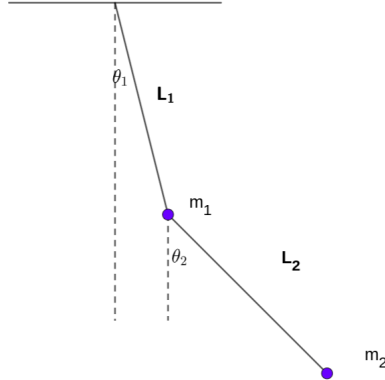


Péndulo Doble

Planteamiento del problema: El péndulo doble está formado por dos péndulos simples de longitudes l_1 y l_2 , de los que cuelgan partículas de masas m_1 y m_2 .

En un instante determinado t , las varillas que conforman a cada péndulo forman ángulos θ_1 y θ_2 con la vertical (Figura).



1. Construya un par integrado RK23 que utilice el RK2 de Ralston-Rabinowitz con un RK3 nuevo.
2. Modele el movimiento del péndulo doble.
3. Para empezar, considere que las masas y las longitudes de las varillas son iguales $L=1.3$ m, $M=1.5$ Kg.
4. Inicialmente los péndulos se liberan desde el reposo cuando los ángulos son iguales $\theta_1(0) = \theta_2(0) = \pi/2$ rad, simule el movimiento del sistema durante 100 segundos, utilizando para esto el par integrado construido previamente con $\Delta t= 0.04$ s. Acompañe sus resultados con las gráficas de θ vs t y de ω vs θ y θ_1 vs θ_2 .

Análisis Preliminar

Básicamente hay que modelar el movimiento de un péndulo con amortiguamiento. Para ello definimos los siguientes elementos del sistema:

1. l_1 : Longitud de la cuerda del pivote a la masa 1
2. l_2 : Longitud de la cuerda de la masa 1 a la masa 2
3. m_1 : Masa del objeto oscilador 1
4. m_2 : Masa del objeto oscilador 2
5. θ_1 : Ángulo que forma la cuerda 1 del péndulo con la horizontal
6. θ_2 : Ángulo que forma la cuerda 2 del péndulo con la horizontal

Con lo anterior podemos establecer las fuerzas que rigen el movimiento. Estas fuerzas son las siguientes:

Para el primer péndulo tenemos:

$$m_1 \ddot{x}_1 = -T_1 \sin \theta_1 + T_2 \sin \theta_2$$

$$m_1 \ddot{y}_1 = T_1 \cos \theta_1 - T_2 \cos \theta_2 - m_1 g$$

Para el segundo péndulo tenemos:

$$m_2 \ddot{x}_2 = -T_2 \sin \theta_2$$

$$m_2 \ddot{y}_2 = T_2 \cos \theta_2 - m_2 g$$

Despejando $\ddot{\theta}_1$ y $\ddot{\theta}_2$ tenemos:

$$\ddot{\theta}_1 = \frac{-g(2m_1 + m_2) \sin \theta_1 - m_2 g \sin(\theta_1 - \theta_2) - 2 \sin(\theta_1 - \theta_2) m_2 (\dot{\theta}_2^2 L_2 + \dot{\theta}_1^2 L_1 \cos(\theta_1 - \theta_2))}{L_1(2m_1 + m_2 - m_2 \cos(2\theta_1 - 2\theta_2))}$$

$$\ddot{\theta}_2 = \frac{2 \sin(\theta_1 - \theta_2) (\dot{\theta}_1^2 L_1 (m_1 + m_2) + g(m_1 + m_2) \cos \theta_1 + \dot{\theta}_2^2 L_2 m_2 \cos(\theta_1 - \theta_2))}{L_2(2m_1 + m_2 - m_2 \cos(2\theta_1 - 2\theta_2))}$$

Para descomponer en un sistema de dos ecuaciones de primer orden tomamos $\dot{\theta}_1 = \omega_1$ y $\dot{\theta}_2 = \omega_2$, de esta forma:

$$\dot{\theta}_1 = \omega_1$$

$$\dot{\theta}_2 = \omega_2$$

$$\dot{\omega}_1 = \frac{-g(2m_1 + m_2) \sin \theta_1 - m_2 g \sin(\theta_1 - \theta_2) - 2 \sin(\theta_1 - \theta_2) m_2 (\omega_2^2 L_2 + \omega_1^2 L_1 \cos(\theta_1 - \theta_2))}{L_1(2m_1 + m_2 - m_2 \cos(2\theta_1 - 2\theta_2))}$$

$$\dot{\omega}_2 = \frac{2 \sin(\theta_1 - \theta_2) (\omega_1^2 L_1 (m_1 + m_2) + g(m_1 + m_2) \cos \theta_1 + \omega_2^2 L_2 m_2 \cos(\theta_1 - \theta_2))}{L_2(2m_1 + m_2 - m_2 \cos(2\theta_1 - 2\theta_2))}$$

Las anteriores son las funciones pendiente que utilizaré para modelar el movimiento del péndulo doble.

Ahora, para desarrollar el RK 23 empleando el RK de Ralston Rabinowitz hay que resolver un sistema de ecuaciones, el cual tiene infinitas soluciones. Sabemos que el RK2 y el RK3 deben de compartir K1 y K2, por lo que tenemos lo siguiente:

$$P_2 = \frac{3}{4}; \quad q_{21} = \frac{3}{4}$$

Para resolver el sistema proponemos que $P_3 = \frac{1}{2}$, de forma que al resolver el sistema de ecuaciones resulta:

$$q_{31} = -\frac{1}{6}, \quad q_{32} = \frac{2}{3}$$

$$a_1 = \frac{2}{9} \quad a_3 = \frac{1}{3}$$

Así, el RK construido tiene la siguiente forma:

$$w_{i+1} = w_i + \frac{h}{9} (2K_1 + 4K_2 + 3K_3)$$

Con un error local dado por la expresión:

$$E = \frac{1}{9} |3K_3 - 2K_2 - K_1|$$

Especificación

Para realizar la simulación, se utilizan un programa principal del cual se mandan los datos de la simulación y un modulo donde contienen las funciones pendiente, funciones auxiliares, una subrutina para aplicar el método de Runge-Kutta (Ralston-Rabinowitz) y una subrutina para actualizar las coordenadas.

Primero establecemos las componentes del objeto de datos del sistema, estos son:

Objeto: DataP

- Posicion/Velocidad angular de la masa oscilatoria superior: MS, vector de 2 dimensiones, Variable de tipo Real.
- Posicion/Velocidad angular de la masa oscilatoria inferior: MI, vector de 2 dimensiones, Variable de tipo Real.
- Magnitud de la masa superior: m1, escalar, Variable de tipo Real.
- Magnitud de la masa superior: m1, escalar, Variable de tipo Real.
- Longitud de la cuerda superior, l1, escalar, Variable de tipo Real.

- Longitud de la cuerda inferior, l2, escalar, Variable de tipo Real.

Función: FP1 (Función pendiente 1), Salida: Posición/Velocidad angular de la masa 1.

- Datos del sistema: Objeto, entrada
- I-ésimo tiempo de la simulación: t, escalar, Variable de tipo Real, entrada.
- Posicion/Velocidad angular de la masa oscilatoria superior: MS, vector de 2 dimensiones, Variable de tipo Real.
- Posicion/Velocidad angular de la masa oscilatoria inferior: MI, vector de 2 dimensiones, Variable de tipo Real.

Función: FP2 (Función pendiente 1), Salida: Posición/Velocidad angular de la masa 1.

- Datos del sistema: Objeto, entrada
- I-ésimo tiempo de la simulación: t, escalar, Variable de tipo Real, entrada.
- Posicion/Velocidad angular de la masa oscilatoria superior: MS, vector de 2 dimensiones, Variable de tipo Real.
- Posicion/Velocidad angular de la masa oscilatoria inferior: MI, vector de 2 dimensiones, Variable de tipo Real.

Función: error (Función que calcula el error de truncamiento para el RK), Salida: error de truncamiento.

- I-ésimo ancho de paso: h0, escalar, Variable de tipo Real, entrada.
- K's del RK: k1, k2, vector de 2 dimensiones, Variable de tipo Real, entrada.

Función: hw (Función que actualiza el valor de h), Salida: Valor de h actualizado.

- I-ésimo ancho de paso: h0, escalar, Variable de tipo Real, entrada.
- I-ésima aproximación: 20, escalar, Variable de tipo Real, entrada.
- I-ésimo error: e0, escalar, Variable de tipo Real, entrada.
- K's del RK: k1, k2, vector de 2 dimensiones, Variable de tipo Real, entrada.

Subrutina: RK23RB1-2 (Aplica el método RK23 Ralston-Rabinowitz), Salida: Valor de h, posición y velocidad angular de cada partícula actualizadas.

- Datos del sistema: Objeto, entrada.
- I-ésimo tiempo de la simulación: t, escalar, Variable de tipo Real, entrada.
- I-ésimo ancho de paso: h, escalar, Variable de tipo Real, entrada/salida.
- Posicion/Velocidad angular de la masa oscilatoria superior: MS, vector de 2 dimensiones, Variable de tipo Real.
- Posicion/Velocidad angular de la masa oscilatoria inferior: MI, vector de 2 dimensiones, Variable de tipo Real.
- K's auxiliares del método: [K1, K2, K3], vector de 2 dimensiones, Variable de tipo Real, Auxiliar.
- I-ésimo error, errors, vector de 2 dimensiones, Variable de tipo Real, Auxiliar.
- Aproximación con método de orden menor, M, vector de 2 dimensiones, Variable de tipo Real, Auxiliar.
- Aproximación con método de orden mayor, B, vector de 2 dimensiones, Variable de tipo Real, Auxiliar.
- auxiliares del método, [e, w0, e1, d], esalar, Variable de tipo Real, Auxiliar.

Subrutina: NewC (Actualiza las coordenadas de la masa oscilatoria), Salida: Archivo de datos con las posiciones actualizadas.

- Datos del sistema: Objeto, entrada.
- Tiempo total de la simulación: T0, escalar, Variable de tipo Real, entrada.
- I-ésimo tiempo de la simulación: t, escalar, Variable de tipo Real, entrada.
- I-ésimo ancho de paso: dt, escalar, Variable de tipo Real, entrada/salida.
- Contadores/Auxiliares, [i, unit0], entero, Variable de tipo Integer, entrada/salida.

****El código está adjunto en la sección de avaus****

Resultados

Para este ejercicio los resultados obtenidos fueron los siguientes:

Simulación 1:

- l_1 : 1.3 m
- l_2 : 1.3 m
- m_1 : 1.5 Kg
- m_2 : 1.5 Kg
- θ_1 : $\frac{\pi}{2}$
- θ_2 : $\frac{\pi}{2}$

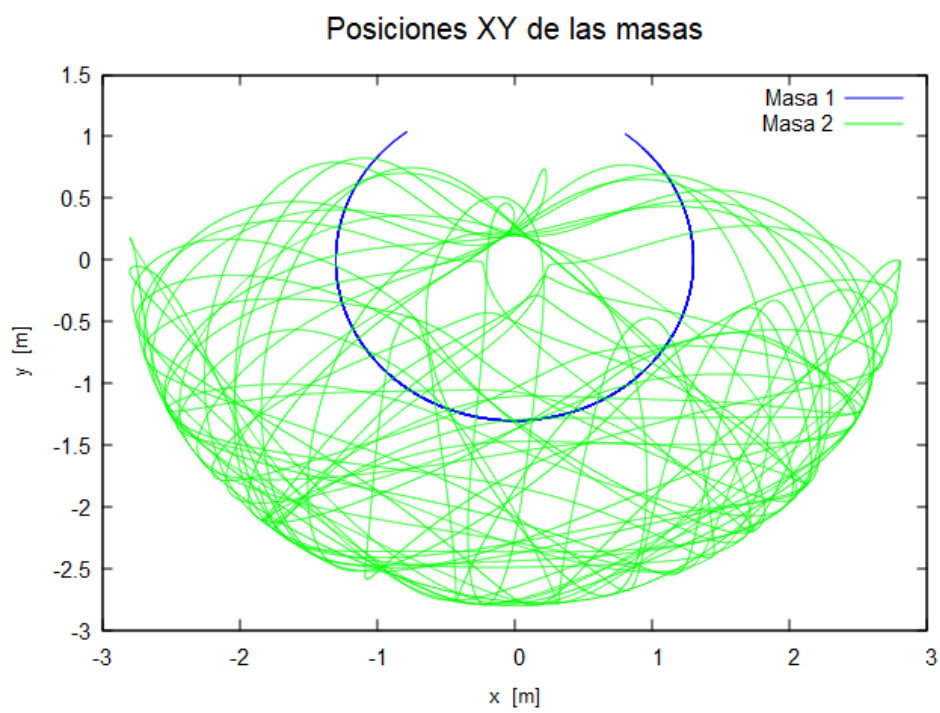


Figura 1: Gráfica de Trayectoria.

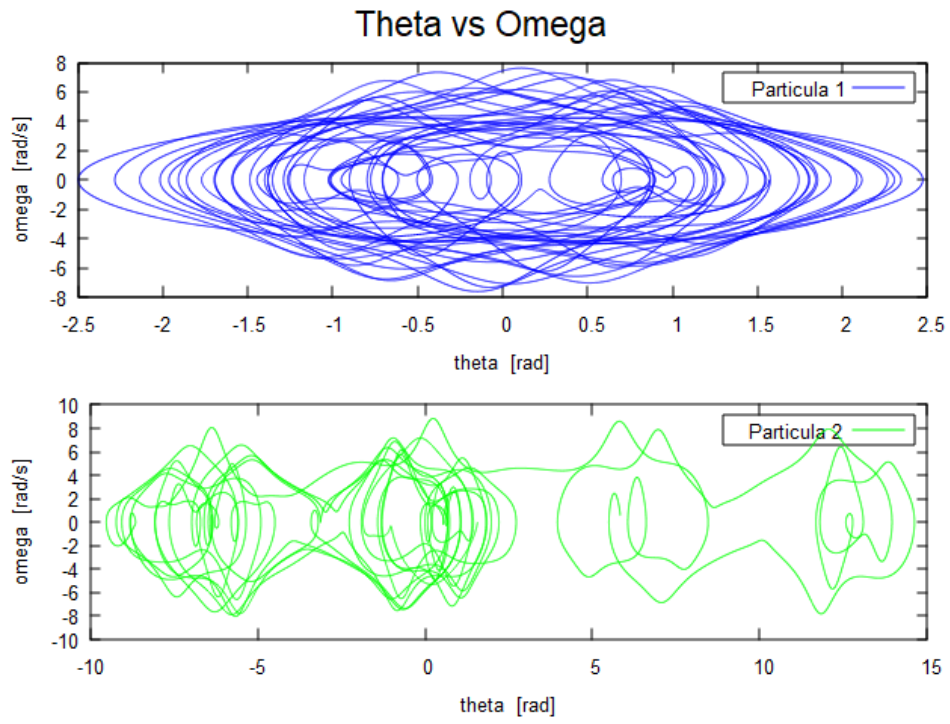


Figura 2: Gráfica de θ_1 vs ω_1 ; θ_2 vs ω_2 .

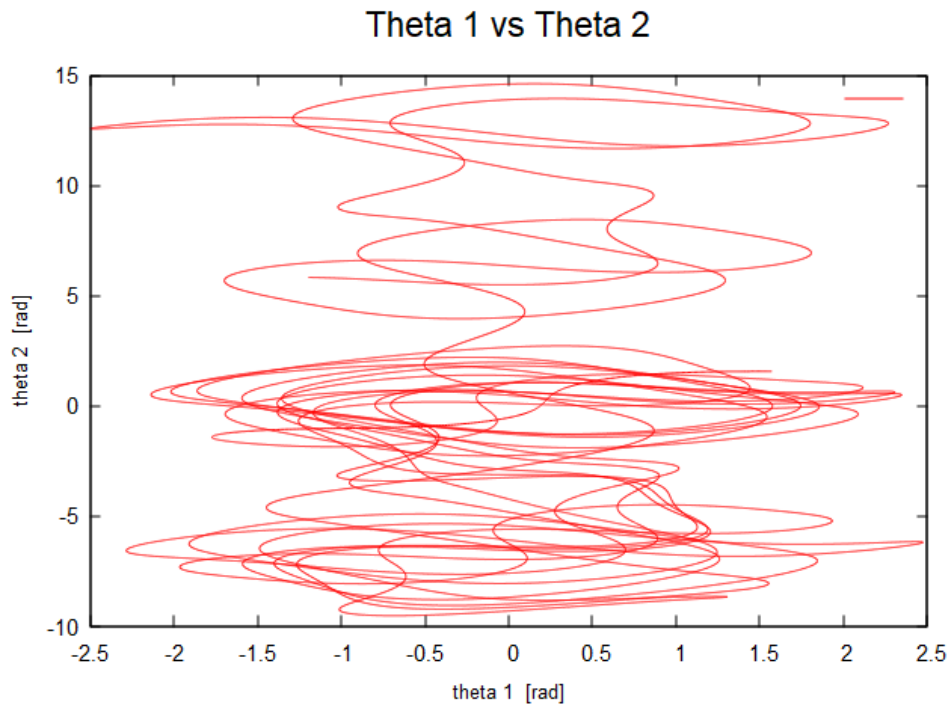


Figura 3: Gráfica de θ_1 vs θ_2 .

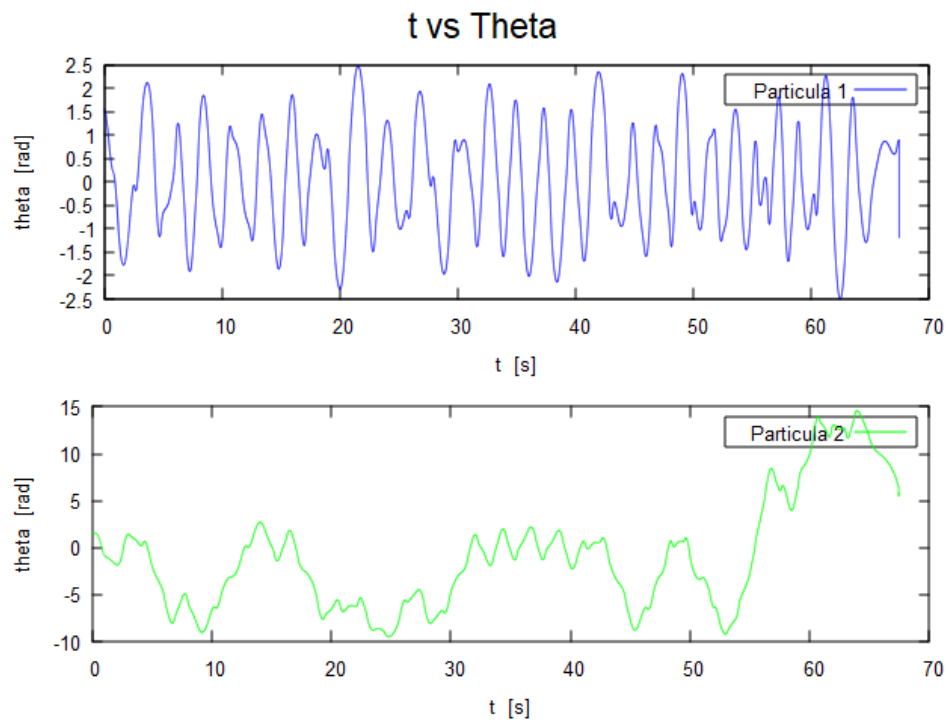


Figura 4: Gráfica de t vs θ_1 ; t vs θ_2 .

Simulación 2:

- l_1 : 2.3 m
- l_2 : 3.5 m
- m_1 : 0.5 Kg
- m_2 : 5.5 Kg
- θ_1 : $\frac{\pi}{2}$
- θ_2 : $\frac{\pi}{2}$

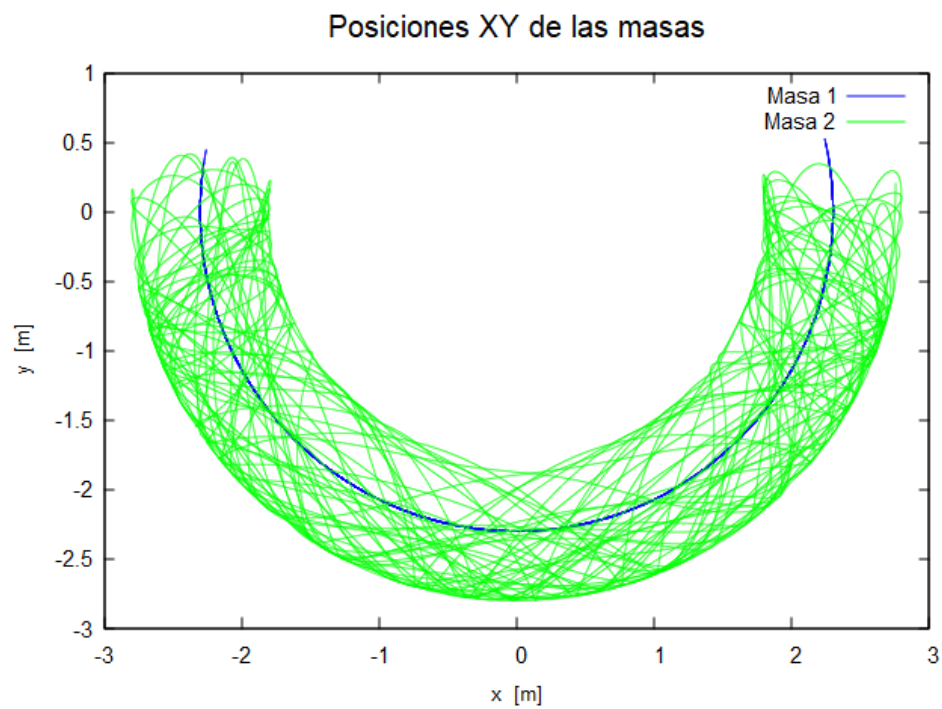


Figura 5: Gráfica de Trayectoria.

Simulación 3:

- l_1 : 0.5 m
- l_2 : 2.5 m
- m_1 : 0.5 Kg
- m_2 : 3.5 Kg
- θ_1 : $\frac{\pi}{2}$
- θ_2 : $\frac{\pi}{2}$

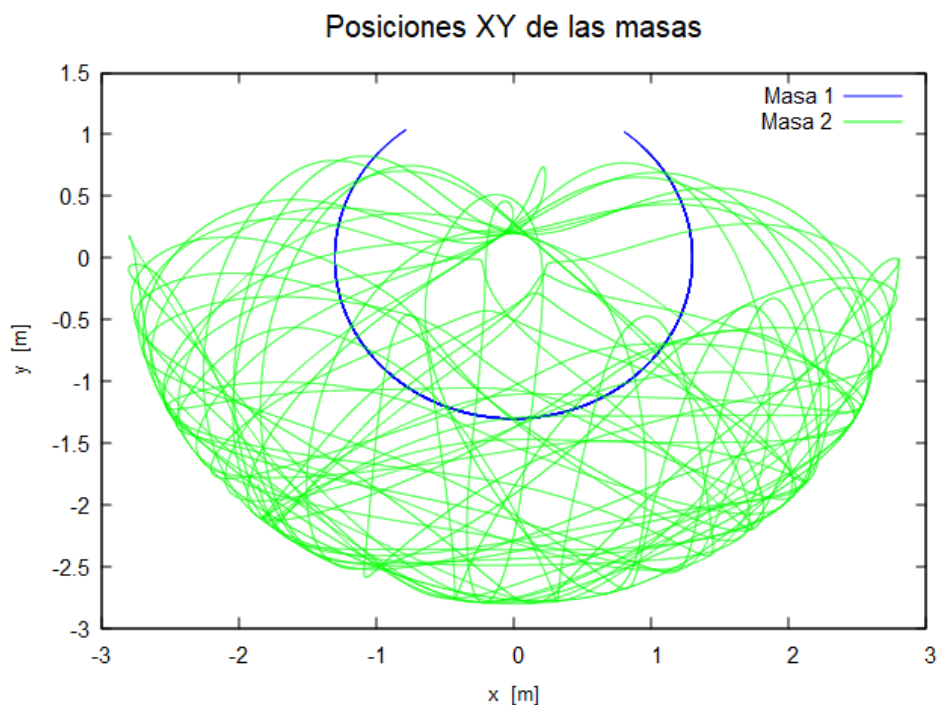


Figura 6: Gráfica de Trayectoria.

Conclusión

Fue una buena práctica, estoy satisfecho con mis resultados, a pesar de que considero que el análisis los resultados no me fue posible de realizar ya que considero que no comprendo bien el fenómeno simulado. Además de que tengo un error bastante significativo en mis resultados que no me fue posible de resolver por razones que desconozco y que espero poder discutir con mis compañeros de clase. De igual forma tuve bastantes problemas descomponer la ecuación de movimiento en un sistema de ecuaciones, pero pude basarme los ejercicios realizados previamente donde simulamos péndulos. El error más significativo de mi programa es que para diferentes situaciones el método no converge en algún lugar, y por más que intente de resolver esta situación no pude.

Respecto a programar, fue muy complicado ya que hay que tener mucho cuidado y poner mucha atención al momento de aplicar los métodos de Runge-Kutta en sistemas de ecuaciones. Sobre todo, al momento de establecer las k 's en el método que me confundía bastante el hecho de que las funciones pendiente no necesariamente dependen del tiempo y que era donde estaban mis ideas erróneas. De igual forma, creo que la forma que emplear funciones apuntadora me confundió bastante, pero pude evadir usar esta forma de empelar el método. Me queda claro que no es lo mismo sólo realizar cálculos manualmente, que sería muy laborioso (sobre todo en trabajos como el presente, donde es necesario realizar una gran cantidad de pasos para llegar al resultado esperado), y a realizarlo mediante la elaboración de programas en algún lenguaje de programación (en este caso Fortran), es necesario analizar con cuidado los datos, interpretarlos, comprobar y contrastar con la teoría con la práctica, así como comprobar la efectividad de los resultados obtenidos.

Por último, hay bastantes cosas que me gustaría cambiar en cuanto a la presentación de resultados, me hubiera gustado modelar el movimiento en cada ejercicio, pero no me es posible de hacer. De igual forma, tuve problemas al usar Gnuplot, pues al ser demasiados datos los que tiene que leer no arrojaba las gráficas, por tal motivo me vi obligado en solamente plasmar en los resultados de las simulaciones 2 y 3 las posiciones de las masas.

Considero que es muy necesario y de gran importancia aplicar la programación en el área de la Física ya que es una gran herramienta que permite facilitar los cálculos y simular fenómenos que pueden llegar a ser difíciles de realizar experimentalmente.