TALLER 4

Lenguajes de Programación 2

En este cuarto taller de lenguajes de programación 2 vamos a colocar en práctica lo visto hasta ahora en el curso. El taller será solucionado en los grupos que ustedes han conformado. Es importante que todos los VIs estén documentados para entender la lógica del código.

Los códigos serán sustentados el próximo jueves 01 de diciembre de forma aleatoria entre los miembros de cada grupo en un pequeño espacio de tiempo. Y hará parte de la calificación final del taller.

El taller debe ser enviado como máximo el jueves 01 de diciembre hasta 10am, si lo tienen antes sería mucho mejor. El archivo debe ser cargado en un Formulario de Google Forms que enviaré para ustedes.

LABVIEW

Métodos de Runge-Kutta

El método de Runge-Kutta no es sólo un único método, sino una importante familia de métodos iterativos, tanto implícitos como explícitos, para aproximar las soluciones de ecuaciones diferenciales ordinarias (E.D.O´s); estas técnicas fueron desarrolladas alrededor de 1900 por los matemáticos *alemanes Carl David Tolmé Runge* y *Martin Wilhelm Kutta*.

Métodos de Runge-Kutta de cuarto orden

Un miembro de la familia de los métodos Runge-Kutta usado ampliamente es el de cuarto orden. Es usado tanto que a menudo es referenciado como «RK4» o como «el método Runge-Kutta».

Definiendo un problema de valor inicial como:

$$y' = f(x, y), \qquad y(x_0) = y_0$$

Entonces el método RK4 para este problema está dado por la siguiente ecuación:

$$y_{i+1} = y_i + \frac{1}{6}\, h\, (k_1 + 2k_2 + 2k_3 + k_4)$$

Donde

$$\begin{cases} k_1 = f(x_i, y_i) \\ k_2 = f(x_i + \frac{1}{2}h, \qquad y_i + \frac{1}{2}k_1 h\,) \\ k_3 = f(x_i + \frac{1}{2}h, \qquad y_i + \frac{1}{2}k_2 h\,) \\ k_4 = f(x_i + h, \qquad y_i + k_3 h\,) \end{cases}$$

Así, el siguiente valor (y_{n+1}) es determinado por el presente valor (y_n) más el producto del tamaño del intervalo (h) por una pendiente estimada. La pendiente es un promedio ponderado de pendientes, donde k_1 es la pendiente al principio del intervalo, k_2 es la pendiente en el punto medio del intervalo, usando $k_1$para determinar el valor de **y** en el punto $x_n + \frac{h}{2}$ usando el método de Euler. k_3 es otra vez la pendiente del punto medio, pero ahora usando k_4 para determinar el valor de **y**; k_4 es la pendiente al final del intervalo, con el valor de **y** determinado por k_3. Promediando las cuatro pendientes, se les asigna mayor peso a las pendientes en el punto medio:

$$pendiente = \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}$$

La ecuación diferencial puede ser calculada en un intervalo [a,b]. Como este es un método iterativo, puede ser implementado dentro de un ciclo FOR donde el número de veces que se repite el ciclo viene dado por:

$$N = \frac{b - a}{h}$$

Se pide crear un VI que resuelva la siguiente ecuación diferencial con valor inicial usando un Runge Kutta de 4to orden:

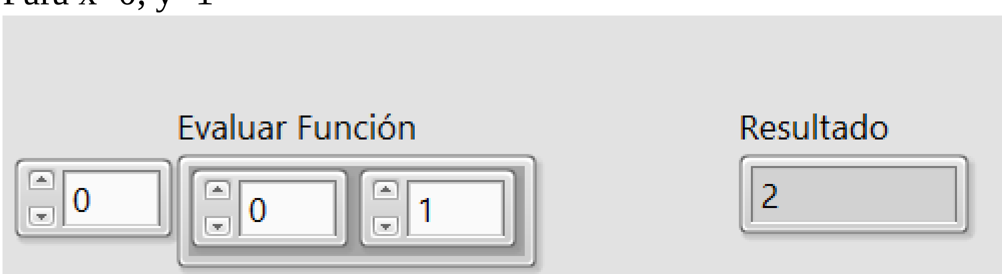
$$\frac{dy}{dx} = (y+1)(x+1)\cos(x^2+2x)\,,\quad y(0) = 4$$

Cuya solución analítica es:

$$y(x) = 5\,\exp\left(\frac{1}{2}sen(x^2+2x)\right) - 1$$

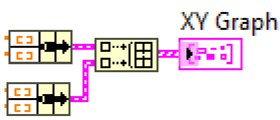
Un procedimiento paso a paso con incrementos de $h = 0.5$ usando Runge Kutta de la ecuación anterior puede ser visto en el siguiente video: https://www.youtube.com/watch?v=Wwf7HrjPcTM

Requisitos

1. Implementar por su cuenta el método de Runge Kutta, si lo desea puede usar el propio bloque de Runge Kutta de LabView para comparar el resultado.
2. Crear un SubVi que contenga la parte derecha de la ecuación diferencial
   $$(y+1)(x+1)\cos(x^2+2x)$$
   Y reciba un **VECTOR** con los valores de *x* y *y* que desean ser evaluados. Aplique todas las prácticas de documentación para el subvi como visto en clase, diseñe un ícono para el subvi y coloque la entrada como conexión requerida. (Se suministra un SubVI de ejemplo llamado **dydx** para que vea como colocando dos puntos xy retorna el resultado de la ecuación. Eleabore ud mismo(a) este subvi)
   Para x=0, y=1
   

3. Crear un SubVi que contenga la solución analítica
   $$5\,\exp\left(\frac{1}{2}sen(x^2+2x)\right) - 1$$
   Y reciba un **VECTOR** con los valores de *x* y *y* que desean ser evaluados. Aplique todas las prácticas de documentación para el subvi como visto en clase, diseñe un ícono para el subvi y coloque la entrada como conexión requerida. (SubVI fanalitica de ejemplo)

4. En el panel frontal, el usuario podrá escoger los intervalos de la ecuación. Estos datos deben ser ingresados en un **VECTOR**.
5. En el panel frontal el usuario puede escoger el tamaño del intervalo (h)
6. En un grafico XY grafique la solución analítica con un tamaño de intervalo de 0.01, con el objetivo que la gráfica analítica se vea suavizada en la gráfica.
7. En el mismo gráfico XY grafique los puntos calculados con Runge Kutta representado por cuadrados rojos y espaciados en el tamaño del intervalo ingresado por el usuario, es decir el intervalo (h)

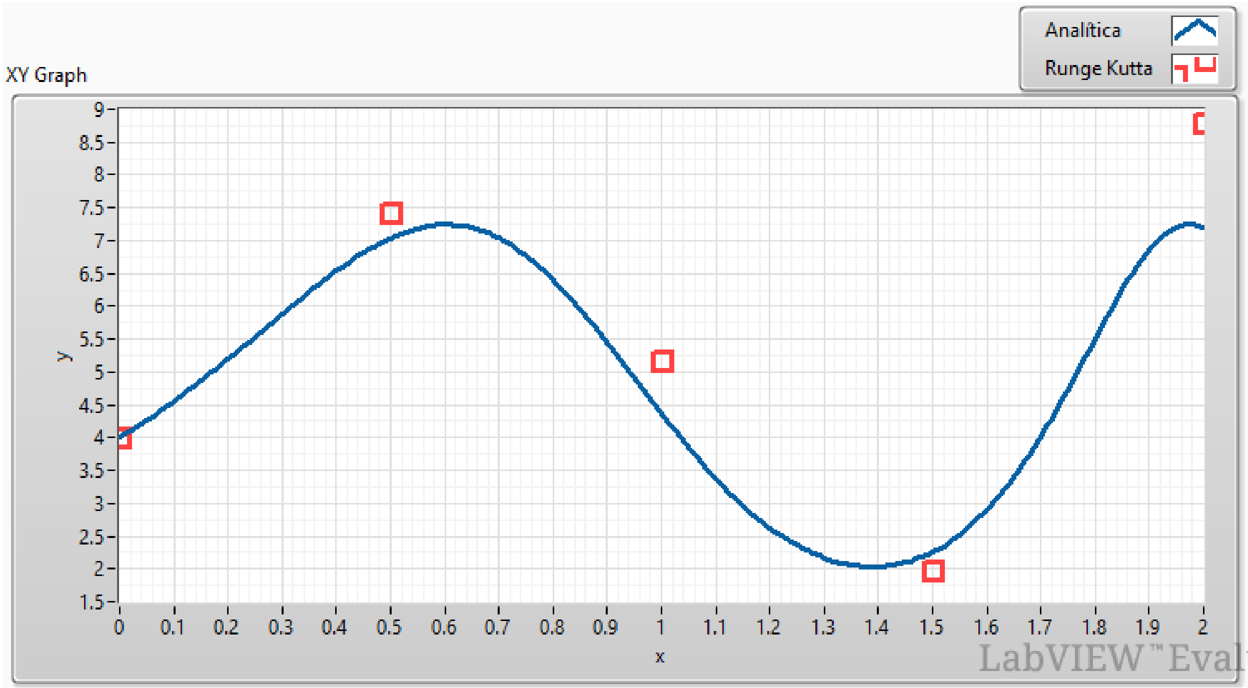


8. En el panel frontal muestre una **MATRIZ** con los **valores x** y los **valores y** de acuerdo con el cálculo hecho por el método RK4
9. En el panel frontal muestre una **MATRIZ** donde cada columna muestre el cálculo de las pendientes k_1, k_2, k_3, k_4 calculadas en cada iteración.

Los valores que tomaría **y** solucionando por el método RK4 para un intervalo [a,b]=[0,2] con un h=0.5 sería:

i	xi	yi
0	0.0	4
1	0.5	7.440973
2	1.0	5.183458
3	1.5	1.986632
4	2.0	8.791971

Note que, para ese caso, el ciclo FOR se ejecuta 5 veces por lo tanto muestra 5 puntos en la gráfica bien próximos de la solución analítica



Data (index 0, 0)

x	y
0	4
0.5	7.44097
1	5.18346
1.5	1.98663
2	8.79197
0	0

constantes K (index 0, 0)

k1	k2	k3	k4
5	6.60879	7.03407	9.00596
3.99244	-7.7988	-5.36325	-4.75852
-12.2432	-4.25142	-6.97155	-3.67282
3.82353	10.4217	14.7822	27.4327
-4.2742	-28.0827	-8.92145	-17.1625
0	0	0	0

Utilice los ciclos FOR con auto-indexing para que se le facilite el tratamiento de vectores, también puede usar los shift register.