



DIVISIÓN DE CIENCIAS EXACTAS  
DEPARTAMENTO DE FÍSICA  
CURSO DE FÍSICA COMPUTACIONAL

## REPORTE

# SOLUCIÓN NUMÉRICA DE ECUACIONES DIFERENCIALES ORDINARIAS

ACTIVIDAD 9

*Autor:*

Miguel Ernesto MEDINA LEÓN

*Profesor:*

Carlos LIZÁRRAGA CELAYA

AÑO ACADÉMICO 2018-2019

# 1 Introducción

El propósito de esta actividad es la de resolver numéricamente un sistema de ecuaciones diferenciales ordinarias que describen el siguiente sistema:

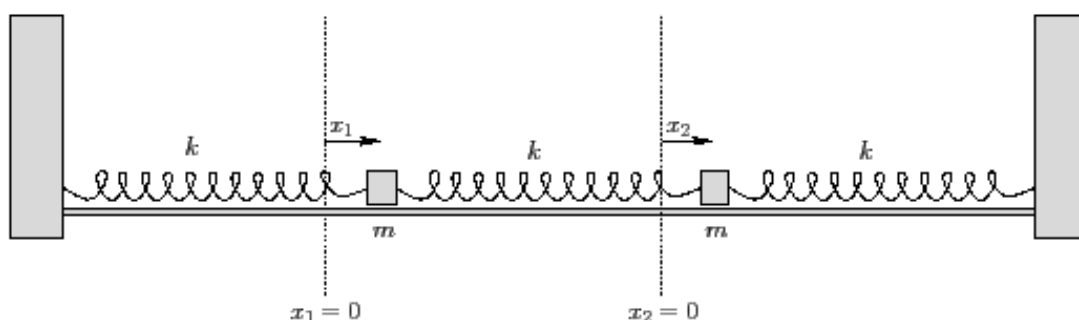


Figura 1: Sistema masa-resorte acoplado.

Tomando como base las notas de Richard Fitzpatrick (profesor de física de la Universidad de Texas que resolvió analíticamente el sistema anteriormente mostrado), se intentará resolver numéricamente dicho sistema utilizando la función *odeint* de SciPy.

Cabe aclarar que se adaptará el código mostrado en el Cookbook de SciPy, para resolver el problema ya mencionado.

El objetivo es conseguir una gráfica como la siguiente:

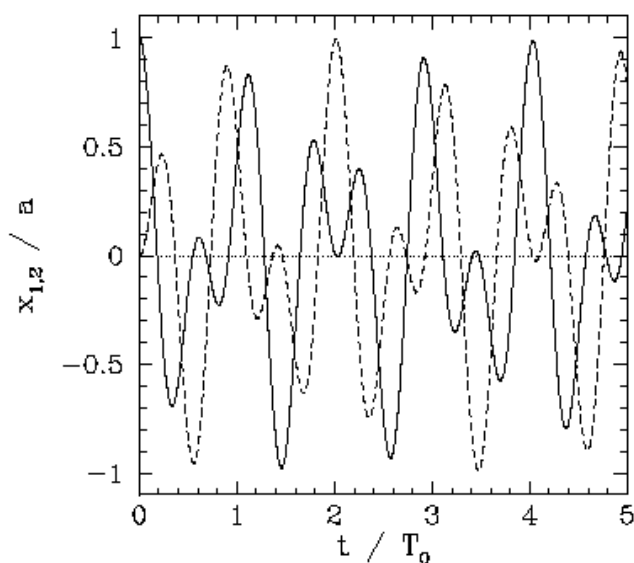


Figura 2: Figura 16 de las notas de R. Fitzpatrick.

## 2 Desarrollo

### 2.1 SciPy

[5] SciPy es un ecosistema basado en Python de software de código abierto para matemáticos, científicos e ingenieros. En particular, algunas de sus paquetes son:

- Numpy
- Pandas
- Matplotlib
- Sympy
- IPython

#### 2.1.1 ODEINT

[6] Las ecuaciones diferenciales son resueltas en Python con el paquete `Scipy.integrate`, por medio de la función `ODEINT`.

ODEINT solo requiere tres inputs:

```
y = odeint(model, y0, t)
```

Siendo

- `model`: Nombre de la función que devuelve los valores de las derivadas, tales que este modelo depende de  $y$  y de  $t$ . Ejemplo:  $dydt = model(y, t)$
- $y_0$ : Condiciones iniciales.
- `t`: Puntos en los que la solución será mostrada.

### 2.2 Marco teórico

[4] Un oscilador armónico es un sistema cuyo desplazamiento describiendo oscilaciones en torno a una posición estable a la cual quiere volver. Según la ley de Hooke, la fuerza descrita por un resorte sobre una masa es la forma

$$F = kx \tag{1}$$

Tal que:

- $k$  es la constante elástica del resorte.
- $x$  es la distancia entre la posición de equilibrio y la masa.
- $m$  es la masa.

De acuerdo a la segunda ley de Newton, se sabe

$$F = ma \quad (2)$$

Reemplazando (1) en (2) y escribiéndolas como derivadas de la posición respecto al tiempo

$$m\ddot{x} = -kx \quad (3)$$

Si se añade la fricción, (3) se actualiza a

$$m\ddot{x} = -kx - b\dot{x} \quad (4)$$

Tal que  $b$  es el coeficiente de fricción.

El sistema modelado en esta actividad es un poco más complejo, pero versa sobre la misma base, solo que se adapte al haber otro resorte unido por otro resorte entre ellos (además del que uno a la otra masa a la "pared").

## 2.3 Metodología

Habiendo entendido las bases del texto de [1] Richard Fitzpatrick y leído el código mostrado en [2] ScyPy Cookbook, se escribió el código y se añadieron distintas condiciones iniciales para adaptarlo a nuestro problema. Las condiciones iniciales fueron las siguientes:

1.  $m_1 = 1.0$

2.  $m_2 = 1.0$

3.  $k_1 = 1.0$

4.  $k_2 = 1.0$

5.  $x_3 = 1.0$

6.  $b_1 = 0.0$

7.  $b_2 = 0.0$

8.  $L_1 = 1.0$

9.  $L_2 = 1.0$

10.  $x_1 = 1.0$

11.  $x_2 = 0.0$

12.  $y_1 = 0.0$

13.  $y_2 = 0.0$

Tal que

- $m_i$  es la masa de cada bloque.
- $k_i$  es la constante de cada resorte.
- $b_i$  es el coeficiente de fricción.
- $L_i$  es la longitud del resorte.
- $x_i$  es la posición horizontal inicial.
- $y_i$  es la posición vertical inicial.

## 2.4 Resultados

En base a la hoja de datos, se hizo la gráfica que se puede ver a continuación:

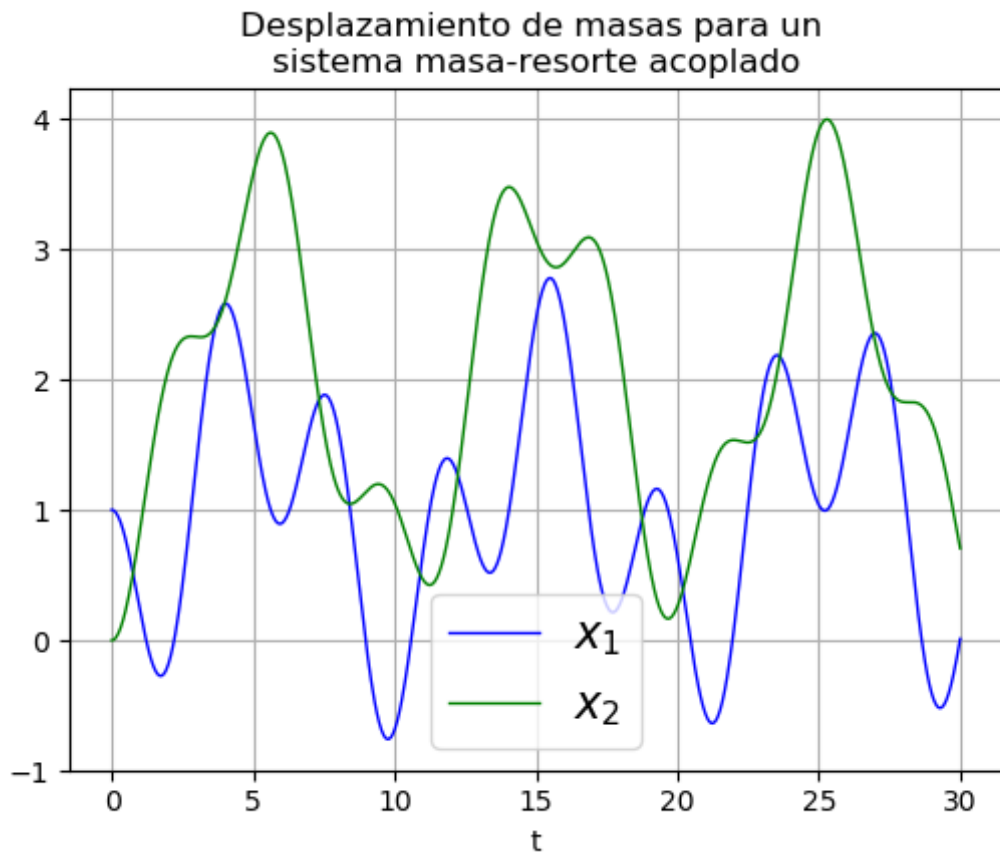


Figura 3:  $x_i$  vs  $t$

## 3 Conclusión

En resumidas palabras, Python con todas sus librerías facilitan muchísimo la modelación de sistemas descritos por ecuaciones diferenciales que antes podían romper la cabeza de tantos estudiantes.

Fue muy gratificante esta actividad, en lo personal me inspiró a buscar más modelaciones e intentarlas hacer por mí mismo para experimentar.

## Riferimenti bibliografici

- [1] Richard Fitzpatrick, Universidad de Texas, "Two Spring-Coupled Masses", 08/04/2013.  
<https://farside.ph.utexas.edu/teaching/315/Waves/node18.html>
- [2] ScyPy Cookbook, "Coupled spring-mass system". <https://scipy-cookbook.readthedocs.io/items/CoupledSpringMassSystem.html>
- [3] ScyPy, "scipy.integrate.odeint". <https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.odeint.html>
- [4] "Modelado con ecuaciones diferenciales de orden superior." <https://wilfridomtz.files.wordpress.com/2014/09/capitulo-5.pdf>
- [5] SciPy. <https://www.scipy.org/>
- [6] Dynamics and control, Brigham Young University, "Solve Differential Equations With Odeint". <https://apmonitor.com/pdc/index.php/Main/SolveDifferentialEquations>