

# Reporte de Actividad 6: Sistema de Resortes Acoplados

Jesús Antonio González Espinosa

Física Computacional 1

Martes, 20 de Marzo del 2018

## 1 Introducción

Para esta actividad hemos trabajado con modelación de fenómenos físicos en python; específicamente en el de oscilaciones, presentando varios ejemplos de sistemas de resortes acoplados.

En este reporte se presenta la síntesis de los primeros dos subtemas del documento "Coupled spring equations" de TEMPLE H. FAY, el cual nos aporta la teoría necesaria para obtener las bases y comprender los conceptos importantes para poder efectuar la actividad. Asimismo, el documento dio las propiedades y condiciones iniciales de las gráficas presentadas en este reporte. Naturalmente, se presenta el cierre al reporte, incluyendo bibliografía, conclusiones y el apéndice perteneciente a la actividad.

## 2 Síntesis

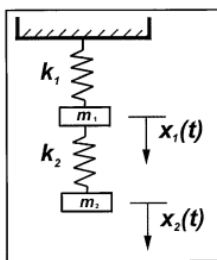
### 2.1 Introducción

En éste artículo se investiga el problema de los dos resortes y dos pesas adjuntas en serie. Bajo la suposición de que las fuerzas de restauración se comportan de acuerdo a la Ley de Hooke, este problema de dos grados de libertad es modelado por un par de ecuaciones diferenciales lineales de segundo orden que al sustituir una en la otra, el movimiento de cada pesa puede ser determinado por una ecuación diferencial lineal de cuarto orden.

Lo que hace interesante a este problema, es que podemos investigar los movimientos de las masas, para ver si están sincronizadas u opuestas entre ellas. También se puede ver que al agregarle factores no lineales, pueden surgir movimientos interesantes. Al modificar los parámetros podemos observar gráficamente los cambios en la periodicidad, la amplitud, la fase, la sensibilidad a las condiciones iniciales, entre otras propiedades.

### 2.2 El modelo de resorte acoplado

El modelo consiste en dos resortes y dos pesas. Un resorte tiene la constante elástica  $k_1$  y está adjunto al techo de un extremo y del otro tiene una pesa de masa  $m_1$ . La misma pesa tiene adjunto un segundo resorte de constante elástica  $k_2$  y al final de éste, una pesa de masa  $m_2$ . Al dejar el sistema llegar al equilibrio, medimos el desplazamiento de cada masa, en función del tiempo, y llamaremos estas medidas como:  $x_1(t)$  y  $x_2(t)$ .



### 2.2.1 Asumiendo la Ley de Hooke

Asumiendo que hay pequeñas oscilaciones, las fuerzas restauradoras son de la forma  $-k_1 l_1$  y  $-k_2 l_2$ , donde  $l_1$  y  $l_2$  son las elongaciones o compresiones de los dos resortes. Como la masa superior está unida a ambos resortes, ésta siente las dos fuerzas restauradoras actuando sobre ella, mientras que la segunda masa solo siente la fuerza restauradora del segundo resorte. Asumiendo que no hay fuerzas de amortiguamiento, la Ley de Newton implica que las ecuaciones que representan los movimientos de las pesas son:

$$\begin{aligned}m_1 \ddot{x}_1 &= -k_1 x_1 - k_2 (x_1 - x_2) \\ m_1 \ddot{x}_2 &= -k_2 (x_1 - x_2)\end{aligned}$$

Entonces para encontrar una ecuación para  $x_1$  que no involucre a  $x_2$ , resolvemos la primera ecuación diferencial para  $x_2$ , y la sustituimos en la primera, donde después de simplificar, obtenemos:

$$m_1 m_2 x_1^{(4)} + (m_2 k_1 + k_2 (m_1 + m_2)) \ddot{x}_1 + k_1 k_2 x_1 = 0$$

Ahora, para encontrar una ecuación que involucre de  $x_2$ , resolvemos la segunda ecuación diferencial para  $x_1$  y lo sustituimos en la primera ecuación, obteniendo:

$$m_1 m_2 x_2^{(4)} + (m_2 k_1 + k_2 (m_1 + m_2)) \ddot{x}_2 + k_1 k_2 x_2 = 0$$

Para poder resolver las ecuaciones, solo es necesario las posiciones iniciales y las velocidades iniciales.

### 2.2.2 Algunos ejemplos con masas idénticas

Consideremos el modelo con dos pesas con la misma masa,  $m_1 = m_2 = 1$ , ignorando el amortiguamiento y sin fuerzas externas.

**Ejemplo 2.1:** Describe el movimiento para las constantes de resorte  $k_1 = 6$  y  $k_2 = 4$  con las condiciones iniciales  $(x_1(0), \dot{x}_1(0), x_2(0), \dot{x}_2(0)) = (1, 0, 2, 0)$ . Al resolverlo de manera analítica obtenemos:

$$\begin{aligned}x_1(t) &= \cos\sqrt{2}t \\ x_2(t) &= 2\cos\sqrt{2}t\end{aligned}$$

El movimiento es sincronizado, por lo tanto, las pesas se mueven en fase una con la otra, teniendo el mismo periodo, pero con amplitudes un poco diferentes.

Para poder graficar esto en Python, utilizamos un código proporcionado. Este código se utilizó durante toda la actividad, solo modificando los parámetros y condiciones iniciales; así como las variables de los ejes coordenados para presentar las diferentes gráficas. En este caso, el código para este ejemplo se ve así:

```
##### Ejemplo 2.1 #####
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
```

```

# Use ODEINT to solve the differential equations defined by the vector field
from scipy.integrate import odeint
import numpy as np
import math

# Parameter values
# Masses:
m1 = 1.0
m2 = 1.0
# Spring constants
k1 = 6.0
k2 = 4.0
# Natural lengths
L1 = 0.0
L2 = 0.0
# Friction coefficients
b1 = 0.0
b2 = 0.0

# Initial conditions
# x1 and x2 are the initial displacements; y1 and y2 are the initial velocities
x1 = 1.0
y1 = 0.0
x2 = 2.0
y2 = 0.0

# ODE solver parameters
abserr = 1.0e-8
relerr = 1.0e-6
stoptime = 50.0
numpoints = 250

# Create the time samples for the output of the ODE solver.
# I use a large number of points, only because I want to make
# a plot of the solution that looks nice.
t = [stoptime * float(i) / (numpoints - 1) for i in range(numpoints)]

# Pack up the parameters and initial conditions:
p = [m1, m2, k1, k2, L1, L2, b1, b2]
w0 = [x1, y1, x2, y2]

# Call the ODE solver.
wsol = odeint(vectorfield, w0, t, args=(p,),
              atol=abserr, rtol=relerr)

with open('ejemplo_2.1.dat', 'w') as f:
    # Print & save the solution.
    for ti, w1 in zip(t, wsol):
        print(ti, w1[0], w1[1], w1[2], w1[3], np.abs((w1[0]-(np.cos(np.sqrt(2)*ti)))/(np.cos(np.sqrt(2)*ti))))

```

```

# Plot the solution that was generated

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

t, x1, y1, x2, y2, e1, e2 = loadtxt('ejemplo_2.1.dat', unpack=True)

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

xlabel('t')
ylabel('x')
grid(True)
lw = 1

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

legend((r'$x_{1s}$', r'$x_{2s}$'), prop=FontProperties(size=16))
title('Ejemplo 2.1: Posición con respecto al Tiempo')
savefig('ejemplo_2.1.1.png', dpi=100)

```

Obteniendo las siguientes gráficas:

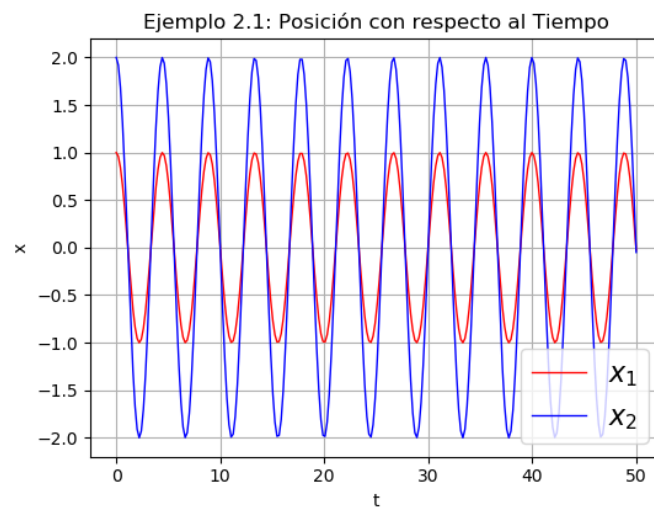
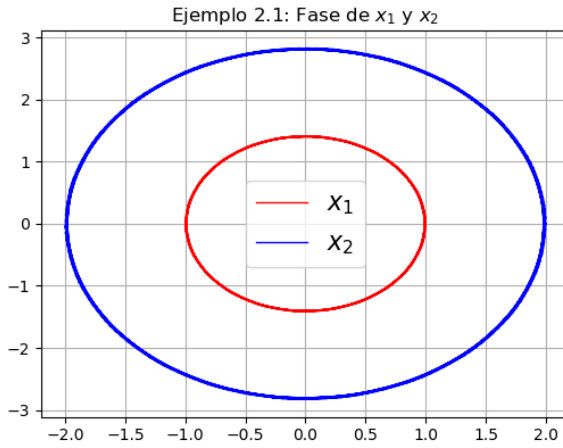
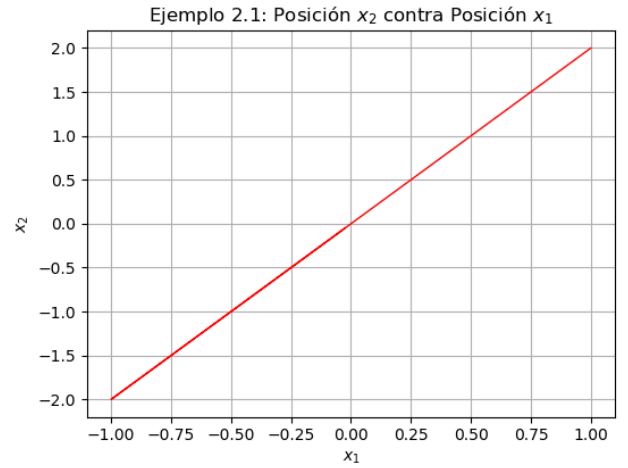


Figure 1: Posición con respecto al Tiempo de  $x_1$  y  $x_2$

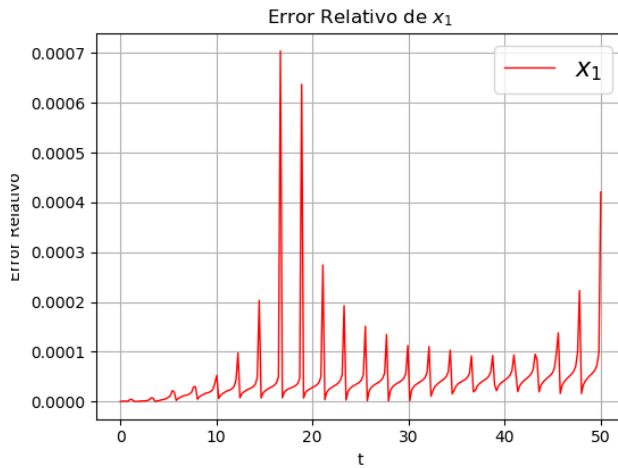


(a) Fase de  $x_1$  y  $x_2$

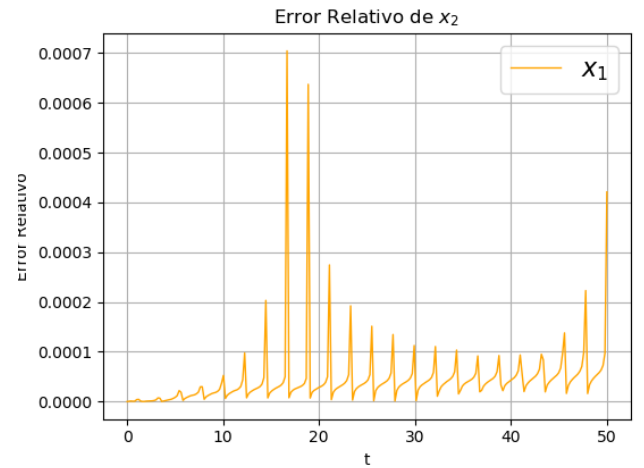


(b) Posición  $x_2$  contra  $x_1$

El error relativo de  $x_1$  y  $x_2$ :



(a) Error Relativo de  $x_1$



(b) Error Relativo de  $x_2$

**Ejemplo 2.2:** Describe el movimiento para las constantes de resorte  $k_1 = 6$  y  $k_2 = 4$  con las condiciones iniciales  $(x_1(0), \dot{x}_1(0), x_2(0), \dot{x}_2(0)) = (-2, 0, 1, 0)$ . Al resolverlo de manera analítica obtenemos:

$$\begin{aligned} x_1(t) &= -2\cos 2\sqrt{3}t \\ x_2(t) &= \cos 2\sqrt{3}t \end{aligned}$$

En este caso, cuando la primera pesa se mueve hacia arriba, la segunda se mueve hacia abajo; aún tienen el mismo periodo, pero ahora un desfase de  $180^\circ$

El código modificado para hacer tales gráficas resulta así:  
Obteniendo las siguientes gráficas:

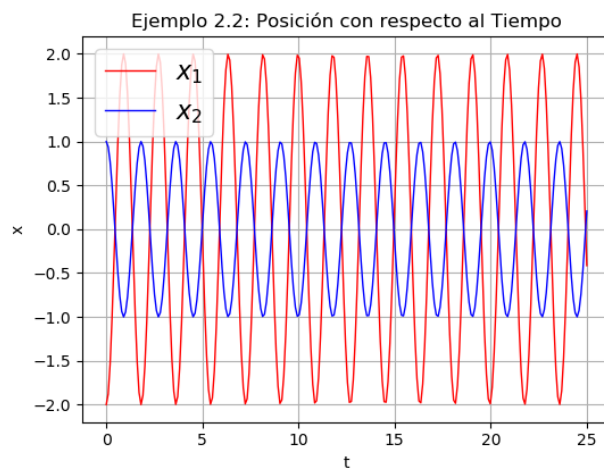
```

# Parameter values
# Masses:
m1 = 1.0
m2 = 1.0
# Spring constants
k1 = 6.0
k2 = 4.0
# Natural lengths
L1 = 0.0
L2 = 0.0
# Friction coefficients
b1 = 0.0
b2 = 0.0

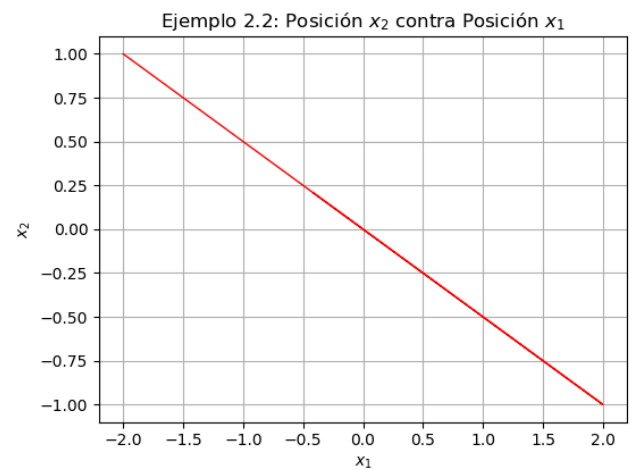
# Initial conditions
# x1 and x2 are the initial displacements; y1 and y2 are the initial velocities
x1 = -2.0
y1 = 0.0
x2 = 1.0
y2 = 0.0

# ODE solver parameters
abserr = 1.0e-8
relerr = 1.0e-6
stoptime = 25.0
numpoints = 250

```

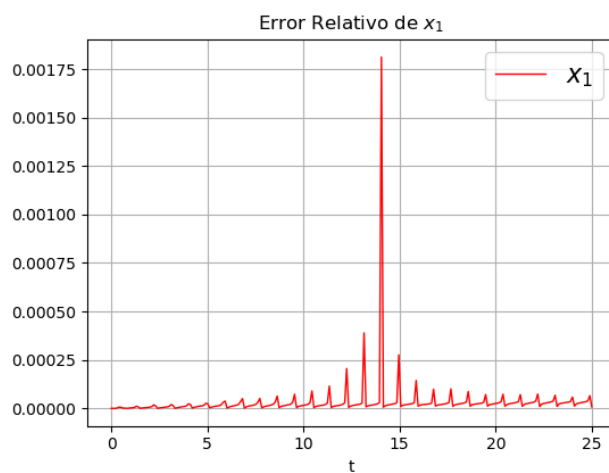


(a) Posición con respecto al Tiempo de  $x_1$  y  $x_2$

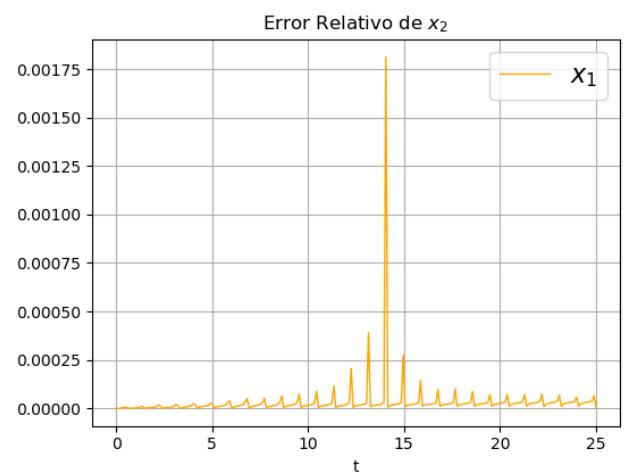


(b) Posición  $x_2$  contra  $x_1$

El error relativo de  $x_1$  y  $x_2$ :



(a) Error Relativo de  $x_1$



(b) Error Relativo de  $x_2$

**Ejemplo 2.3:** Describe el movimiento para las constantes de resorte  $k_1 = 0.4$  y  $k_2 = 1.808$  con las condiciones iniciales  $(x_1(0), \dot{x}_1(0), x_2(0), \dot{x}_2(0)) = (1/2, 0, -1/2, 7/10)$ .

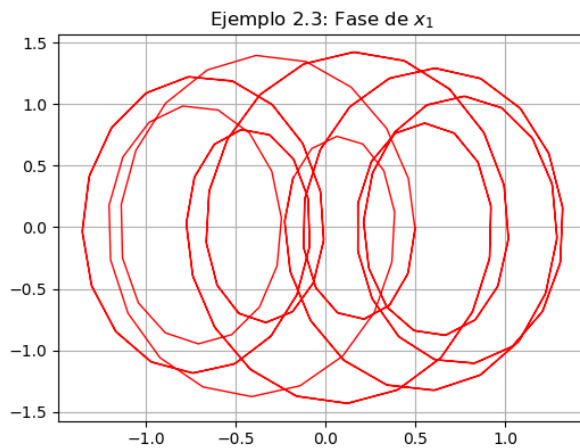
En este problema, los valores de las constantes del resorte determinan el periodo y por lo tanto, la frecuencia; mientras que las condiciones iniciales solo afectan la amplitud y la fase de las soluciones. El código modificado para hacer tales gráficas resulta así:

```
# Parameter values
# Masses:
m1 = 1.0
m2 = 1.0
# Spring constants
k1 = 0.4
k2 = 1.808
# Natural lengths
L1 = 0.0
L2 = 0.0
# Friction coefficients
b1 = 0.0
b2 = 0.0

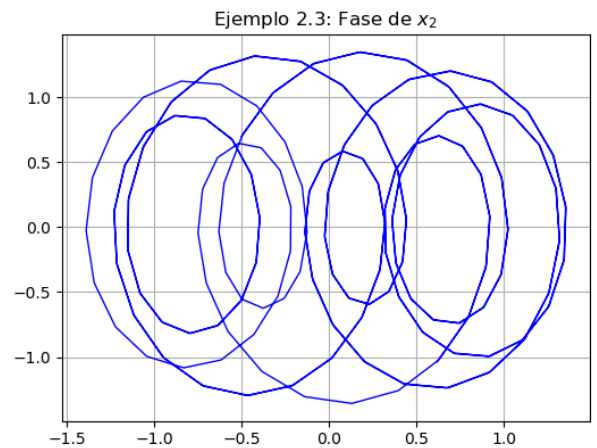
# Initial conditions
# x1 and x2 are the initial displacements; y1 and y2 are the initial velocities
x1 = 1.0/2.0
y1 = 0.0
x2 = -1.0/2.0
y2 = 7.0/10.0

# ODE solver parameters
abserr = 1.0e-8
relerr = 1.0e-6
stoptime = 50.0
numpoints = 250
```

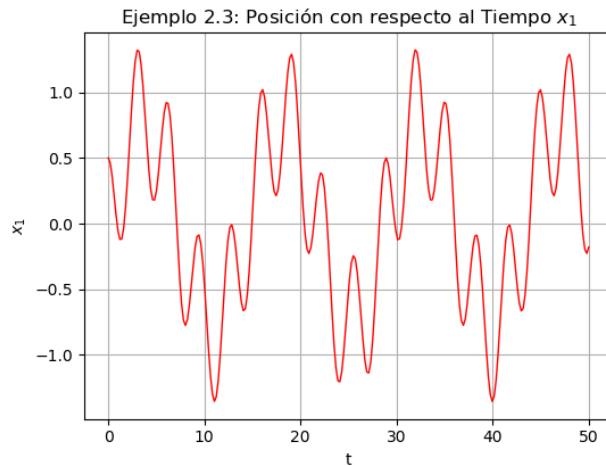
Al graficar las soluciones, podemos ver varios patrones interesantes:



(a) Fase para  $x_1$



(b) Fase para  $x_2$



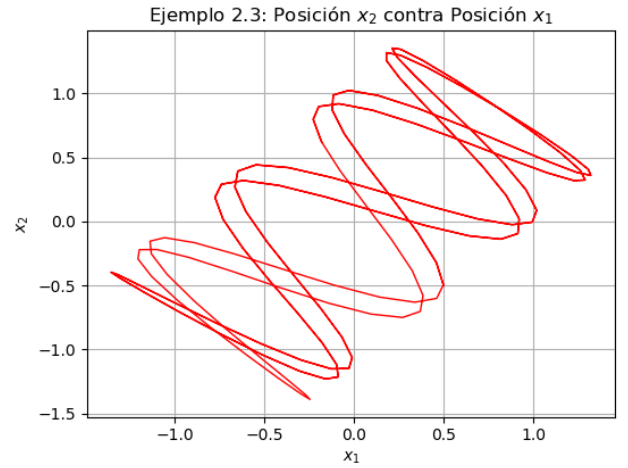
(c) Posición con respecto al tiempo de  $x_1$



(d) Posición con respecto al tiempo de  $x_2$



(a) Posición con respecto al tiempo de  $x_1$  y  $x_2$



(b) Posición  $x_2$  contra  $x_1$

### 2.2.3 Amortiguamiento

El tipo de amortiguamiento más común, son los de viscosidad, donde la fuerza amortiguadora es proporcional a la velocidad. El amortiguamiento de la primera masa depende solamente de su velocidad y no de la velocidad de la segunda masa y vice versa. Para modelar esto, agregamos los términos  $-\delta_1 \dot{x}_1$  a la primera ecuación y  $-\delta_2 \dot{x}_2$  a la segunda. Asumiendo que los coeficientes de amortiguamiento  $\delta_1$  y  $\delta_2$  son pequeños, el modelo se da por:

$$\begin{aligned} m_1 \ddot{x}_1 &= -\delta_1 \dot{x}_1 - k_1 x_1 - k_2 (x_1 - x_2) \\ m_2 \ddot{x}_2 &= -\delta_2 \dot{x}_2 - k_2 (x_2 - x_1) \end{aligned}$$

Ahora, para hacer una ecuación que solo dependa de  $x_2$ , podemos resolver la segunda ecuación para  $x_1$  y sustituirlo en la primera, obteniendo una ecuación de cuarto orden:

$$m_1 m_2 x_2^{(4)} + (m_1 \delta_1 + m_2 \delta_2) \ddot{x}_2 + (m_2 k_1 + k_2 (m_1 + m_2) + \delta_1 \delta_2) \dot{x}_2 + (k_1 \delta_2 + k_2 (\delta_1 + \delta_2)) x_2 + k_1 k_2 x_2 = 0$$

Podemos hacer el mismo procedimiento para que dependa de  $x_1$ . Con esto, obtenemos una ecuación diferencial lineal que representa el movimiento de ambas pesas.

**Ejemplo 2.4:** Asuma que  $m_1 = m_2 = 1$ . Describe el movimiento para las constantes de resorte  $k_1 = 0.4$  y  $k_2 = 1.808$ , los coeficientes de amortiguamiento  $\delta_1 = 0.1$  y  $\delta_2 = 0.2$  con las condiciones iniciales  $(x_1(0), \dot{x}_1(0), x_2(0), \dot{x}_2(0)) = (1, 1/2, 2, 1/2)$ .

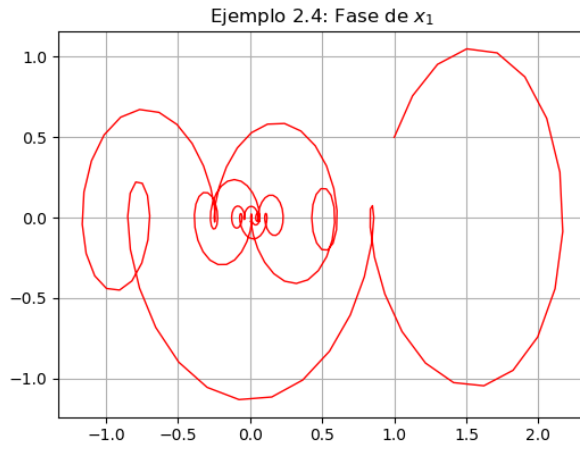
Se puede observar un patrón regular en donde el movimiento avanza con una disminución en su amplitud. El código para graficar las soluciones resulta así:

```
# Parameter values
# Masses:
m1 = 1.0
m2 = 1.0
# Spring constants
k1 = 0.4
k2 = 1.808
# Natural lengths
L1 = 0.0
L2 = 0.0
# Friction coefficients
b1 = 0.1
b2 = 0.2

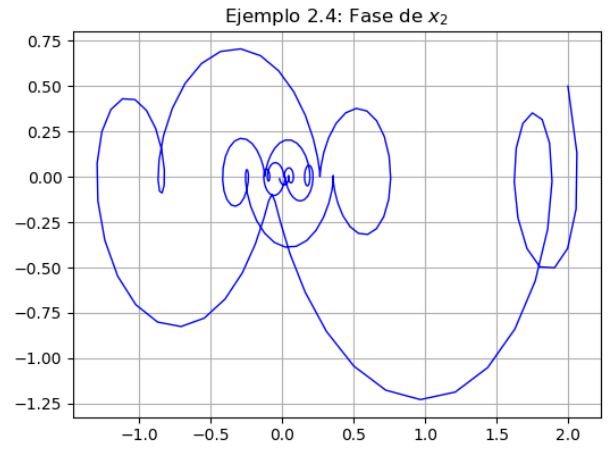
# Initial conditions
# x1 and x2 are the initial displacements; y1 and y2 are the initial velocities
x1 = 1.0
y1 = 1.0/2.0
x2 = 2.0
y2 = 1.00/2.00

# ODE solver parameters
abserr = 1.0e-8
relerr = 1.0e-6
stoptime = 50.0
numpoints = 250
```

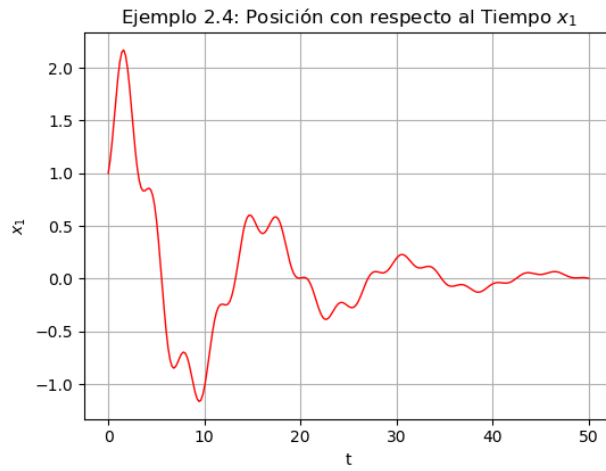
Resultando en las siguientes graficas



(a) Fase para  $x_1$



(b) Fase para  $x_2$



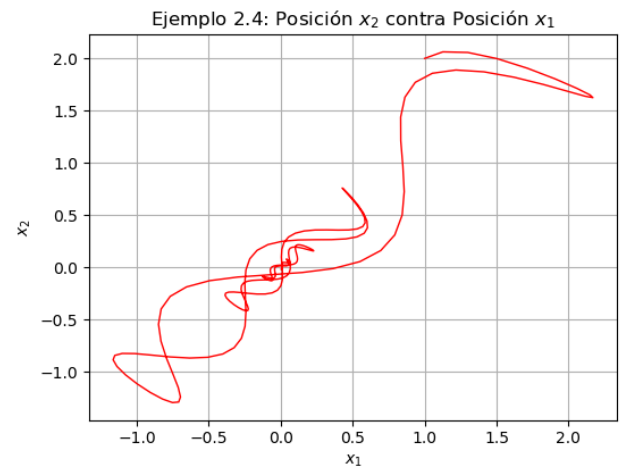
(c) Posición con respecto al tiempo de  $x_1$



(d) Posición con respecto al tiempo de  $x_2$



(e) Posición con respecto al tiempo de  $x_1$  y  $x_2$



(f) Posición  $x_2$  contra  $x_1$



### 3 Conclusión

Ha sido muy interesante trabajar con esta actividad, porque es la primera vez utilizando ecuaciones diferenciales en la programación, aunque el código ya estaba hecho, el ver como trabaja y entender como funciona fue algo nuevo. Además de que se unieron con conceptos vistos en cursos anteriores, lo que a mi parecer, lo hizo más llamativo. Por otra parte, Jupyter Lab ha sido de más agrado que Jupyter Notebook, por cosas como la interfase, hasta el explorador de archivos que incluye a la izquierda, lo hizo más agradable y fácil de utilizar. En pocas palabras, la actividad ha sido realizada con éxito y sin mucha dificultad, y se espera poder continuar trabajando con esta seguridad y constancia en las futuras actividades del curso.

### 4 Bibliografía

1. Fay, T., Graham, S. (2003) Coupled Spring Equations. Recuperado el 17 de Marzo del 2018 desde [http://math.oregonstate.edu/~gibsonn/Teaching/MTH323-010S15/Supplements/coupled\\_spring.pdf](http://math.oregonstate.edu/~gibsonn/Teaching/MTH323-010S15/Supplements/coupled_spring.pdf)

### 5 Apéndice

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

La actividad me pareció muy interesante, fue muy entretenido ver como pequeños cambios iban afectando las gráficas. Lo único que no me gustó fue que el código ya estaba totalmente hecho, hubiera estado más interesante si hubiera sido más como una plantilla y no que ya estuviera totalmente desarrollado.

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

Sí, estuvo muy bien y fácil de entender, ya que eran conceptos previamente estudiados en el curso de mecánica 2, aplicados en la programación.

3. **¿Cuál es tu primera impresión de Jupyter Lab?**

Parece una versión más avanzada de Jupyter Notebook, además de que tener los archivos a la mano en la sección de la izquierda, hace todo más fácil, por lo que fue muy agradable trabajar con esta herramienta.

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

Habíamos visto integración numérica antes en otros cursos que incluían programación en FORTRAN, pero nunca con Python.

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

Sí, pero con mucha ayuda del profesor y sin entender muy bien que sucedía, pues aún no veíamos los temas necesarios en el curso de Ecuaciones Diferenciales.

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

La práctica tal y como esta estuvo bien, no fue ni muy tediosa, ni complicado, pero tampoco muy fácil. En pocas palabras, fue una actividad muy amena.