

# Tarea 2

Métodos Numéricos para la Ciencia e Ingeniería

Integrante: Matías Méndez Zenteno  
Profesor: Valentino González C.  
Auxiliares: Vicente Donaire Aguilera  
Benjamín Navarrete

Fecha de realización: 17 de octubre de 2020

Fecha de entrega: 17 de octubre de 2020

Santiago, Chile

# 1. Introducción

En el presente informe se muestra el desarrollo hecho para la Tarea 2 del curso *Métodos Numéricos para la Ciencia e Ingeniería*. Consistente de dos problemas, el primer objetivo es integrar la ecuación de movimiento de un péndulo simple en un medio viscoso con condiciones de borde, implementado el método de RK4. La ecuación de movimiento es

$$\ddot{\phi} = -\gamma\dot{\phi} - \frac{g}{l}\sin(\phi)$$

Donde  $l = 5.427$  [m],  $g = 9.8$  m/s<sup>2</sup> y  $\gamma = 2.427$  s<sup>-1</sup>. Las condiciones de borde son  $\phi(t=0) = \phi_0 = 0.0001$  y  $\dot{\phi}(t=0) = 0$

El resultado se compara con la solución del péndulo para pequeñas oscilaciones. Y además, se integra nuevamente para una condición de borde  $\phi_0 = \pi/2.427$ . Se estudia la energía en este caso, y también se compara con el caso de las pequeñas oscilaciones.

El segundo objetivo es integrar mediante un algoritmo RK4 para un tiempo determinado el sistema de Lorentz, que corresponde al set de ecuaciones diferenciales ordinarias

$$\frac{dx}{ds} = \sigma(y - x)$$

$$\frac{dy}{ds} = x(\rho - z) - y$$

$$\frac{dz}{ds} = xy - \beta z$$

Con parámetros  $\sigma = 10$ ,  $\beta = 8/3$  y  $\rho = 28$ . Junto a las condiciones iniciales  $(x_0, y_0, z_0) = (1, 1, 1)$ . Y además se estudia la solución cuando se perturba  $y_0 \rightarrow y_0 + 0.0427$ , comparándose con la primera solución.

## 2. Desarrollo

### 2.1. Problema 1

Para pequeñas oscilaciones tenemos que  $\sin(\phi) \approx \phi$ , quedando la ecuación de movimiento como

$$\ddot{\phi} = -\gamma\dot{\phi} - \frac{g}{l}\sin(\phi) \implies \ddot{\phi} = -\gamma\dot{\phi} - \frac{g}{l}\phi$$

Esta EDO de segundo orden tiene una solución analítica del tipo  $\phi(t) = A \cdot e^{-\frac{\gamma}{2}t} \cdot \sin(\omega t + \varphi)$  con  $\omega = \sqrt{\frac{g}{l} - \left(\frac{\gamma}{2}\right)^2}$ ,  $\tan\phi = \frac{\phi_0 \cdot \omega}{\dot{\phi}_0 + \frac{\gamma}{2}\phi_0}$  y  $A = \sqrt{\phi_0^2 + \left(\frac{\dot{\phi}_0 + \frac{\gamma}{2}\phi_0}{\omega}\right)^2}$ .

De esta forma para resolver la ecuación de movimiento y aplicar el algoritmo de RK4, se transforma la ecuación en una EDO vectorial de primer orden. Para eso definimos  $\vec{y} = \begin{pmatrix} \phi \\ w \end{pmatrix}$  tal que  $\frac{d}{dt}\phi = w$ , con esto aparece la siguiente EDO

$$\frac{d}{dt}\vec{y} = \vec{f}(t, \vec{y}) = \begin{pmatrix} w \\ -\gamma w - \frac{g}{l}\sin(\phi) \end{pmatrix}$$

Así tenemos  $\frac{d}{dt}\vec{y} = \vec{f}(t, \vec{y})$ , con la condición inicial  $\vec{y}_0 = (0.0001, 0)$ . Entonces esta es la EDO que se resuelve, por lo que se puede aplicar RK4. Tenemos que el algoritmo usado para calcular un punto  $y_n$  con un intervalo de tiempo  $dt$ , es

$$y_{n+1}^{\vec{}} = \vec{y}_n + \frac{1}{6}(\vec{k}_1 + \vec{k}_2 + \vec{k}_3 + \vec{k}_4)$$

donde

$$\begin{aligned} \vec{k}_1 &= \vec{f}(t_n, \vec{y}_n) \\ \vec{k}_2 &= \vec{f}(t_n + 0.5 \cdot dt, \vec{y}_n + 0.5 \cdot \vec{k}_1 \cdot dt) \\ \vec{k}_3 &= \vec{f}(t_n + 0.5 \cdot dt, \vec{y}_n + 0.5 \cdot \vec{k}_2 \cdot dt) \\ \vec{k}_4 &= \vec{f}(t_n + dt, \vec{y}_n + \vec{k}_3 \cdot dt) \end{aligned}$$

Entonces, lo que se implementa en el código es el cálculo de las constantes  $\vec{k}_1$ ,  $\vec{k}_2$ ,  $\vec{k}_3$  y  $\vec{k}_4$  para cada punto. Para luego, a partir del punto  $\vec{y}_0$ , hacer una iteración de  $n$  pasos tal que llegue a un tiempo

determinado, en este caso se llegará hasta el segundo 10 con un intervalo de tiempo  $dt = 0.001$ . El método de RK4 tiene un error por paso del orden de  $O(dt^4)$ , por lo que se espera llegar a tener 9 decimales de precisión en el resultado.

Este algoritmo nos entrega por cada punto el valor de  $\phi$ , y su velocidad  $\dot{\phi}$ . Para el caso de la perturbación, se realiza el mismo procedimiento solo que ahora la condición inicial es  $\vec{y}_0 = (\pi/2.427, 0)$ .

Para el estudio del péndulo cuando se realiza una perturbación, se utiliza la energía total para comparar su valor entre la solución numérica y analítica. Así, se tiene que la energía total de un péndulo es

$$E = \frac{1}{2}ml^2\dot{\phi}^2 - mgl\cos(\phi)$$

## 2.2. Problema 2

Como en el problema anterior, para resolver este set de EDOs es necesario transformarla lo anterior en una EDO vectorial de primer orden. Así, se define el vector  $\vec{r}$ ,

$$\vec{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

De esta forma queda la siguiente EDO vectorial

$$\frac{d}{dt}\vec{r} = \vec{f}(t, \vec{r}) = \begin{pmatrix} \sigma(y - x) \\ x(\rho - z) - y \\ xy - \beta z \end{pmatrix}$$

En este caso, se utiliza el método de RK45 de la librería de *scipy-integrate*, el cual utiliza el algoritmo RK4 junto a RK5 como paso adaptativo. Este algoritmo crea un objeto clase *scipy.integrate.RK45*.

Los inputs para este objeto son:

- $\vec{f}(t, \vec{r})$ , función a integrar
- $t_0 = 0$ , tiempo inicial
- $\vec{r}_0 = (1, 1, 1)$ , condición inicial
- $rtol = 10^{-13}$ , precisión relativa
- $t_{bound} = 50$ , tiempo máximo a integrar
- `vectorized= True`, indica que es vectorial

Luego teniendo el objetivo, aplicando la función `step()` de la clase, se avanza hacia el punto  $\vec{y}_1$ . De esta forma, se hace una iteración para avanzar N puntos de tal forma que se llegó hasta el momento  $t = 50$ . Para la perturbación se realiza, los mismos pasos anteriores solamente que ahora la condición inicial es  $\vec{r}_0 = (1, 1 + 0.0427)$ .

## 3. Resultados

### 3.1. Problema 1

En la figura 1 se muestran dos gráficos, el primero se refiere a la solución numérica de la ecuación de movimiento del péndulo versus la solución analítica. Luego, en el segundo gráfico se muestra la diferencia entre estas dos soluciones.

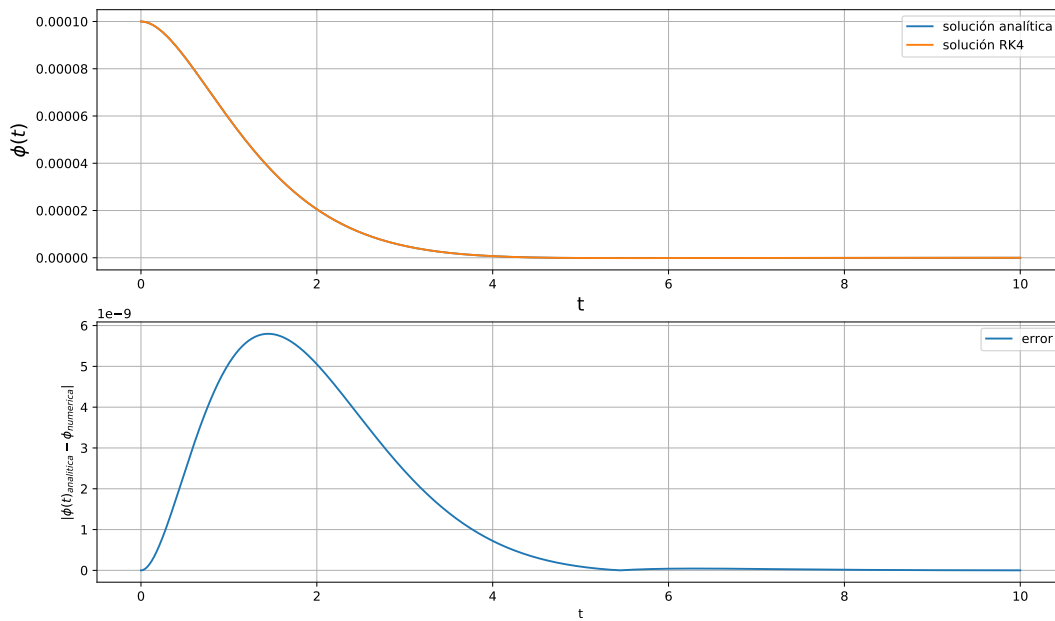


Figura 1: Solución numérica junto a la analítica para la ecuación de movimiento de un péndulo simple en un medio viscoso.

En la figura 2 se muestra el gráfico con la solución numérica versus la solución analítica para una condición inicial  $\phi_0 = \beta/2.427$ .

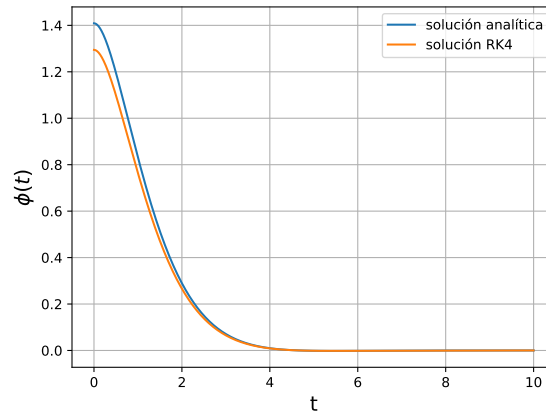


Figura 2: Solución numérica junto a la analítica para la ecuación de movimiento de un péndulo simple en un medio viscoso.

Finalmente, en la figura 3 se muestra la energía total para las soluciones anteriores.

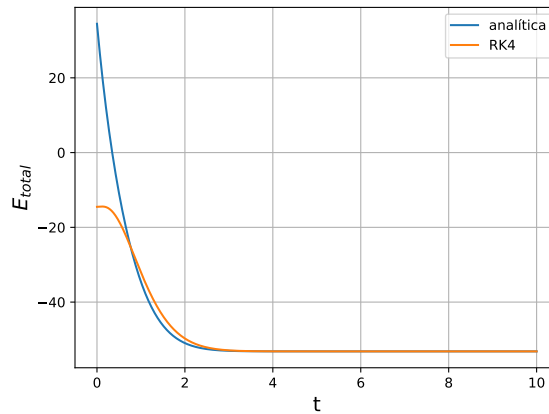
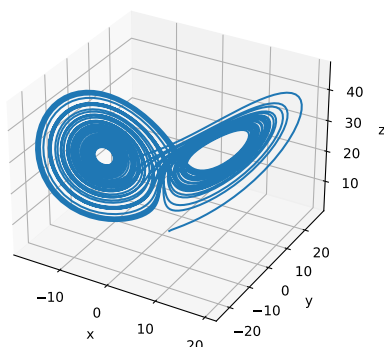


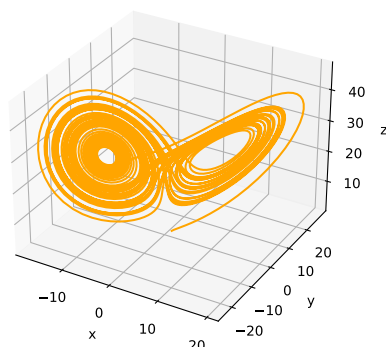
Figura 3: Energía total de la solución numérica versus la solución analítica para la ecuación de movimiento de un péndulo simple en un medio viscoso, con una condición inicial  $\phi_0 = \pi/2.427$  y  $\dot{\phi}_0 = 0$ .

### 3.2. Problema 2

En la figura 4, se muestran dos gráficos en 3D, correspondientes a las soluciones para el sistema de Lorentz. La figura 4 (a) corresponde a la solución numérica con la condición inicial (1,1,1), mientras que la figura 4 (b) corresponde a la solución numérica con una perturbaciones, de tal forma que la condición inicial es (1,1+0.0427,1).



(a) Solución numérica sin perturbación.



(b) Solución numérica con perturbación.

Figura 4: Soluciones numéricas hechas con RK4 de la librería scipy-integrate para el sistema de Lorenz, con y sin perturbación en Y para un tiempo de 50 segundos aproximado.

En la figura 5 se muestran las soluciones numéricas con y sin perturbación para el sistema de Lorenz en gráficos por coordenadas y tiempo respectivamente.

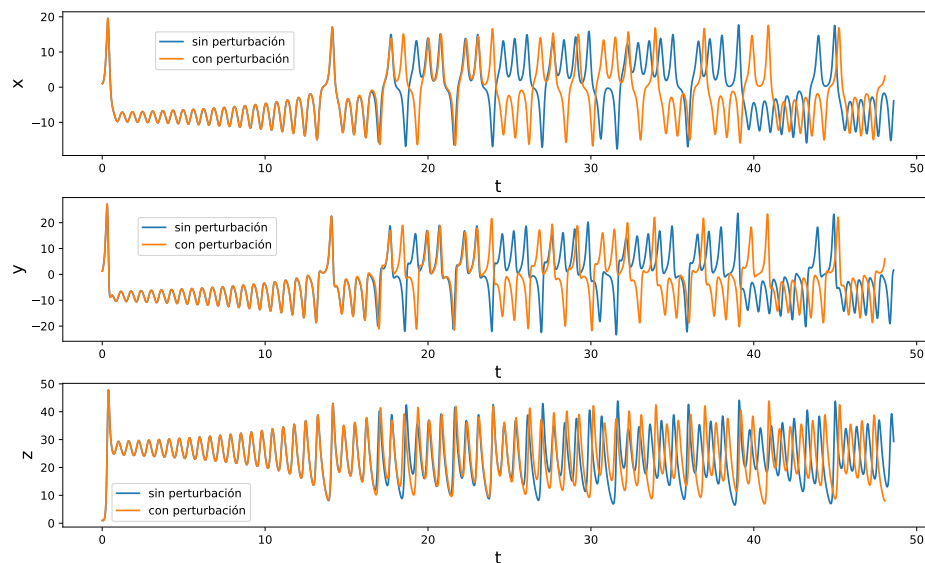


Figura 5: Gráficos para la solución numérica con y sin perturbación para un tiempo de integración de 50 segundos.

Finalmente, en la figura 6 se muestran los gráficos hechos en la figura 4 vistas desde 3 planos distintos para un tiempo de integración aproximado de 50 segundos.

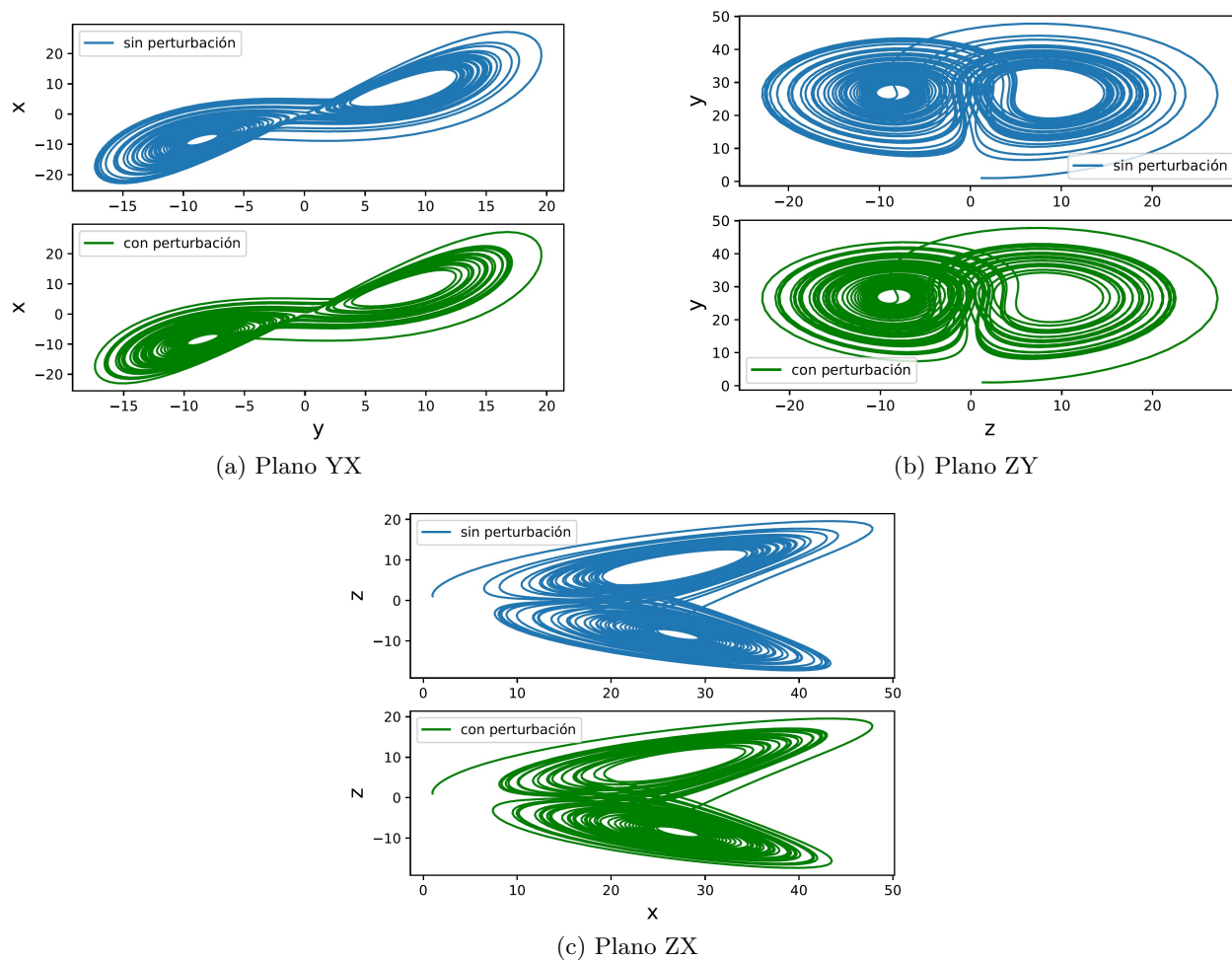


Figura 6: Planos de las soluciones numéricas del sistema de Lorenz con y sin perturbaciones



## 4. Discusión y Conclusiones

Respecto al problema 1, se tiene que la solución numérica para la ecuación de movimiento con el algoritmo de RK4 implementado es bastante precisa, a simple vista no se ve la diferencia entre la solución numérica y la analítica del caso de pequeñas oscilaciones. La precisión, como se observa en la figura 1, es de 9 decimales por lo que fue la precisión esperada del algoritmo, y se logra implementar bien RK4.

Para la nueva condición inicial impuesta, se observa, en la figura 2, que las soluciones numéricas y analítica se diferencian por poco en los primeros dos segundos de integración. Mientras que en el resto del tiempo se observa que son bastante parecidas. Respecto a la energía total de estas, figura 3, se ve una considerable diferencia de energía en los dos primeros segundos de integración. Así se ve que la solución numérica pierde energía de forma más rápida. Esta diferencia en los dos primeros se puede deber a la nueva condición inicial  $\phi_0$ , al ser esta mayor que 1 no se comporta al principio como el movimiento del péndulo para pequeñas oscilaciones.

Respecto al problema 2, se observa en los gráficos de las figuras 4, 5 y 6 la correcta implementación de RK4 de la librería `scipy-integrate`. Por otro lado, se observa una diferencia entre los comportamientos con y sin perturbación, el comportamiento con perturbación presenta una mayor concentración de líneas en el centro de los agujeros de figura, mientras que el otro presenta una mayor concentración afuera del centro de los agujeros. De la figura 5, se observa que en los ejes X e Y ambos comportamientos son similares hasta el segundo 15 de integración aproximadamente, luego comienzan a tener comportamientos opuestos. Por otro lado, en el eje Z muestran un comportamiento bastante similar durante todo el tiempo de integración.