


ELC-30524
Sistemas de Potencia II



Anexo 2.2.
Métodos Numéricos

Prof. Francisco M. González-Longatt

fglongatt@ieee.org

<http://www.giaelec.org/fglongatt/SP2.htm>

Ejemplo

- Para introducir de forma directa al uso de los métodos numéricos en la resolución de ecuaciones diferenciales, considere la siguiente ecuación.

$$\begin{cases} \frac{dy(t)}{dt} = y(t) & t \in [0,1] \\ y(0) = 1 \end{cases}$$

A. Método de Euler

- Inicialmente se resuelve aplicando el método de Euler hacia delante, para un $h = 0.2$, $n = 5$

$$Y_{i+1} = Y_i + hF_i$$

- Se tiene:

Solución por el Método de Euler con $h = 0.2$, $n = 5$.

i	t_i	Y_i
1	0.2	1.2
2	0.4	1.44
3	0.6	1.728
4	0.8	2.0736
5	1.0	2.48832

A. Método de Euler

- Si se compara con la solución exacta:

$$e^1 - Y(1) = 0.23$$

- Se tiene que el error absoluto es:

$$\mathcal{E}_{abs} = 0.23$$

- Ahora si se considera un paso de integración más pequeño, $h = 0.02$, el error se reduce:

$$\mathcal{E}_{abs} = 0.0267$$

A. Método de Euler

- Ahora si se considera un paso de integración más pequeño, $h = 0.02$.

Solución por el Método de Euler con $h = 0.02$, $n = 50$.

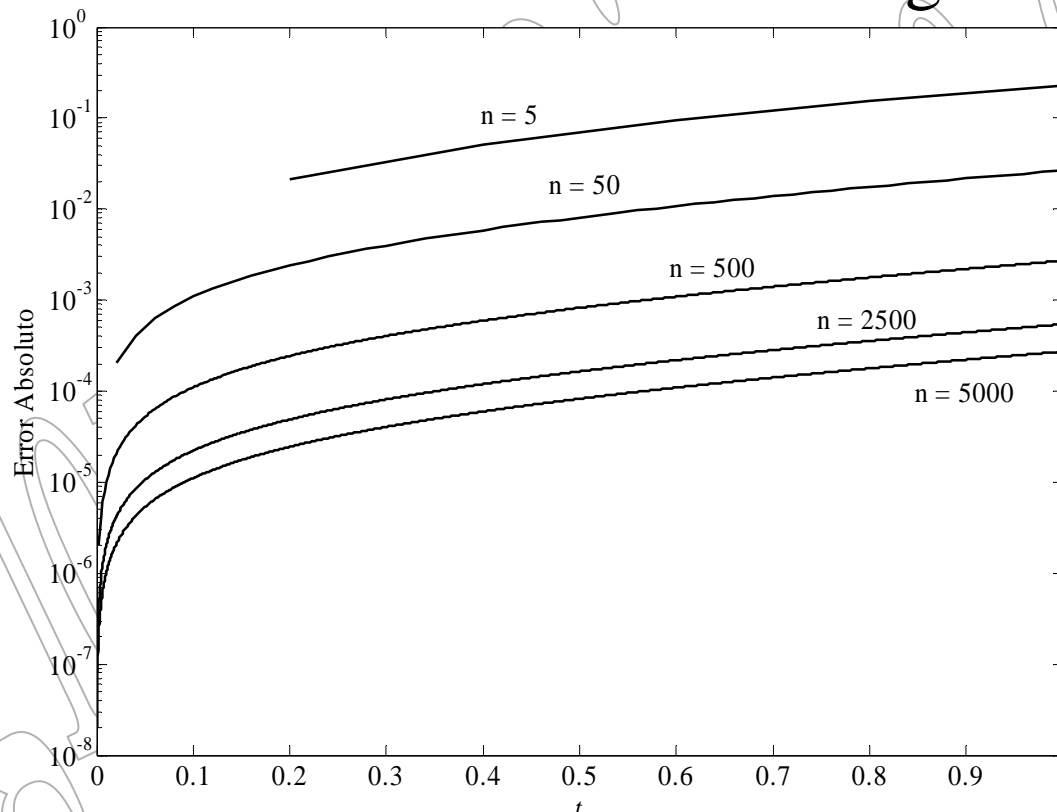
i	t_i	Y_i
1	0.2	1.02
2	0.4	1.48595
3	0.6	1.81136
4	0.8	2.20804
5	1.0	2.69159

- El error se reduce:.

$$\varepsilon_{abs} = 0.0267$$

A. Método de Euler

- El trazado del error absoluto para un número superior de sub-intervalos se muestra en la Figura.



Comparación del error absoluto con el método de Euler,
medido a la solución analítica, para $n = \{5, 50, 500, 2500, 5000\}$

B. Método de Runge-Kutta de 2^{do} Orden

- Empleando Runge-Kutta de segundo orden:

$$\begin{cases} \bar{Y}_{i+\frac{1}{2}} = Y_{i+1} + \frac{h}{2} F(t_i, Y_i) \\ Y_{i+1} = Y_{i+1} + \frac{h}{2} F\left(t_i + \frac{h}{2}, \bar{Y}_{i+\frac{1}{2}}\right) \end{cases}$$

- Para varios números de sub-intervalos.

B. Método de Runge-Kutta de 2^{do} Orden

- Para varios números de sub-intervalos, los resultados se muestran en la Tabla

Solución por el Método De Runge-kutta de 2^{do} orden para varios valores de n

i	t_i	$Y_i (n=5)$	$Y_i (n=50)$	$Y_i (n=500)$
1	0.2	1.2	1.22139	1.2214
2	0.4	1.4884	1.49179	1.49182
3	0.6	1.8150	1.82205	1.82212
4	0.8	2.215	2.22542	2.22554
5	1.0	2.7027	2.7181	2.71828

B. Método de Runge-Kutta de 2^{do} Orden

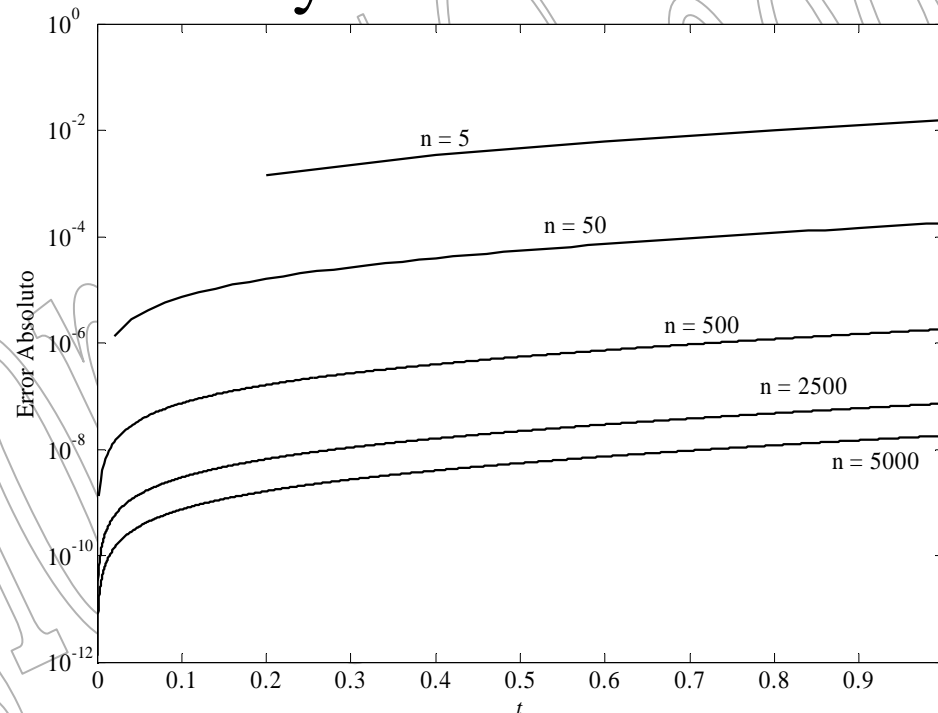
- El error absoluto resulta ser:

$$\varepsilon_{abs} = 0.0156$$

- Para el caso particular de $n = 5$, lo cual es muy inferior al obtenido para el mismo paso de integración con el *Método de Euler hacia delante*.

B. Método de Runge-Kutta de 2^{do} Orden

- En la Figura, se muestra la grafica del error absoluto medido desde la solución analítica para un número de sub-intervalos mayor



Comparación del error absoluto con el método de Runge-Kutta de 2^{do} orden, medido a la solución analítica, para $n = \{5, 50, 500, 2500, 5000\}$

C. Método de Runge-Kutta de 4^{to} Orden

- Se procede a la solución por el método de Runge-Kutta de 4to orden:

$$\left\{ \begin{array}{l} Y_{i+1} = Y_i + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 = F(t_i, Y_i) \\ k_2 = F\left(t_i + \frac{h}{2}, Y_i + \frac{1}{2}hk_1\right) \\ k_3 = F\left(t_i + \frac{h}{2}, Y_i + \frac{h}{2}k_2\right) \\ k_4 = F(t_i + h, Y_i + hk_3) \end{array} \right.$$

C. Método de Runge-Kutta de 4^{to} Orden

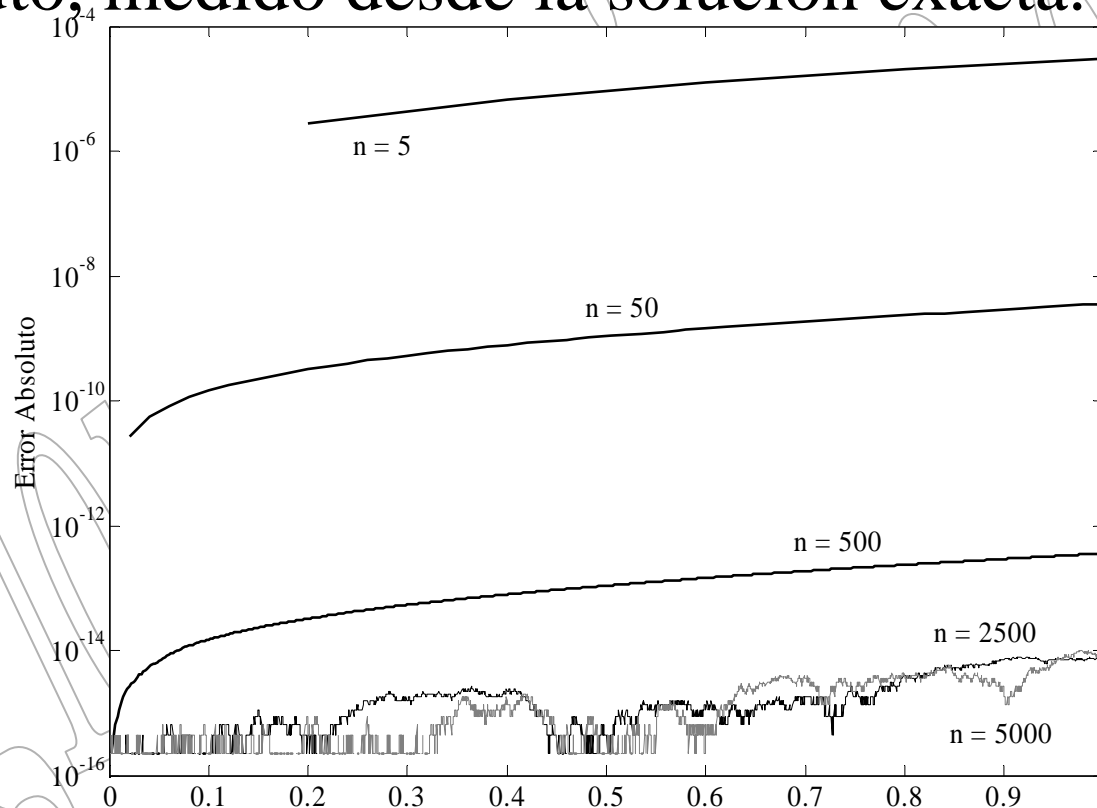
- Se procede a la solución por el método de Runge-Kutta de 4to orden con varios pasos de integración, los resultados se muestran en la Tabla:

Solución por el Método de Runge-Kutta de 2^{do} orden
para varios valores de n

i	t_i	$Y_i (n=5)$	$Y_i (n=50)$	$Y_i (n=500)$
1	0.2	1.2214	1.2214	1.2214
2	0.4	1.4918	1.49182	1.49182
3	0.6	1.8221	1.82212	1.82212
4	0.8	2.2255	2.22554	2.22554
5	1.0	2.7183	2.71828	2.71828

C. Método de Runge-Kutta de 4^{to} Orden

- En la Figura siguiente se ha graficado el error absoluto, medido desde la solución exacta.



Comparación del error absoluto con el método de Runge-Kutta de 4^{to}, medido a la solución analítica, para $n = \{5, 50, 500, 2500, 5000\}$

C. Método de Runge-Kutta de 4^{to} Orden

- El error absoluto para $n = 5$ resulta:

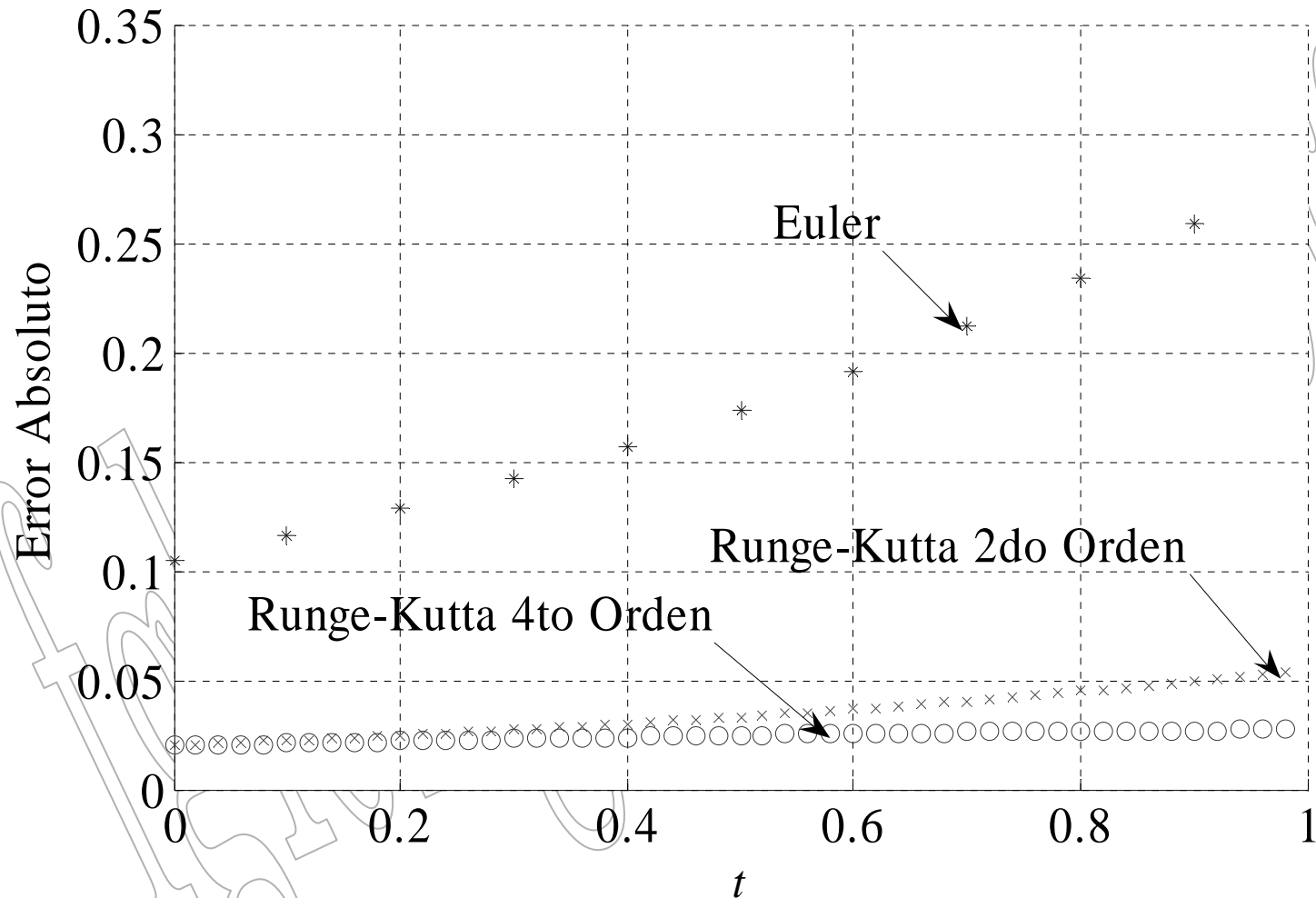
$$\varepsilon_{abs} = 1.8175 \times 10^{-5}$$

- El cual es el menor error comparado con los métodos de Euler y Runge-Kutta de menor orden.

D. Comparación entre Métodos

- Al efectuar el trazado del error absoluto en la solución de la ecuación diferencial para un mismo paso de integración ($n=10$), para los diferentes métodos de integración: Euler, Runge-Kutta de 2^{do} y 4^{to} orden.
- Se evidencia, que los métodos que presentan mayor error son: Euler, Runge Kutta de 2^{do} y de 4^{to} orden en ese mismo orden de mayor a menor.

D. Comparación entre Métodos



Comparación del Error Absoluto, para varios métodos para $n = 10$.

D. Comparación entre Métodos

- Se evidencia, que los métodos que presentan mayor error son: Euler, Runge Kutta de 2^{do} y de 4^{to} orden en ese mismo orden de mayor a menor.
- Esto es consistente con la teoría que afirma que el error de Euler es $O(h)$, mientras que para Runge Kutta de 2do y 4to orden son $O(h^3)$ y $O(h^5)$, respectivamente.

D. Comparación entre Métodos

Resultados con los Diferentes Métodos Numéricos

i	x_i	$Y(x_i)$ <i>Euler</i>	$Y(x_i)$ <i>Runge-Kutta 2^{do}</i> <i>Orden</i>	$Y(x_i)$ <i>Runge-Kutta 4^{do}</i> <i>Orden</i>
0	0	1.0000	1.0000	1.0000
1	0.2	1.2000	1.2200	1.2214
2	0.4	1.4400	1.4884	1.4918
3	0.6	1.7280	1.8158	1.8221
4	0.8	2.03736	2.2153	2.2255
5	1	2.4883	2.7027	2.7183

Conclusiones

- Se ha enfrentado el problema de los métodos numéricos para resolver ecuaciones diferenciales en forma simple y directa
- El tratamiento dado aquí es simple y es un intento para introducir al análisis numérico a las personas no expertas.
- En este caso se resolvió la ecuación diferencial lineal ordinaria de primer orden $y'(t)=y$, $y(0)=1$, $t \in [0,1]$ en forma rápida y simple por tres métodos numéricos: Euler, Runge-Kutta 2^{do} orden, y Runge-Kutta 4^{to} orden; y variando el paso de integración.

Conclusiones

- Se empleo Matlab™.
- Los resultados son consecuentes con la teoría, mostrando la conveniencia para resultados de menor error el Método de Runge-Kutta de 4^{to} orden.