Laboratorio 8 EDO

1. Nociones Básicas

El objetivo de ese laboratorio es que aprendan a implementar la resolución de ecuaciones diferenciales ordinarias (EDO) en Matlab, utilizando para aquello los conocimientos aprendidos en clase más la información que se complementará en este laboratorio. En este laboratorio miraremos el Método de Euler, el Método de Taylor de orden superior y Runge Kutta de Orden 2.

- ### 1.1. Método de Euler

El método de euler aproxima la ecuación diferencial de la forma:

$$y' = f(x, y), \quad y(x_o) = y_o \tag{1}$$

Luego partiendo desde la condición inicial, deseamos encontrar una solución. Usamos pequeños intervalos (o saltos) $h = \frac{b-a}{N}$, donde $[a, b]$ es el intervalo donde queremos encontrar la solución,tal que si utilizamos la condición inicial como punto de partida, tenemos que la solución es:

$$x_{n+1} = x_n + h \tag{2}$$

$$y_{n+1} = y_n + hf(x_n, y_n) \tag{3}$$

Y el error asociado a esta aproximación es de orden de:

$$E_a = \frac{f'(x, y)}{2!} h^2 \tag{4}$$

Haga una función en Matlab que realice el método de Euler y resuelva la siguiente EDO en el intervalo $x \in [0, 1]$ con un $N = [10, 50, 100, 500]$:

$$y' = 3 + x - y \quad y(0) = 2 \tag{5}$$

Cuando la haga la función *euler.m* piense que debe dar como resultado un conjunto de puntos que son la respuesta a la EDO.
Una vez resuelta la función, represéntela en un gráfico y junto a ella la solución de la EDO 5 que es:

$$y = 2 + x - e^{-x} \tag{6}$$

Calcule el error de su aproximación, para esto utilice el siguiente comando: `norm(yr-ya)`, donde `yr` es el vector de la solución real e `ya` el vector de la solución aproximada. Gráfique el error respecto a N.

1.2. Método de Taylor

El método de Taylor consiste en utilizar la expansión de polinomios de Taylor para resolver el problema:

$$\frac{dy}{dt} = f(t,y) \quad a \leq t \leq b \tag{7}$$
$$y(a) = \alpha \tag{8}$$

Veamos la expansión de Taylor de y(t):

$$y(t) = y(t_i) + y'(t_i)(t-t_i) + y''(t_i)\frac{(t-t_i)^2}{2} + y^{(3)}(t_i)\frac{(t-t_i)^3}{6} + ... + \frac{(t-t_i)^{n+1}}{(n+1)!}y^{n+1}(t_i) \tag{9}$$

Luego consideremos si evaluamos en $t = t_{i+1}$ y tenemos que$h = t_{i+1} - t_i$, además sabemos que $y'(t_i) = f(t_i, y_i)$, por lo que tenemos:

$$y(t_{i+1}) = y(t_i) + f(t_i,y_i)h + f'(t_i,y_i)\frac{h^2}{2} + f(t_i,y_i)''\frac{h^3}{6} + ... + \frac{h^{n+1}}{(n+1)!}f^{(n)}(t_i,y_i) \tag{10}$$

Esta última ecuación define el método de Taylor de orden n. Tal que:

$$y_0 = \alpha \tag{11}$$
$$y_{i+1} = y_i + f(t_i,y_i)h + f'(t_i,y_i)\frac{h^2}{2} + f(t_i,y_i)''\frac{h^3}{6} + ...f^{(n-1)}(t_i,y_i)\frac{h^n}{n!}, i = 0,..,N-1 \tag{12}$$

Veamos el método de Taylor de Orden 2:

$$y_0 = \alpha \tag{13}$$
$$y(t_{i+1}) = y(t_i) + f(t_i,y_i)h + f'(t_i,y_i)\frac{h^2}{2} \tag{14}$$

El error local de esta formula será del orden del término desconocido, o sea $O(h^3)$. Luego para el orden 4:

$$y_0 = \alpha \tag{15}$$
$$y(t_{i+1}) = y(t_i) + f(t_i,y_i)h + f'(t_i,y_i)\frac{h^2}{2} + f(t_i,y_i)''\frac{h^3}{6} + f(t_i,y_i)'''\frac{h^4}{4!} \tag{16}$$

En este caso el error local es del orden de $O(h^5)$. Es fácil notar que el método de Euler es equivalente al método de Taylor de orden 1. Además se requiere calcular la derivada de la función.
Ahora se le pide crear una función para Taylor de Orden 2 y resuelva la ecuación 5, calculando el error correspondiente. Además resuelvan la siguiente ecuación:

$$y' = ty, \quad 0 \leq t \leq 2 \tag{17}$$
$$y(0) = 1 \tag{18}$$

Cuya solución es $y(t) = e^{\frac{t^2}{2}}$.

Tomar en consideración que MATLAB puede calcular la derivada **sólo** si definimos la función de manera simbólica tal que:

```
syms x
```
$f = x^3 + 4x^2$;
```
diff(f,1)
```
Esto entregaría la primera derivada de f. Si no tenemos la función simbólica debemos calcular la derivada nosotros, por lo que en ese caso habrá que hacer una función específica para cada EDO.

1.3. Método de Runge Kutta Orden 2

El problema que existe del método de Taylor es el cálculo de la derivada de la función $f(x, y)$. Es por esto que para lograr una aproximación del mismo orden que Taylor, pero sin utilizar derivadas, se utiliza el método de Runge-Kutta, el cual tiene una forma parecida a la del método de euler $y_{i+1} = y_i + hf(t_i, y_i)$ donde se reemplaza f por un promedio ponderado de valores de f. Veamos el caso de Runge-Kutta de orden 2 para el problema de EDO clásico en el intervalo $[a, b]$ y $h = \frac{b-a}{N}$ siendo N el número de subdivisiones:

$$y_0 = \alpha \tag{19}$$
$$k_1 = hf(t_i, y_i) \tag{20}$$
$$k_2 = hf(t_i + h, y_i + k_1) \tag{21}$$
$$y_{i+1} = y_i + \frac{1}{2}(k_1 + k_2) \tag{22}$$

Se le pide generar una función llamada *runge2.mat* que realice la integración por el método explicado anteriormente y resuelva las ecuaciones 5, 18 y además la siguiente ecuación:

$$y' = \frac{5x^2 - y}{e^{x+y}} \quad 0 \leq x \leq 1 \tag{23}$$
$$y(0) = 1 \tag{24}$$

Con $N = 10$ para la ecuación 24 y con $N = [10, 50, 100]$ para las ecuaciones 5 y 18.
¿Podrían solucionar el problema 24 de manera analítica?