Cálculo Numérico 2021
Entregable Número 4

Ejercicio Entregable: 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 (x1(t), x2(t)), donde las funciones x1 y x2 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 partıicula 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

$$\int_0^T \sqrt{x_1'(t)^2 + x_2'(t)^2}\ dt$$

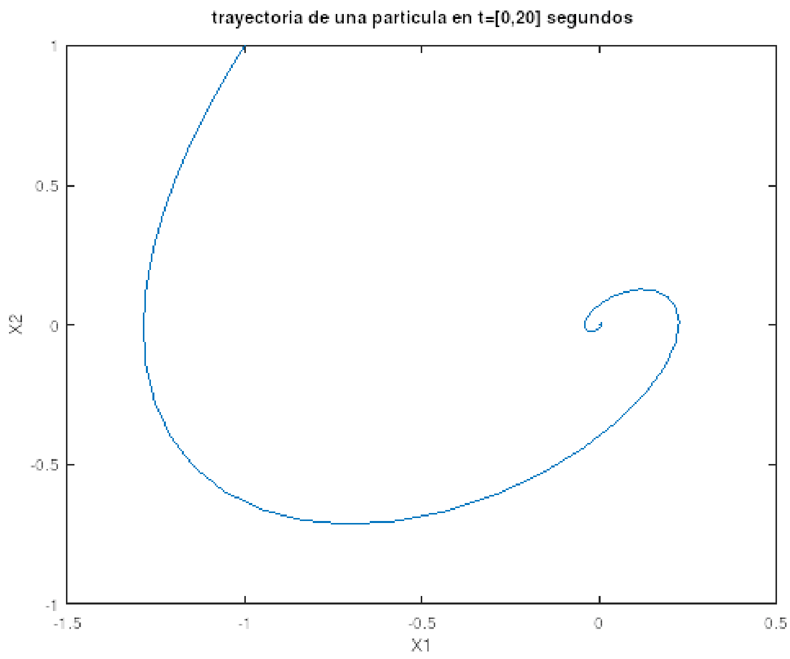
Calcule la distancia recorrida por la partícula durante esta dada por 0 los primeros tres segundos. Explique cómo lo hizo.
(e) Determine el instante de tiempo a partir del cual la partícula está siempre a una distancia menor a 0.01 del origen de coordenadas.

Resolución:

```
% TP Entregable 4
close all; clear; clc;
format long;

% Ecuacion diferencial:
f = @(t,x) [-t*x(2); t*x(1)-t*x(2) ];

% Inciso A --------------------------------------------------------------------
h = 0.1; y0 = [-1 ; 1];

% usamos runge_kutta_4 para graficar la trayectoria de la particula
[t,XY,~] = runge_Kutta_4(f,0,20,h,y0);

figure;plot(XY(1,:),XY(2,:));

% Inciso B --------------------------------------------------------------------
figure;plot(XY(1,1:31),XY(2,1:31));

disp('posicion de la particula a los tres segundos:');
disp(['posicion en X: ' num2str(XY(1,31))]);
disp(['posicion en Y: ' num2str(XY(2,31))]);
```

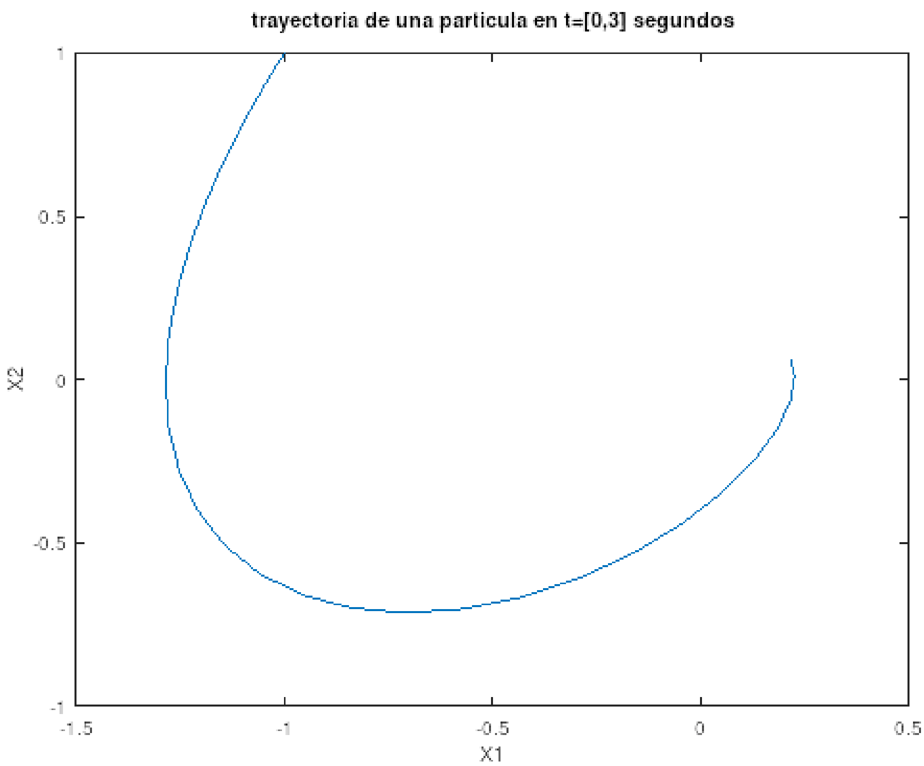
```
V = f(t(31),XY(:,31));
disp('');
disp('velocidad de la particula a los tres segundos:');
disp(['velocidad X: ' num2str(V(1))]);
disp(['velocidad Y: ' num2str(V(2))]);

disp('');
disp('rapidez de la particula a los tres segundos:');
disp(['rapidez: ' num2str(norm(V,2))]);

% Inciso C -------------------------------------------------------------------
hc = [1/10;1/20;1/40;1/80;1/160];

for i = 1:length(hc)
  [tc,XYc,~] = runge_Kutta_4(f,0,3,hc(i),y0);
  disp(['Soluciones con RK4 h: ' num2str(hc(i))]);
  XYc(:,end)'
end

% Inciso D -------------------------------------------------------------------
df = @(t,x) [-t.*x(2,:); t.*x(1,:)-t.*x(2,:) ];

% calculamos las derivadas de X1 y X2
Aux = df(t(1:31),XY(:,1:31));
% sqrt(dX1^2+dX2^2);
Xsq = sqrt( Aux(1,:).^2 + Aux(2,:).^2 );

disp(['longitud de la trayectoria (simpson): '
num2str(simpsonCompuestaDatos(t(1:31),Xsq))]);
disp(['longitud de la trayectoria (trapezoidal): '
num2str(trapezoidalCompuestaDatos(t(1:31),Xsq))]);


% Inciso E -------------------------------------------------------------------
distancias = zeros(1,length(t));
for i = 1:length(t)
  distancias(i) = norm(XY(:,i),2); % distancias con respecto al origen
end

it = find(distancias<0.01);
disp(['tiempo a partir de que la distancia se hace menor a 0.01: '
num2str(t(it(1))) 's']);
```

Para la realización del primer ejercicio se utilizó el método de runge-Kutta de orden 4 para la resolución de un sistema de ecuaciones diferenciales. para ello se utilizó un h = 0.1, y así tener también los datos para realizar el inciso b.



En la imagen anterior podemos observar cómo fue variando la posición de la partícula en los primeros 20 segundos partiendo desde el punto (-1,1).

Como ya tenemos las posiciones de la partícula en los primeros 20 segundos, de ahí tomamos los primeros 31 posiciones que se corresponderá a los primeros 3 segundos, dando así la siguiente gráfica:



y obteniendo así los siguientes resultados:

```
posicion de la particula a los tres segundos:
posicion en X: 0.21544
posicion en Y: 0.064772

velocidad de la particula a los tres segundos:
velocidad X: -0.19432
velocidad Y: 0.45199

rapidez de la particula a los tres segundos:
rapidez: 0.49199
```

para la realización del inciso c, la implementación que se ve en el código consiste en calcular la posiciones de la particula a los 3 segundos a traves del metodo de runge-kutta de orden 4 con distintos h, partiendo de h = 0.1 a h=0.00625, obteniendo así lo siguiente:

```
Soluciones con RK4 h: 0.1
ans =

   2.154364265128252e-01   6.477243089481563e-02

Soluciones con RK4 h: 0.05
ans =

   2.081344953390552e-01   5.632330775021896e-02

Soluciones con RK4 h: 0.025
ans =

   2.048906746581178e-01   5.231309189994903e-02

Soluciones con RK4 h: 0.0125
ans =

   2.033628626450482e-01   5.036363695764285e-02

Soluciones con RK4 h: 0.00625
ans =

   2.026215712269286e-01   4.940302374126489e-02
```

Sabemos que a medida que disminuye h, el error de truncamiento disminuye. Por lo que podemos apreciar que el la diferencia entre el mayor h y el menor h empiezan a partir del segundo decimal, por lo que podemos decir que tiene dos dígitos correctos a partir del punto decimal.

Para la realización del inciso D, se calculó las derivadas de x1 y x2 en los primeros 3 segundos con un salto de 0.1, luego se sumaron los cuadrados de estos y se obtuvo la raíz, con el fin de obtener los valores para integrar numéricamente en el intervalo de tiempo [0,3]. Y así obtener la longitud de la trayectoria. (para realizar la integral se utilizó el método del trapecio y el método de simpson para comparar los resultados).

```
longitud de la trayectoria (simpson): 3.3905
longitud de la trayectoria (trapezoidal): 3.3878
```

Por último en la realización del inciso E se calculó la norma dos de los puntos con respecto al origen, con el fin obtener la distancia a este. Se busco el tiempo en el cual la distancia con respecto al origen sea menor a 0.01, dando los siguientes resultados:

```
tiempo a partir de que la distancia se hace menor a 0.01: 4.6s
```

Otras funciones utilizadas:

```
% -------------------------------------------------------------------
function [t,y,n] = runge_Kutta_4(f,a,b,h,y0)
  [fil,~] = size(y0); % tiene que ser un vector columna

  n = (b-a)/h;
  t = zeros(1,n+1);
  y = zeros(fil,n+1);
  t(1) = a;
  y(:,1) = y0;

  for i = 1:n
    K1 = h*f(t(i)      , y(:,i));
    K2 = h*f(t(i)+0.5*h, y(:,i)+0.5*K1);
    K3 = h*f(t(i)+0.5*h, y(:,i)+0.5*K2);
    K4 = h*f(t(i)+h    , y(:,i)+0.5*K3);

    y(:,i+1) = y(:,i)+(K1+2*K2+2*K3+K4)/6;
    t(i+1) = t(i)+h;
  end

end

% -------------------------------------------------------------------
function I = simpsonCompuestaDatos(X,Y)
  n = length(X)-1;         % n numero de subintervalos
  h = (X(end)-X(1))/n;     % (b-a)/n

  f_x0 = Y(1);
  f_xn = Y(end);
  Y = Y(2:n);

  % impares
  si = sum(Y(2*[1:n/2]-1));
  % pares
  sp = sum(Y(2*[2:n/2]-2));

  % formula de Simpson
  I = (h/3)*(f_x0+2*sp+4*si+f_xn);
end
```

```
% --------------------------------------------------------------------
function I = trapezoidalCompuestaDatos(X,Y)
  N = length(X)-1;
  I = 0;
  n = 1:N;
  I = 0.5*(sum( (X(n+1)-X(n)).*(Y(n+1)+Y(n)) ));
end
```