Trabajo Práctico Entregable N°3

Problemas de Valores Iniciales

Facultad de Ingeniería y Ciencias Hídricas.

Ingeniería Informática

Cálculo Numérico

Alumno: Garcia, Lucas Enedín

INTRODUCCIÓN

En este trabajo pondremos a prueba uno de los métodos para resolver problemas de valores iniciales de forma numérica, más precisamente, resolveremos un sistema de ecuaciones diferenciales ordinarias con valores iniciales cuyas soluciones tienen interpretación física.

MÉTODOS NUMÉRICOS PARA PROBLEMAS DE VALOR INICIAL (PVI)

Describimos algunos de los métodos de un paso, los cuales permiten evaluar la solución numérica y_{i+1} en la abscisa x_{i+1}, con las formulas del tipo:

$$y_{i+1} = y_i + h\,\Phi$$

Donde Φ es una aproximación a $\frac{y(x_{i+1})-y(x_i)}{h}$.

Método de Euler:

Es un método de un paso y explicito debido a que Φ no depende de y_{i+1}. Esta es la técnica de aproximación más básica para resolver PVI de la forma:

$$\begin{cases} y' = f(x,y), \quad a \leq x \leq b \\ y(a) = \overline{y_0} \end{cases}$$

El método de Euler define a y_i la solución numérica obtenida para aproximar a $y(x_i)$.

De esta manera:

$$\left| \begin{array}{l} y_0 = y(a) = \overline{y_0} \\ y_{i+1} = y_i + h\, f(x_i, y_i) \quad i = 0,1,2 \dots, n-1 \end{array} \right.$$

Este método se denomina también Método de Euler Progresivo o *Hacia Adelante (forward Euler).*

Si la formula se plantea de la siguiente forma:

$$y_{i+1} = y_i + h\, f(x_{i+1}, y_{i+1})$$

en este caso la solución x_{i+1} depende de y_{i+1} por lo que este método es implícito y se denomina Euler *Regresivo* o *Hacia atrás (backward Euler)*.

Método de Crank-Nicholson (o del Trapecio)

Es un método implícito y es más preciso que el de Euler, utiliza la siguiente formula:

$$y_{i+1} = y_i + \frac{1}{2} h \left(f(x_i, y_i) + f(x_{i+1}, y_{i+1}) \right)$$

Método de Taylor:

Es un método explicito para el cual la función $f(x, y)$ debe poder derivarse varias veces, a partir del desarrollo en series de Taylor se llega a:

$$y(x_{i+1}) \simeq y(x_i) + h\ f(x_i, y(x_i)) + \frac{h^2}{2}\ f'(x_i, y(x_i)) + \ldots + \frac{h^n}{n!}\ f^{(n-1)}(x_i, y(x_i))$$

Por lo tanto, este método de orden n utiliza la expresión anterior para integrar la ecuación diferencial, partiendo de $y_0 = \overline{y_0}$.

A pesar de que se puede reducir el error de truncamiento su desventaja es que hay que derivar f(x,y) varias veces.

El método de Euler es un Método de Taylor de orden 1.

Runge-kutta:

Los métodos de Runge-Kutta son un conjunto de métodos genéricos iterativos, algunos son explícitos y otros implícitos. El más usado es Runge-Kutta de 4° Orden, existen muchas fórmulas para este método, una de ellas es:

$$y_{i+1} = y_i + \frac{1}{6}(F_1 + F_2 + F_3 + F_4)$$

Donde:

$$F_1 = h\ f(x_i, y_i)$$

$$F_2 = h\ f\left(x_i + \tfrac{1}{2}h,\ y_i + \tfrac{1}{2}F_1\right)$$

$$F_3 = h\ f\left(x_i + \tfrac{1}{2}h,\ y_i + \tfrac{1}{2}F_2\right)$$

$$F_4 = h\ f(x_i + h\ ,\ y_i + F_3)$$

Es de 4° orden porque contempla los términos hasta h^4. El error es de orden h^5.

ENUNCIADO

La trayectoria de una partícula que inicialmente se encuentra en el punto (−1, 1) y se mueve en el plano está dada por la curva $(x_1(t), x_2(t))$, donde las funciones x_1 y x_2 son la solución del siguiente sistema de ecuaciones diferenciales:

$$\begin{cases} x_1'(t) = -tx_2(t) \\ x_2'(t) = tx_1(t) - tx_2(t) \end{cases}$$

(a) Grafique la trayectoria de la partícula durante los primeros 20 segundos.

(b) Utilice el método de Runge-Kutta 4 con paso $h = 0.1$ para determinar la posición de la partícula y su rapidez a los tres segundos.

(c) ¿Cuántos dígitos correctos tienen los resultados del ítem anterior? Explique cómo lo determinó.

(d) Recuerde que la longitud de la trayectoria de la partícula durante los T primeros segundos está dada por $\int_0^T \sqrt{x_1'(t)^2 + x_2'(t)^2}\,dt$.

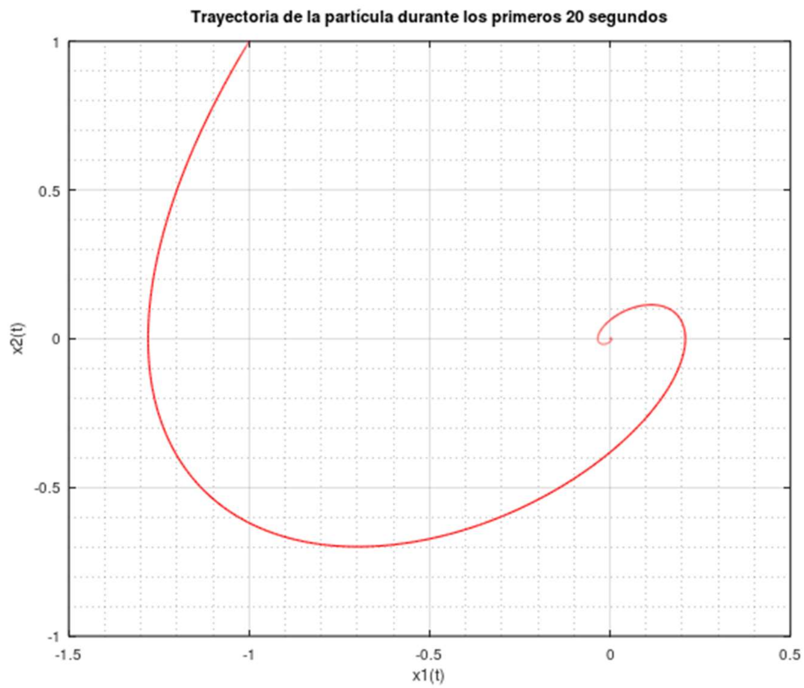
i) Realice interpolación de $x_1(t)$ y de $x_2(t)$ con los datos obtenidos en (b), por medio de spline cúbico sujeto. Explique cómo obtiene las condiciones de las derivadas en los extremos.

ii) Utilice la interpolación realizada anteriormente para estimar la distancia recorrida por la partícula durante los primeros tres segundos.

(e) Determine el instante de tiempo a partir del cual la partícula está siempre a una distancia menor a 0.01 del origen de coordenadas.

SOLUCIÓN

(a) Gráfica de la trayectoria de la partícula en el intervalo: $0 \leq t \leq 20$



Para obtener la gráfica se calcularon los valores de x_1 y x_2 en los primeros 20 segundos, utilizando el algoritmo de Runge-Kutta de orden 4 con un valor de h=0.01 con el fin de obtener mayor cantidad de puntos y así poder graficar, con el comando **'plot'** de Octave, una curva mejor definida.

(b) Posición y rapidez de la partícula cuando t = 3 segundos:

Utilizando nuevamente el algoritmo de Runge-Kutta 4 pero con un h = 0.1 obtenemos los valores de x_1 y x_2 en el instante $t = 3$ segundos.

$$x_1(3) = 2.019150006882205e - 01\ [m]$$
$$x_2(3) = 4.847275127510394e - 02\ [m]$$

La velocidad de la partícula se representa mediante el vector $\vec{v}$, el cual tiene la forma:

$$\vec{v} = [x_1'(t);\ x_2'(t)]$$
$$\vec{v} = [-tx_2(t)\ ;\ tx_1(t) - tx_2(t)]$$

La velocidad de la partícula en el instante t = 3 segundos es:

$$\vec{v} = [-3x_2(3)\ ;\ 3x_1(3) - 3x_2(3)]$$

$$\vec{v} = [-1.454182538253122e - 01\ ;\ 4.603267482393496e - 01]$$

La rapidez de la partícula para el instante dado es el módulo del vector $\vec{v}$ evaluado para t=3:

$$Rapidez = \left|\overrightarrow{v(3)}\right| = 4.827496076541300e - 01\ [m/_s]$$

Respuesta: Luego de 3 segundos de haber iniciado el recorrido, la partícula se encontraba en el punto $(0.201915\,, 0.048473)$ y se desplazaba con una rapidez de 0.482749 $[m/_s]$ aproximadamente.

(c) Dígitos correctos:

Para determinar la cantidad de dígitos correctos en el resultado obtenido en el punto anterior podemos repetir los cálculos con distinta cantidad de subintervalos (L), de manera que al ir incrementando estos subintervalos se obtienen resultados más precisos. De la misma forma, se llega a resultados más precisos decrementando el valor de h, debido a que L depende de este valor. $L = \frac{b-a}{h}$, donde a y b definen el intervalo [a, b].

Para el ítem (b) se nos pide un valor h = 0.1, por lo que

$$L = \frac{3-0}{0.1} = 30$$

L	$x_1(3)$	$x_2(3)$	Rapidez | v(3)|
1	-0.437500000000000	17.3125000000000	74.38458446647397
3	0.381537543402778	-0.262152777777778	2.085078362149295
6	0.222864565373206	0.05486367801153957	0.530197031181193
9	0.205510631535657	0.05058960733457615	0.488915625632897
12	0.202921425537341	0.04920892054999562	0.484191561740417
15	0.202283409144205	0.04877480732237871	0.483213052352318
18	0.202071546114576	0.04861039663493102	0.482928209515218
21	0.201986050106121	0.04853817357545782	0.482824860536715
24	0.201946484185603	0.04850267868271267	0.482781111456802
27	0.201926220468870	0.04848366012499470	0.482760356301136
30	0.201915000688221	0.04847275127510394	0.482749607654130

Analizando la tabla podemos observar que, en los últimos valores, tanto para x_1 y la rapidez |v| en el instante t = 3 segundos, pareciera que se obtienen tienen 4 cifras significativas ya que estas coinciden con el L = 27. Sin embargo, si calculamos con un L mucho mayor, L = 150 por ejemplo, los resultados son:

x_1 = 0.201895045861167

x_2 = 0.04845191994024090

|v| = 0.482733293801585

Vemos que el cuarto digito decimal de x_1 cambia para un valor grande de L, por lo que solo se debe considerar que para L = 30, x_1 y x_2 tienen 3 cifras significativas (o dígitos correctos), mientras que para |v| se tiene 4 dígitos correctos.

(d) I) interpolación con Spline cubico sujeto.
II) Estimar la distancia recorrida en los 3 primeros segundos.

Teniendo en cuenta la fórmula para la longitud de la trayectoria de la partícula en los primeros T segundos podemos determinar la distancia de la partícula en los primeros 3 segundos haciendo:

$$Distancia\ recorrida\ = \int_0^3 g(t)\,dt = \int_0^3 \sqrt{x_1'(t)^2 + x_2'(t)^2}dt$$

Donde $x_2'(t) = \ tx_1(t) - tx_2(t)$ y $x_1'(t) = \ -tx_2(t)$. (Ecuaciones dadas en el enunciado).

Existen varias formas de evaluar y obtener un resultado aproximado de esta integral (utilizando el método del Trapecio, método de Simpson, Cuadratura de Gauss, entre otros). Aquí se utilizó el algoritmo de Cuadratura de Gauss, el cual es bastante preciso. Sin embargo, para poder utilizarlo, debemos conocer primeramente las funciones $x_1(t)$ y $x_2(t)$ para luego poder definir la función g(t). Como no tenemos la representación analítica de estas soluciones, sino que solo tenemos una serie de puntos para cada solución (obtenidos con el método de Runge-Kutta), lo que se puede hacer es desarrollar una interpolación con dichos puntos. (El método que se utilizó para ello, fue el trazador cúbico sujeto). De esa manera se obtienen 2 funciones S_1 y S_2 que representan las soluciones $x_1\ y\ x_2$.

Una vez obtenidas las funciones S_1 y S_2 podemos representar las derivadas $x_1'(t)$ y $x_2'(t)$ de la siguiente forma:

$$x_1'(t) = \ -tS_2(t) \quad \text{y} \quad x_2'(t) = \ tS_1(t) - tS_2(t)$$

Por lo tanto, la función g(t) es:

$$g(t) = \ \sqrt{(-tS_2(t))^2 + (\ tS_1(t) - tS_2(t))^2}$$

La cual no es más que el módulo de la rapidez en función de t:

$$g(t) = \ \left|\overrightarrow{v(t)}\right|$$

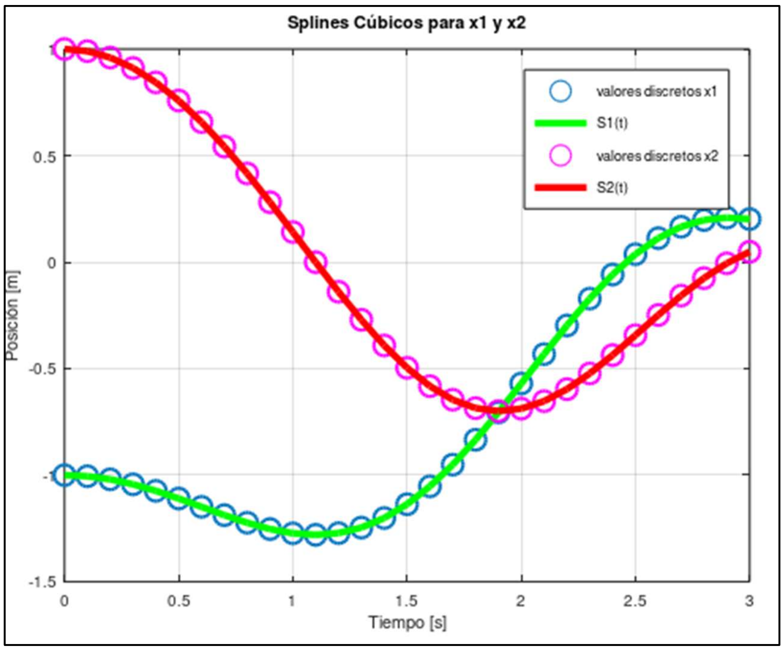
Con esta expresión de g utilizamos el algoritmo de Cuadratura de Gauss en el intervalo [0;3] para obtener el siguiente resultado:

$$Distancia\ recorrida\ (Cuad, Gauss) = \int_0^3 g(t)\,dt = 3.353123068095401\ [m]$$

Si comparamos este resultado y el obtenido al utilizar el Comando **integral(g,0,3)** de Octave vemos que los primeros 5 dígitos coinciden.

$$Dist.recorrida\ (Comando\ 'integral') = \int_0^3 g(t)\ dt\ = 3.353101039582637\ [m]$$

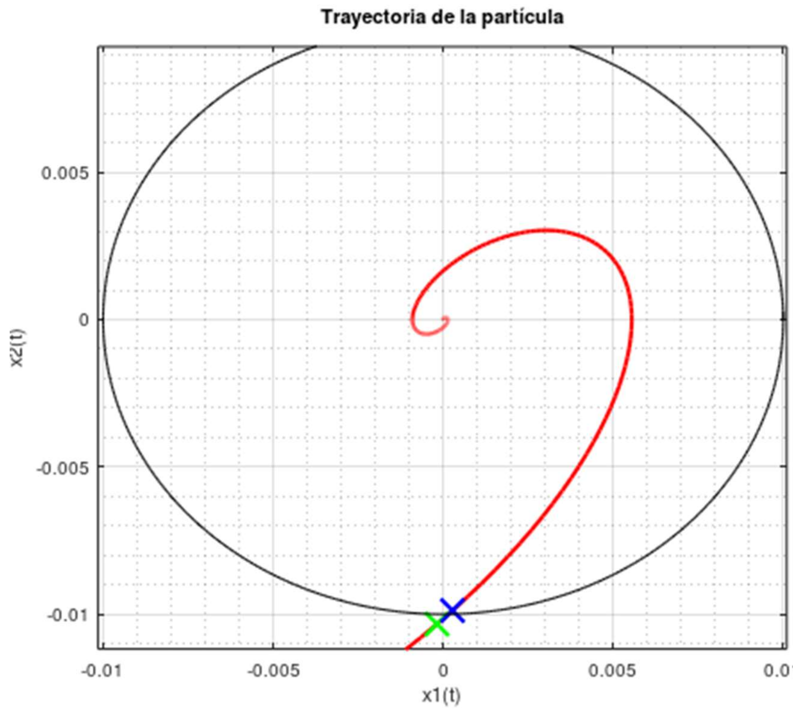
En la siguiente gráfica se observan las Spline $S_1(t)$ y $S_2(t)$ que interpolan los valores de $x_1(t)$ y $x_2(t)$ respectivamente.



(e) Tiempo a partir la distancia al origen es menor a 0.01

En la siguiente gráfica podemos observar la trayectoria de la partícula (en rojo), la cual inicia en el punto (-1,1) y luego se va acercando al origen. Además vemos un círculo (negro) que encierra la distancia de 0.01 al origen. Y dos equis que marcan el momento donde la partícula está a punto de llegar a esta distancia del origen (equis verde) y la otra equis (azul) marca un instante después a partir del cual la partícula está siempre a una distancia menor a ésta.

El valor de t en ese momento es aproximadamente de 4.54 segundos (para un valor de L=2000).



CÓDIGOS REALIZADOS EN OCTAVE PARA LA RESOLUCIÓN DEL EJERCICIO:

- Script para la consigna (a):

```
%TP3 Entregable 2022
%Script 1 para item (a)
clear all; clc; clf;

f = @(t,x) [-t*x(2);
            t*(x(1)-x(2))];

x0 = [-1;1]; %Punto Inicial

a = 0;        %t inicial
b = 20;       %t "final"
h = 0.01;     %paso
L = (b-a)/h;  %subintervalos

[t_RK4, x_RK4] = RungeKutta4(f,[a b],x0,L); %Obtengo las soluciones

#---------------------Grafico----------------------------------------
figure(1)

%Curva
plot(x_RK4(:,1),x_RK4(:,2),'r')

%Detalles de la gráfica
title('Trayectoria de la partícula durante los primeros 20 segundos')
xlabel('x1(t)')
ylabel('x2(t)')
grid on
grid minor
```

- <u>Script para la consigna (b) y (c):</u>

```
%Script 2 para item (b) y (c)
clear all; clc;
f = @(t,x) [-t*x(2);
            t*(x(1)-x(2))];
x0 = [-1;1];

#----------------------------inciso b)-----------------------------------
a = 0;
b = 3;
h = 0.1;
L = (b-a)/h   %L = 30
[t_RK4, x_RK4] = RungeKutta4(f,[a b],x0,L);
x1_t3 = x_RK4(end,1)                  %Posición en x1 para t=3s
x2_t3 = x_RK4(end,2)                  %Posición en x2 para t=3s
v = norm([-3*x2_t3; 3*x1_t3-3*x2_t3]) %Rapidez en t=3s
#-----------------------------------------------------------------------
#----------------------------inciso c)-----------------------------------
%Para ver cuantos digitos correctos tienen los valores anteriores
% variamos el L, cuanto mas grande sea mas preciso será el
% resultado, compararemos cuantos digitos coinciden con los
% resultados anteriores.
for i=1:50
  h = 1/i
  L = (b-a)/h

  [t_RK4, x_RK4] = RungeKutta4(f,[a b],x0,L);
  x1_t3 = x_RK4(end,1)                  %Posición en x1 para t=3s
  x2_t3 = x_RK4(end,2)                  %Posición en x2 para t=3s
  v = norm([-3*x2_t3; 3*x1_t3-3*x2_t3])   %Rapidez en t=3s
endfor
#-----------------------------------------------------------
```

- Script para la consigna (d):

```
%Script 3 para item (d)
f = @(t,x) [-t*x(2);
            t*(x(1)-x(2))];
x0 = [-1;1];
h = 0.1;
a = 0;
b = 3;
L = (b-a)/h;

[t_RK4, x_RK4] = RungeKutta4(f,[a b],x0,L);
x1 = x_RK4(:,1);
x2 = x_RK4(:,2);

#-----------------------------Inciso d-i)------------------------------------
#Calculo los coeficientes y desarrollo el Spline para x1 y para x2
[a1,b1,c1,d1] = Spline_Cubico_Sujeto(t_RK4,x1,0,0);
[a2,b2,c2,d2] = Spline_Cubico_Sujeto(t_RK4,x2,0,0);

xd= t_RK4;
n = length(a1);
S1 = @(w) a1(1)*(w==xd(1));
S2 = @(w) a2(1)*(w==xd(1));
for i = 1:n
  S1 = @(w) S1(w) + ...
  (a1(i) + b1(i).*(w - xd(i)) + c1(i).*(w - xd(i)).^2 + d1(i).*(w - xd(i)).^3).*(w>xd(i)).*(w<=xd(i+1));

  S2 = @(w) S2(w) + ...
  (a2(i) + b2(i).*(w - xd(i)) + c2(i).*(w - xd(i)).^2 + d2(i).*(w - xd(i)).^3).*(w>xd(i)).*(w<=xd(i+1));
endfor
xx = linspace(xd(1),xd(end),31);
#Grafico los Spline y observo si pasan por los puntos de x1(t) y x2(t)
figure(1);
plot(xd,x1,'o',"linewidth",1);
hold on
plot(xx,S1(xx),'g-',"linewidth",2);
plot(xd,x2,'om',"linewidth",1);
plot(xx,S2(xx),'r-',"linewidth",2);
title("Splines Cúbicos para x1 y x2");
xlabel("Tiempo [s]");
ylabel("Posición [m]");
legend('valores discretos x1','S1(t)','valores discretos x2','S2(t)');
hold off
#----------------------------------------------------------------------------
```

```
#-----------------------------Inciso d-ii)------------------------------------
%defino las derivadas a partir de la Ecuac Diferencial y las Splines
dx1 = @(t) (-t.*S2(t));
dx2 = @(t) t.*S1(t)- t.*S2(t);

g = @(t) (sqrt(dx1(t).^2 + dx2(t).^2)); %Integrando

I_Gauss = Cuadratura_Gauss_Comp (g,a,b,L,2)
I = integral(g,a,b)
#----------------------------------------------------------------------------
```

- <u>Script para la consigna (e):</u>

```
clear all; clc;
#Script 4 para el item (e):
f = @(t,x) [-t*x(2);
            t*(x(1)-x(2))];
x0 = [-1;1];
h = 0.01;
a = 0;
b = 20;
L = (b-a)/h

[t_RK4, x_RK4] = RungeKutta4(f,[a b],x0,L);

dist_origen = norm(x_RK4,'rows')  %Equivale a: sqrt(x_RK4(:,1).^2+x_RK4(:,2).^2)

pos = find(dist_origen<=0.01); %Encuentra el indice (pos) en el que la distancia al origen es menor o igual a 0.01
t_RK4(pos(1))
figure(2)
plot(x_RK4(:,1),x_RK4(:,2),'r-', "linewidth",1);  %Gráfica de la trayectoria (x1(t),x2(t))
hold on;
plot(x_RK4(pos(1)-1,1),x_RK4(pos(1)-1,2),'xg',"linewidth",1); %marca con una X un instante antes
                                                                % de llegar a la distancia de 0.01
plot(x_RK4(pos(1),1),x_RK4(pos(1),2),'xb',"linewidth",1); %marca un instante despues para el cuál la
                                                           % distancia es menor a 0.01
title('Trayectoria de la particula')
xlabel('x1(t)')
ylabel('x2(t)')

R = 0.01;
tita = (0:0.01:2.01*pi);
x_c = R*cos(tita);
y_c = R*sin(tita);
plot(x_c,y_c,'-k')      %Grafica un circulo encerrando la distancia que se pide (0.01) respecto al origen.
grid on
grid minor
pos(1)     %Primera posición en la que dist_origen<=0.01
t_RK4(pos(1)-1) %Valor antes de llegar a una distancia de 0.01 del origen
t_RK4(pos(1))  %Valor buscado. t aproximadamante 4.54 seg
```

CONCLUSIÓN

Con este trabajo pudimos observar como se obtiene una gráfica mejor definida cuando el paso ‘h’ es más pequeño, como también así una mejor precisión en los resultados.

Todos los incisos del problema se resolvieron aplicando el método de Runge-Kutta de orden 4. Sin embargo, se realizaron pruebas con otros métodos, como el de Forward Euler por ejemplo, pero se observó que tiene menor precisión por lo que se optó por usar Runge-Kutta 4.