1. Tarea #05 - Métodos numéricos

1.1. Runge-Kutta

Se implementó un algoritmo en Jupyter para resolver *ecuaciones diferenciales ordinarias* empleando el método Runge-Kutta de cuarto orden. El modelo matemático para el método Runge-Kutta fue tomado de [1].

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k2_2 + 2k_3 + k_4) \tag{1}$$

$$\begin{aligned} k_1 &= f(x_i, y_i) \\ k_2 &= f(x_i + \frac{h}{h}, y_i + \frac{h}{2}k1) \\ k_3 &= f(x_i + \frac{h}{h}, y_i + \frac{h}{2}k2) \\ k_4 &= f(x_i + h, y_i + hk_3) \end{aligned} \tag{2}$$

Se evaluó el algoritmo para las funciones $\sin(x)$ y $\cos(x)$ con múltiples condiciones iniciales. Los detalles de la simulación se pueden observar en la tabla 1.

Tamaño de paso (h)	0.1
Intervalo de integración (t)	[0, 2π]
Condiciones de inicio (x_0)	[-3, 3]

Cuadro 1: Condiciones de la simulación.

Al observar el comportamiento de las ecuaciones en las figuras 1 y 2, se pueden identificar valores a los cuales ambas funciones tienden a estabilizarse. Para el caso de la función seno, existen tres valores a los que tiende la función: $-\pi, 0, \pi$. Para el caso de la función coseno los valores son $\frac{\pi}{2}, -\frac{5\pi}{2}$

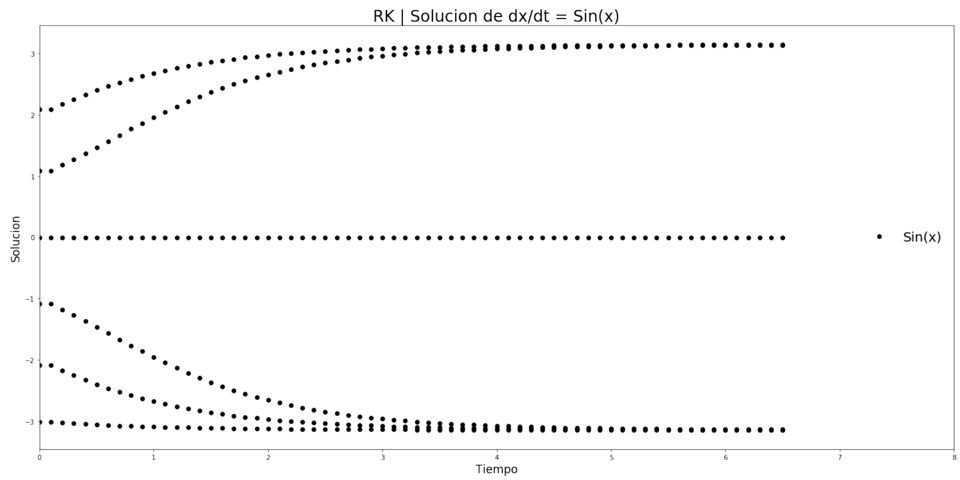

Figura 1: Resultados de evaluación empleando el método de Runge Kutta.

1.2. lsoda

El kernel de SageMath, empleado dentro de Jupyter, incluye los solucionadores numéricos de SciPy. Uno de estos solucionadores es *lsoda* que fue adaptado de la librería provista por FORTRAN. Se evaluó el solucionador para las funciones seno y coseno, empleando las mismas condiciones de simulación que el solucionador Runge-Kutta. Los resultados de dichas simulaciones se muestran en las figura 3 y 4. Se observa que de igual manera, las funciones tienden a los mismos valores que el solucionador Runge-Kutta.

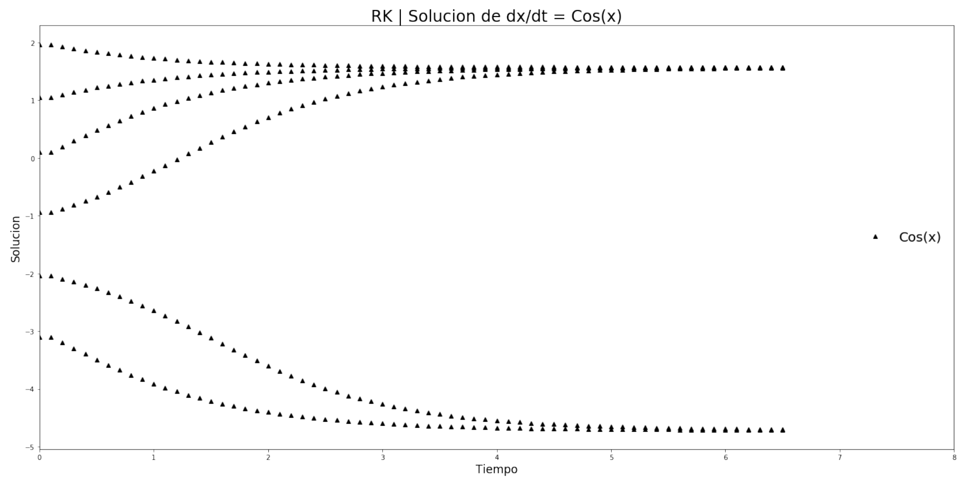

Figura 2: Resultados de evaluación empleando el método de Runge Kutta para la función seno.

1.3. Comparación

Para la comparación de los solucionadores se tomó exclusivamente el comportamiento de las funciones con las condiciones de inicio $x(0) = 3, \dot{x}(0) = 3$. Se observa que la función tiende a un valor único conforme el tiempo t tiende a infinito. Es posible identificar diferencias entre solucionadores al inspeccionar los primeros puntos de evaluación en las figuras 5 y 6.

Referencias

[1] *Numerical Methods Using Python*. `http://people.bu.edu/andasari/courses/numericalpython/python.html`. Accesado: 2019-10-08.

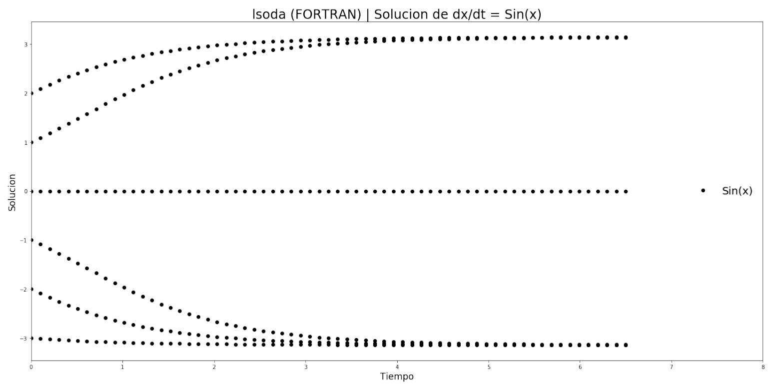

Figura 3: Resultados de evaluación empleando el solucionador numérico *lsoda* para la función seno.

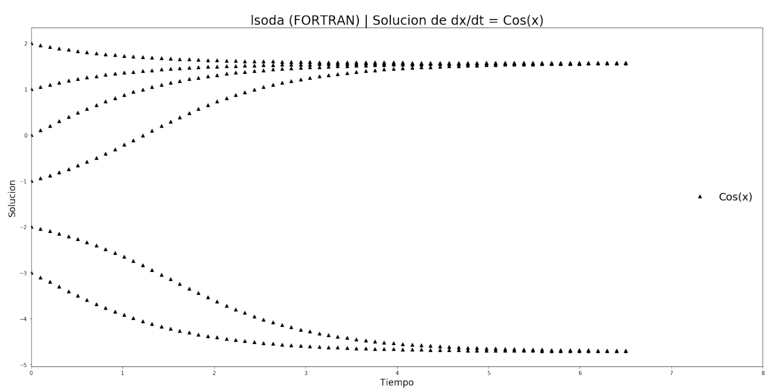

Figura 4: Resultados de evaluación empleando el solucionador numérico *lsoda* para la función coseno.

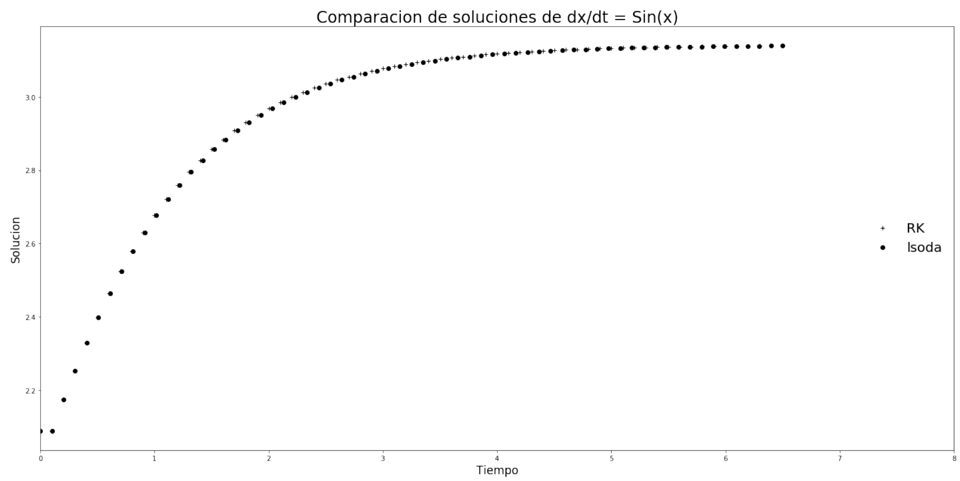

Figura 5: Comparación de resultados entre solucionadores para la función seno.

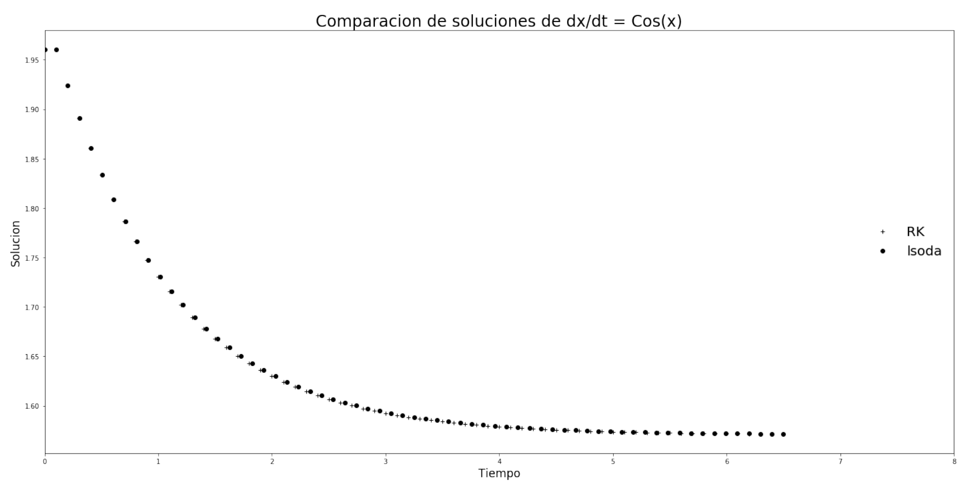

Figura 6: Comparación de resultados entre solucionadores para la función coseno.