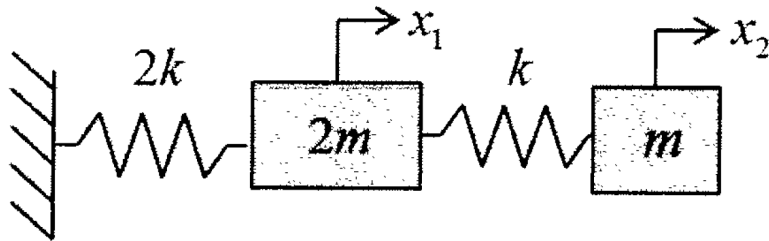


# Reporte de Actividad 6

Valenzuela Terán Jonás

March 25, 2018



## 1 Introducción

El objetivo de la actividad es aprender a modelar un fenómeno físico, en este caso, como ejemplo el sistema de doble resorte, a través de una simulación, hecha en el entorno que ha sido trabajado en clase últimamente, python en jupyter, pero esta vez en jupyter lab, que da a disposición un poco más de herramientas. Después, realizar distintas gráficas como las realizadas anteriormente para visualizar su comportamiento, y ver su confiabilidad mostrando el error relativo.

## 2 Resumen de artículo: Coupled spring equations, pt. 1 y 2

### 2.1 Introducción

La perspectiva general con la que se toma a las ecuaciones diferenciales esta cambiando mucho, de realizar técnicas para encontrar soluciones analíticas a visualizar y resolver numéricamente estas con las herramientas disponibles.

En el artículo, se trabaja con el problema de 2 resortes y 2 masas conectadas en serie, suponiendo que obedecen la Ley de Hooke. Esto introduce 2 ecuaciones diferencial lineal de segundo orden. Hacer esto permite tener una interpretación física clara de la fase, amplitud, periodicidad, y la sensibilidad del sistema a las condiciones iniciales al introducir no linealidad.

### 2.2 El modelo del doble resorte

El modelo consiste en 2 resortes  $k_1$  y  $k_2$ , y 2 masas  $m_1$  y  $m_2$  conectados en serie.

#### 2.2.1 Asumiendo la Ley de Hooke

Suponiendo que cumple la Ley de Hooke en el sistema, las fuerzas restauradoras producto de las elongaciones son de la forma  $-k_1 l_1$  y  $-k_2 l_2$  ( $l$  siendo la elongación de cada resorte), la masa inferior solo siente la fuerza del resorte 2  $-k_2(x_2 - x_1)$ , donde  $x_2 - x_1 = l_2$ , mientras que la masa superior siente esta y la del resorte superior  $-k_1 x_1$ , donde  $x_1 = l_1$ . Escribimos las ecuaciones para las fuerzas de la forma:

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

Se tiene un par de ecuaciones diferenciales lineales de segundo orden, para obtener una ecuación de  $x_1$  sin involucrar a  $x_2$ , se despeja de la ecuación:

$$x_2 = -\frac{m_1 \ddot{x}_1}{k_2} + \frac{k_1 + k_2}{k_2} x_1$$

una ecuación de segundo orden, la sustituimos en la segunda ecuación:

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

análogamente para  $x_2$

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

Las 2 masas obedecen la misma ecuación, que solo necesita las 2 velocidades y posiciones iniciales para resolverla. Podemos usar esta ecuación en la forma de 2 ecuaciones diferenciales de primer grado, con  $v_1 = \dot{x}_1$ ,  $v_2 = \dot{x}_2$ .

$$\begin{aligned} v_1 &= -\frac{k_1}{m_1} - \frac{k_2}{m_1} (x_1 - x_2) \\ v_2 &= -\frac{k_2}{m_2} (x_2 - x_1) \end{aligned}$$

## 2.2.2 Algunos ejemplos con masas iguales

Se considera un modelo con  $m_1 = m_2 = 1$ , en el caso de no tener amortiguamiento, la ecuación característica es:

$$m^4 + (k_1 + 2k_2)m_2 + k_1 k_2 = 0$$

que tiene raíces:

$$\pm \sqrt{-\frac{1}{2}k_1 - k_2 \pm \frac{1}{2}\sqrt{k_1^2 + 4k_2^2}}$$

El ejemplo 2.1 plantea el sistema con  $k_1 = 6$ ,  $k_2 = 4$ , y  $(x_1(0), v_1(0), x_2(0), v_2(0)) = (1, 0, 2, 0)$ . Se obtiene, mediante lo desarrollado, la solución analítica:

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

Se observa que el movimiento para ambas masas es en fase, solo sus amplitudes son diferentes, las gráficas del movimiento y fase muestran patrones simples, estas se encontrarán en la sección *Implementación en python (jupyter lab)*.

El ejemplo 2.2 tiene condiciones iniciales  $k_1 = 6$ ,  $k_2 = 4$ , y  $(x_1(0), v_1(0), x_2(0), v_2(0)) = (-2, 0, 1, 0)$ , que resulta en un movimiento completamente fuera de fase, las soluciones analíticas obtenidas, útiles en nuestro contexto para verificar el error de soluciones numéricas, son:

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

El ejemplo 2.3 tiene condiciones iniciales  $k_1 = 0.4$ ,  $k_2 = 1.808$ , y  $(x_1(0), v_1(0), x_2(0), v_2(0)) = (0.5, 0, -0.5, 0.7)$ , aquí ocurre un movimiento más elaborado que los 2 anteriores, debido a que no entran completamente en fase o no fase, además, los valores de  $k$  fueron seleccionados cuidadosamente para que ocurriera.

## 2.2.3 Amortiguamiento

Agregamos el amortiguamiento en las ecuaciones, introduciendo una fuerza que depende solamente de la velocidad de la masa, y coeficientes de fricción  $\delta_1$  y  $\delta_2$ .

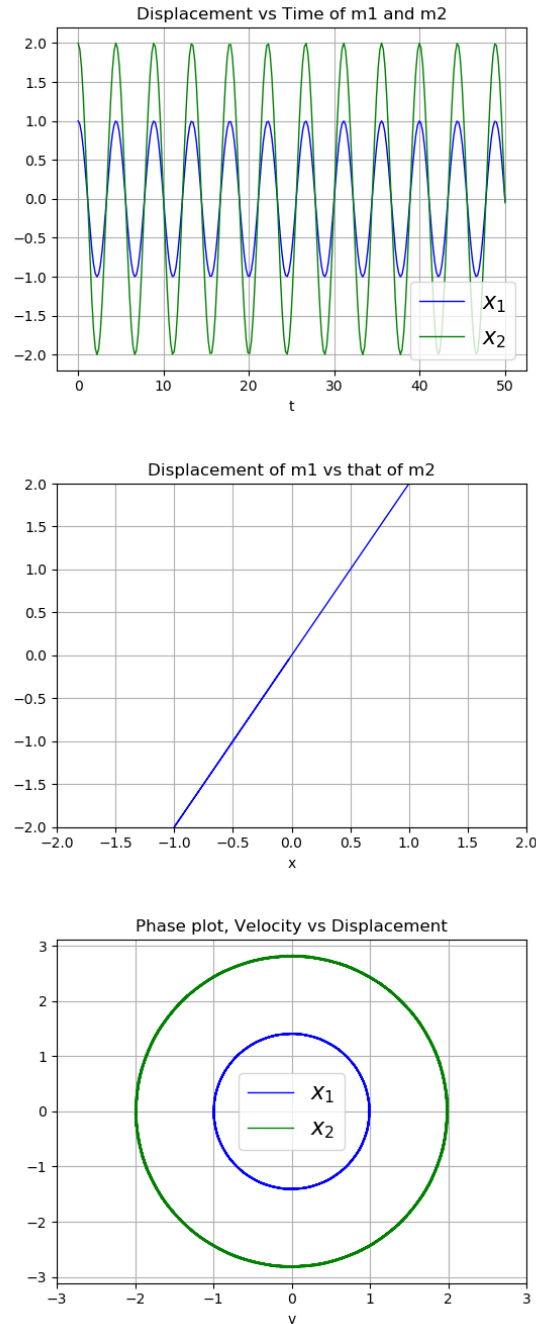
$$\begin{aligned} m_1 \ddot{x}_1 &= -\delta_1 v_1 - k_1 x_1 - k_2 (x_2 - x_1) \\ m_2 \ddot{x}_2 &= -\delta_2 v_2 - k_2 (x_2 - x_1) \end{aligned}$$

El ejemplo 2.4 introduce amortiguamiento, con condiciones iniciales  $k_1 = 0.4$ ,  $k_2 = 1.808$ ,  $\delta_1 = 0.1$ ,  $\delta_2 = 0.2$  y  $(x_1(0), v_1(0), x_2(0), v_2(0)) = (1, 0.5, 2, 0.5)$ . Se observa como el movimiento es periódico pero pierde amplitud, sin embargo, la fase varía considerablemente.

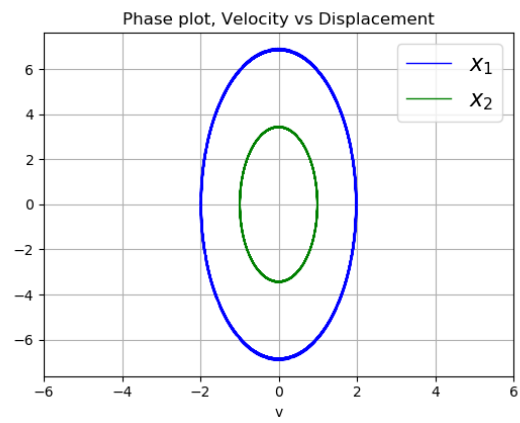
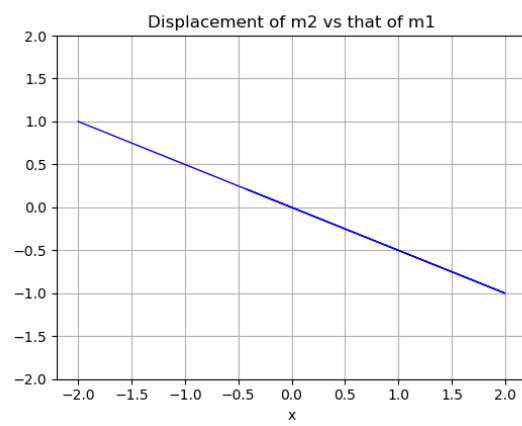
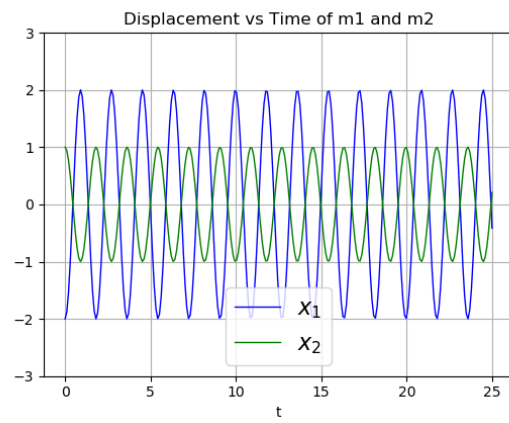
### 3 Implementación en python (jupyter lab)

Se realizó una simulación de los ejemplos 2.1 al 2.4, generando puntos mediante una solución numérica y realizando gráficas los datos para producir figuras que muestren la posición contra tiempo, desplazamiento de  $m_1$  contra el de  $m_2$  y fase (desplazamiento contra velocidad). Todo esto mediante el entorno de programación jupyter lab, con lenguaje de programación Python y bibliotecas importadas para graficación y solución numérica.

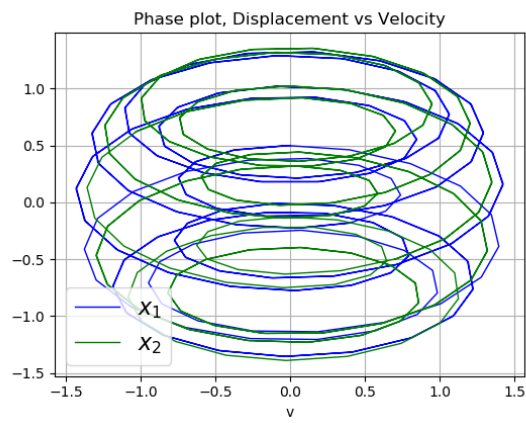
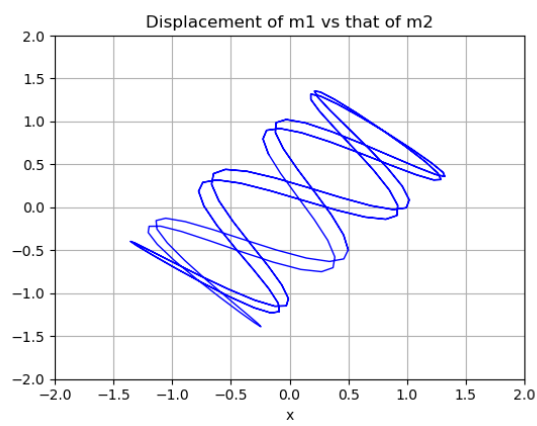
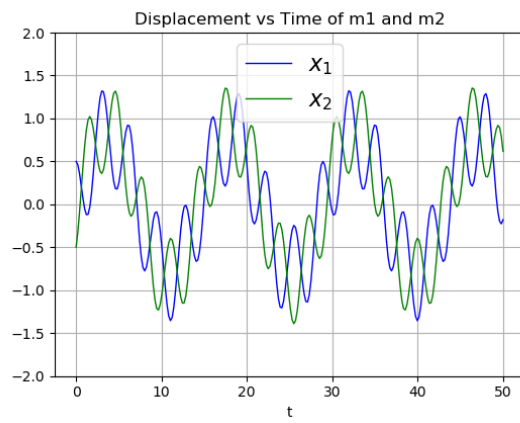
*Ejemplo 2.1*



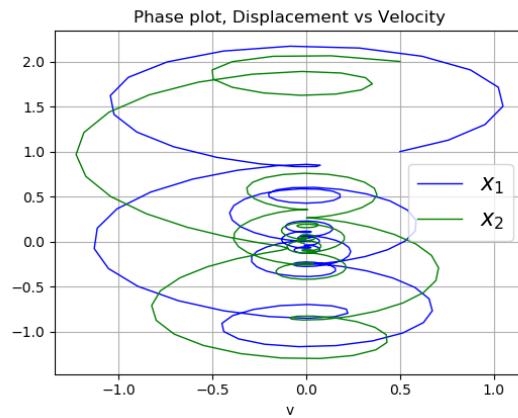
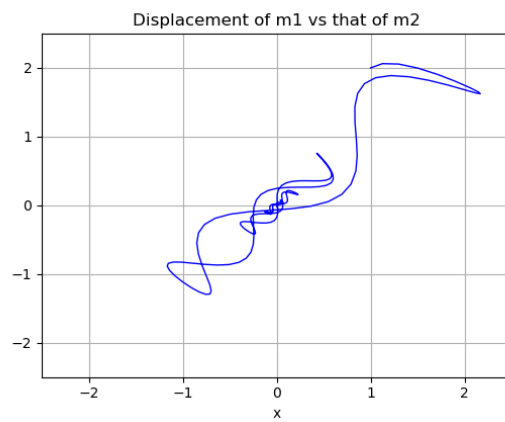
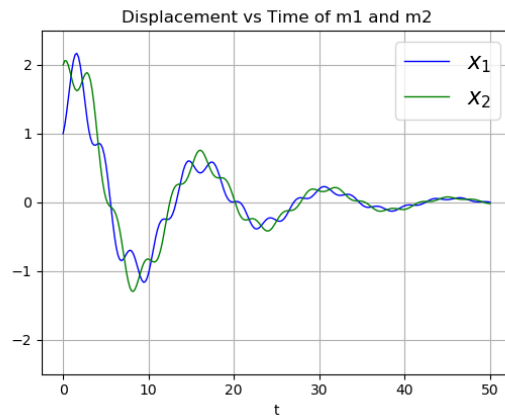
*Ejemplo 2.2*



*Ejemplo 2.3*



*Ejemplo 2.4*



Para realizar cada ejemplo se siguen 3 pasos:

- Definir el modelo, declarando parámetros y especificando la función involucrada
- Brindar valor a los parámetros, condiciones iniciales, y especificar el tiempo, tolerancia y número de puntos que se generan, escribiendo estos en un archivo.
- Recuperar los resultados del archivo y especificar los ejes de lo que se desea graficar.

## 4 Conclusiones

Se encontró que el modelo si es efectivo en simular el sistema de doble resorte, aunque no produce resultados completamente confiables, esto es por la resolución numérica y discretización del movimiento, sin embargo, son datos y visualizaciones muy útiles para comprender el sistema.

## 5 Bibliografía

- SARAH DUNCAN GRAHAM. (2003). Coupled spring equations. 8 de Marzo, 2018, de University of Southern Mississippi Sitio web: [http://math.oregonstate.edu/~gibsonn/Teaching/MTH323-010S15/Supplements/coupled\\_spring.pdf](http://math.oregonstate.edu/~gibsonn/Teaching/MTH323-010S15/Supplements/coupled_spring.pdf)
- (2009). Coupled spring-mass system. 8 de marzo, 2018, de SciPy Cookbook Sitio web: <http://scipy-cookbook.readthedocs.io/items/CoupledSpringMassSystem.html>

## 6 Apéndice

- ¿En general te pareció interesante esta actividad de modelación matemática? ¿Qué te gustó mas? ¿Qué no te gustó?  
Si me pareció muy interesante, se puede tener una mejor visualización del comportamiento del modelo físico.
- La cantidad de material te pareció ¿bien?, ¿suficiente?, ¿demasiado?  
Opino que recibimos una cantidad buena de información, no poca ni demasiada.
- ¿Cuál es tu primera impresión de Jupyter Lab?  
Es más útil que jupyter notebook, ya que te permite mover las celdas, y ver los archivos producidos en el mismo lugar.
- Respecto al uso de funciones de SciPy, ¿ya habías visto integración numérica en tus cursos anteriores? ¿Cuál es tu experiencia?  
Si, en Cálculo II, Fortran y Análisis Numérico, en general funcionan bien y rápidamente, pero nunca realizamos un modelo con muchas iteraciones.
- El tema de sistema de masas acopladas con resortes, ¿ya lo habías resuelto en tu curso de Mecánica 2?  
Fue mencionado, pero no estudiado.
- ¿Qué le quitarías o agregarías a esta actividad para hacerla más interesante y divertida?  
La dejaría como es actualmente.