Comparación de Métodos Numéricos para la Solución Ecuación Diferencial de 1er Orden

Francisco M. Gonzalez-Longatt

***Resumen*—Este documento presenta un comparación de los resultados obtenidos por tres métodos numéricos: Euler, Runge-Kutta 2do orden, y Runge-Kutta 4to orden, en la solución de la ecuación diferencial lineal ordinaria de primer orden y´(x)=y, y(0)=1. La comparación es efectuada para diferentes números de subintervalos, mostrado el mejor**

***Índice de Términos*— Ecuaciones diferenciales, métodos numericos, Euler, Runge-Kutta.**

I. Generalidades

Considere la Ecuación Diferencial Ordinaria de Primer Orden a condiciones iniciales:

$$\begin{cases} \dfrac{dy(x)}{dx} = y(x) & x \in [0,1] \\ y(0) = 1 \end{cases} \qquad (1)$$

Se supone que la función *y(x)* es continua sobre $x \in [0,1]$.

II. Método de Euler

Para n = 5, empleando el método de Euler se tiene:

```
X =
      0    0.2000    0.4000    0.6000    0.8000
Y =
   1.0000    1.2000    1.4400    1.7280    2.0736    2.4883
```

Comparando con el valor real $e^1 - Y(1) = 0.23$. El error absoluta resulta ser: $\varepsilon_{abs} = 0.23$. Para n = 50; se tiene:

```
Y =

  Columns 1 through 9
   1.0000    1.0200    1.0404    1.0612    1.0824    1.1041    1.1262    1.1487    1.1717
  Columns 10 through 18
   1.1951    1.2190    1.2434    1.2682    1.2936    1.3195    1.3459    1.3728    1.4002
  Columns 19 through 27
   1.4282    1.4568    1.4859    1.5157    1.5460    1.5769    1.6084    1.6406    1.6734
  Columns 28 through 36
   1.7069    1.7410    1.7758    1.8114    1.8476    1.8845    1.9222    1.9607    1.9999
  Columns 37 through 45
   2.0399    2.0807    2.1223    2.1647    2.2080    2.2522    2.2972    2.3432    2.3901
  Columns 46 through 51
   2.4379    2.4866    2.5363    2.5871    2.6388    2.6916
```

El error resulta ser: $\varepsilon_{abs} = 0.0267$. El trazado del error absoluto para un número superior de subintervalos resulta:

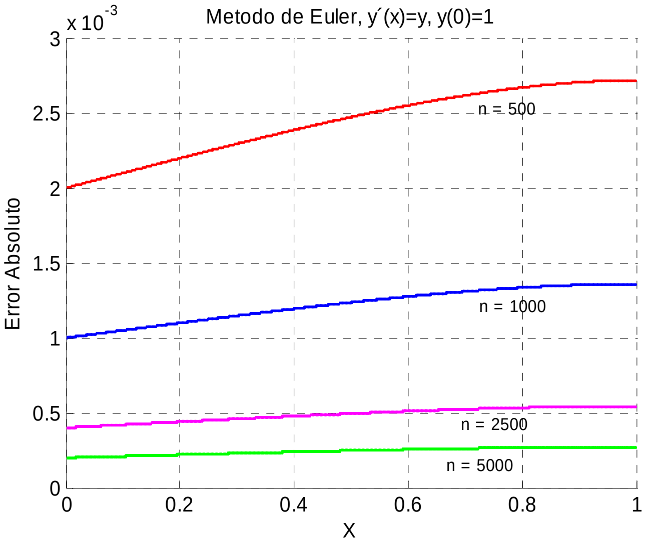

Fig. 1. Comparación del error absoluto con el método de Euler, medido a la solución analítica, para n = { 500, 1000, 2500, 5000}

III. Método de Runge-Kutta de 2DO Orden

Empleando Runge-Kutta de segundo orden, para n = 5:

```
X =
      0    0.2000    0.4000    0.6000    0.8000
Y =
   1.0000    1.2200    1.4884    1.8158    2.2153    2.7027
```

El error absoluta resulta ser: $\varepsilon_{abs} = 0.0156$. Para n = 50; se tiene:

```
Y =
  Columns 1 through 9
   1.0000    1.0202    1.0408    1.0618    1.0833    1.1052    1.1275    1.1503    1.1735
  Columns 10 through 18
   1.1972    1.2214    1.2461    1.2712    1.2969    1.3231    1.3498    1.3771    1.4049
  Columns 19 through 27
   1.4333    1.4622    1.4918    1.5219    1.5527    1.5840    1.6160    1.6487    1.6820
  Columns 28 through 36
   1.7159    1.7506    1.7860    1.8220    1.8589    1.8964    1.9347    1.9738    2.0137
  Columns 37 through 45
   2.0543    2.0958    2.1382    2.1814    2.2254    2.2704    2.3162    2.3630    2.4108
  Columns 46 through 51
   2.4595    2.5091    2.5598    2.6115    2.6643    2.7181
```

Manuscrito terminado el 23 de Febrero de 2005.

F. G. L. Está con la Universidad Experimental Politécnica de la Fuerza Armada, Carretera Nacional Maracay-Mariara, Departamento de Ingeniería Eléctrica, Maracay, Estado Aragua, Venezuela, Tlf. +58-243-5546951, Fax: +58-243-5546921, E-mail: fglongatt@ieee.org.

Es candidato a Doctor en Ciencias de la Ingeniería de la Universidad Central de Venezuela, Los Chaguaramos, Caracas, Venezuela, Tlf. +58-414-4572832, E-mail: flongatt@elecrisc.ing.ucv.ve.

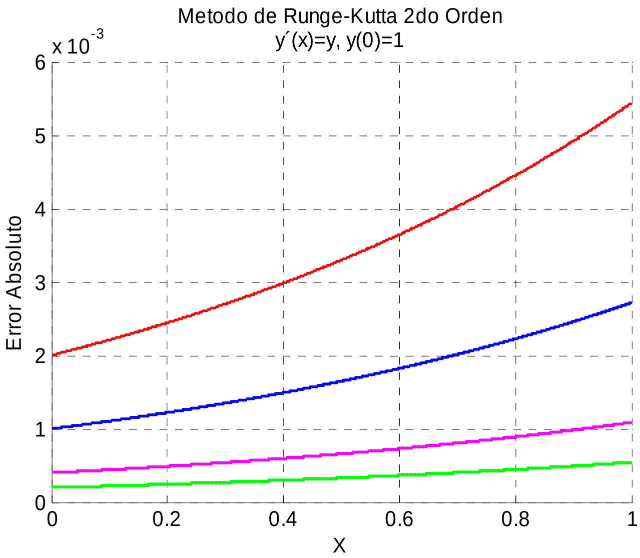

Fig. 2. Comparación del error absoluto con el método de Runge-Kutta 2do Orden, medido a la solución analítica, para n = { 500, 1000, 2500, 5000}

IV. Método de Runge-Kutta de 4TO Orden

Empleando Runge-Kutta de cuarto Orden n = 5:

X =
0 0 0.2000 0.4000 0.6000 0.8000
Y =
1.0000 1.2214 1.4918 1.8221 2.2255 2.7183

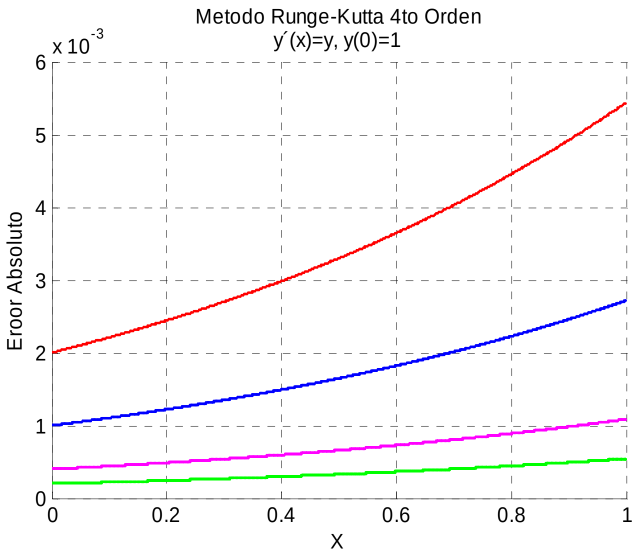

Fig. 3. Comparación del error absoluto con el método de Runge-Kutta 4do Orden, medido a la solución analítica, para n = { 500, 1000, 2500, 5000}

El error absoluta resulta ser: $\varepsilon_{abs} = 1.8175\times10^{-5}$. Para n = 50; se tiene:

Y =
Columns 1 through 9
1.0000 1.0202 1.0408 1.0618 1.0833 1.1052 1.1275 1.1503 1.1735
Columns 10 through 18
1.1972 1.2214 1.2461 1.2712 1.2969 1.3231 1.3499 1.3771 1.4049
Columns 19 through 27
1.4333 1.4623 1.4918 1.5220 1.5527 1.5841 1.6161 1.6487 1.6820
Columns 28 through 36
1.7160 1.7507 1.7860 1.8221 1.8589 1.8965 1.9348 1.9739 2.0138
Columns 37 through 45
2.0544 2.0959 2.1383 2.1815 2.2255 2.2705 2.3164 2.3632 2.4109
Columns 46 through 51
2.4596 2.5093 2.5600 2.6117 2.6645 **2.7183**

V. Comparación

Al efectuar el trazado del error absoluto en la solución de la ecuación diferencial (1) para un mismo paso de integración (n=10), para los diferentes métodos de integración: Euler, Runge-Kutta de 2do y 4to orden, se evidencia, que los métodos que presentan mayor error son: Euler, Runge Kutta de 2do y de 4to orden en ese mismo orden de mayor a menor.

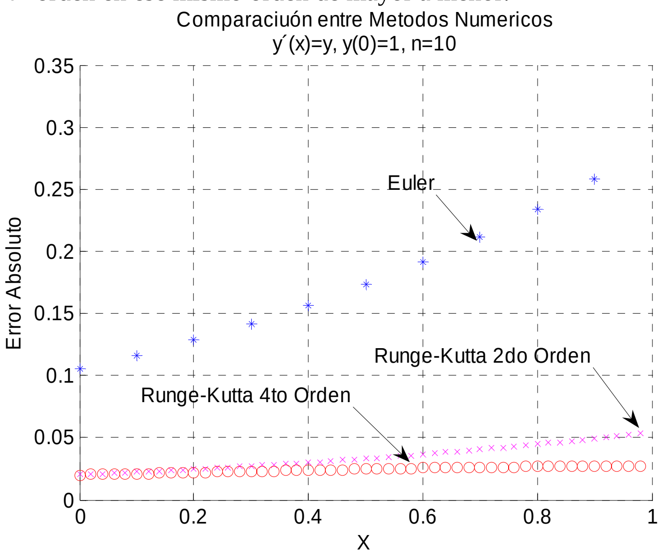

Fig. 4.

Y eso 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.

Tabla 1
Resultados con los Diferentes Métodos Numéricos

i	x_i	$Y(x_i)$ Euler	$Y(x_i)$ Runge-Kutta 2do Orden	$Y(x_i)$ Runge-Kutta 4do Orden
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

Referencias

[1] "Apuntes de Computación Científica II Métodos EDO - Problemas de valores iniciales". Disponible en: http://www.ii.uam.es/~pedro/ccii/teoria/MetodosEDOInicial/MetodosEDOInicial.htm

Anexos

Se cree adecuado agregar a este documento, un resumen de los diferentes métodos numéricos aplicados a la resolución de ecuaciones diferenciales. Este anexo a sido tomado por completo de [1].

A. Método de Taylor

1. **Resolución de EDO de primer orden:**
$$\frac{dy}{dx} = f(x,y)$$

2. **Uso del desarrollo de Taylor de la función:**
$$y_{i+1} = y_i + hf_i + \frac{1}{2!}h^2 f_i' + \ldots + \frac{1}{n!}h^n f_i^{(n-1)}$$
$$f_i = f(x_i, y_i) = \left[\frac{dy}{dx}\right]_{(x_i,y_i)}$$
$$f_i' = f'(x_i, y_i) = \frac{df}{dx} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\frac{dy}{dx} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}f = \left[\frac{d^2y}{dx^2}\right]_{(x_i,y_i)}$$
…
(concepto de derivada total)

3. **Error (de truncamiento) de orden n+1:**
$$y(x_{i+1}) = y(x_i + h) = y_{i+1} + \frac{h^{n+1}}{(n+1)!}\xi = y_{i+1} + O(h^{n+1})$$

4. **Problemas**
   1. Diferenciación de f(x,y) puede ser complicada: uso de programas de cálculo simbólico

B. Método de Euler (forward)

1. **Método de Taylor de orden 1 (error de orden $O(h^2)$):**
$$y_{i+1} = y_i + hf_i \quad ; \quad y(x_{i+1}) = y_{i+1} + O(h^2)$$

2. **Interpretación geométrica:**
   1. Aproximación de la solución por la recta tangente
   2. Solución calculada es polilínea con lados igual a rectas con pendiente f_i en el intervalo $[x_i, x_{i+1}]$

3. **Problemas de estabilidad y de exactitud (convergencia)**

4. **Método de Euler backward (ecuación implícita)**
$$y_{i+1} = y_i + hf_{i+1}$$

C. Métodos de Runge-Kutta

1. **Objetivo: evitar cálculo de derivadas de f(x,y), a base de calcular f(x,y) en más puntos, manteniendo el orden del error y la complejidad de cálculo**
   1. métodos de orden 2, 3 y 4 requieren el cálculo de f(x,y) en 2, 3 y 4 puntos respectivamente
   2. métodos de orden m > 4 requieren el cálculo de f(x,y) en más de m puntos (no más de m+2)

2. **Uso de función de incremento $\phi(x_i, y_i, h)$, aproximación de f(x,y) en intervalo $[x_i, x_{i+1}]$**
$$y_{i+1} = y_i + h\phi(x_i, y_i, h)$$

3. **Deducción de Runge-Kutta de orden 2 (RK-2)**

4. $$\phi = ak_1 + bk_2$$
$$k_1 = f(x_i, y_i)$$
$$k_2 = f(x_i + ph, y_i + qhf(x_i, y_i)) = f(x_i + ph, y_i + qhk_1)$$
$$y_{i+1} = y_i + h(ak_1 + bk_2)$$
   1. Se expande por Taylor k_2, y se iguala desarrollo de Taylor de y_{i+1} en función de parámetros a, b, p y q con el desarrollo estándar término a término
   2. $a + b = 1 \; ; \; bp = \frac{1}{2} \; ; \; bq = \frac{1}{2}$

5. **Métodos de Runge-Kutta de orden 2 (error de $O(h^3)$)**
   1. Para a=b=1/2: promedio de tangentes en extremos (RK2-promedio)
$$\bar{y}_{i+1} = y_i + hf(x_i, y_i)$$
$$y_{i+1} = y_i + \frac{h}{2}\left(f(x_i, y_i) + f(x_i + h, \bar{y}_{i+1})\right)$$
   2. Para a=0, b=1: estimación de tangente en punto medio (*midpoint method*) (RK2-punto-medio)
$$\bar{y}_{i+\frac{1}{2}} = y_i + \frac{h}{2}f(x_i, y_i)$$
$$y_{i+1} = y_i + hf(x_i + \frac{h}{2}, \bar{y}_{i+\frac{1}{2}})$$
   3. Métodos **predictor-corrector**
   4. **¡OJO!** Al implementar el algoritmo, no usar las mismas variables para y_i y para $\bar{y}_{i+\frac{1}{2}}$; ambas son necesarias para calcular y_{i+1}

6. **Métodos de Runge-Kutta de orden 3 (error de $O(h^4)$)**
   1. Promedio de tangente en extremos y punto medio
$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 4k_2 + k_3)$$
$$k_1 = f(x_i, y_i)$$
$$k_2 = f(x_i + \frac{h}{2}, y_i + \frac{1}{2}hk_1)$$
$$k_3 = f(x_i + h, y_i + 2hk_2 - hk_1)$$

7. **Métodos de Runge-Kutta de orden 4 (error de $O(h^5)$)**
   1. Método atribuido a Kutta
$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$
$$k_1 = f(x_i, y_i)$$
$$k_2 = f(x_i + \frac{h}{2}, y_i + \frac{1}{2}hk_1)$$
$$k_3 = f(x_i + \frac{h}{2}, y_i + \frac{1}{2}hk_2)$$
$$k_4 = f(x_i + h, y_i + hk_3)$$

8. **RK3 y RK4 se reducen a fórmula de Simpson de integración (interpolación parabólica de 3 puntos), cuando f(x,y) es función sólo de x:**

$$\int_a^b f(x)dx = \frac{b-a}{6}\left(f(a) + 4f(\frac{a+b}{2}) + f(b)\right) + O(h^5 f^{(4)})$$

$$a = x_i \quad ; \quad b = x_i + h$$

$$\int_a^b f = y_{i+1} - y_i$$

D. Método de Runge-Kutta adaptativo

1. **Duplicación de paso (step doubling)**
    1. Se repite estimación de y(x+2h) con 2h (y^1) y con h (y^2)
    2. Estimación del error: $\Delta = y^2 - y^1$
    3. Extrapolación local para aumentar el orden del método
       Ejemplo de RK4:
       $y(x+2h) = y^2 + \frac{\Delta}{15} + O(h^6)$
2. **RK embebido (métodos Runge-Kutta-Fehlberg)**
    1. Se aprovechan las evaluaciones de f(x,y) (las k_i) de un método de orden superior para calcular una aproximación de orden inferior (método embebido en el anterior)

3. Ejemplo de RK5 con RK4 embebido

$$y_{i+1}^1 = y_i + h(c_1k_1 + c_2k_2 + c_3k_3 + c_4k_4 + c_5k_5 + c_6k_6) + O(h^6)$$

$$y_{i+1}^2 = y_i + h(c_1^*k_1 + c_2^*k_2 + c_3^*k_3 + c_4^*k_4 + c_5^*k_5 + c_6^*k_6) + O(h^5)$$

$$\Delta = y_{i+1}^1 - y_{i+1}^2 = h\sum_{n=1}^{6}(c_n - c_n^*)k_n$$

    1. Extrapolación local al tomar y_{i+1}^1cuando en realidad el error se le estima a y_{i+1}^2

4. **Estimación de siguiente paso de integración h**
    1. Si $\Delta \propto h^5$(ejemplos anteriores)
       $h_{siguiente} = h_{actual}\left|\frac{\Delta_{deseado}}{\Delta_{actual}}\right|^{1/5}$
    2. Estimación nos da nuevo h para recalcular y_{i+1} si error era muy grande (mayor que la precisión deseada), o para calcular y_{i+2} si era muy pequeño
5. **Errores fraccionales (i.e., número de cifras significativas) para fijar $\Delta_{deseado}$: respecto de valor actual de y, máximo de y (conocido o estimado), etc.**

E. Método del punto medio (midpoint) modificado

1. **Método del punto medio también conocido como "centered difference" o "leapfrog"**
   $y_{i+1} = y_{i-1} + 2hf(x_i, y_i)$
    1. Método de segundo orden con sólo una evaluación de f(x,y)
2. **Método del punto medio modificado (en los extremos)**

$$z_0 = y(x_0)$$

$$z_1 = z_0 + hf(x_0, z_0)$$

$$z_{i+1} = z_{i-1} + 2hf(x_0 + ih, z_i)$$

$$y_n = \frac{1}{2}[z_n + z_{n-1} + hf(x_0 + nh, z_n)]$$

3. **Error sólo depende de potencias pares de h**
   $y_n - y(x_0 + nh) = \sum_{i=1}^{\infty} a_i h^{2i}$
    1. Con técnicas similares a la integración de Simpson se puede aumentar el orden de la aproximación con poco coste
    2. Ejemplo de orden 4: $\frac{4y_n^h - y_{n/2}^{2h}}{3}$
4. **Aplicación en el método de Richardson**

F. Método de extrapolación de Richardson

1. **Idea similar a paso de integración de Simpson a la de Romberg (aproximación diferida al límite de Richardson)**
    1. Se estima valor de $y(x_0+nh)$ para distintos pasos de integración (y distintos valores de n: H=nh=cte):
       n=2,4,6,8,10,...2j...
    2. Se extrapolan los valores obtenidos a h=0
2. **Extrapolación por funciones racionales (Burlisch-Stoer) o polinómicas (peores pero más rápidas)**
    1. Extrapolación da además estimación de error: uso de algoritmos incrementales de extrapolación (Neville)
    2. Límite en valor máximo de n: si se llega a él, se reduce intervalo de integración (H=nh)
3. **Uso de métodos cuyo error depende de potencias de h^2:**
    1. Convergencia más rápida: extrapolación depende de h^2, y al subdividir más, se aumenta aproximación en 2 órdenes

4. **H ya no es 'pequeño' (10h, 100h)**

5. **Importancia de evaluación de $\Delta_{deseado}$ por error fraccional; escalado de variables**

G. Métodos predictor-corrector

1. **Métodos multipaso con iteración funcional**

2. **Método explícito para estimar y_{i+1} (predictor):**
$$y_{i+1} = y_i + h(c^p_0 f_i + c^p_{-1} f_{i-1} + c^p_{-2} f_{i-2} + \dots) \quad ; \quad f_i = f(x_i, y_i), \quad \dots$$

3. **Ecuación implícita para valor final (corrector), que usa y_{i+1} anterior para evaluar f_{i+1}:**
$$y_{i+1} = y_i + h(c^c_{+1} f_{i+1} + c^c_0 f_i + c^c_{-1} f_{i-1} + c^c_{-2} f_{i-2} + \dots) \quad ; \quad f_i = f(x_i, y_i), \quad \dots$$

4. **Diferencia entre los y_{i+1} da estimación del error de truncamiento local**

5. **Se itera hasta reducir este error a los niveles deseados**

6. **Método Adams-Bashforth-Moulton de tercer orden ($O(h^4)$):**
$$c^p = \frac{23}{12}, \frac{-16}{12}, \frac{5}{12}$$
$$c^c = \frac{5}{12}, \frac{8}{12}, \frac{-1}{12}$$

7. **Si número de iteraciones fijas, se convierte en método explícito: no muy aconsejable**

8. **Método corrector multipaso con ecuación implícita resuelta por método de Newton es mejor para problemas stiff**

H. Métodos multivalor (multivalue)

1. **En lugar de calcular y_{i+1}, calculamos en cada paso y_{i+1} y sus derivadas:**
$$\overline{y_i} = \begin{pmatrix} y_i \\ h y_i' \\ \frac{h^2}{2} y_i'' \\ \frac{h^3}{6} y_i''' \end{pmatrix} \quad ; \quad y_i' = f(x_i, y_i)$$

2. **Fórmula:**
$$\overline{y_{i+1}} = B \cdot \overline{y_i} + \alpha_{i+1} \, \overline{r}$$
$$B = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$
$$\alpha_{i+1} = h f(x_{i+1}, y_{i+1}) - (B \cdot \overline{y_i})_2$$
$$r_2 = 1$$

    1. B se obtiene al derivar:
$$y(x) = y_n + (x - x_n) y_n' + \frac{(x - x_n)^2}{2} y_n'' + \frac{(x - x_n)^3}{6} y_n'''$$

    2. las condiciones sobre α_{i+1} y r_2 se obtienen al forzar
$$y_{i+1}' = f(x_{i+1}, y_{i+1})$$

3. **Resto componentes de r (r_1, r_3 , r_4) son libres y determinan orden y estabilidad del método**

    1. Adams-Moulton (corrector) corresponde a 5/12, 3/4, 1/6

4. **Resolución de método implícito por predictor-corrector o Newton**

5. **Son equivalentes a métodos multipaso, pero resuelven dificultades de estos métodos:**

    1. Adaptación de paso de integración; incluso se puede cambiar de orden dinámicamente, cambiando r

    2. Arranque del método, empezando con orden 1

I. Comparación entre los métodos

1. **Runge-Kutta:**

    1. Bueno para casi todo

    2. Cuando no se exige mucha eficiencia ni exactitud

2. **Richardson:**

    1. Para mejorar eficiencia y exactitud, aunque puede dar más problemas que RK

    2. Falla cuando hay singularidades o si las soluciones son poco suaves

3. **Predictor-corrector:**

    1. Richardson los está dejando obsoletos

    2. Útiles para f(x,y) complicada, pero con funciones suaves y se desea alta precisión

Conclusiones

Este artículo presentó una vista general del desarrollo de las turbinas de viento, describe el estado actual y futuros retos para esta tecnología. Esta es una notable historia que comienza en el siglo diecinueve y entonces se acelera hasta las últimas dos décadas del siglo veinte con un curso muy similar a los primeros días de la aeronáutica. La historia esta lejos de terminar pero ciertamente comenzó y esta en evolución.

References

[2] 35th IEA Topical Expert Meeting “Long Term R&D Needs 2000 – 2020” The Netherlands, March 2001.
[3] www.wind-energy-network.org