienda a infinito, manteniéndose como una constante fija en cada iteración posterior a ese valor crítico.

Si bien no es correcto decir que lo anterior sucede únicamente debido a la histéresis, el hecho de que el fenómeno se repita para valores de ida y de vuelta es una muestra de que, en efecto, dicho fenómeno está presente al momento de actualizar para cada nuevo valor de condiciones iniciales.

Por otro lado, los resultados observados en la segunda gráfica describen un cambio notable en la frecuencia, una cierta *curvatura* en la gráfica, para valores de β distintos a cero. La razón de esto se debe a que cuando $\beta = 0$, la frecuencia del oscilador es lineal; sin embargo, cuando $\beta \neq 0$, la frecuencia pierde su linealidad.

Lo que sucede en este último caso es que el oscilador se torna o más rígido o más suelto, hecho que explica la curvatura en la gráfica. Es decir, la fuerza restauradora β determina integralmente la forma en la que el oscilador regresará a su forma original; a valores grandes de β , la curva comienza a curvarse hacia la derecha y la amplitud máxima comienza a achataarse, ya que la fuerza restauradora impide al resorte alejarse mucho de su posición de reposo.

Referencias

- [1] *Histéresis* (s.f) Wikipedia. Recuperado el 20 de mayo de 2019 de: <https://es.wikipedia.org/wiki/Histéresis>
- [2] *Histéresis* (s.f) Hyper Physics. Recuperado el 20 de mayo de 2019 de: <http://hyperphysics.phy-astr.gsu.edu/hbasees/Solids/hyst.html>
- [3] Laboratorio de Materiales Magnéticos. (2016) *Ciclo de histéresis de materiales ferromagnéticos*. Universidad Complutense de Madrid. Recuperado el 21 de mayo de 2019 de: https://www.ucm.es/data/cont/media/www/pag-17833/MaterialesMagnéticos_2016-17.pdf
- [4] *Oscillateur de duffing*. (2001) École Normale Supérieure. Recuperado el 22 de mayo de 2019 de: <http://www.physique.enscachan.fr/laboratoire/experiences/fichiers/Duffing.pdf>
- [5] *Duffing equation* (s.f) Wikipedia. Recuperado el 20 de mayo de 2019 de: https://en.wikipedia.org/wiki/Duffing_equation
- [6] Rand, R. (2005) *Lecture notes on Nonlinear Vibrations*. Cornell University: Dept. Theoretical and applied mechanics. Recuperado el 23 de mayo de 2019 de: <http://audiophile.tam.cornell.edu/randdocs/nlvibe52.pdf>

- [7] *Teoría de las catástrofes* (s.f) Wikipedia. Recuperado el 23 de mayo de 2019 de:
https://es.wikipedia.org/wiki/Teor%C3%ADa_de_las_cat%C3%A1strofes