Universidad de Sonora

Departamento de ciencias exactas

Física Computacional

Sistema de resortes acoplados

Alumno:
Martha Anahí Iñiguez Beltrán

214202804

20 de Marzo del 2018

Contents

1 Introducción

En la siguiente actividad nos centraremos en la modelación de fenómenos físicos con el lenguaje de programación de python. Para el desarrollo de la actividad usaremos como ejemplo el caso de un sistema de resortes acoplados, el cual estará adaptado al problema expuesto en el artículo de Temple Fay y Sarah Graham sobre dos resortes acoplados.

A continuación haremos una breve síntesis del artículo antes mencionado para exponer de manera más completa el desarrollo de nuestro código.

2 Síntesis

El siguiente problema es el de dos resortes con dos pesos unidos en serie y colgando de un techo. Suponiendo que los resortes se comportan de acuerdo a la Ley de Hooke, este problema, con dos grados de libertad está modelado por un par de ecuaciones diferenciales lineales acopladas de segundo orden. Al diferenciar y sustituir una ecuación en la otra, se puede mostrar que el movimiento de cada peso es determinado por una ecuación diferencial lineal de cuarto orden.

Podemos observar, mediante diversos ejemplos, que pueden surgir movimientos interesantes y diversos al aplicar una pequeña no linealidad como lo es un forzamiento o amortiguamiento en el sistema.

2.1 Modelo del Resorte Acoplado

El modelo consiste en dos resortes y dos pesos. Una primavera, teniendo primavera constante k1, se adjunta al techo y un peso de masa m1 se adjunta a la extremo inferior de esta primavera. A este peso, se le agrega un segundo resorte con resorte constante k2. En la parte inferior de este segundo muelle, se adjunta un peso de masa m2 y todo el sistema aparece como se ilustra en la figura 1.

Permitiendo que el sistema descanse en equilibrio, medimos el desplazamiento del centro de masa de cada peso del equilibrio, como una función del tiempo, y denotan estas medidas por x1(t) y x2(t), respectivamente.

De aui tomaremos cuatro ejemplos donde consideramos que las masas son iguales y el valor de ellas es 1. Para cada ejemplo utilizaremos los siguientes parámetros:

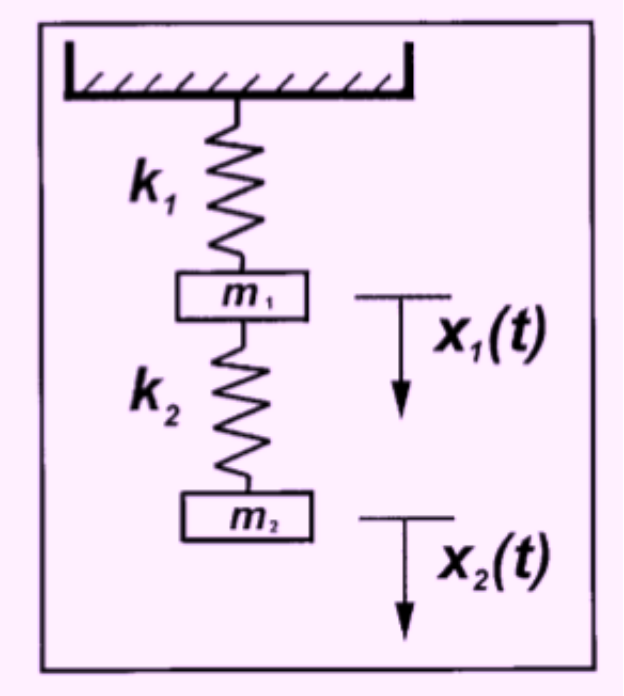

Figure 1: Sistema de Dos Resortes

Ejemplo 2.1:
k1 = 6, k2 = 4.
Condiciones iniciales (x1(0), v1(0), x2(0), v2(0))= (1,0,2,0)

Ejemplo 2.2:
k1 = 6, k2 = 4.
Condiciones iniciales (x1(0), v1(0), x2(0), v2(0))= (-2,0,1,0)

Ejemplo 2.3:
k1 = 0.4, k2 = 1.808.
Condiciones iniciales (x1(0), v1(0), x2(0), v2(0))= (1/2,0,-1/2,7/10)

Ejemplo 2.4:
k1 = 0.4, k2 = 1.808, f1 = 0.1, f2 = 0.2.
Condiciones iniciales (x1(0), v1(0), x2(0), v2(0))= (1,1/2,2,1/2)

En cada uno de los ejemplos, v1 y v2 se refiere a las velocidades respectivas a cada sistema masa-resorte. f1 y f2 del ejemplo 2.4 son las fricciones.

3 Código

En ésta parte analizaremos lo que es nuestro código base, con el cual nos guiamos para interpretar cada uno de los ejemplos anteriormente expuestos.

3.1 Código Base

Empezamos el código definiendo lo que serán los vectores, los cuales serán distintos entre ellos por los valores y las ecuaciones diferenciales que los componen:

```
def vectorfield(w, t, p):
    """
    Defines the differential equations for the coupled spring-mass system.

    Arguments:
        w :  vector of the state variables:
                  w = [x1,y1,x2,y2]
        t :  time
        p :  vector of the parameters:
                  p = [m1,m2,k1,k2,L1,L2,b1,b2]
    """
    x1, y1, x2, y2 = w
    m1, m2, k1, k2, L1, L2, b1, b2 = p

    # Create f = (x1',y1',x2',y2'):
    f = [y1,
         (-b1 * y1 - k1 * (x1 - L1) + k2 * (x2 - x1 - L2)) / m1,
         y2,
         (-b2 * y2 - k2 * (x2 - x1 - L2)) / m2]
    return f
```

ahora usaremos la función ODEINT para resolver las ecuaciones diferenciales definidas por el campo vectorial:

```
from scipy.integrate import odeint

# Valor de los parámetros
# Masa:
m1 = 1.0
```

```
m2 = 1.5
# Constante del resorte:
k1 = 8.0
k2 = 40.0
# Longitudes naturales:
L1 = 0.5
L2 = 1.0
# Coeficientes de fricción:
b1 = 0.8
b2 = 0.5

# Condiciones iniciales:
# x1 y x2 son los desplazamientos iniciales. y1 y y2 son las velocidades iniciales
x1 = 0.5
y1 = 0.0
x2 = 2.25
y2 = 0.0

# Parametros solucionadores ODE
abserr = 1.0e-8
relerr = 1.0e-6
stoptime = 10.0
numpoints = 250

# Cree las muestras de tiempo para la salida del solucionador ODE.

t = [stoptime * float(i) / (numpoints - 1) for i in range(numpoints)]

# Empaca los parametros y las condiciones iniciales:
p = [m1, m2, k1, k2, L1, L2, b1, b2]
w0 = [x1, y1, x2, y2]

# Llama al solucionador ODE:
wsol = odeint(vectorfield, w0, t, args=(p,),
              atol=abserr, rtol=relerr)

with open('two_springs.dat', 'w') as f:
    # Print & save the solution.
    for t1, w1 in zip(t, wsol):
        print (t1, w1[0], w1[1], w1[2], w1[3], file=f)
```

A continuación daremos valores a cada uno de los parámetros así como los valores de los errores al momento de hacer cálculos. Con la función ODEINT resolvemos las ecuaciones diferenciales de los vectores definidos al inicio y guardamos los resultados en un archivo de datos el cual nombraremos two-springs.dat.

Por último, en ésta misma sección del código, lo que haremos es graficar los datos guardados en el archivo el cual nos dará los valores de los desplazamientos respecto al tiempo, como se muestra en la figura 2:

```
from numpy import loadtxt
from pylab import figure, plot, xlabel, grid, hold, legend, title, savefig
from matplotlib.font_manager import FontProperties
%matplotlib inline

t, x1, xy, x2, y2 = loadtxt('two_springs.dat', unpack=True)

figure(1, figsize=(6, 4.5))

xlabel('t')
grid(True)
#hold(True)
lw = 1

plot(t, x1, 'b', linewidth=lw)
plot(t, x2, 'g', linewidth=lw)

legend((r'$x_1$', r'$x_2$'), prop=FontProperties(size=16))
title('Mass Displacements for the\nCoupled Spring-Mass System')
savefig('two_springs.png', dpi=100)
```

4 Resultados

A continuación se muestran lás gráficas obtenidas al resololver cada uno de los ejemplos mostrados en la sección 2.1 de éste reporte:

Ejemplo 2.1:

Ejemplo 2.2

Ejemplo 2.3

Ejemplo 2.4

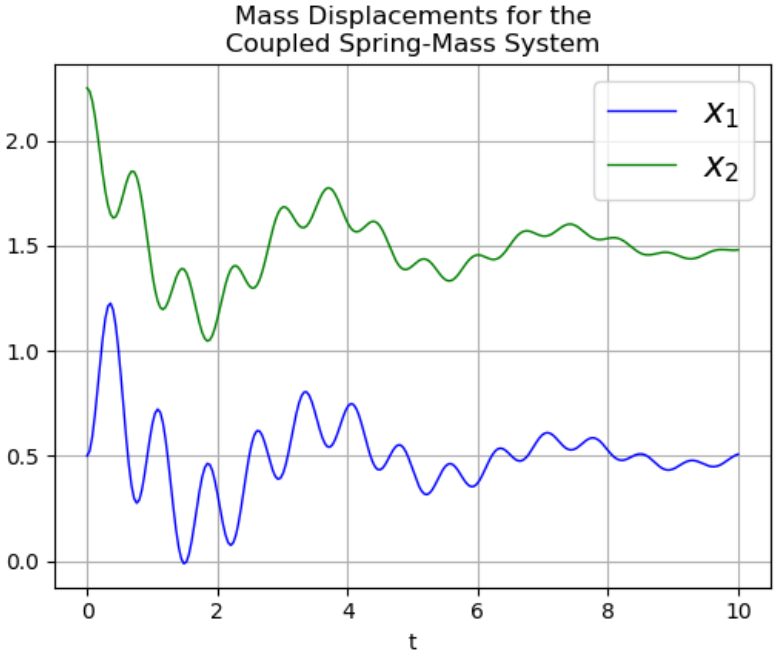

Figure 2: Desplazamiento del sistema de resortes acoplados

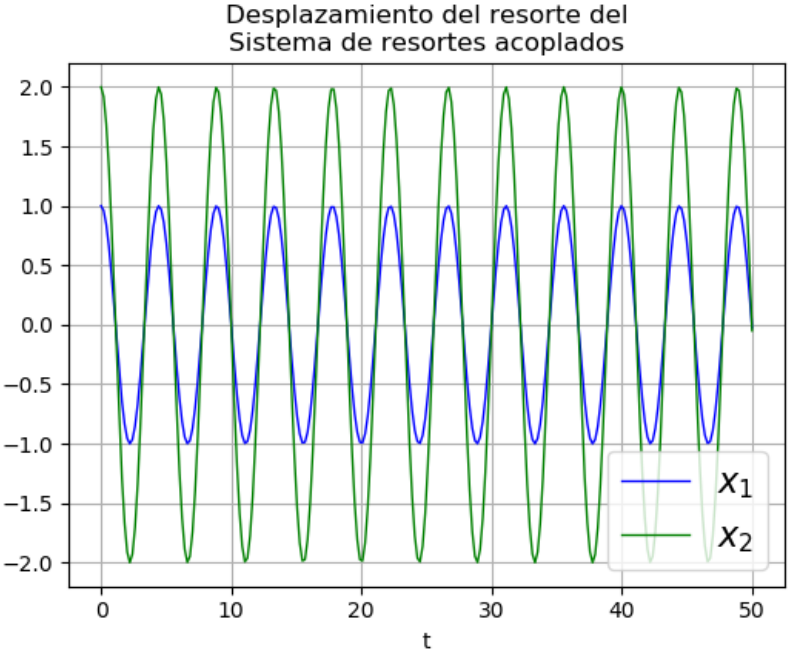

Figure 3: Desplazamiento del Sistema de Dos Resortes

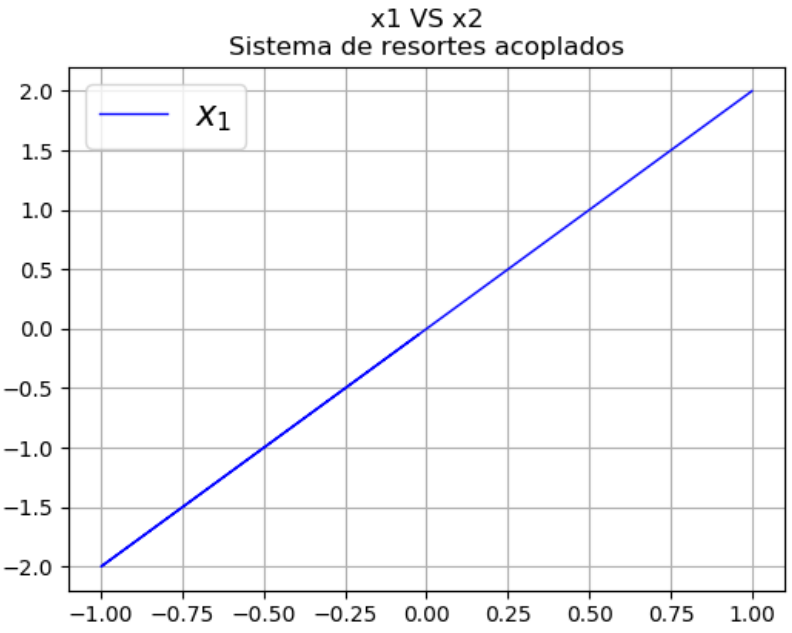

Figure 4: x1 vs x2

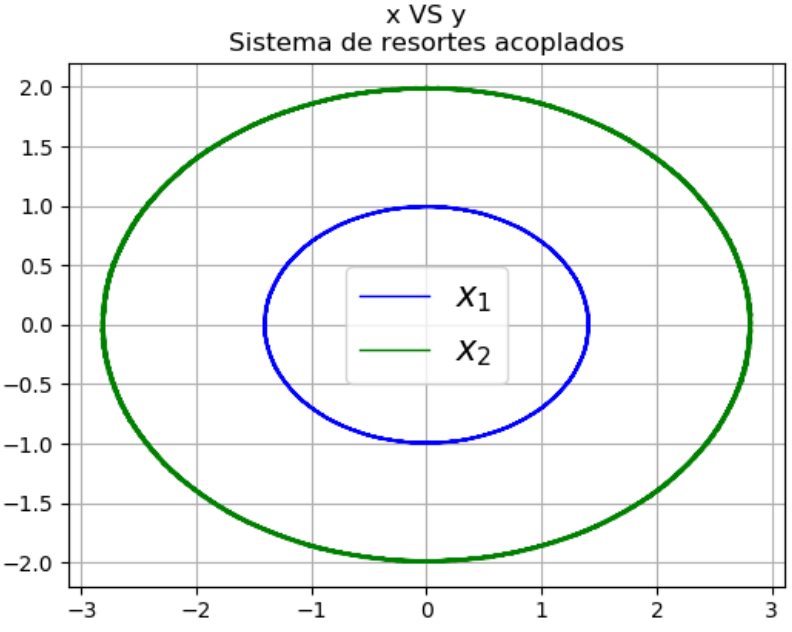

Figure 5: x vs y

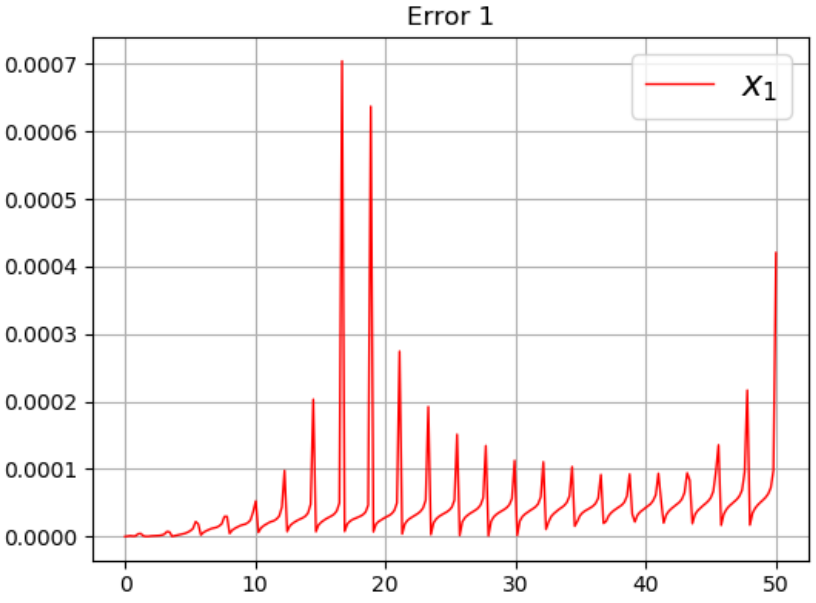

Figure 6: Error de la masa 1

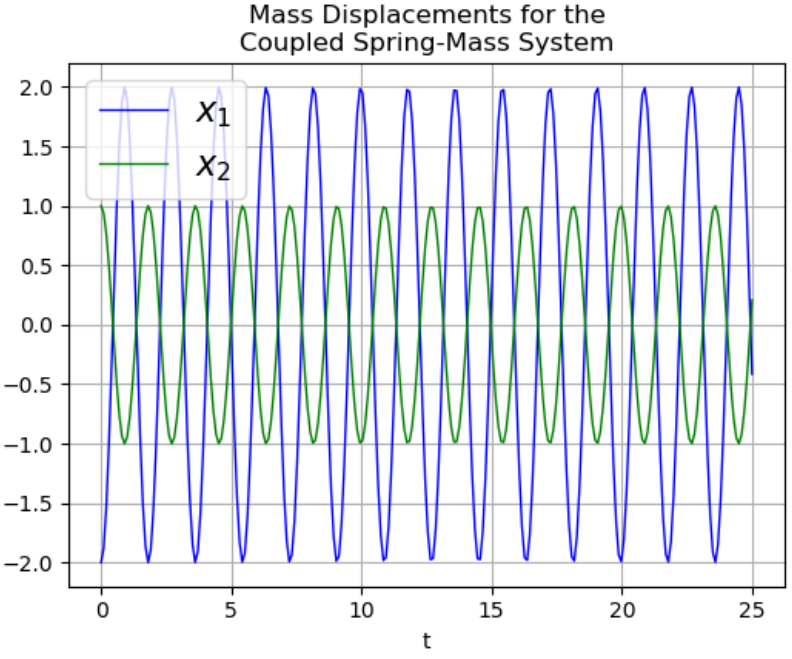

Figure 7: Desplazamiento del Sistema de Dos Resortes

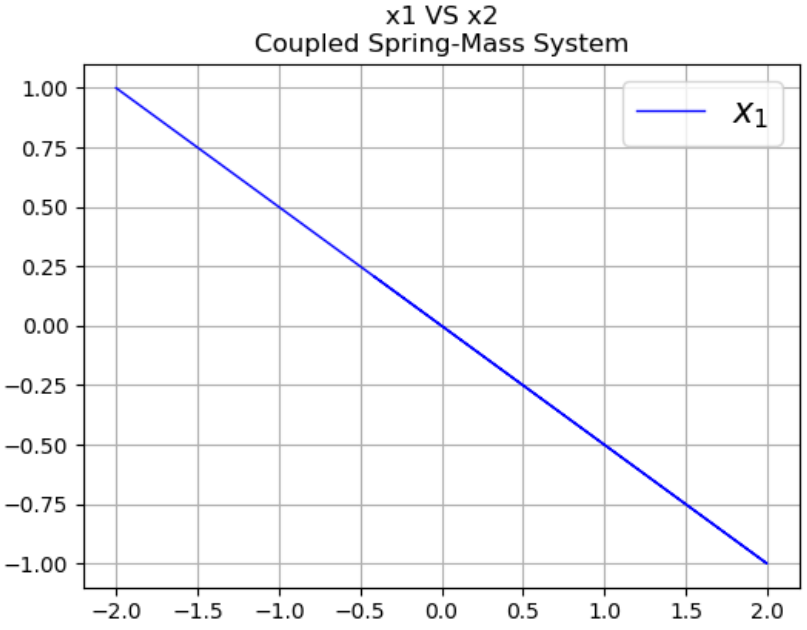

Figure 8: x1 vs x2

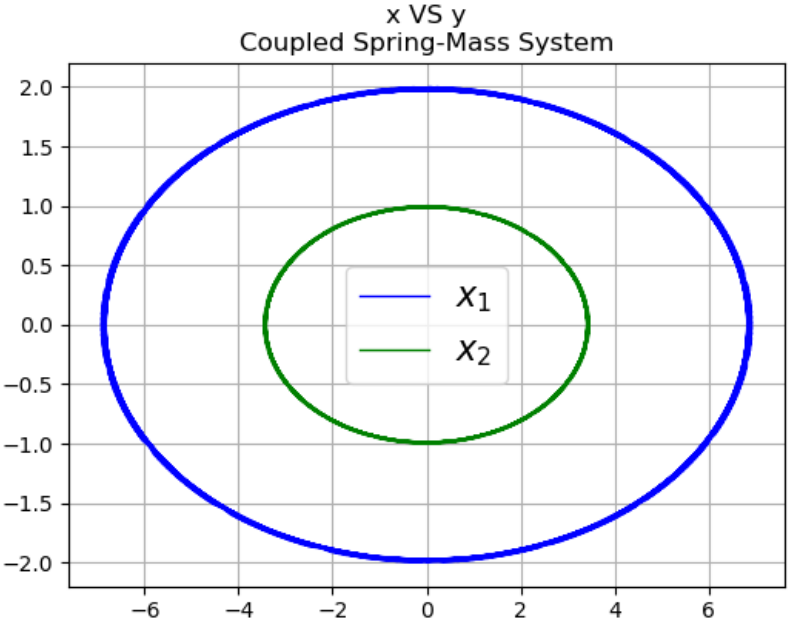

Figure 9: x vs y

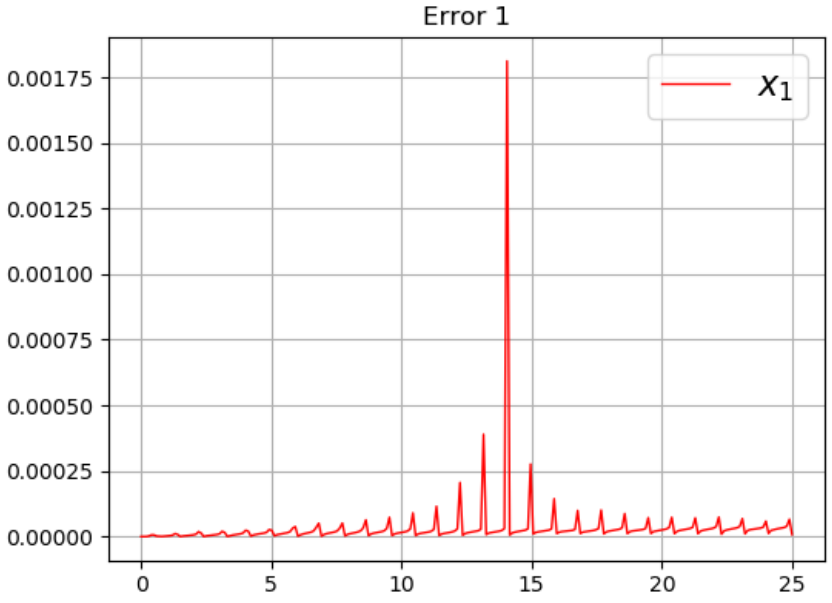

Figure 10: Error de la masa 1

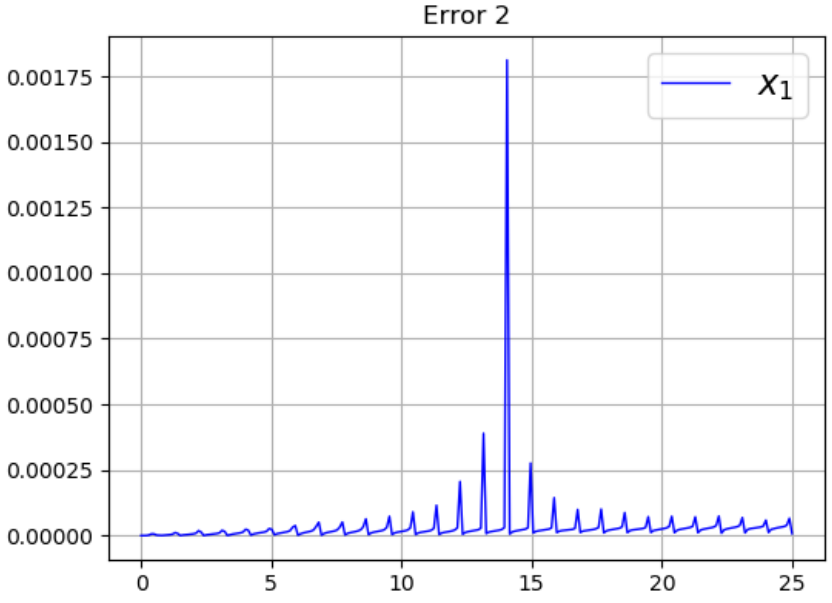

Figure 11: Error de la masa 2

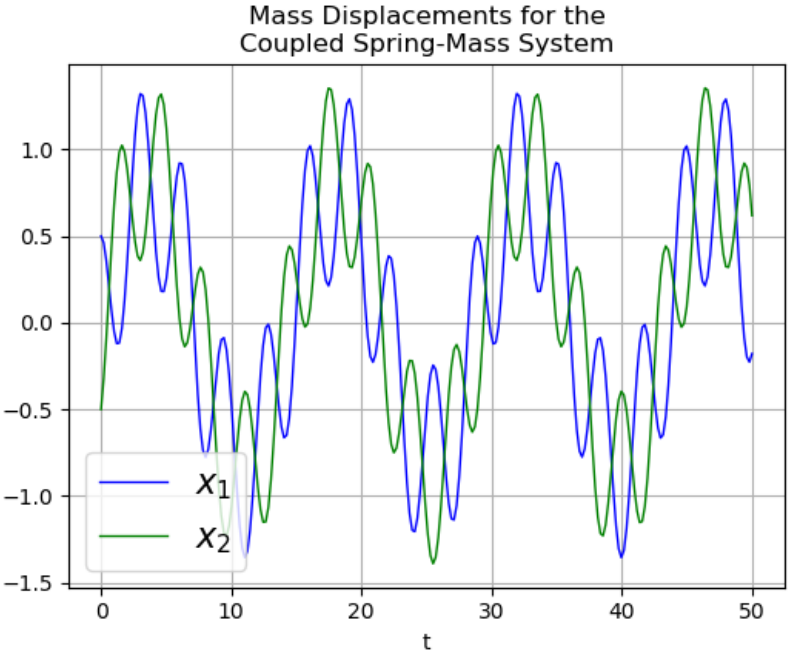

Figure 12: Velocidad contra posicion x1

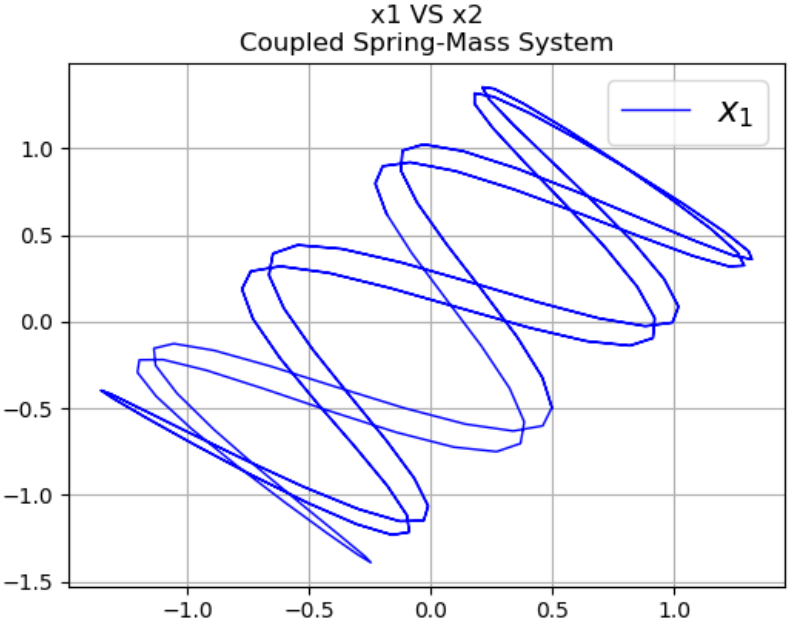

Figure 13: Velocidad contra posicion x2

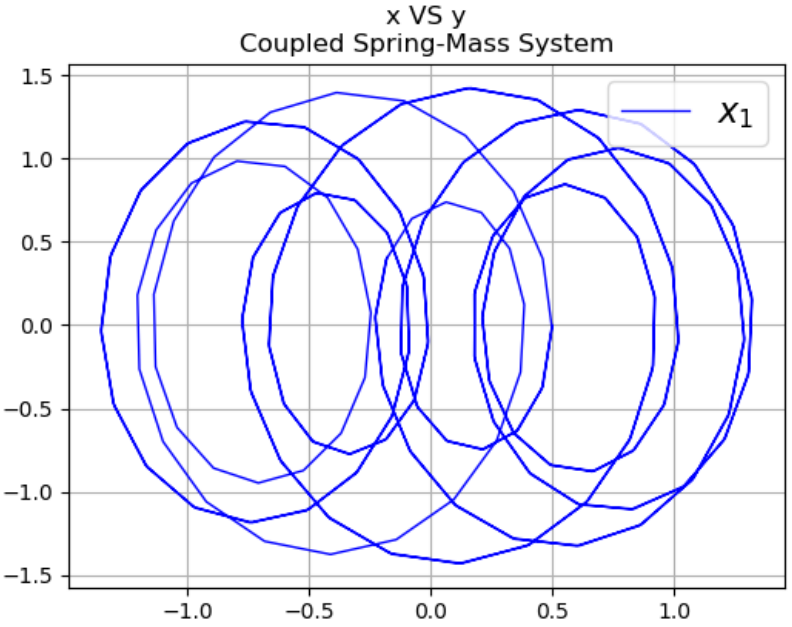

Figure 14: x1 y x2 contra tiempo

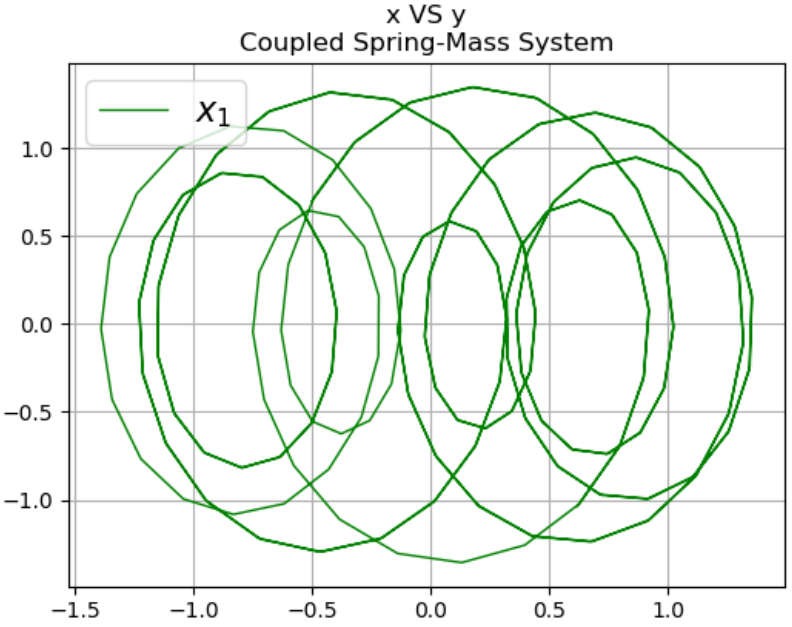

Figure 15: grafica x2 contra x1

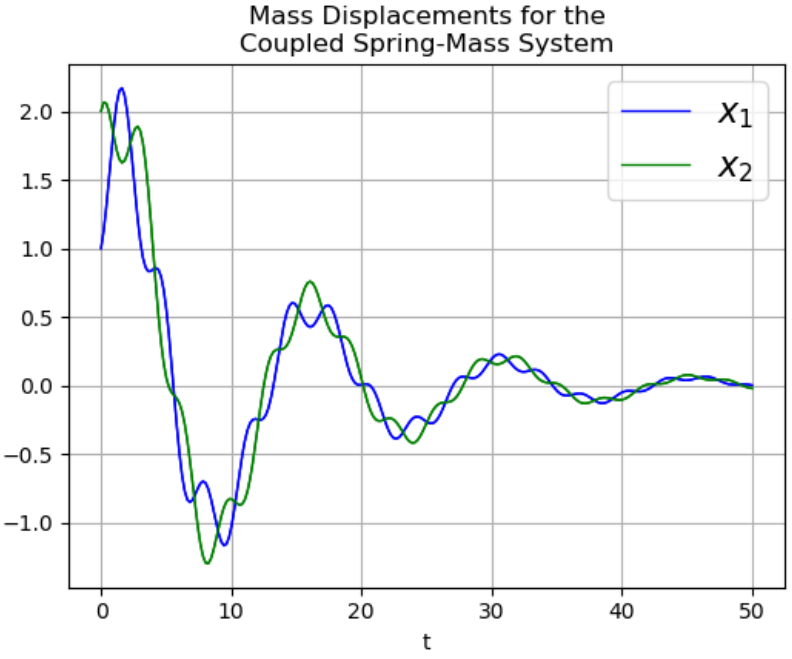

Figure 16: x1 y x2 contra tiempo

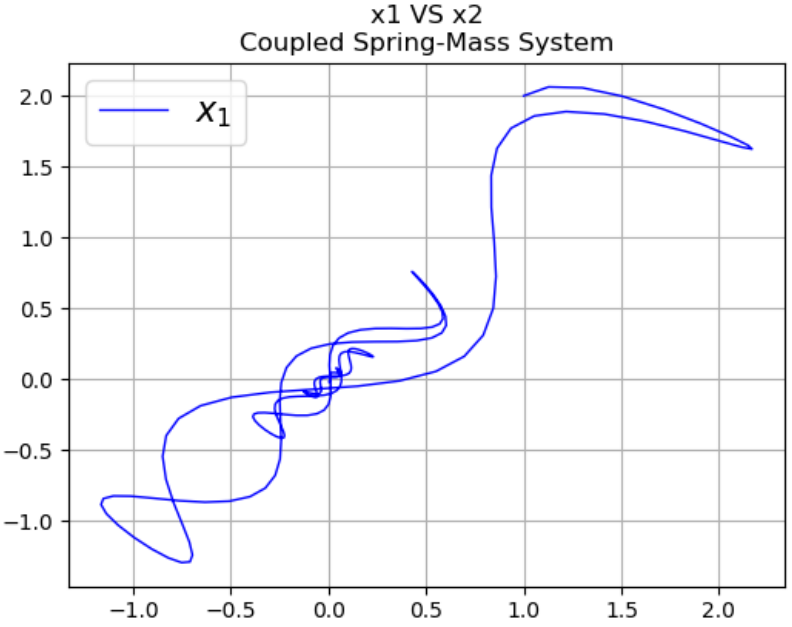

Figure 17: grafica x2 contra x1

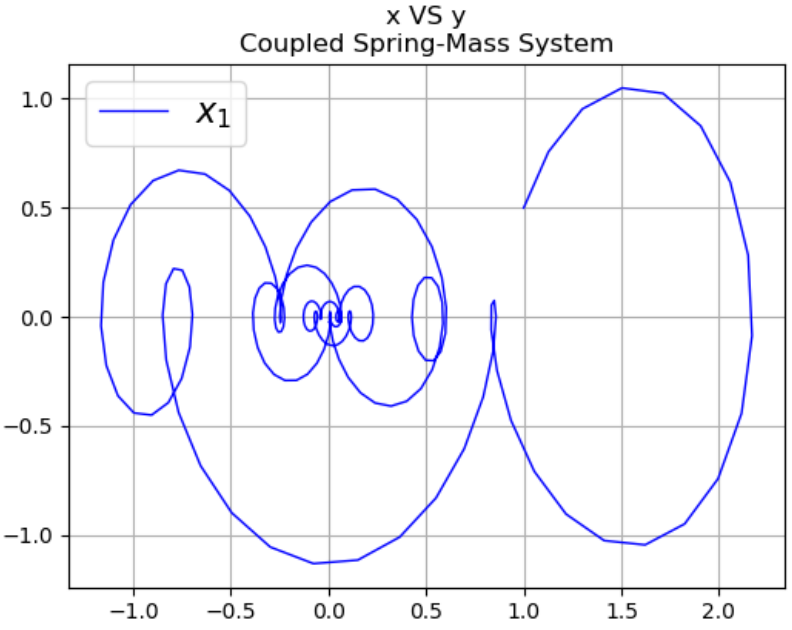

Figure 18: x1: x vs y

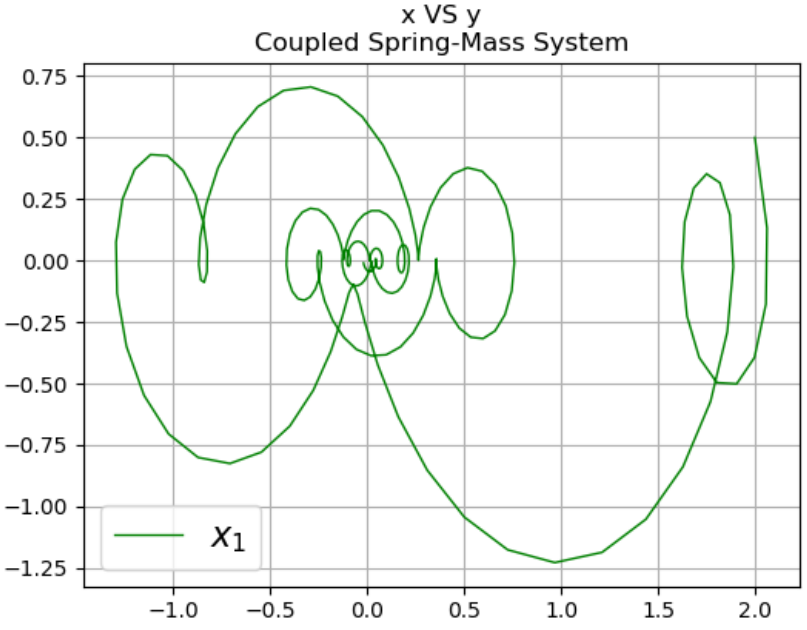

Figure 19: x2: x vs y

5 Conclusión

El uso de Jupyter lab para el desarrollo de esta actividad fue de gran apoyo ya que se puede tener un mejor orden del código así como varias pestañas para recrear el mismo código con distintos valores.

6 Bibliografía

- H. FAY, T., & DUNCAN GRAHAM, S. (2003). Coupled Spring ecuations (pp. 65-79). Int. J. Math. Educ. Sci. Technol. Recuperado el 20 de Marzo del 2018 desde http://math.oregonstate.edu/ gibsonn/Teaching/MTH323-010S15/Supplements/coupled_spring.pdf

7 Apéndice

¿En general te pareció interesante esta actividad de modelación matemática? ¿Qué te gustó mas? ¿Qué no te gustó?

Me pareció interesante la actividad por la parte de la simulación de los resortes acoplados así como en el uso de Jupyter lab.

La cantidad de material te pareció ¿bien?, ¿suficiente?, ¿demasiado?

Suficiente para desarrollar la actividad en diferentes ejemplos.

¿Cuál es tu primera impresión de Jupyter Lab?

Es muy ordenado, me gusta el uso de pestañas para no confundir los ejemplos vistos.

Respecto al uso de funciones de SciPy, ¿ya habías visto integración numérica en tus cursos anteriores? ¿Cuál es tu experiencia?.

Nunca había utilizado integración númerica en lenguajes de programación. La sintaxis es sencilla y entendible.

El tema de sistema de masas acopladas con resortes, ¿ya lo habías resuelto en tu curso de Mecánica 2?

Si.

¿Qué le quitarías o agregarías a esta actividad para hacerla más interesante y divertida?

En mi opinion, la actividad así está completa.