



Universidad de Valladolid

Facultad de Ciencias

TRABAJO FIN DE GRADO

Grado en Matemáticas

Interpolación racional. Aplicación a la integración de problemas de valores iniciales de ecuaciones diferenciales ordinarias.

Autor: Eduardo Trujillo Puente

Tutor: Luis María Abia Llera

Prefacio

En este trabajo Fin de Grado abordamos partiendo de primeros principios el problema de la interpolación racional. Una aplicación de estas técnicas es su uso en la integración del problema de valores iniciales de ecuaciones diferenciales ordinarias utilizando los llamados métodos de extrapolación.

En el Capítulo 1, planteamos el problema general de interpolación para una función racional discutiendo resultados preliminares sobre la existencia del interpolante. Rápidamente abordamos la construcción efectiva de los interpolantes racionales así como su evaluación en un punto. Esto nos obliga a realizar una breve incursión en la teoría de fracciones continuas.

Con vistas a la aplicación de los algoritmos de interpolación racional a la integración de problemas de valores iniciales de ecuaciones diferenciales ordinarias, abordamos en el Capítulo 2 la existencia de desarrollos asintóticos del error global para métodos de un paso explícitos. Como el marco de aplicación de estos resultados es la extrapolación al límite de métodos numéricos que admitan un desarrollo asintótico del error global en potencias pares de la longitud de paso, se introducen los métodos simétricos, de los cuales la regla modificada del punto medio es un ejemplo. La existencia del desarrollo asintótico en potencias pares del parámetro de la discretización para la regla del punto medio modificada es un resultado clave para la efectividad de los métodos de extrapolación que se basan en ella.

En el Capítulo 3, se introducen los métodos de extrapolación al límite y se describe la estrategia general para un código de paso y orden variable basado en extrapolaciones de aproximaciones obtenidas con la regla del punto medio modificado (algoritmo GBS, por Gragg-Bulirsch-Stoer). Se implementan estos métodos en Matlab (que no ofrece en su biblioteca de funciones un integrador basado en métodos de extrapolación), y se realiza una experimentación numérica con algunos problemas tests.

Añadimos un Apéndice con los programas utilizados en Matlab (no ofrecidos en su biblioteca) para los ejemplos mencionados en el Capítulo 3.

En Valladolid, a 12 de julio de 2019.

Índice general

Índice general	v
1. Interpolación por funciones racionales	1
1.1. El problema general de interpolación racional	1
1.2. Procedimientos recursivos de interpolación racional . . .	5
1.2.1. Diferencias inversas y recíprocas. Fracción continua de Thiele	6
1.2.2. Algoritmos tipo Neville	12
2. El desarrollo asintótico del error global	17
2.1. Introducción	17
2.2. Convergencia de métodos de un paso	18
2.3. El desarrollo asintótico del error global	21
2.4. El desarrollo asintótico en métodos simétricos	24
3. Métodos de extrapolación y su aplicación a problemas de valores iniciales	27
3.1. Introducción	27
3.2. Definición del método	28
3.3. Extrapolación de Richardson	30
3.4. Extrapolación racional	30
3.5. Extrapolación con métodos simétricos	31
3.6. El método Gragg-Bulirsch-Stoer(GBS)	31
3.7. Control de orden y del tamaño del paso	36
3.8. Experimentos numéricos	38
3.9. Conclusiones	56
Bibliografía	57
A. Programas de Matlab	59

Capítulo 1

Interpolación por funciones racionales

1.1. El problema general de interpolación racional

El problema general de interpolación racional se plantea en los siguientes términos: Dado un conjunto de pares de puntos (x_i, f_i) , $i = 0, 1, 2, \dots, N$, donde $f_i = f(x_i)$, y dados enteros $n \geq 0$, $m \geq 0$, determinar una función racional de la forma

$$\Phi^{n,m}(x) = \frac{P^{n,m}(x)}{Q^{n,m}(x)} = \frac{a_0 + a_1x + \dots + a_nx^n}{b_0 + b_1x + \dots + b_mx^m}$$

que satisfaga las condiciones de interpolación

$$\Phi^{n,m}(x_i) = f_i, \quad i = 0, 1, \dots, N.$$

La forma de $\Phi^{n,m}(x)$ está fijada por el par (n, m) , al que nos referiremos como grados de $\Phi^{n,m}$.

La función racional $\Phi^{n,m}$ esta determinada por $n + m + 2$ coeficientes. Por otro lado $\Phi^{n,m}$ determina estos coeficientes salvo un factor de proporcionalidad en dichos coeficientes, por lo que $\Phi^{n,m}(x)$ depende esencialmente de $n + m + 1$ coeficientes. Para fijar estos coeficientes parece natural fijar $n + m + 1$ condiciones de interpolación

$$\Phi^{n,m}(x_i) = f_i, \quad i = 0, 1, \dots, n + m. \quad (1.1)$$

Llamaremos $A^{n,m}$ al problema de interpolación así planteado. Si tiene solución, es obvio que los coeficientes $a_i, i = 0, \dots, n$ y $b_i, i = 0, \dots, m$,

deben satisfacer el sistema lineal homogéneo

$$P^{n,m}(x_i) - f_i Q^{n,m}(x_i) = 0, \quad i = 0, 1, \dots, n + m. \quad (1.2)$$

Denotamos este sistema como $S^{n,m}$. Sin embargo, una solución del sistema de ecuaciones $S^{n,m}$ puede definir una función racional $\Phi^{n,m}(x)$ que no satisfaga todas las condiciones de interpolación. Por ejemplo, si buscamos determinar $\Phi^{1,1}(x)$, que satisfaga las condiciones $\Phi(0) = 1$, $\Phi(1) = 2$ y $\Phi(2) = 2$, entonces la solución del sistema $S^{1,1}$ correspondiente determina

$$\Phi^{1,1}(x) = \frac{2x}{x},$$

que obviamente no resuelve $A^{1,1}$.

Para poder continuar, tenemos que distinguir entre las diferentes representaciones de una misma función racional $\Phi^{n,m}$. Se dice que dos expresiones racionales

$$\Phi_1 = \frac{P_1(x)}{Q_1(x)}, \quad \Phi_2 = \frac{P_2(x)}{Q_2(x)}, \quad Q_1(x) \neq 0, \quad Q_2(x) \neq 0$$

son equivalentes ($\Phi_1 \sim \Phi_2$) si

$$P_1(x)Q_2(x) = P_2(x)Q_1(x).$$

Se dice que una expresión racional es irreducible si el numerador y el denominador son primos entre sí. En el caso de que la expresión racional no sea irreducible se puede reducir a una irreducible cancelando los factores polinomiales comunes del numerador y denominador.

Teorema 1.1. *El sistema lineal homogéneo $S^{n,m}$ siempre tiene soluciones no triviales. Para cada una de tales soluciones:*

$$\Phi^{n,m} = \frac{P^{n,m}(x)}{Q^{n,m}(x)}$$

donde $Q^{n,m}(x)$ es distinto del polinomio idénticamente nulo.

Demostración. El sistema lineal homogéneo $S^{n,m}$ tiene $n + m + 1$ ecuaciones para $n + m + 2$ incógnitas. Como tiene más incógnitas que ecuaciones, $S^{n,m}$ tiene soluciones no triviales

$$(a_0, a_1, \dots, a_n, b_0, b_1, \dots, b_m) \neq (0, \dots, 0)$$

Si ocurriera que

$$Q^{n,m}(x) = b_0 + b_1x + \dots + b_mx^m \equiv 0$$

entonces implicaría que el polinomio $P^n(x) = a_0 + a_1x + \dots + a_nx^n$ tiene todos los ceros

$$P^{n,m}(x_i) = 0, \quad i = 0, 1, \dots, n + m$$

Como $P^{n,m}(x)$ es a lo sumo de grado n , y tiene $n + m + 1$ raíces, debe ser el polinomio idénticamente nulo. Esto contradice que

$$(a_0, a_1, \dots, a_n, b_0, b_1, \dots, b_m) \neq (0, \dots, 0)$$

□

Veamos ahora que si el problema de interpolación racional tiene solución, esta solución es única.

Teorema 1.2. Si Φ_1 y Φ_2 son ambas soluciones del sistema lineal homogéneo $S^{n,m}$, entonces, son equivalentes ($\Phi_1 \sim \Phi_2$).

Demostración. Si tanto $\Phi_1 = \frac{P_1(x)}{Q_1(x)}$ como $\Phi_2 = \frac{P_2(x)}{Q_2(x)}$ son soluciones de $S^{n,m}$, construimos entonces el polinomio

$$P(x) = P_1(x)Q_2(x) - P_2(x)Q_1(x)$$

Las ecuaciones $S^{n,m}$ implican que $P(x)$ tiene $n + m + 1$ ceros en los $x_i, i = 0, \dots, n + m$, pues

$$\begin{aligned} P(x_i) &= P_1(x_i)Q_2(x_i) - P_2(x_i)Q_1(x_i) \\ &= f_iQ_1(x_i)Q_2(x_i) - f_iQ_2(x_i)Q_1(x_i) = 0. \end{aligned}$$

Como $P(x)$ es a lo sumo de grado $n + m$, entonces $P(x) \equiv 0$ y por lo tanto $\Phi_1 \sim \Phi_2$.

□

Nota: No se cumple el recíproco del teorema. Si tenemos $\Phi_1 \sim \Phi_2$ y Φ_1 es solución de $S^{n,m}$, no implica que Φ_2 sea solución de $S^{n,m}$. Veremos que esta situación es típica de problemas de interpolación racional sin solución.

Combinando los dos teoremas anteriores, tenemos que existe una única función racional para cada problema de interpolación racional $A^{n,m}$, la cual se representa por una expresión racional $\Phi^{n,m}$ que resuelve el correspondiente sistema $S^{n,m}$.

Ahora, o bien, dicha expresión racional satisface (1.1), de modo que resuelve $A^{n,m}$, o $A^{n,m}$ no tiene solución. En este último caso, existe algún par (x_i, f_i) tal que $Q^{n,m}(x_i) = 0$, y por tanto no satisface la condición

interpolatoria $\Phi^{n,m}(x_i) = 0$. Diremos que tal par es inaccesible. El problema de interpolación $A^{n,m}$ tiene solución si y sólo no existen pares inaccesibles entre los $(x_i, f_i), i = 0, \dots, n + m$.

Sea $\Phi^{n,m} = \frac{P^{n,m}(x)}{Q^{n,m}(x)}$ una solución de $S^{n,m}$. Para cada $i \in \{0, 1, \dots, n + m\}$ se distinguen dos casos:

- (1) $Q^{n,m}(x_i) \neq 0$,
- (2) $Q^{n,m}(x_i) = 0$.

En el primer caso, claramente $\Phi^{n,m}(x_i) = f_i$. Sin embargo, en el segundo caso el par (x_i, f_i) es inaccesible. Entonces, por (1.2)

$$P^{n,m}(x_i) = 0$$

Por lo tanto, $P^{n,m}$ y $Q^{n,m}$ contienen el factor $(x - x_i)$ y no pueden ser primos entre sí. En consecuencia hemos demostrado:

Proposición 1.1. *Si $S^{n,m}$ tiene una solución $\Phi^{n,m}$ irreducible, entonces no hay pares inaccesibles y por lo tanto $A^{n,m}$ tiene solución.*

Dado $\Phi^{n,m}$, sea $\Theta^{n,m}$ una expresión racional equivalente que es irreducible. Entonces tenemos el resultado general.

Teorema 1.3. *Supongamos que $\Phi^{n,m}$ resuelve el sistema $S^{n,m}$. Entonces $A^{n,m}$ tiene solución (y $\Phi^{n,m}$ representa dicha solución), sí y solo sí $\Theta^{n,m}$ resuelve $S^{n,m}$.*

Demostración. Si $\Theta^{n,m}$ resuelve $S^{n,m}$, entonces $A^{n,m}$ tiene solución por la proposición (1.1). Si $\Theta^{n,m}$ no resuelve $S^{n,m}$, entonces su correspondiente función racional no resuelve $A^{n,m}$. □

Definición 1.1. *Decimos que los pares $(x_i, f_i), i = 1, 2, \dots, k$ están en posición especial si se pueden interpolar por una expresión racional de grados (i, j) , con $i + j < k$. En otras palabras, el problema de interpolación racional tiene solución para una combinación de grados del numerador y el denominador inferior a la que sugiere el número de pares dados.*

Observamos entonces que:

Proposición 1.2. *Los pares accesibles de un problema de interpolación racional $A^{n,m}$ que no tiene solución están en una posición especial.*

Demostración. Sean i_1, \dots, i_a los subíndices de los pares inaccesibles, y sea $\Phi^{n,m}$ una solución de $S^{n,m}$. Entonces el numerador y el denominador de $\Phi^{n,m}$ tienen los factores comunes $(x - x_{i_1}), \dots, (x - x_{i_a})$, cuya cancelación nos deja una función racional equivalente $\Phi^{r,s}$ con $r = n - a$, $s = m - a$. $\Phi^{r,s}$ resuelve el problema de interpolación $A^{r,s}$, asociado a los $n + m + 1 - a$ pares accesibles. Como

$$r + s + 1 = n + m + 1 - 2a < n + m + 1 - a,$$

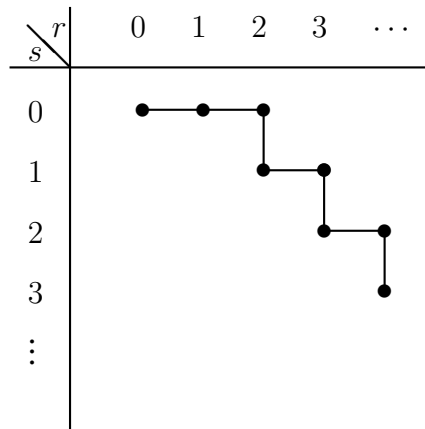
se concluye que los pares accesibles de $A^{n,m}$ están en una posición especial. □

En lo que sigue, nos vamos a centrar en los problemas donde no hay subconjuntos de los pares de puntos que estén en posición especial. En este caso no solo $A^{n,m}$ tiene solución, sino que todos los problemas $A^{r,s}$ con $r + s + 1$ pares extraídos del conjunto original, donde $r + s \leq n + m$ también tienen solución.

1.2. Procedimientos recursivos de interpolación racional

En lo que queda de capítulo consideraremos distintos procedimientos recursivos, análogos a algoritmos existentes para la interpolación polinómica de Lagrange, para la construcción y evaluación de los interpolantes $\Phi^{n,m}(x)$. Todos estos procedimientos recursivos computan sucesivos $\Phi^{r,s}(x)$, con $r + s = 0, 1, 2, \dots$, interpolando de forma sucesiva los datos de interpolación $(x_0, f_0), (x_1, f_1), \dots, (x_{r+s}, f_{r+s})$ de forma que en cada paso se aumenta en una unidad el grado del numerador o el grado del denominador del interpolante.

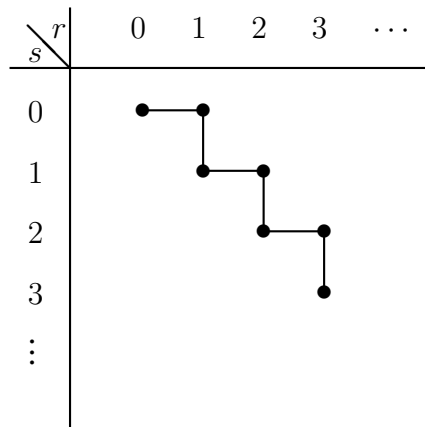
La variedad de posibilidades para ir aumentando el grado en una unidad del numerador o del denominador del interpolante $\Phi^{n,m}(x)$ produce una diversidad de algoritmos que pueden identificarse con un diagrama como el siguiente.



Distinguiremos dos tipos de algoritmos. El primero es análogo a la forma de Newton del polinomio de interpolación de Lagrange: Se construirá una tabla de cantidades análoga a las diferencias divididas, a partir de la cual se generaran los coeficientes para la expresión del interpolante racional. El segundo corresponde a un algoritmo tipo Aitken-Neville para generar las aproximaciones en una abscisa x fija de interpolantes de grados cada vez más altos.

1.2.1. Diferencias inversas y reciprocas. Fracción continua de Thiele

El algoritmo que vamos a tratar en esta sección es el que corresponde a ir incrementado de forma alternativa un grado del numerador y un grado del denominador.



(1.3)

Empezando por los pares (x_i, f_i) , $i = 0, 1, \dots$, construimos la tabla de diferencias inversas

i	x_i	f_i			
0	x_0	f_0			
1	x_1	f_1	$\varphi(x_0, x_1)$		
2	x_2	f_2	$\varphi(x_0, x_2)$	$\varphi(x_0, x_1, x_2)$	
3	x_3	f_3	$\varphi(x_0, x_3)$	$\varphi(x_0, x_1, x_3)$	$\varphi(x_0, x_1, x_2, x_3)$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots

Estas diferencias inversas están definidas recursivamente:

$$\begin{aligned}
 \varphi(x_i, x_j) &= \frac{x_i - x_j}{f_i - f_j} \\
 \varphi(x_i, x_j, x_k) &= \frac{x_j - x_k}{\varphi(x_i, x_j) - \varphi(x_i, x_k)} \\
 &\vdots \\
 \varphi(x_i, \dots, x_l, x_m, x_n) &= \frac{x_m - x_n}{\varphi(x_i, \dots, x_l, x_m) - \varphi(x_i, \dots, x_l, x_n)}
 \end{aligned} \tag{1.4}$$

En ocasiones, algunas de las diferencias inversas son ∞ ya que el denominador en (1.4) puede ser cero. También hay que tener en cuenta que las diferencias inversas no son, en general, funciones simétricas respecto a sus argumentos (como ocurre con las diferencias divididas). Debemos notar, que las diferencias inversas $\varphi(x_i, x_{i+1}, \dots, x_l, x_m, x_n)$ son funciones simétricas con respecto a sus dos últimos argumentos.

Sean P^r , Q^s polinomios cuyo grado viene dado por r y s . Queremos intentar usar las diferencias inversas para encontrar una expresión racional

$$\Phi^{n,n}(x) = \frac{P^n(x)}{Q^n(x)}$$

con

$$\Phi^{n,n}(x_i) = f_i \quad \text{para } i = 0, 1, \dots, 2n.$$

Por lo tanto debemos tener

$$\begin{aligned}
 \frac{P^n(x)}{Q^n(x)} &= f_0 + \frac{P^n(x)}{Q^n(x)} - \frac{P^n(x_0)}{Q^n(x_0)} \\
 &= f_0 + (x - x_0) \frac{P^{n-1}(x)}{Q^n(x)} = f_0 + \frac{x - x_0}{Q^n(x)/P^{n-1}(x)}.
 \end{aligned}$$

La expresión racional $Q^n(x)/P^{n-1}(x)$ satisface

$$\frac{Q^n(x)}{P^{n-1}(x)} = \frac{x_i - x_0}{f_i - f_0} = \varphi(x_0, x_i)$$

son en particular las reducidas de grado $(0, 0), (1, 0), (1, 1), (2, 1), \dots$ de la fracción continua (1.5)

$$\begin{aligned}\Phi^{0,0}(x) &= f_0 \\ \Phi^{1,0}(x) &= f_0 + \frac{x - x_0}{\varphi(x_0, x_1)} \\ \Phi^{1,1}(x) &= f_0 + \frac{x - x_0}{\varphi(x_0, x_1) + \frac{x - x_1}{\varphi(x_0, x_1, x_2)}} \\ &\vdots\end{aligned}$$

Debido a la ausencia de simetría en las diferencias inversas, se suelen preferir las llamadas diferencias recíprocas

$$\rho(x_i, x_{i+1}, \dots, x_{i+k}).$$

Por definición,

$$\rho(x_i, x_{i+1}, \dots, x_{i+k}) = \varphi(x_i, x_{i+1}, \dots, x_{i+k}) + \varphi(x_i, x_{i+1}, \dots, x_{i+k-2}) + \dots$$

donde el último sumando del miembro de la derecha es $\varphi(x_i)$, si k es par, y $\varphi(x_i, x_{i+1})$ si k es impar. Es obvio de esta definición que

$$\rho(x_i, x_{i+1}, \dots, x_{i+k}) - \rho(x_i, x_{i+1}, \dots, x_{i+k-2}) = \varphi(x_i, x_{i+1}, \dots, x_{i+k}). \quad (1.6)$$

Para ver que las diferencias recíprocas son funciones simétricas de sus argumentos utilizaremos las recurrencias que definen numerador y denominador de las sucesivas reducidas de un desarrollo en fracción continua. Antes de formular estas recurrencias introducimos la fracción continua

$$r = b_0 + \frac{a_1}{b_1 + \frac{a_2}{b_2 + \frac{a_3}{b_3 + \frac{a_4}{b_4 + \dots}}}}$$

y su convergente o reducida de orden n

$$r_n = \frac{p_n}{q_n} = b_0 + \frac{a_1}{b_1 + \frac{a_2}{b_2 + \frac{a_3}{b_3 + \dots \frac{a_n}{b_n}}}}$$

Proposición 1.3. *La n -ésima convergente $r_n = p_n/q_n$, $n \geq 1$, verifica las siguientes fórmulas recursivas:*

$$\begin{aligned}p_n &= b_n p_{n-1} + a_n p_{n-2}, & p_{-1} &= 1, & p_0 &= b_0, \\ q_n &= b_n q_{n-1} + a_n q_{n-2}, & q_{-1} &= 0, & q_0 &= 1.\end{aligned}$$

Demostración. Utilizamos inducción sobre n para probar las fórmulas. Para $n = 1$ obtenemos

$$\frac{p_1}{q_1} = \frac{b_1 p_0 + a_1 p_{-1}}{b_1 q_0 + a_1 q_{-1}} = \frac{b_1 b_0 + a_1}{b_1 + 0} = b_0 + \frac{a_1}{b_1} = r_1.$$

Supongamos que las fórmulas son válidas hasta p_{n-1} , q_{n-1} para toda fracción continua. Notemos que p_n/q_n se obtiene de p_{n-1}/q_{n-1} mediante la sustitución de $b_{n-1} + a_n/b_n$ por b_{n-1} . Por tanto

$$\begin{aligned} \frac{p_n}{q_n} &= \frac{(b_{n-1} + a_n/b_n)p_{n-2} + a_{n-1}p_{n-3}}{(b_{n-1} + a_n/b_n)q_{n-2} + a_{n-1}q_{n-3}} = \frac{b_n(b_{n-1}p_{n-2} + a_{n-1}p_{n-3}) + a_n p_{n-2}}{b_n(b_{n-1}q_{n-2} + a_{n-1}q_{n-3}) + a_n q_{n-2}} \\ &= \frac{b_n p_{n-1} + a_n p_{n-2}}{b_n q_{n-1} + a_n q_{n-2}}. \end{aligned}$$

Esto demuestra que las fórmulas son válidas para p_n y q_n . □

Cuando $a_k = (x - x_{k-1})$, y $b_k = \varphi(x_0, x_1, \dots, x_k)$, $k = 1, 2, \dots$, las recurrencias anteriores se transforman en

$$\begin{aligned} M_{k+1}(x) &= \varphi(x_0, x_1, \dots, x_{k+1}) M_k(x) + (x - x_k) M_{k-1}(x) \\ M_{-1}(x) &= 1, \quad M_0(x) = y_0 \\ N_{k+1}(x) &= \varphi(x_0, x_1, \dots, x_{k+1}) N_k(x) + (x - x_k) N_{k-1}(x) \\ N_{-1}(x) &= 0, \quad N_0(x) = 1. \end{aligned}$$

Notemos que $M_{2k}(x)$ y $M_{2k-1}(x)$ son a lo sumo de grado k y N_{2k} y N_{2k+1} son a lo sumo de grado k . Además $\Phi^{k,k}(x) = \frac{M_{2k}(x)}{N_{2k}(x)}$, y $\Phi^{k,k-1}(x) = \frac{M_{2k-1}(x)}{N_{2k-1}(x)}$, son reducidas del interpolante racional.

Es fácil verificar que el coeficiente director de $M_k(x)$ es 1 si k es impar, y es $\rho(x_0, x_1, \dots, x_k)$ si k es par. Análogamente, el coeficiente director de $N_k(x)$ es 1 si k es par y es $\rho(x_0, \dots, x_k)$ si k es impar. De esta manera relacionamos las diferencias recíprocas con el interpolante racional Φ . Concretamente, $\rho(x_0, \dots, x_{2k})$ es el cociente entre los coeficientes principales del numerador y denominador del interpolante racional $\Phi^{k,k}(x)$ que coincide con $f(x)$ en las abscisas x_0, \dots, x_{2k} . Análogamente, $\rho(x_0, \dots, x_{2k-1})$ es el inverso de dicho cociente para el interpolante racional $\Phi^{k,k-1}(x)$. Como claramente los coeficientes principales de numerador y denominador del interpolante racional no depende del orden en que se toman las abscisas de interpolación, se tiene así una prueba indirecta de la simetría de las diferencias recíprocas con respecto a todos sus argumentos.

Teorema 1.4. *Las diferencias recíprocas satisfacen la siguiente fórmula recursiva*

$$\begin{aligned}
 \rho(x_i) &= f_i \\
 \rho(x_i, x_k) &= \frac{x_i - x_k}{f_i - f_k} \\
 &\vdots \\
 \rho(x_i, x_{i+1}, \dots, x_{i+k}) &= \frac{x_i - x_{i+k}}{\rho(x_i, x_{i+1}, \dots, x_{i+k-1}) - \rho(x_{i+1}, x_{i+2}, \dots, x_{i+k})} \\
 &\quad + \rho(x_{i+1}, x_{i+2}, \dots, x_{i+k-1}) \tag{1.7}
 \end{aligned}$$

Demostración. Los casos $k = 0$ y $k = 1$ son meras comprobaciones. Para el caso general, de (1.6) tenemos

$$\begin{aligned}
 \rho(x_i, x_{i+1}, \dots, x_{i+k}) &= \rho(x_{i+1}, \dots, x_{i+k-1}, x_i, x_{i+k}) \\
 &= \rho(x_{i+1}, \dots, x_{i+k-1}) + \varphi(x_{i+1}, \dots, x_{i+k-1}, x_i, x_{i+k}). \tag{1.8}
 \end{aligned}$$

Utilizando la fórmula recursiva para las diferencias inversas,

$$\begin{aligned}
 \varphi(x_{i+1}, \dots, x_{i+k-1}, x_i, x_{i+k}) &= \\
 &= \frac{x_i - x_{i+k}}{\varphi(x_{i+1}, \dots, x_{i+k-1}, x_i) - \varphi(x_{i+1}, \dots, x_{i+k-1}, x_{i+k})} \\
 &= \frac{x_i - x_{i+k}}{\rho(x_{i+1}, \dots, x_{i+k-1}, x_i) - \rho(x_{i+1}, \dots, x_{i+k-1}, x_{i+k})} \\
 &= \frac{x_i - x_{i+k}}{\rho(x_i, x_{i+1}, \dots, x_{i+k-1}) - \rho(x_{i+1}, x_{i+2}, \dots, x_{i+k})}
 \end{aligned}$$

Basta sustituir esta expresión en (1.8) para concluir la prueba. □

Las diferencias recíprocas se pueden organizar en la siguiente tabla

$$\begin{array}{c|cccc}
 x_0 & f_0 & & & \\
 & \rho(x_0, x_1) & & & \\
 x_1 & f_1 & \rho(x_0, x_1, x_2) & & \\
 & \rho(x_1, x_2) & & \rho(x_0, x_1, x_2, x_3) & \\
 x_2 & f_2 & \rho(x_1, x_2, x_3) & \vdots & \\
 & \rho(x_2, x_3) & \vdots & & \\
 x_3 & f_3 & \vdots & & \\
 \vdots & \vdots & & &
 \end{array} \tag{1.9}$$

Sustituyendo las diferencias inversas por las diferencias recíprocas en (1.5) obtenemos la fracción continua de Thiele del interpolante racional:

$$\Phi^{n,n}(x) = f_0 + \frac{x - x_0}{\rho(x_0, x_1) + \frac{x - x_1}{\rho(x_0, x_1, x_2) - \rho(x_0) + \frac{x - x_2}{\rho(x_0, x_1, x_2, x_3) - \rho(x_0, x_1) + \dots + \frac{x - x_{2n-1}}{\rho(x_0, \dots, x_{2n}) - \rho(x_0, \dots, x_{2n-2})}}}}$$

1.2.2. Algoritmos tipo Neville

Ahora vamos a deducir un algoritmo para la interpolación racional que es análogo al algoritmo de Neville para la interpolación polinómica. Usaremos

$$\Phi_s^{n,m}(x) = \frac{P_s^{n,m}(x)}{Q_s^{n,m}(x)}$$

para denotar la expresión racional con

$$\Phi_s^{n,m}(x_i) = f_i \quad \text{para } i = s, s+1, \dots, s+n+m$$

$P_s^{n,m}(x)$, $Q_s^{n,m}(x)$ son polinomios cuyos grados no son mayores que n y m respectivamente. Sean $p_s^{n,m}$ y $q_s^{n,m}$ los coeficientes principales de dichos polinomios:

$$P_s^{n,m}(x) = p_s^{n,m}x^n + \dots, \quad Q_s^{n,m}(x) = q_s^{n,m}x^m + \dots$$

Para abreviar usaremos la siguiente notación:

$$a_i = x - x_i \quad \text{y} \quad T_s^{n,m}(x, y) = P_s^{n,m}(x) - yQ_s^{n,m}(x).$$

Fijémonos que

$$T_s^{n,m}(x_i, f_i) = 0, \quad i = s, s+1, \dots, s+n+m.$$

Teorema 1.5. *Empezando con*

$$P_s^{0,0}(x) = f_s, \quad Q_s^{0,0}(x) = 1$$

se verifican las siguientes recurrencias:

(a) *Transición* $(n-1, m) \rightarrow (n, m)$:

$$\begin{aligned} P_s^{n,m}(x) &= a_s q_s^{n-1,m} P_{s+1}^{n-1,m}(x) - a_{s+n+m} q_{s+1}^{n-1,m} P_s^{n-1,m}(x), \\ Q_s^{n,m}(x) &= a_s q_s^{n-1,m} Q_{s+1}^{n-1,m}(x) - a_{s+n+m} q_{s+1}^{n-1,m} Q_s^{n-1,m}(x). \end{aligned}$$

(b) *Transición* $(n, m-1) \rightarrow (n, m)$:

$$\begin{aligned} P_s^{n,m}(x) &= a_s p_s^{n,m-1} P_{s+1}^{n,m-1}(x) - a_{s+n+m} p_{s+1}^{n,m-1} P_s^{n,m-1}(x), \\ Q_s^{n,m}(x) &= a_s p_s^{n,m-1} Q_{s+1}^{n,m-1}(x) - a_{s+n+m} p_{s+1}^{n,m-1} Q_s^{n,m-1}(x), \end{aligned}$$

Demostración. Probaremos solo (a) ya que (b) es análogo. Supongamos que las expresiones racionales $\Phi_s^{n-1,m}$ y $\Phi_{s+1}^{n-1,m}$ verifican los requisitos de interpolación

$$\begin{aligned} T_s^{n-1,m}(x_i, f_i) &= 0 \quad \text{para } i = s, s+1, \dots, s+n+m-1, \\ T_{s+1}^{n-1,m}(x_i, f_i) &= 0 \quad \text{para } i = s+1, \dots, s+n+m. \end{aligned} \quad (1.10)$$

Si definimos $P_s^{n,m}(x)$, $Q_s^{n,m}(x)$ mediante (a), claramente el grado de $P_s^{n,m}(x)$ no excede n . La expresión polinómica para $Q_s^{n,m}(x)$ contiene un término con x^{m+1} , pero el coeficiente que lo acompaña se anula. Por lo tanto el polinomio $Q_s^{n,m}(x)$ es como mucho de grado m . Finalmente,

$$T_s^{n,m}(x, y) = a_s q_s^{n-1,m} T_{s+1}^{n-1,m}(x, y) - a_{s+n+m} q_{s+1}^{n-1,m} T_s^{n-1,m}(x, y).$$

Por esto y por (1.10),

$$T_s^{n,m}(x_i, f_i) = 0 \quad \text{para } i = s, s+1, \dots, s+n+m.$$

Bajo la hipótesis general de que ninguna combinación (n, m, s) es inaccesible, lo anterior prueba que (a) define el numerador y el denominador de $\Phi_s^{n,m}$. □

Lamentablemente, como la recursión del teorema anterior todavía contiene los coeficientes $p_s^{n,m-1}$, $q_s^{n-1,m}$, las formulas son poco adecuadas para el cálculo de $\Phi_s^{n,m}(x)$ para un x dado. Sin embargo, podemos eliminar esos coeficientes con los siguientes teoremas.

Teorema 1.6.

$$(a) \quad \Phi_s^{n-1,m}(x) - \Phi_{s+1}^{n-1,m-1}(x) = k_1 \frac{(x - x_{s+1}) \cdots (x - x_{s+n+m-1})}{Q_s^{n-1,m}(x) Q_{s+1}^{n-1,m-1}(x)}$$

$$\text{con } k_1 = -p_{s+1}^{n-1,m-1} q_s^{n-1,m},$$

$$(b) \quad \Phi_{s+1}^{n-1,m}(x) - \Phi_{s+1}^{n-1,m-1}(x) = k_2 \frac{(x - x_{s+1}) \cdots (x - x_{s+n+m-1})}{Q_{s+1}^{n-1,m}(x) Q_{s+1}^{n-1,m-1}(x)}$$

$$\text{con } k_2 = -p_{s+1}^{n-1,m-1} q_{s+1}^{n-1,m}.$$

Demostración. El numerador de la expresión racional

$$\begin{aligned} \Phi_s^{n-1,m}(x) - \Phi_{s+1}^{n-1,m-1}(x) &= \\ &= \frac{P_s^{n-1,m}(x) Q_{s+1}^{n-1,m-1}(x) - P_{s+1}^{n-1,m-1}(x) Q_s^{n-1,m}(x)}{Q_s^{n-1,m}(x) Q_{s+1}^{n-1,m-1}(x)} \end{aligned}$$

es como mucho de grado $n - 1 + m$ y tiene $n + m - 1$ raíces diferentes

$$x_i, \quad i = s + 1, s + 2, \dots, s + n + m - 1$$

por definición de $\Phi_s^{n-1,m}$ y $\Phi_{s+1}^{n-1,m-1}$. Por lo tanto debe ser de la forma

$$k_1(x - x_{s+1}) \cdots (x - x_{s+n+m-1}) \quad \text{con} \quad k_1 = -p_{s+1}^{n-1,m-1} q_s^{n-1,m}.$$

Esto prueba (a), (b) se hace de forma análoga. □

Teorema 1.7. Para $n \geq 1, m \geq 1$,

$$(a) \quad \Phi_s^{n,m}(x) = \Phi_{s+1}^{n-1,m}(x) + \frac{\Phi_{s+1}^{n-1,m}(x) - \Phi_s^{n-1,m}(x)}{\frac{a_s}{a_{s+n+m}} \left[1 - \frac{\Phi_{s+1}^{n-1,m}(x) - \Phi_s^{n-1,m}(x)}{\Phi_{s+1}^{n-1,m}(x) - \Phi_{s+1}^{n-1,m-1}(x)} \right] - 1},$$

$$(b) \quad \Phi_s^{n,m}(x) = \Phi_{s+1}^{n,m-1}(x) + \frac{\Phi_{s+1}^{n,m-1}(x) - \Phi_s^{n,m-1}(x)}{\frac{a_s}{a_{s+n+m}} \left[1 - \frac{\Phi_{s+1}^{n,m-1}(x) - \Phi_s^{n,m-1}(x)}{\Phi_{s+1}^{n,m-1}(x) - \Phi_{s+1}^{n-1,m-1}(x)} \right] - 1}.$$

Demostración. Por el teorema (1.5),

$$\Phi_s^{n,m}(x) = \frac{a_s q_s^{n-1,m} P_{s+1}^{n-1,m}(x) - a_{s+n+m} q_{s+1}^{n-1,m} P_s^{n-1,m}(x)}{a_s q_s^{n-1,m} Q_{s+1}^{n-1,m}(x) - a_{s+n+m} q_{s+1}^{n-1,m} Q_s^{n-1,m}(x)}.$$

Asumimos que $p_{s+1}^{n-1,m-1} \neq 0$, y multiplicamos el numerador y el denominador de la fracción anterior por

$$\frac{-p_{s+1}^{n-1,m-1}(x - x_{s+1})(x - x_{s+2}) \cdots (x - x_{s+n+m-1})}{Q_{s+1}^{n-1,m}(x) Q_s^{n-1,m}(x) Q_{s+1}^{n-1,m-1}(x)}.$$

Teniendo en cuenta el teorema (1.6), llegamos a que

$$\Phi_s^{n,m}(x) = \frac{a_s \Phi_{s+1}^{n-1,m}(x) []_1 - a_{s+n+m} \Phi_s^{n-1,m}(x) []_2}{a_s []_1 - a_{s+n+m} []_2}, \quad (1.11)$$

donde

$$[]_1 = \Phi_s^{n-1,m}(x) - \Phi_{s+1}^{n-1,m-1}(x),$$

$$[]_2 = \Phi_{s+1}^{n-1,m}(x) - \Phi_{s+1}^{n-1,m-1}(x).$$

Y claramente (a) se obtiene de una sencilla transformación. La prueba para (b) se haría de forma análoga. □

las cuales surgen del teorema anterior si definimos $\Phi_{s+1}^{-1,m-1}(x) = 0$.

La experiencia nos dice que la secuencia de grados

$$(0, 0) \rightarrow (0, 1) \rightarrow (1, 1) \rightarrow (1, 2) \rightarrow (2, 2) \rightarrow \dots$$

(que está representada en el diagrama (1.12) por la línea de puntos) tiene ventajas, sobretodo en los métodos de extrapolación, donde el interés se centra en el valor de $\Phi_s^{n,m}(x)$ cuando $x = 0$. Si nos referimos a esta recurrencia en particular, es suficiente con indicar la suma de $n + m$, lo cual permite una notación más corta

$$T_{ik} = \Phi_s^{n,m}(x) \quad \text{con} \quad i = s + n + m, \quad k = n + m.$$

Las fórmulas del último teorema combinadas con (1.13) nos dan el siguiente algoritmo

$$\begin{aligned} T_{i0} &= f_i, \quad T_{i,-1} = 0, \\ T_{ik} &= T_{i,k-1} + \frac{T_{i,k-1} - T_{i-1,k-1}}{\frac{x - x_{i-k}}{x - x_i} \left[1 - \frac{T_{i,k-1} - T_{i-1,k-1}}{T_{i,k-1} - T_{i-1,k-2}} \right] - 1} \end{aligned} \quad (1.14)$$

para $1 \leq k \leq i$, $i = 0, 1, \dots$

Si ponemos los valores T_{ik} en la tabla de abajo, de tal forma que i contabilice las diagonales ascendentes y k las columnas, entonces en cada fórmula de (1.14) se utilizan las cuatro esquinas de los rombos formados:

$$\begin{array}{ccccccc} (n, m) = & (0, 0) & (0, 1) & (1, 1) & (1, 2) & \dots & \\ \hline & f_0 = T_{00} & & & & & \\ 0 = T_{0,-1} & & T_{11} & & & & \\ & f_1 = T_{10} & & T_{22} & & & \\ 0 = T_{1,-1} & & T_{21} & & T_{33} & & \\ & f_2 = T_{20} & & T_{32} & \vdots & \ddots & \\ 0 = T_{2,-1} & & T_{31} & \vdots & & & \\ \vdots & f_3 = T_{30} & \vdots & & & & \\ & \vdots & & & & & \end{array}$$

Si se está interesado en hallar la función racional en si misma, los métodos a partir de diferencias inversas y recíprocas son los más adecuados. Sin embargo, si lo que buscamos es el valor de la función de interpolación en un único punto, entonces el algoritmo de Neville, basado en el teorema (1.7) y en (1.14) es el más indicado. La fórmula (1.14), como ya comentamos anteriormente, es particularmente útil en los métodos de extrapolación, que veremos más adelante.

Capítulo 2

El desarrollo asintótico del error global

2.1. Introducción

Antes de poder continuar con los métodos de extrapolación, es necesario introducir unos cuantos resultados teóricos, en relación con los errores de los métodos numéricos para el problema de valores iniciales de un sistema de ecuaciones diferenciales ordinarias, que nos permitirán cimentar las bases de la extrapolación.

Dado el problema

$$\begin{cases} y' &= f(x, y), \quad x \in [a, b] \\ y(a) &= y_0, \end{cases}$$

seleccionamos una división del intervalo $[a, b]$ adecuada, tal que $a = x_0 < x_1 < \dots < x_N = b$.

En los métodos explícitos de un paso para aproximar la solución del problema anterior, la solución numérica se avanza paso a paso mediante una fórmula que escribiremos en la forma

$$y_{n+1} = y_n + h_n \Phi(x_n, y_n, h_n), \quad n = 0, 1, \dots, N-1, \quad (2.1)$$

donde $h_n := x_{n+1} - x_n$ y cada $y_n \approx y(x_n)$, $n = 0, 1, \dots, N$. Supondremos con carácter general que $\Phi(x, y, h)$ es una función continua de sus argumentos, y que además es Lipschitziana con respecto a y

$$\|\Phi(x, y_1, h) - \Phi(x, y_2, h)\| \leq \mathcal{L} \|y_1 - y_2\|, \quad \forall x \in [a, b], \quad \forall y_1, y_2 \in \mathbb{R}^n.$$

Esta condición no es difícil de establecer para los métodos más usuales si suponemos que la función f que define el problema es continua y Lipschitziana con respecto a su segundo argumento, una hipótesis que supondremos en todo lo que sigue.

La otra propiedad determinante del método es que el error de truncación local del método, en x , definido por

$$d(x) := y(x+h) - y(x) - \Phi(x, y(x), h), \quad x, x+h \in [a, b]$$

converja a cero cuando $h = \max_{n=0, \dots, N-1} (h_n) \rightarrow 0$, uniformemente en h_n , para $h_n \in (0, h_{\min}]$. $d(x)$ representa lo que se aparta la solución numérica de la solución exacta en un paso de longitud h cuando se parte de un punto de la solución exacta $y(x)$.

Diremos que el método es de orden p si para cada $x \in [a, b]$ el error de truncación local $d(x)$ en x tiene un desarrollo asintótico en potencias de h

$$\begin{aligned} y(x+h) - y(x) - h\Phi(x, y(x), h) = & \quad (2.2) \\ d_{p+1}(x)h^{p+1} + \dots + d_{N+1}(x)h^{N+1} + o(h^{N+1}), & (h \rightarrow 0) \end{aligned}$$

suponiendo que la solución $y = y(x)$ es suficientemente diferenciable.

2.2. Convergencia de métodos de un paso

Sean x e y puntos fijos arbitrarios, y sea $z(t)$ la solución exacta del problema de valor inicial

$$z'(t) = f(t, z(t)), \quad z(x) = y,$$

con valores iniciales x, y . Entonces la función

$$\Delta(x, y, h) := \begin{cases} \frac{z(x+h) - y}{h} & \text{si } h \neq 0, \\ f(x, y) & \text{si } h = 0 \end{cases}$$

representa el cociente de diferencias de la solución exacta $z(t)$ para paso h , mientras que $\Phi(x, y, h)$ es el cociente de diferencias para paso h de la solución aproximada producida por el método. La diferencia

$$\tau(x, y, h) := \Delta(x, y, h) - \Phi(x, y, h)$$

nos indica como de bien cumple el valor $z(x+h)$, con respecto a la solución exacta en $x+h$ con dato inicial (x, y) , la ecuación del método de un paso. Es decir, mide la calidad de la aproximación del método. Se suele llamar a $\tau(x, y, h)$ el error de discretización local en el punto (x, y) de dicho método.

Antes de pasar al teorema del error global probaremos un lema que nos será de utilidad en la demostración de dicho teorema.

Lema 2.1. *Si los números ξ_i satisfacen*

$$|\xi_{i+1}| \leq (1 + \delta)|\xi_i| + B, \quad \delta > 0 \quad B \geq 0, \quad i = 0, 1, 2, \dots,$$

entonces

$$|\xi_n| \leq e^{n\delta}|\xi_0| + \frac{e^{n\delta} - 1}{\delta}B.$$

Demostración. Por hipótesis tenemos que

$$\begin{aligned} |\xi_1| &\leq (1 + \delta)|\xi_0| + B, \\ |\xi_2| &\leq (1 + \delta)^2|\xi_0| + B(1 + \delta) + B, \\ &\vdots \\ |\xi_n| &\leq (1 + \delta)^n|\xi_0| + B [1 + (1 + \delta) + (1 + \delta)^2 + \dots + (1 + \delta)^{n-1}] \\ &= (1 + \delta)^n|\xi_0| + B \frac{(1 + \delta)^n - 1}{\delta} \\ &\leq e^{n\delta}|\xi_0| + B \frac{e^{n\delta} - 1}{\delta}, \end{aligned}$$

con $0 < 1 + \delta \leq e^\delta$ para $\delta > -1$, que es lo que queríamos probar. □

Teorema 2.1. *Dado el problema de valor inicial*

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

donde $x_0 \in [a, b]$, $y_0 \in \mathbb{R}$, y con solución exacta $y(x)$. Sea la función Φ continua en

$$G := \{(x, y, h) | a \leq x \leq b, |y - y(x)| \leq \gamma, 0 \leq |h| \leq h_0\}, \quad h_0 > 0, \quad \gamma > 0,$$

y donde existen constantes M y N tales que

$$|\Phi(x, y_1, h) - \Phi(x, y_2, h)| \leq M|y_1 - y_2|$$

para todo $(x, y_i, h) \in G$, $i = 1, 2$, y

$$|\tau(x, y(x), h)| = |\Delta(x, y(x), h) - \Phi(x, y(x), h)| \leq N|h|^p, \quad p > 0,$$

para todo $x \in [a, b]$, $|h| \leq h_0$. Entonces existe h' , $0 < h' \leq h_0$, tal que para el error de discretización global $e(x, h) = y_h(x) - y(x)$,

$$|e(x, h_n)| \leq |h_n|^p N \frac{e^{M|x-x_0|} - 1}{M}$$

para todo $x \in [a, b]$ y todo $h_n = (x - x_0)/n$, $n = 1, 2, \dots$, con $|h_n| \leq h'$. Si $\gamma = \infty$, entonces $h' = h_0$.

Demostración. La función

$$\tilde{\Phi}(x, y, h) := \begin{cases} \Phi(x, y, h) & \text{if } (x, y, h) \in G, \\ \Phi(x, y(x) + \gamma, h) & \text{if } x \in [a, b], |h| \leq h_0, y \geq y(x) + \gamma, \\ \Phi(x, y(x) - \gamma, h) & \text{if } x \in [a, b], |h| \leq h_0, y \leq y(x) - \gamma, \end{cases}$$

claramente es continua en $\tilde{G} := \{(x, y, h) / x \in [a, b], y \in \mathbb{R}, |h| \geq h_0\}$ y satisface la condición

$$|\tilde{\Phi}(x, y_1, h) - \tilde{\Phi}(x, y_2, h)| \leq M|y_1 - y_2|$$

para todo $(x, y_i, h) \in G$, $i = 1, 2$, y, como $\tilde{\Phi}(x, y(x), h) = \Phi(x, y(x), h)$, también cumple la condición

$$|\Delta(x, y(x), h) - \tilde{\Phi}(x, y(x), h)| \leq N|h|^p, \quad \text{para } x \in [a, b], |h| \leq h_0.$$

Cojamos ahora el método de un paso generado por $\tilde{\Phi}$ que nos proporciona los valores de aproximación $\tilde{y}_i := \tilde{y}(x_i, h)$ para $y_i := y(x_i)$, $x_i := x_0 + ih$:

$$\tilde{y}_{i+1} = \tilde{y}_i + h\tilde{\Phi}(x_i, \tilde{y}_i, h).$$

Teniendo en cuenta

$$y_{i+1} = y_i + h\Delta(x_i, y_i, h),$$

obtenemos del error $\tilde{e}_i := \tilde{y}_i - y_i$, restando, la recurrencia

$$\tilde{e}_{i+1} = \tilde{e}_i + h \left[\tilde{\Phi}(x_i, \tilde{y}_i, h) - \tilde{\Phi}(x_i, y_i, h) \right] + h \left[\tilde{\Phi}(x_i, y_i, h) - \Delta(x_i, y_i, h) \right].$$

Ahora de las dos desigualdades iniciales tenemos que

$$\begin{aligned} |\tilde{\Phi}(x_i, \tilde{y}_i, h) - \tilde{\Phi}(x_i, y_i, h)| &\leq M|\tilde{y}_i - y_i| = M|\tilde{e}_i|, \\ |\tilde{\Phi}(x_i, y_i, h) - \Delta(x_i, y_i, h)| &\leq N|h|^p, \end{aligned}$$

y por lo tanto gracias a la recurrencia anterior llegamos a

$$|\tilde{e}_{i+1}| \leq (1 + |h|M)|\tilde{e}_i| + N|h|^{p+1}.$$

El lema anterior, teniendo en cuenta que $\tilde{e}_0 = \tilde{y}_0 - y_0 = 0$, nos dice que

$$|\tilde{e}_k| \leq N|h|^p \frac{e^{k|h|M} - 1}{M}.$$

Ahora, sea $x \in [a, b]$, $x \neq x_0$, fijo y $h := h_n = (x - x_0)/n$, $n > 0$ entero. Entonces $x_n = x_0 + nh = x$ y de la desigualdad anterior con $k = n$, donde $\tilde{e}(x, h_n) = \tilde{e}_n$ llegamos finalmente a que

$$|\tilde{e}(x, h_n)| \leq |h_n|^p N \frac{e^{M|x-x_0|} - 1}{M}$$

para todo $x \in [a, b]$ y h_n con $|h_n| \leq h_0$. Como $|x - x_0| \leq |b - a|$ y $\gamma > 0$, existe un h' , $0 < h' \leq h_0$, tal que $|\tilde{e}(x, h_n)| \leq \gamma$ para todo $x \in [a, b]$, $|h_n| \leq h'$, es decir, para el método de un paso generado por Φ ,

$$\begin{aligned} \eta_0 &= y_0 \\ \eta_{i+1} &= \eta_i + \Phi(x_i, \eta_i, h), \end{aligned}$$

de acuerdo con la definición de $\tilde{\Phi}$, para $|h| \leq h'$,

$$\tilde{y}_i = \eta_i, \quad \tilde{e}_i = e_i, \quad \text{and} \quad \tilde{\Phi}(x_i, \tilde{y}_i, h) = \Phi(x_i, \eta_i, h).$$

Finalmente hemos probado el teorema, ya que

$$|e(x, h_n)| \leq |h_n|^p N \frac{e^{M|x-x_0|} - 1}{M}$$

para todo $x \in [a, b]$ y todo $h_n = (x - x_0)/n$, $n = 1, 2, \dots$, con $|h_n| \leq h'$. □

2.3. El desarrollo asintótico del error global

Teorema 2.2. *Sea $f(x, y)$ una función con derivadas parciales hasta orden $N + 1$ continuas y acotadas en la banda $S = \{(x, y) / a \leq x \leq b, y \in \mathbb{R}^n\}$ y sea $y_h(x)$ la solución aproximada que se obtiene al usar un*

método numérico de un paso de orden p , $p \leq N$, para la solución $y(x)$ del problema de valores iniciales

$$\begin{cases} y' &= f(x, y), & x \in [a, b] \\ y(a) &= y_0, & a = x_0. \end{cases}$$

Entonces $y_h(x)$ tiene un desarrollo asintótico de la forma

$$y_h(x) - y(x) = e_p(x)h^p + e_{p+1}(x)h^{p+1} + \cdots + e_N(x)h^N + E_{N+1}(x, h)h^{N+1}, \quad \text{con } e_k(x_0) = 0 \quad \text{para } k \geq p,$$

el cual es válido para todo $x \in [a, b]$ y todo $h = h_n = (x - x_0)/n$, $n = 1, 2, \dots$. Las funciones $e_i(x)$ son diferenciables e independientes de h , y el término residual $E_{N+1}(x, h)$ está acotado para x fijo y todos los $h = h_n = (x - x_0)/n$, $n = 1, 2, \dots$.

Demostración. Supongamos que el método de un paso viene dado por $\Phi(x, y, h)$. Si el método es de orden p , y f cumple las hipótesis del enunciado, entonces sabemos que

$$\begin{aligned} y(x+h) - y(x) - h\Phi(x, y, h) \\ = d_{p+1}(x)h^{p+1} + \cdots + d_{N+1}(x)h^{N+1} + O(h^{N+2}). \end{aligned}$$

Nosotros usaremos solo que

$$y(x+h) - y(x) - h\Phi(x, y, h) = d_{p+1}(x)h^{p+1} + O(h^{p+2})$$

y veremos entonces que hay una función diferenciable $e_p(x)$ tal que

$$y_h(x) - y(x) = e_p(x)h^p + O(h^{p+1}), \quad e_p(x_0) = 0.$$

Con este fin consideramos la función

$$\tilde{y}_h(x) := y_h(x) - e_p(x)h^p, \tag{2.3}$$

donde la elección de $e_p(x)$ sigue estando abierta. Es fácil ver que $\tilde{y}_h(x)$ se puede ver como el resultado de otro método de un paso,

$$\tilde{y}_h(x+h) = \tilde{y}_h(x) + h\tilde{\Phi}(x, \tilde{y}_h(x), h),$$

si $\tilde{\Phi}$ se define como

$$\tilde{\Phi}(x, y, h) := \Phi(x, y + e_p(x)h^p, h) - (e_p(x+h) - e_p(x))h^{p-1}.$$

Nuestro objetivo es encontrar $e_p(x)$, con $e_p(x_0) = 0$, tal que el método con la función incremento $\tilde{\Phi}$ sea de orden $p + 1$. Desarrollando el error local del método de un paso $\tilde{\Phi}$ en potencias de h llegamos a

$$\begin{aligned} y(x+h) - y(x) - h\tilde{\Phi}(x, y(x), h) &= \\ &= \left(d_{p+1}(x) - \frac{\partial f}{\partial y}(x, y(x))e_p(x) + e'_p(x) \right) h^{p+1} + O(h^{p+2}) \end{aligned}$$

donde hemos usado que

$$\frac{\partial \Phi}{\partial y}(x, y, 0) = \frac{\partial f}{\partial y}(x, y).$$

Por lo tanto el término en h^{p+1} desaparece si $e_p(x)$ lo definimos como solución de

$$e'_p(x) = d_{p+1}(x) - f_y(x, y(x))e_p(x), \quad e_p(x_0) = 0. \quad (2.4)$$

Con esta elección de $e_p(x)$, aplicando el teorema (2.1) a $\tilde{\Phi}$ llegamos a que

$$\tilde{y}_h(x) - y(x) = y_h(x) - y(x) - e_p(x)h^p = O(h^{p+1})$$

osea que

$$y_h(x) - y(x) = e_p(x)h^p + O(h^{p+1}).$$

Repitiendo estos argumentos con $\tilde{\Phi}$ en vez de con Φ se completa la prueba. □

El teorema puede extenderse a la situación en la que la longitud de paso es variable mediante el siguiente artificio. Supondremos que existe una función $\tau(x) > 0$ tal que la longitud de paso depende de ella y de un parámetro h

$$x_{i+1} - x_i = \tau(x_i)h.$$

Entonces el desarrollo asintótico (2.2) del error de truncación local pasa a ser

$$y(x + \tau(x)h) - y(x) - h\tau(x)\Phi(x, y(x), \tau(x)h) = d_{p+1}(x)\tau^{p+1}(x)h^{p+1} + \dots$$

y en la definición de $\tilde{\Phi}$ en la demostración anterior ponemos

$$\tilde{\Phi}(x, y, \tau(x)h) := \Phi(x, y + e_p(x)h^p, \tau(x)h) - (e_p(x + \tau(x)h) - e_p(x))\frac{h^p}{\tau(x)h}.$$

Desarrollando el error local del método de un paso $\tilde{\Phi}$ en potencias de h obtenemos

$$\begin{aligned} y(x+\tau(x)h) - y(x) - h\tau(x)\tilde{\Phi}(x, y(x), \tau(x)h) &= \\ &= \tau(x) \left(d_{p+1}(x)\tau^p(x) - \frac{\partial f}{\partial y}(x, y(x))e_p(x) + e'_p(x) \right) h^{p+1} + O(h^{p+2}) \end{aligned}$$

y se puede generalizar con unas ligeras modificaciones del teorema (2.2).

2.4. El desarrollo asintótico en métodos simétricos

Los métodos de extrapolación más importantes usan desarrollos asintóticos con potencias pares de h . Para poder dar una base teórica para estos métodos, tenemos que explicar antes el sentido de $y_h(x)$ para h negativo.

Si reescribimos (2.1) como

$$y_h(x+h) = y_h(x) + h\Phi(x, y_h(x), h)$$

y reemplazamos h por $-h$ obtenemos

$$y_{-h}(x-h) = y_{-h}(x) - h\Phi(x, y_{-h}(x), -h).$$

sustituyendo ahora x por $x+h$

$$y_{-h}(x) = y_{-h}(x+h) - h\Phi(x+h, y_{-h}(x+h), -h). \quad (2.5)$$

Obtenemos entonces una ecuación implícita para $y_{-h}(x+h)$, que poseerá una única solución para h suficientemente pequeño. Si expresamos dicha solución así

$$y_{-h}(x+h) = y_{-h}(x) - h\tilde{\Phi}(x, y_{-h}(x), h), \quad (2.6)$$

comparando (2.5) y (2.6) (con $A = y_{-h}(x+h)$, $B = y_{-h}(x)$), tenemos la siguiente definición:

Definición 2.1. Sea $\Phi(x, y, h)$ la función incremento de un método. Entonces, nosotros definimos la función incremento $\Phi^*(x, y, h)$ del método adjunto con el par de fórmulas siguientes:

$$\begin{aligned} B &= A - h\Phi(x+h, A, -h), \\ A &= B + h\Phi^*(x, B, h). \end{aligned} \quad (2.7)$$

Pasemos a ver unas cuantas propiedades de los método adjuntos.

Teorema 2.3. $\Phi^{**} = \Phi$.

Demostración. Sustituyendo en (2.7) h por $-h$ y luego x por $x + h$, B por A y A por B se ve claramente. □

Teorema 2.4. *El método adjunto tiene el mismo orden que el método original. El término principal del error es el termino principal del error del original multiplicado por $(-1)^p$.*

Demostración. Reemplazamos h por $-h$ en (2.2), posteriormente x por $x+h$ y reorganizamos los términos. Esto nos lleva a (usando que $d_{p+1}(x+h) = d_{p+1}(x) + O(h)$)

$$\begin{aligned} y(x) + d_{p+1}(x)h^{p+1}(-1)^p + O(h^{p+2}) \\ = y(x+h) - h\Phi(x+h, y(x+h), -h). \end{aligned}$$

Denotamos por A a la parte izquierda de la igualdad, con $B = y(x+h)$, y usamos (2.7). Nos lleva a que

$$y(x+h) = y(x) + d_{p+1}(x)h^{p+1}(-1)^p + h\Phi^*(x, y(x), h) + O(h^{p+2}),$$

que es lo que pedía el teorema. □

Teorema 2.5. *El método adjunto tiene el mismo desarrollo asintótico que el método original, donde reemplazamos h por $-h$.*

Demostración. Repetimos el procedimiento con el que probamos el teorema del desarrollo asintótico, pero con h negativo. De tal forma que acabamos llegando a que

$$y_{-h}(x) - y(x) = e_p(x)(-h)^p + O(h^{p+1}),$$

con lo que tendríamos el primer término, ya que la solución de (2.4) con valor inicial $e_p(x_0) = 0$ tiene los mismos cambios de signo que $d_{p+1}(x)$.

Para poder finalizar tenemos que ver que la transformación (2.3) conmuta con el operador adjunto, es decir, que

$$\tilde{\Phi}^* = (\tilde{\Phi}^*).$$

Para ello, por (2.3) y por la definición de $\tilde{\Phi}$ tenemos que

$$\begin{aligned} y_h(x+h) - e_p(x+h)h^p \\ = y_h(x) - e_p(x)h^p + h\tilde{\Phi}(x, y_h(x) - e_p(x)h^p, h). \end{aligned}$$

Sustituimos ahora h por $-h$ seguido de la sustitución de x por $x+h$. Finalmente, aplicamos (2.7), donde $B = y_{-h}(x) - e_p(x)(-h)^p$ y $A = y_{-h}(x+h) - e_p(x+h)(-h)^p$, para obtener

$$\begin{aligned} y_{-h}(x+h) - e_p(x+h)(-h)^p \\ = y_{-h}(x) - e_p(x)(-h)^p + h\tilde{\Phi}^*(x, y_{-h}(x) - e_p(x)(-h)^p, h). \quad (2.8) \end{aligned}$$

Por otro lado, gracias al teorema (2.4), si realizamos la transformación

$$\tilde{y}_{-h}(x) = y_{-h}(x) - e_p(x)(-h)^p$$

y esto lo introducimos en (2.6), obtenemos de nuevo (2.8), pero con $(\tilde{\Phi}^*)$ en vez de $\tilde{\Phi}^*$, con lo que hemos terminado la demostración. \square

Tras estas propiedades de los métodos adjuntos que eran necesarias dar a conocer, vamos a acabar el capítulo con la definición de los métodos simétrico y su desarrollo asintótico.

Definición 2.2. *Un método es simétrico si $\Phi = \Phi^*$.*

Teorema 2.6. *Si añadimos al teorema (2.2) la condición de que el método numérico es simétrico, entonces el desarrollo asintótico solo contiene potencias pares de h :*

$$y_h(x) - y(x) = e_{2q}(x)h^{2q} + e_{2q+2}(x)h^{2q+2} + \dots$$

con $e_{2j} = 0$.

Demostración. Si $\Phi = \Phi^*$, tenemos que $y_{-h}(x) = y_h(x)$. Entonces por (2.6) y por el teorema (2.5) tenemos que el resultado es evidente. \square

Capítulo 3

Métodos de extrapolación y su aplicación a problemas de valores iniciales

3.1. Introducción

Sea

$$\begin{cases} y' &= f(x, y), \quad x \in [a, b] \\ y(a) &= y_0 \end{cases}$$

un sistema diferencial dado, si denotamos por e_0 la solución del problema en un instante $x_0 + H$, aproximando por un método numérico de orden p , el error de dicho método admite, para soluciones suficientemente regulares, un desarrollo de Taylor

$$F(h) = e_0 + e_p h^p + e_{p+1} h^{p+1} + \dots \quad (3.1)$$

Los métodos de extrapolación eliminan distintos términos $e_i h^i$ mediante combinaciones lineales de $F(h)$ en distintos h , de tal forma que el método de extrapolación tiene como desarrollo de Taylor del error

$$\tilde{F}(h) = e_0 + \tilde{e}_k h^k + \tilde{e}_{k+1} h^{k+1} + \dots$$

con $k > p$.

3.2. Definición del método

En primer lugar, se tiene que escoger una secuencia de enteros positivos

$$n_1 < n_2 < n_3 < \dots$$

de tal forma que con ella definimos las correspondiente longitudes de paso

$$h_1 > h_2 > h_3 > \dots, \quad \text{donde} \quad h_i = \frac{H}{n_i}.$$

Posteriormente escogemos un método numérico de orden p y hallamos los resultados para nuestro problema de valores iniciales realizando n_i pasos de longitud h_i para así obtener

$$y_{h_i}(x_0 + H) = T_{i,1} \equiv F(h_i).$$

Luego, eliminando términos de (3.1), llegamos al polinomio interpolador

$$p(h) = e_0 + e_p h^p + e_{p+1} h^{p+1} + \dots + e_{p+k-2} h^{p+k-2} \quad (3.2)$$

de tal forma que

$$p(h_i) = T_{i,1}, \quad i = j, \dots, k - j + 1. \quad (3.3)$$

Finalmente, extarpolando sucesivamente, lo que estamos logrando es eliminar los términos de (3.1) con h de grado menor o, lo que es lo mismo, estamos haciendo que $h \rightarrow 0$ cuando $j \rightarrow \infty$ y por lo tanto

$$p(0) = e_0 \equiv T_{i,j}$$

usando $h = 0$ como aproximación numérica.

Teorema 3.1. *El valor $T_{i,j}$ representa un método numérico de orden $p + j - 1$.*

Demostración. Usando el desarrollo asintótico (donde $N = p + j - 1$)

$$y_h(x) = y(x) + e_p(x)h^p + e_{p+1}(x)h^{p+1} + \dots + e_N(x)h^N + E_{N+1}(x)h^{N+1}$$

evaluado en $x_0 + H$ y teniendo en cuenta (3.2) y (3.3) llegamos a

$$T_{i,1} = y(x_0 + H) + e_p(x_0 + H)h_i^p + \dots + e_{p+j-2}(x_0 + H)h_i^{p+j-2} + \Delta_i,$$

donde

$$\Delta_i = e_N(x_0 + H)h_i^N + E_{N+1}(x_0 + H)h_i^{N+1} = O(H^{p+j})$$

ya que $e_N(x_0) = 0$ y $h_i \leq H$. Esto que se nos presenta aquí es un sistema de ecuaciones lineal para las incógnitas

$$y(x_0 + H), e_p(x_0 + H)H^p, \dots, e_{p+j-2}(x_0 + H)H^{p+j-2}$$

con matriz

$$A = \begin{bmatrix} 1 & \frac{1}{n_j^p} & \frac{1}{n_j^{p+1}} & \cdots & \frac{1}{n_j^{p+j-2}} \\ 1 & \frac{1}{n_{j+1}^p} & \frac{1}{n_{j+1}^{p+1}} & \cdots & \frac{1}{n_{j+1}^{p+j-2}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \frac{1}{n_{k-j+1}^p} & \frac{1}{n_{k-j+1}^{p+1}} & \cdots & \frac{1}{n_{k-j+1}^{p+j-2}} \end{bmatrix}.$$

Tenemos entonces el sistema

$$A \begin{bmatrix} y(x_0 + H) \\ e_p(x_0 + H)H^p \\ \vdots \\ e_{p+j-2}(x_0 + H)H^{p+j-2} \end{bmatrix} = \begin{bmatrix} T_{j,1} \\ T_{j+1,1} \\ \vdots \\ T_{k-j+1,1} \end{bmatrix} - \begin{bmatrix} \Delta_j \\ \Delta_{j+1} \\ \vdots \\ \Delta_{k-j+1} \end{bmatrix},$$

que claramente es similar a (3.3) pero con el lado derecho perturbado por los términos Δ_i . Como la matriz A es claramente de Vandermonde, entonces sabemos que es invertible y por lo tanto, restando con $h = 0$ en (3.2), obtenemos

$$|y(x_0 + H) - e_0| \leq \|A^{-1}\|_{\infty} \max_{i=j, \dots, k-j+1} |\Delta_i| = O(H^{p+j}).$$

como error de truncación y por lo tanto el error global es $O(H^{p+j-1})$. \square

Veamos ahora los pasos numéricos más comunes que se usan para los n_i :

$$1, 2, 4, 8, 16, 32, \dots \quad (3.4)$$

$$1, 2, 3, 4, 6, 8, 12, 16, 24, 32, \dots \quad (3.5)$$

$$1, 2, 3, 4, 5, 6, 7, \dots \quad (3.6)$$

La sucesión (3.4) da resultados muy precisos, pero es demasiado cara computacionalmente, ya que reduciendo la longitud de paso a la mitad el trabajo para el ordenador se duplica. Por otro lado la sucesión (3.6) es muy asequible computacionalmente pero lleva a inestabilidades en ciertos algoritmos de extrapolación ya que

$$\liminf \left(\frac{n_{i+1}}{n_i} \right) \rightarrow 1, \quad \text{cuando } i \rightarrow \infty.$$

Por el contrario, tanto para la sucesión (3.4), como para la sucesión (3.5), el límite permanece alejado de 1. Además la sucesión (3.5) no tiene un coste computacional tan alto a la hora de avanzar con el método en comparación con (3.4). Otra particularidad de (3.4) y (3.5) es que para problemas $y' = f(x)$ (cuadratura de Romberg), muchos valores de funciones se pueden guardar y volver a usar para valores de h_i más pequeños.

3.3. Extrapolación de Richardson

$$\begin{array}{cccc} T_{11} & & & \\ T_{21} & T_{22} & & \\ T_{31} & T_{32} & T_{33} & \\ \vdots & \vdots & \vdots & \ddots \end{array}$$

Las entradas de la tabla se evalúan de tal forma que para la primera columna se usa (3.3) y posteriormente el algoritmo

$$T_{i,j+1} = T_{i,j} + \frac{T_{i,j} - T_{i-1,j}}{\frac{n_i}{n_{i-j}} - 1} \quad (3.7)$$

$$j = 1, \dots, k-1; i = j+1, \dots, k-j.$$

3.4. Extrapolación racional

Este tipo de extrapolación se basa en lo visto en el primer capítulo (donde ya explicamos ampliamente la tabla y el algoritmo). En este caso se trata de sustituir (3.7) del algoritmo anterior por

$$T_{i,j+1} = T_{i,j} + \frac{T_{i,j} - T_{i-1,j}}{\frac{n_i}{n_{i-j}} \left[1 - \frac{T_{i,j} - T_{i-1,j}}{T_{i,j} - T_{i-1,j-1}} \right] - 1} \quad (3.8)$$

donde

$$T_{i,0} = 0.$$

Por lo general la extapolación racional no es mas ventajosa que la extapolación de Richardson (o polinómica).

3.5. Extrapolación con métodos simétricos

Si el método numérico que usamos es simétrico, entonces el desarrollo asintótico (3.1) es en potencias de h^2

$$F(h) = e_0 + e_p(h^2)^p + e_{p+1}(h^2)^{p+1} + \dots,$$

y entonces cada paso de la extrapolación haría que se eliminasen dos potencia de h .

Ahora, reemplazando en (3.2) h por h^2 , se genera un cambio en los algoritmos (3.7) y (3.8) de tal forma que (3.7) quedaría tal que así

$$T_{i,j+1} = T_{i,j} + \frac{T_{i,j} - T_{i-1,j}}{\left(\frac{n_i}{n_{i-j}}\right)^2 - 1} \quad (3.9)$$

y (3.8) así

$$T_{i,j+1} = T_{i,j} + \frac{T_{i,j} - T_{i-1,j}}{\left(\frac{n_i}{n_{i-j}}\right)^2 \left[1 - \frac{T_{i,j} - T_{i-1,j}}{T_{i,j} - T_{i-1,j-1}}\right] - 1}. \quad (3.10)$$

3.6. El método Gragg-Bulirsch-Stoer(GBS)

Para poder aplicar extrapolación en la situación anterior a problemas de valores iniciales, necesitamos un método numérico, como base de la extrapolación, que verifique que

$$F(h) = e_0 + e_1h^2 + e_2h^4 + e_3h^6 + \dots \quad (3.11)$$

Esto quiere decir que si $y(x)$ es solución de la ecuación diferencial sujeta a $y(x_0) = y_0$ e $y_i, i = 1, \dots, N$, es la solución aproximada en el punto $x_i = x_0 + ih$, entonces para cualquier punto fijo $x = x_0 + Nh$, la solución aproximada $y_h(x)$ se puede expresar como

$$y_h(x) = y(x) + a_1h^2 + a_2h^4 + a_3h^6 + \dots \quad (3.12)$$

Gragg llegó a la conclusión de que $y_h(x)$, producido por el siguiente algoritmo

$$\begin{cases} y_1 &= y_0 + hf(x_0, y_0) \\ y_{i+1} &= y_{i-1} + 2hf(x_i, y_i), \quad i = 1, \dots, 2n \\ y_h(x) &= \frac{1}{4}(y_{2n-1} + 2y_{2n} + y_{2n+1}) \end{cases} \quad (3.13)$$

donde $x = x_0 + 2nh$ y $x_i = x_0 + ih$, tenía un desarrollo en potencias de pares de h como en (3.11) y unas excelentes propiedades de estabilidad. Pero la prueba de este resultado se antoja excesivamente larga y compleja. Para soslayar este problema, Stetter tuvo la gran idea de reinterpretar (3.13) como un único algoritmo donde los índices pares e impares están separados. Para ello hay que definir primero los siguientes elementos:

$$\begin{aligned} h^* &= 2h, \quad x_k^* = x_0 + kh^*, \quad u_0 = v_0 = y_0, \\ u_k &= y_{2k}, \quad v_k = y_{2k+1} - hf(x_{2k}, y_{2k}) = \frac{1}{2}(y_{2k+1} + y_{2k-1}). \end{aligned} \quad (3.14)$$

Entonces podemos reescribir (3.13)

$$\begin{aligned} u_k = y_{2k} &\Rightarrow u_{k+1} = y_{2(k+1)} = y_{(2k+1)+1} = y_{(2k+1)-1} + 2hf(x_{2k+1}, y_{2k+1}), \\ u_{k+1} &= y_{2k} + 2hf(x_{2k+1}, y_{2k+1}) \\ \Rightarrow u_{k+1} &= u_k + 2hf(x_0 + (2k+1)h, v_k + hf(x_{2k}, y_{2k})) \\ &= u_k + h^*f(x_0 + 2kh + h, v_k + \frac{h^*}{2}f(x_0 + 2kh, u_k)) \\ &= u_k + h^*f(x_0 + kh^* + \frac{h^*}{2}, v_k + \frac{h^*}{2}f(x_0 + kh^*, u_k)) \\ &= u_k + h^*f(x_k^* + \frac{h^*}{2}, v_k + \frac{h^*}{2}f(x_k^*, u_k)) \\ v_k = y_{2k+1} - hf(x_{2k}, y_{2k}) &\Rightarrow v_{k+1} = y_{2(k+1)+1} - hf(x_{2(k+1)}, y_{2(k+1)}) \\ \Rightarrow v_{k+1} &= y_{(2k+2)+1} - hf(x_{2k+2}, y_{2k+2}) \\ &= y_{(2k+2)-1} + 2hf(x_{2k+2}, y_{2k+2}) - hf(x_{2k+2}, y_{2k+2}) \\ &= y_{2k+1} + hf(x_{2k+2}, y_{2k+2}) \\ &= y_{2k+1} - hf(x_{2k}, y_{2k}) + hf(x_{2k}, y_{2k}) + hf(x_{2k+2}, y_{2k+2}) \\ &= v_k + \frac{h^*}{2} [f(x_0 + 2kh, u_k) + f(x_0 + (2k+2)h, u_{k+1})] \\ &= v_k + \frac{h^*}{2} [f(x_k^*, u_k) + f(x_k^* + h^*, u_{k+1})] \end{aligned}$$

Y escribiendolo en forma matricial

$$\begin{bmatrix} u_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} u_k \\ v_k \end{bmatrix} + h^* \begin{bmatrix} f(x_k^* + \frac{h^*}{2}, v_k + \frac{h^*}{2} f(x_k^*, u_k)) \\ \frac{1}{2} [f(x_k^* + h^*, u_{k+1}) + f(x_k^*, u_k)] \end{bmatrix} \quad (3.15)$$

Este método que a partir del par (u_k, v_k) nos lleva en el par (u_{k+1}, v_{k+1}) nos va a facilitar la prueba de las dos propiedades que nos interesan de (3.13). Primero veamos que (3.15) es un método simétrico, y por lo tanto, (3.13) sería también un método simétrico.

Demostración. Tenemos que ver que cambiando u_{k+1} por u_k , v_{k+1} por v_k , h^* por $-h^*$ y x_k^* por $x_k^* + h^*$ (3.15) no varía.

$$\begin{bmatrix} u_k \\ v_k \end{bmatrix} = \begin{bmatrix} u_{k+1} \\ v_{k+1} \end{bmatrix} - h^* \begin{bmatrix} f(x_k^* + h^* - \frac{h^*}{2}, v_{k+1} - \frac{h^*}{2} f(x_k^* + h^*, u_{k+1})) \\ \frac{1}{2} [f(x_k^* + h^* - h^*, u_k) + f(x_k^* + h^*, u_{k+1})] \end{bmatrix},$$

despejando u_{k+1} y v_{k+1}

$$\begin{bmatrix} u_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} u_k \\ v_k \end{bmatrix} + h^* \begin{bmatrix} f(x_k^* + \frac{h^*}{2}, v_{k+1} - \frac{h^*}{2} f(x_k^* + h^*, u_{k+1})) \\ \frac{1}{2} [f(x_k^*, u_k) + f(x_k^* + h^*, u_{k+1})] \end{bmatrix},$$

ahora solo nos queda ver que

$$v_{k+1} - \frac{h^*}{2} f(x_k^* + h^*, u_{k+1}) = v_k + \frac{h^*}{2} f(x_k^*, u_k).$$

$$\begin{aligned} & v_{k+1} - \frac{h^*}{2} f(x_k^* + h^*, u_{k+1}) \\ &= y_{2(k+1)+1} - h f(x_{2(k+1)}, y_{2(k+1)}) - \frac{h^*}{2} f(x_k^* + h^*, u_{k+1}) \\ &= v_k + \frac{h^*}{2} [f(x_k^*, u_k) + f(x_k^* + h^*, u_{k+1})] - \frac{h^*}{2} f(x_k^* + h^*, u_{k+1}) \\ &= v_k + \frac{h^*}{2} f(x_k^*, u_k). \end{aligned}$$

Por lo tanto tenemos entonces (3.15) y hemos probado la simetría. \square

Una vez vista la simetría del método, nos queda probar que el método de Gragg de verdad tiene un desarrollo asintótico en potencias de h^2 .

Teorema 3.2. *Sea $f(x, y) \in C^{2l+2}$, entonces la solución numérica dada por (3.13) tiene un desarrollo asintótico de la forma*

$$y(x) - y_h(x) = \sum_{j=1}^l e_j(x)h^{2j} + h^{2l+2}C(x, h) \quad (3.16)$$

con $e_j(x_0) = 0$ y $C(x, h)$ acotado por $x_0 \leq x \leq x_0 + 2nh$ y por $0 \leq h \leq H$.

Demostración. Ya hemos visto que realizando las sustituciones (3.14) en (3.13) pasamos a tener el sistema (3.15). Este sistema es consistente con la ecuación diferencial

$$\begin{cases} u' = f(x, v), & u(x_0) = y_0, \\ v' = f(x, u), & v(x_0) = y_0, \end{cases}$$

cuando $h^* \rightarrow 0$ (donde la solución exacta es $u(x) = v(x) = y(x)$). Como ya hemos visto, el método es simétrico, y por lo tanto los desarrollos asintóticos de los errores son tal que

$$\begin{aligned} y(x) - u_{h^*}(x) &= \sum_{j=1}^l a_j(x)(h^*)^{2j} + (h^*)^{2l+2}A(x, h^*), \\ y(x) - v_{h^*}(x) &= \sum_{j=1}^l b_j(x)(h^*)^{2j} + (h^*)^{2l+2}B(x, h^*), \end{aligned}$$

con $a_j(x_0) = b_j(x_0) = 0$ y $A(x, h^*), B(x, h^*)$ acotados por $x_0 \leq x \leq x_0 + 2nh$ y por $0 \leq \frac{h^*}{2} \leq H$. Sumando ambas expresiones y usando que $h^* = 2h$

$$\begin{aligned} 2y(x) - u_{2h}(x) - v_{2h}(x) \\ = \sum_{j=1}^l (a_j(x) - b_j(x))2^{2j}h^{2j} + 2^{2l+2}h^{2l+2}(A(x, 2h) + B(x, 2h)), \end{aligned}$$

entonces dividiendo por dos la expresión anterior

$$\begin{aligned} y(x) - \frac{1}{2}(u_{2h}(x) + v_{2h}(x)) \\ = \sum_{j=1}^l (a_j(x) - b_j(x))2^{2j-1}h^{2j} + 2^{2l+1}h^{2l+2}(A(x, 2h) + B(x, 2h)) \end{aligned}$$

y denotando a $(a_j(x) - b_j(x))2^{2j-1}$ como $e_j(x)$ y a $2^{2l+1}(A(x, 2h) + B(x, 2h))$ por $C(x, h)$, como

$$\begin{aligned} y_h(x_0 + 2nh) &= \frac{1}{4}(y_{2n-1} + 2y_{2n} + y_{2n+1}) = \frac{1}{2}\left(\frac{1}{2}(y_{2n+1} + y_{2n-1}) + y_{2n}\right) \\ &= \frac{1}{2}(v_n + u_n), \end{aligned}$$

llegamos a que

$$y(x) - y_h(x) = \sum_{j=1}^l e_j(x)h^{2j} + h^{2l+2}C(x, h),$$

donde $e_j(x_0) = 0$ y $C(x, h)$ está acotado por $x_0 \leq x \leq x_0 + 2nh$ y por $0 \leq h \leq H$. □

El método GBS se puede usar para la extrapolación de la misma manera que el método simétrico anterior: primero elegimos una secuencia de longitudes de paso, pero con la condición de que cada n_j sea par, es decir, por ejemplo

$$\begin{aligned} &2, 4, 8, 16, 32, 64, \dots \\ &2, 4, 6, 8, 12, 16, \dots \\ &2, 4, 6, 8, 10, 12, \dots, \end{aligned}$$

que son las secuencias vistas con anterioridad, pero donde hemos eliminado los n_j impares. Posteriormente fijamos que

$$T_{i,1} = y_{h_i}(x_0 + H)$$

y calculamos las expresiones $T_{i,j}$, que ahora son en términos de h^2 , mediante la extrapolación del algoritmo (3.9) o (3.10).

Aunque no tiene mayor relevancia para el algoritmo numérico que hemos visto, para completar un poco la sección, vamos a probar la existencia del desarrollo asintótico en términos de h^2 para y_{2k+1} .

Teorema 3.3. *Para $x = x_0 + (2k + 1)h$ tenemos que*

$$y(x) - y_h(x) = \sum_{j=1}^l b_j(x)h^{2j} + h^{2l+2}B(x, h)$$

donde $b_j(x_0) \neq 0$.

Demostración. y_{2k+1} se puede hallar tanto avanzando un paso a partir de v_k , como retrocediendo un paso desde v_{k+1} . Con motivo de la simetría, usamos la media de ambas expresiones, y obtenemos

$$y_{2k+1} = \frac{1}{2}(v_k + v_{k+1}) + \frac{h}{2}(f(x_k^*, u_k) - f(x_{k+1}, u_{k+1})).$$

Posteriormente restamos la solución exacta

$$\begin{aligned} 2(y_h(x) - y(x)) &= v_{2h}(x-h) - y(x-h) \\ &\quad + v_{2h}(x+h) - y(x+h) + y(x-h) - 2y(x) + y(x+h) \\ &\quad + h[f(x-h, u_{2h}(x-h)) - f(x+h, u_{2h}(x+h))]. \end{aligned}$$

Debido a la simetría de $u_{2h}(x)$ y de $v_{2h}(x)$, la expresión anterior es simétrica respecto a h . Y por lo tanto la expresión asintótica de y_{2k+1} no tiene potencias impares de h . □

3.7. Control de orden y del tamaño del paso

Sea

$$\varepsilon_{k+1,k} = ||T_{k+1,k} - T_{k+1,k+1}||$$

el estimativo del error de la k -ésima columna y la $(k-1)$ -ésima fila de la tabla de extrapolación. Entonces, para que el paso sea aceptado, tiene que ocurrir que

$$\varepsilon_{k+1,k} < tol \tag{3.17}$$

donde tol es la tolerancia dada al programa. Como nosotros vamos a utilizar para nuestro programa la sucesión de números pares,

$$n = 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, \dots,$$

entonces el orden del método es $2k + 1$, es decir,

$$\varepsilon_{k+1,k} \sim H^{2k+1}$$

donde H es el tamaño del paso que vamos a dar ese momento. Por lo tanto, para estimar un nuevo H_k , de tal forma que converja para la columna k , tendría que ser

$$H_k = H \left(\frac{tol}{\varepsilon_{k+1,k}} \right)^{\frac{1}{2k+1}}.$$

Ahora que ya tenemos claro cual es la longitud de paso adecuada para k , tenemos que ver que k es el más óptimo, es decir, el orden del método en cada paso. Para ello vamos a comparar el trabajo requerido para cada k . Consideraremos que el trabajo es principalmente el coste de las evaluaciones de función definidas por la parte derecha de la ecuación diferencial. Si tenemos n_k divisiones de H , entonces podemos calcular el número de evaluaciones de función con la recurrencia

$$\begin{aligned} A_1 &= n_1 + 1, \\ A_{k+1} &= A_k + n_{k+1}. \end{aligned}$$

Por lo tanto, el trabajo por paso para obtener la columna k es A_{k+1}/H_k , que nosotros no dimensionalizamos con H

$$W_k = \frac{A_{k+1}}{H_k} H = A_{k+1} \left(\frac{\varepsilon_{k+1,k}}{tol} \right) \frac{1}{2k+1}.$$

Entonces la columna óptima será el q tal que

$$W_q = \min_{k=1, \dots, k_f} W_k$$

donde k_f es la columna final tal que (3.17) se cumple.

De todo este proceso surgen dos problemas:

-Si el H actual es “demasiado pequeño”, k_f será “demasiado pequeño” y por lo tanto lo mismo con q . Entonces será necesario aumentar H para poder tener convergencia con $q > k_f$.

-Si el H actual es “demasiado grande”, puede ocurrir que no tengamos convergencia en ningún caso y haya que disminuir H . Lo que buscaremos será controlar los $\varepsilon_{k+1,k}$ para cada k de tal forma que podamos parar el paso actual cuanto antes.

Deuffhard [2] usó ideas de teoría de comunicación para poder soslayar estos dos problemas. Para ello generó unos factores $\alpha(k, q)$ que solo dependían de la tolerancia y de la serie de números utilizada en la extrapolación:

$$\alpha(k, q) = tol \frac{A_{k+1} - A_{q+1}}{(2k+1)(A_{q+1} - A_1 + 1)}, \quad \text{para } k < q.$$

Entonces, en el caso del primer problema, supongamos que $q = k_f$. Nosotros queremos incrementar H_q de tal forma que tengamos convergencia en en la columna $q+1$, pero no podemos calcular H_{q+1} , entonces, lo que hacemos es estimarlo con estos factores $\alpha(k, q)$

$$H_{q+1} = H_q \alpha(q, q+1).$$

Esta estimación solo tiene sentido si

$$W_q = \frac{A_{q+1}}{H_q} > \frac{A_{q+2}}{H_{q+1}} = W_{q+1}$$

o lo que es lo mismo

$$A_{q+1}\alpha(q, q+1) > A_{q+2}.$$

Como en en estas desigualdades solo intervienen A y α , esto se puede determinar al principio del programa, y así obtener k_{max} (columna máxima permitida). En la práctica, limitaremos k_{max} a 8 para evitar problemas con el ordenador.

Para el segundo problema, el de reducir el paso, calculamos los estimativos de longitudes de paso

$$\tilde{H}_k \equiv H_k\alpha(k, q), \quad k = 1, \dots, q-1.$$

Si alguno de ellos es “demasiado pequeño”, abandonamos el paso actual y empezamos de nuevo con \tilde{H}_k . Para ver si un \tilde{H}_k es “demasiado pequeño” comprobamos si

$$H_k\alpha(k, q+1) < H.$$

Durante el primer paso del problema, esto se comprueba para todo k . En los pasos sucesivos se comprueba solo en el intervalo

$$\max(1, q) \leq k \leq \min(k_{max}, q+1).$$

3.8. Experimentos numéricos

Se ha implementado el método de Gragg-Bulirsch-Stoer con extrapolación de Richardson, y racional, sobre una batería de problemas test tomados de [9] y un problema test propio, que pasamos a describir.

Problema test 1: Este es un problema de fácil integración,

$$y' = -y, \quad y(0) = 1, \quad x \in [0, 10],$$

cuya solución exacta es $y(x) = \exp(-x)$.

Problema test 2: Este problema modela el caso general de oscilaciones forzadas con amortiguamiento, con un grado de libertad. La ecuación diferencial que define el movimiento es:

$$y'' + \lambda y' + \omega^2 y = \epsilon \cos(\Omega x), \quad y(0) = 0, \quad y'(0) = 1, \quad x \in [0, 10].$$

Los valores de los parámetros considerados son: $\lambda = 2$, $\omega = 2$, $\Omega = 5$ y los valores $\epsilon = 0,01$ y $\epsilon = 3$.

Problema test 3: Este problema integra las ecuaciones de Euler del sólido rígido en ausencia de fuerzas externas,

$$\begin{aligned}y_1' &= y_2 y_3, & y_1(0) &= 0, \\y_2' &= -y_1 y_3, & y_2(0) &= 1, \\y_3' &= -0,51 y_1 y_2, & y_3(0) &= 1\end{aligned}$$

Problema test 4: El problema de Kepler que describe la órbita en el problema de dos cuerpos reducido

$$\begin{aligned}y_1' &= y_3, & y_1(0) &= 1 - \epsilon, \\y_2' &= y_4, & y_2(0) &= 0, \\y_3' &= -y_1/(y_1^2 + y_2^2)^{3/2}, & y_3(0) &= 0, \\y_4' &= -y_2/(y_1^2 + y_2^2)^{3/2}, & y_4(0) &= \sqrt{\frac{1 + \epsilon}{1 - \epsilon}}\end{aligned}$$

con ϵ denotando la excentricidad de la órbita. Se ha experimentado con los valores $\epsilon = 0,1$, $\epsilon = 0,5$ y $\epsilon = 0,9$.

Problema test 5: Este problema se toma de [3],

$$\begin{aligned}y'' &= y - xy' + xe^x - |x|(6 - 12x + 2x^2 - 3x^3), & x &\in [-1, 1] \\y(-1) &= e^{-1} - 2, & y'(-1) &= e^{-1} + 7.\end{aligned}$$

El interés de este problema es que la segunda derivada de la solución presenta una discontinuidad en $x = 0$, que somete el mecanismo de control de paso y orden del método a condiciones de regularidad de la solución que únicamente justifican desarrollos asintóticos de orden bajo.

En todos los problemas test se ha comparado los resultados de los códigos de extrapolación estudiados, con los resultados obtenidos con el código `ode113` y `ode45` de Matlab, usando los mismos parámetros de inicio en la integración y las mismas tolerancias para el control del error local.

Hemos considerado sistemáticamente tres tolerancias: 10^{-3} , 10^{-6} y 10^{-9} . Para cada problema test sólo presentamos la gráfica de la solución numérica obtenida con el código de extrapolación racional, ya que no hay casi diferencia gráficamente con la polinómica, con tolerancia

$\text{tol} = 1,0e^{-9}$ junto a una gráfica de la solución (tomada como exacta) obtenida con `ode113` con una tolerancia más exigente. A efectos de comparación de la eficiencia construiremos gráficas de las longitudes de paso frente a la abscisa de integración para cada tolerancia, y las presentaremos conjuntamente, lo mismo para el orden del método.

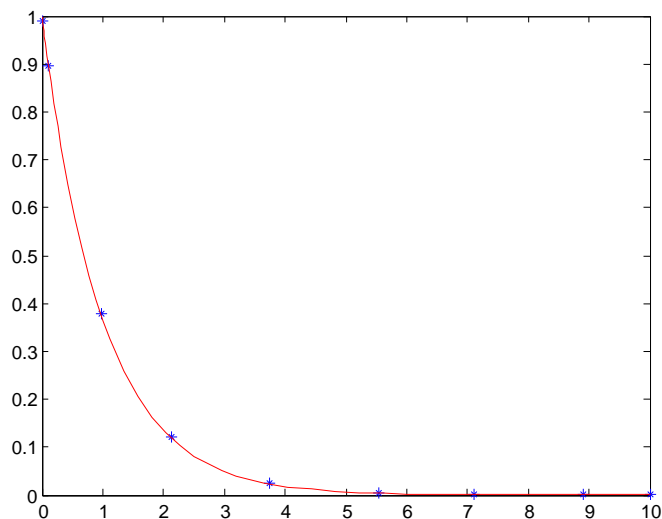


Figura 3.1: Solución numérica mediante extrapolación racional (*).

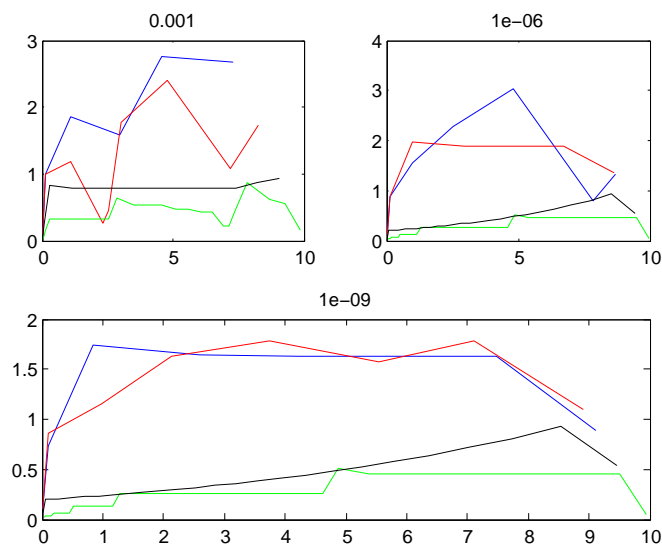


Figura 3.2: Longitudes de paso de los métodos GBS(pol.)—, GBS(rac.)—, ode45—, y ode113— (tolerancias 10^{-3} , 10^{-6} , 10^{-9}).

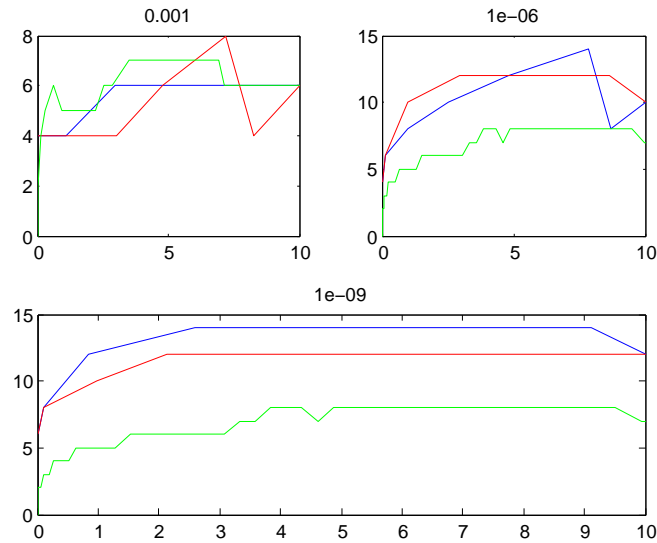


Figura 3.3: Orden local de los métodos GBS(pol.)—, GBS(rac.)—, y ode113— (tolerancias 10^{-3} , 10^{-6} , 10^{-9}).

10^{-3}	<i>Rac.</i>	<i>Pol.</i>	ode113	ode45
<i>Pasos</i>	10	7	28	15
<i>Rechazos</i>	3	1	1	0
<i>Evaluaciones</i>	32	22	58	91
10^{-6}	<i>Rac.</i>	<i>Pol.</i>	ode113	ode45
<i>Pasos</i>	8	8	50	26
<i>Rechazos</i>	4	0	3	0
<i>Evaluaciones</i>	59	36	104	157
10^{-9}	<i>Rac.</i>	<i>Pol.</i>	ode113	ode45
<i>Pasos</i>	9	9	50	26
<i>Rechazos</i>	3	0	3	0
<i>Evaluaciones</i>	65	54	104	157

En este primer problema se puede observar que tanto la longitud de paso como el orden tienden a estabilizarse en todos los métodos cuanto menor es la tolerancia a excepción de ode45 que parece que es capaz de ir aumentando la longitud de paso lentamente. También se puede observar que el orden es mucho mayor en los métodos de extrapolación que en ode113 y lo mismo pasa con la longitud de paso con respecto a ode113 y a ode45. Esto es algo que vamos a observar a lo largo de todos los problemas.

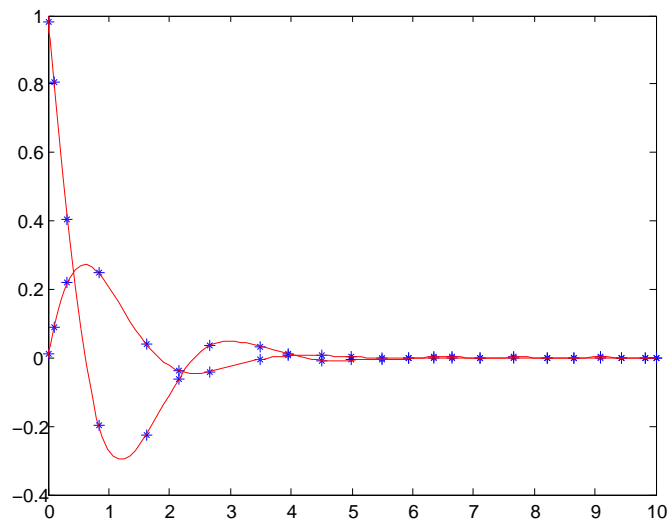


Figura 3.4: Solución numérica mediante extrapolación racional (*).

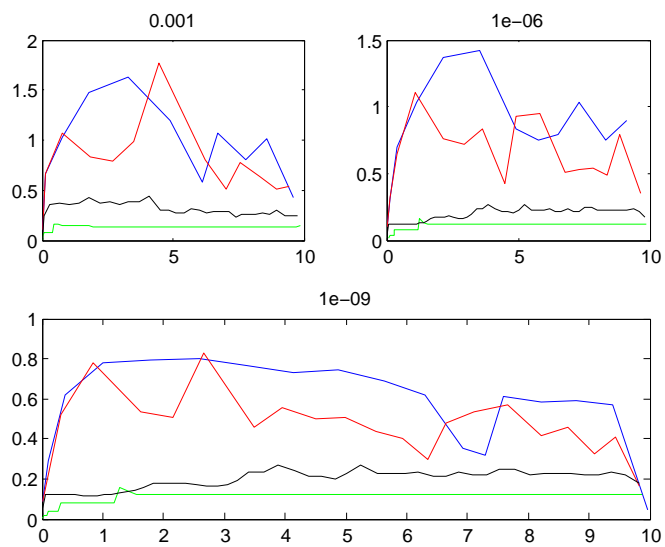


Figura 3.5: Longitudes de paso de los métodos GBS(pol.)—, GBS(rac.)—, ode45—, y ode113— (tolerancias 10^{-3} , 10^{-6} , 10^{-9}).

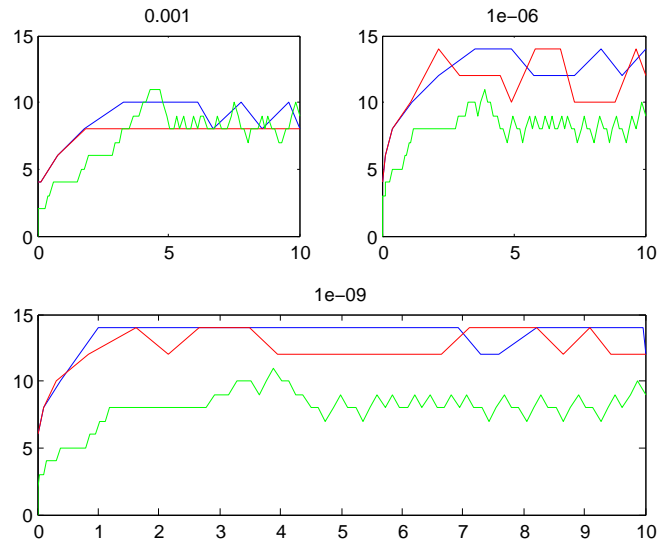


Figura 3.6: Orden local de los métodos GBS(pol.)—, GBS(rac.)—, y ode113— (tolerancias 10^{-3} , 10^{-6} , 10^{-9}).

10^{-3}	<i>Rac.</i>	<i>Pol.</i>	ode113	ode45
<i>Pasos</i>	14	12	85	34
<i>Rechazos</i>	5	0	4	4
<i>Evaluaciones</i>	79	48	175	229
10^{-6}	<i>Rac.</i>	<i>Pol.</i>	ode113	ode45
<i>Pasos</i>	17	13	100	54
<i>Rechazos</i>	4	1	4	7
<i>Evaluaciones</i>	113	78	205	367
10^{-9}	<i>Rac.</i>	<i>Pol.</i>	ode113	ode45
<i>Pasos</i>	23	19	100	54
<i>Rechazos</i>	7	1	4	7
<i>Evaluaciones</i>	183	127	205	367

Con el problema dos en el caso de $\epsilon = 0,01$, a pesar de que la gráfica de la solución tiende a estabilizarse, la longitud de paso varia ligeramente al igual que orden, aunque en intervalos muy reducidos.

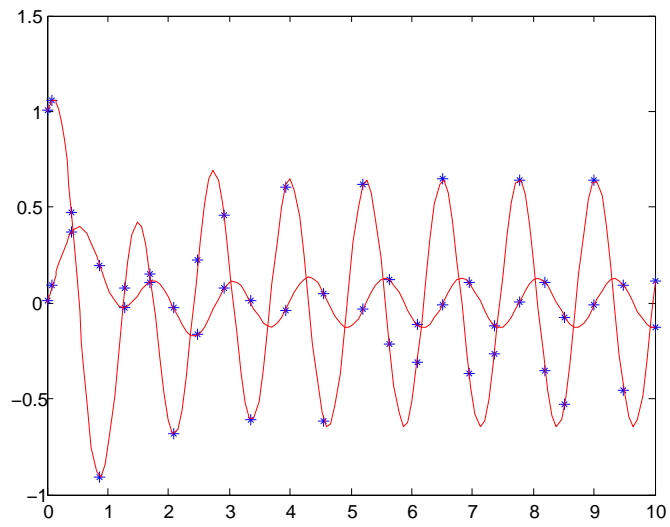


Figura 3.7: Solución numérica mediante extrapolación racional (*).

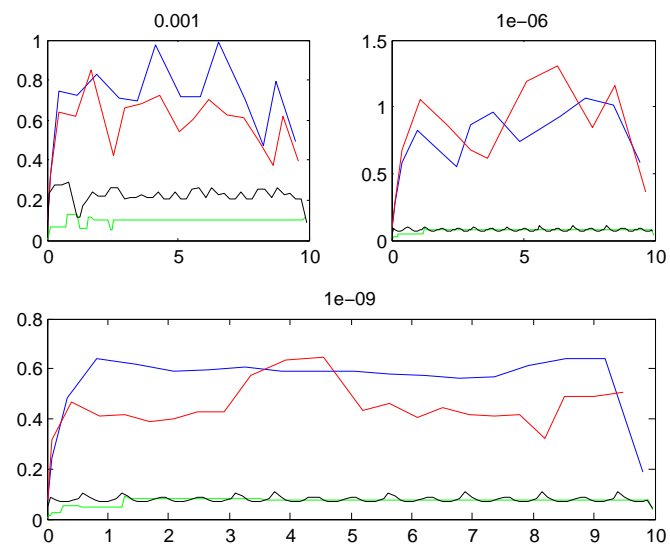


Figura 3.8: Longitudes de paso de los métodos GBS(pol.)—, GBS(rac.)—, ode45—, y ode113— (tolerancias 10^{-3} , 10^{-6} , 10^{-9}).

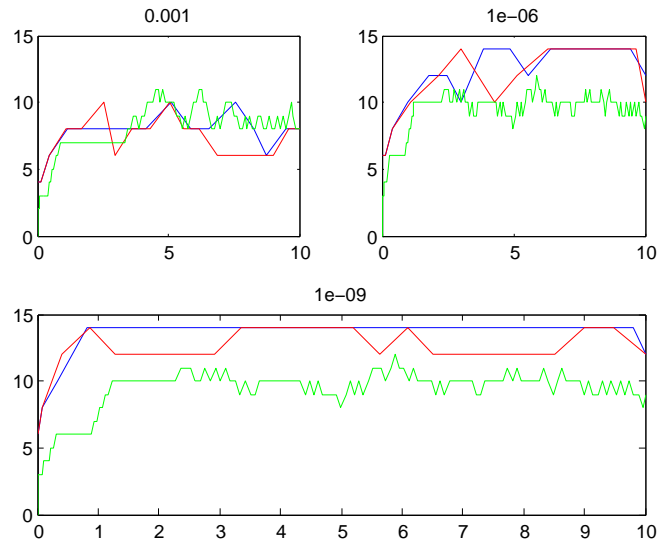


Figura 3.9: Orden local de los métodos GBS(pol.)—, GBS(rac.)—, y ode113— (tolerancias 10^{-3} , 10^{-6} , 10^{-9}).

10^{-3}	<i>Rac.</i>	<i>Pol.</i>	ode113	ode45
<i>Pasos</i>	19	16	112	47
<i>Rechazos</i>	5	4	5	9
<i>Evaluaciones</i>	87	76	230	337
10^{-6}	<i>Rac.</i>	<i>Pol.</i>	ode113	ode45
<i>Pasos</i>	14	15	155	129
<i>Rechazos</i>	3	2	4	8
<i>Evaluaciones</i>	97	98	315	823
10^{-9}	<i>Rac.</i>	<i>Pol.</i>	ode113	ode45
<i>Pasos</i>	24	20	155	129
<i>Rechazos</i>	8	0	4	7
<i>Evaluaciones</i>	196	130	315	817

En este caso, con $\epsilon = 3$, a pesar de que la solución es periódica y las soluciones son de mayor amplitud que en el caso $\epsilon = 0,01$, el comportamiento del orden y de la longitud de paso es muy similar al caso anterior, con la salvedad de que la longitud de paso en los métodos de extrapolación parece que se estabiliza más que en el otro caso y ode45 oscila con más frecuencia.

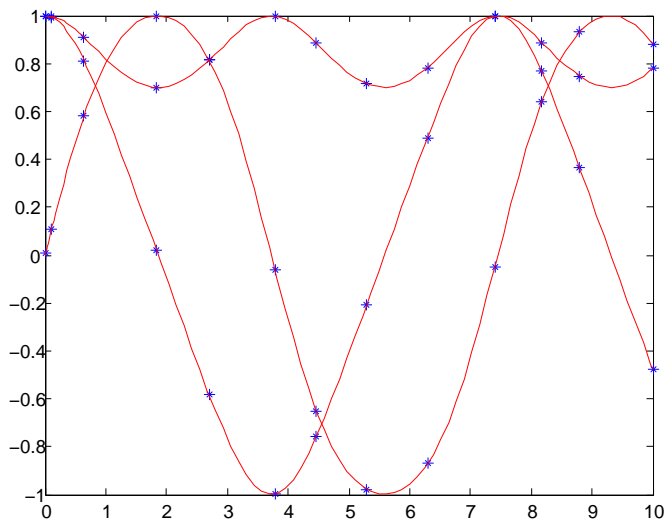


Figura 3.10: Solución numérica mediante extrapolación racional (*).

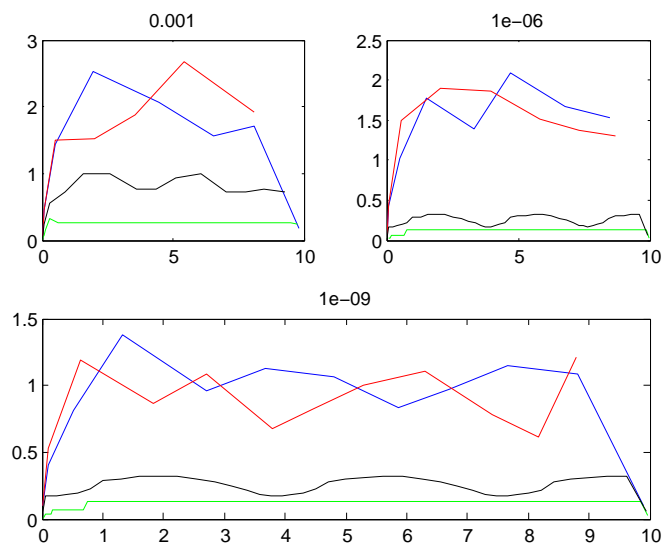


Figura 3.11: Longitudes de paso de los métodos GBS(pol.)—, GBS(rac.)—, ode45—, y ode113— (tolerancias 10^{-3} , 10^{-6} , 10^{-9}).

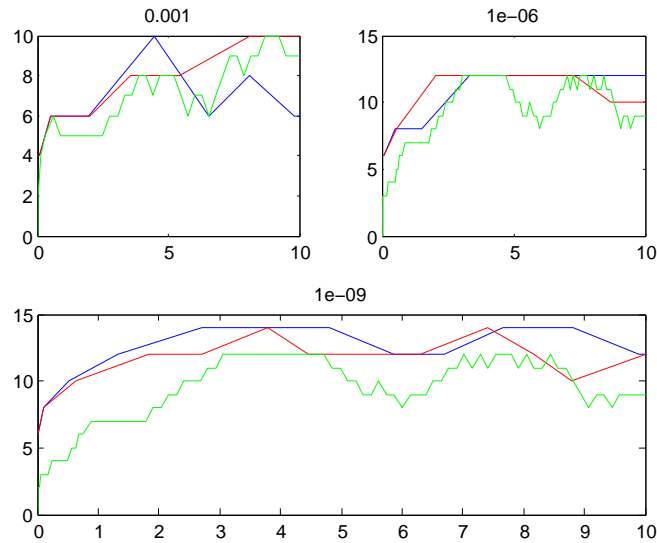


Figura 3.12: Orden local de los métodos GBS(pol.)—, GBS(rac.)—, y ode113— (tolerancias 10^{-3} , 10^{-6} , 10^{-9}).

10^{-3}	<i>Rac.</i>	<i>Pol.</i>	ode113	ode45
<i>Pasos</i>	8	9	41	15
<i>Rechazos</i>	1	1	0	2
<i>Evaluaciones</i>	32	31	83	103
10^{-6}	<i>Rac.</i>	<i>Pol.</i>	ode113	ode45
<i>Pasos</i>	9	9	95	43
<i>Rechazos</i>	2	0	3	0
<i>Evaluaciones</i>	56	44	194	259
10^{-9}	<i>Rac.</i>	<i>Pol.</i>	ode113	ode45
<i>Pasos</i>	13	13	95	43
<i>Rechazos</i>	4	2	3	0
<i>Evaluaciones</i>	98	89	194	259

En el problema tres, que integra las ecuaciones de Euler, la solución también es periódica, pero al contrario de lo que pasaba en el problema anterior, tanto la longitud de paso como el orden de los métodos oscilan bastante. Mientras que ode113 mantiene la longitud de paso casi constante.

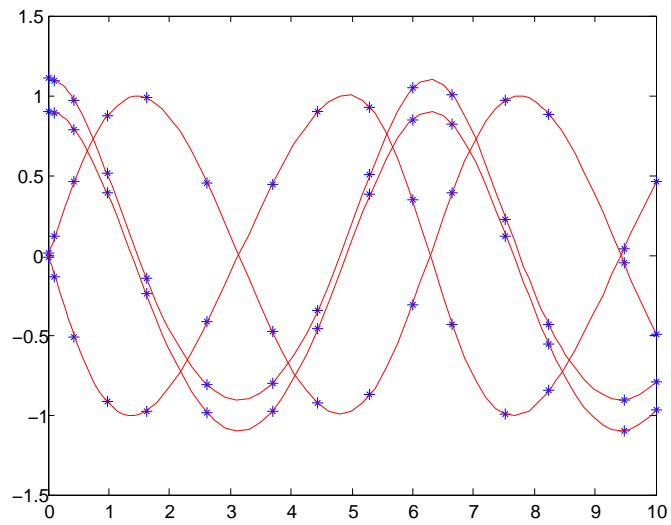


Figura 3.13: Solución numérica mediante extrapolación racional (*).

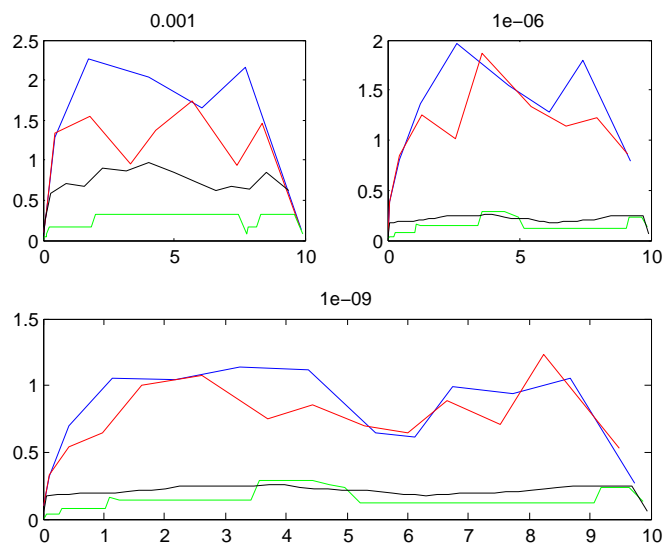


Figura 3.14: Longitudes de paso de los métodos GBS(pol.)—, GBS(rac.)—, ode45—, y ode113— (tolerancias 10^{-3} , 10^{-6} , 10^{-9}).

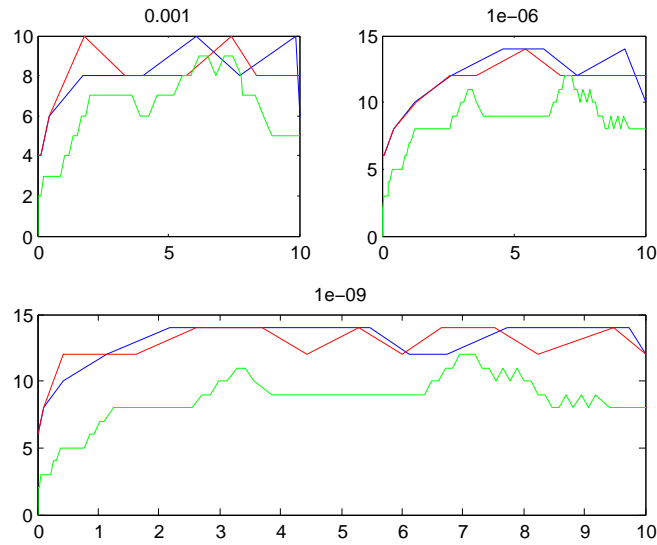


Figura 3.15: Orden local de los métodos GBS(pol.)—, GBS(rac.)—, y ode113— (tolerancias 10^{-3} , 10^{-6} , 10^{-9}).

10^{-3}	<i>Rac.</i>	<i>Pol.</i>	ode113	ode45
<i>Pasos</i>	11	9	47	16
<i>Rechazos</i>	2	1	3	0
<i>Evaluaciones</i>	50	36	98	97
10^{-6}	<i>Rac.</i>	<i>Pol.</i>	ode113	ode45
<i>Pasos</i>	11	10	86	49
<i>Rechazos</i>	1	0	5	0
<i>Evaluaciones</i>	64	53	178	295
10^{-9}	<i>Rac.</i>	<i>Pol.</i>	ode113	ode45
<i>Pasos</i>	15	14	86	49
<i>Rechazos</i>	3	1	5	0
<i>Evaluaciones</i>	112	92	178	295

En el problema 4 con excentricidad 0,1 pasa lo mismo que el caso anterior, con la excepción de que parece que en este caso la longitud de paso de ode113 varía ampliamente en ciertos momentos.

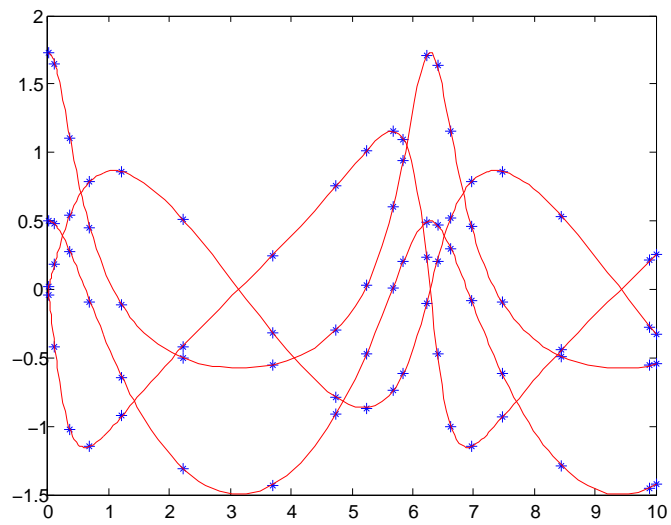


Figura 3.16: Solución numérica mediante extrapolación racional (*).

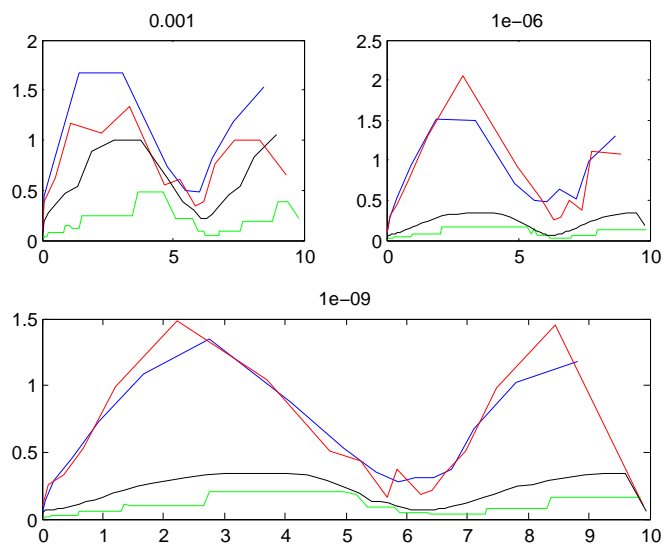


Figura 3.17: Longitudes de paso de los métodos GBS(pol.)—, GBS(rac.)—, ode45—, y ode113— (tolerancias 10^{-3} , 10^{-6} , 10^{-9}).

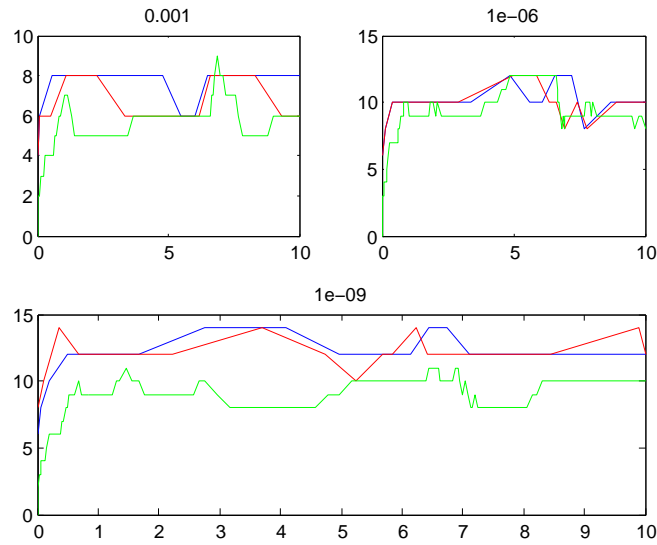


Figura 3.18: Orden local de los métodos GBS(pol.)—, GBS(rac.)—, y ode113— (tolerancias 10^{-3} , 10^{-6} , 10^{-9}).

10^{-3}	<i>Rac.</i>	<i>Pol.</i>	ode113	ode45
<i>Pasos</i>	15	12	76	21
<i>Rechazos</i>	3	3	5	4
<i>Evaluaciones</i>	57	51	158	151
10^{-6}	<i>Rac.</i>	<i>Pol.</i>	ode113	ode45
<i>Pasos</i>	15	14	140	58
<i>Rechazos</i>	3	3	7	0
<i>Evaluaciones</i>	88	83	288	349
10^{-9}	<i>Rac.</i>	<i>Pol.</i>	ode113	ode45
<i>Pasos</i>	19	18	138	59
<i>Rechazos</i>	4	4	6	1
<i>Evaluaciones</i>	140	131	283	361

Ahora como pasamos a tener excentricidad 0,5 se puede empezar a ver que hay zonas en la gráfica donde la curva es muy suave y zonas donde la curva es más cerrada y se acumulan muchos puntos. Esto causa que la longitud de paso se reduzca bastante para así poder pasar la tolerancia fijada. El orden varía, pero de una forma muy similar a problemas vistos anteriormente.

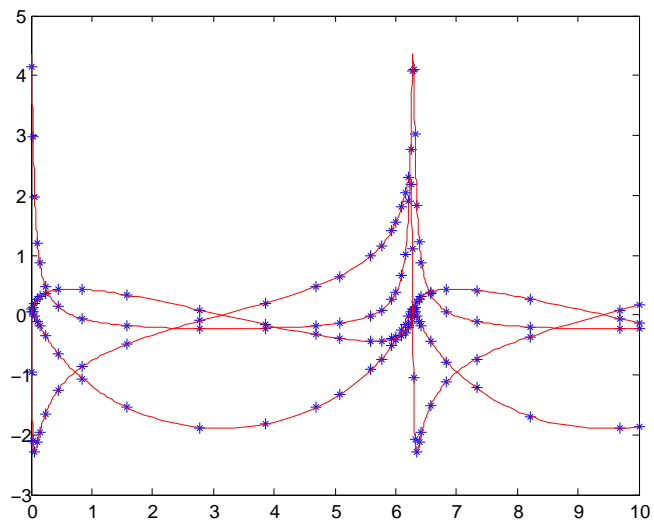


Figura 3.19: Solución numérica mediante extrapolación racional (*).

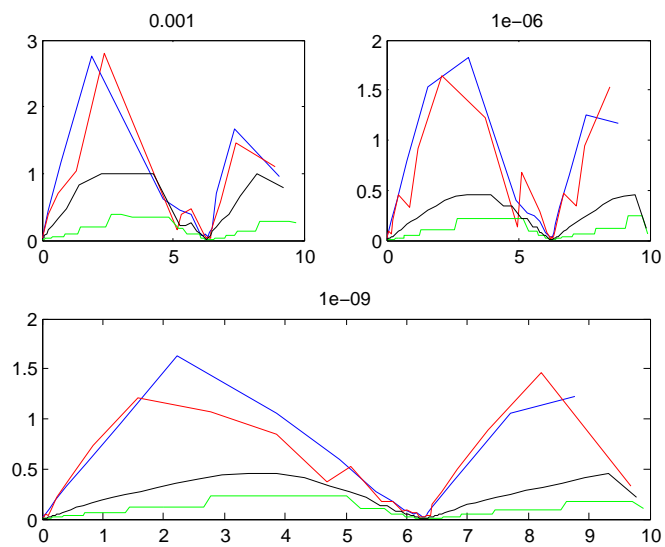


Figura 3.20: Longitudes de paso de los métodos GBS(pol.)—, GBS(rac.)—, ode45—, y ode113— (tolerancias 10^{-3} , 10^{-6} , 10^{-9}).

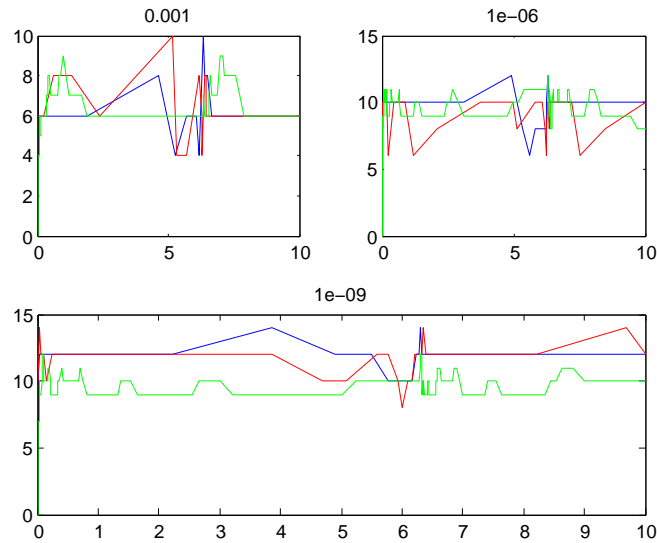


Figura 3.21: Orden local de los métodos GBS(pol.)—, GBS(rac.)—, y ode113— (tolerancias 10^{-3} , 10^{-6} , 10^{-9}).

10^{-3}	<i>Rac.</i>	<i>Pol.</i>	ode113	ode45
<i>Pasos</i>	22	19	144	37
<i>Rechazos</i>	6	4	12	7
<i>Evaluaciones</i>	89	79	301	253
10^{-6}	<i>Rac.</i>	<i>Pol.</i>	ode113	ode45
<i>Pasos</i>	30	26	254	96
<i>Rechazos</i>	8	9	13	8
<i>Evaluaciones</i>	165	164	522	625
10^{-9}	<i>Rac.</i>	<i>Pol.</i>	ode113	ode45
<i>Pasos</i>	33	30	262	101
<i>Rechazos</i>	9	9	11	6
<i>Evaluaciones</i>	242	225	536	643

Como pasamos a tener una excentricidad de 0,9, mucho mayor a la anterior, se observa, como en el problema con excentricidad 0,5, una acumulación de puntos y la consiguiente reducción de la longitud de paso en el subintervalo que corresponde a puntos de la trayectoria elíptica con mayor curvatura, mientras que el orden de los métodos casi no se ve afectado por el cambio de excentricidad.

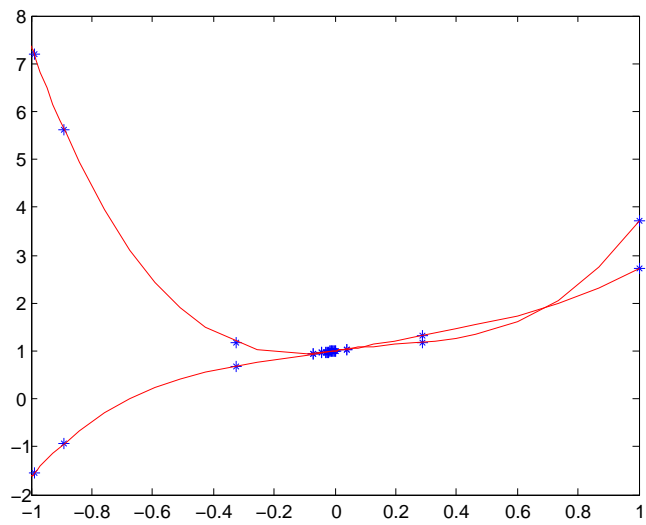


Figura 3.22: Solución numérica mediante extrapolación racional (*).

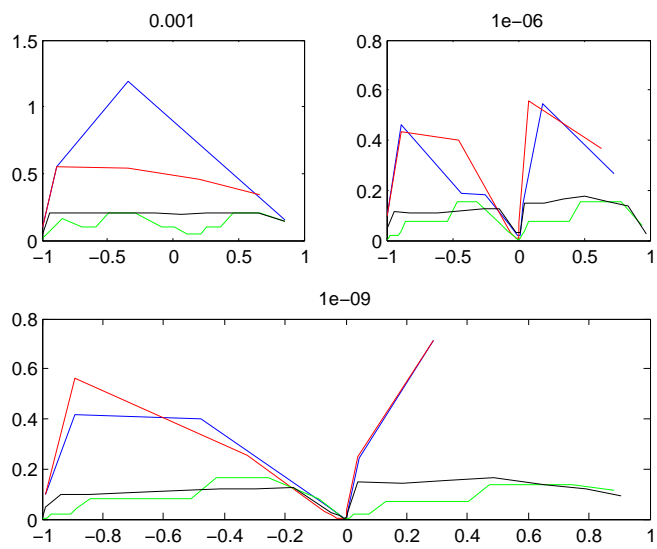


Figura 3.23: Longitudes de paso de los métodos GBS(pol.)—, GBS(rac.)—, ode45—, y ode113— (tolerancias 10^{-3} , 10^{-6} , 10^{-9}).

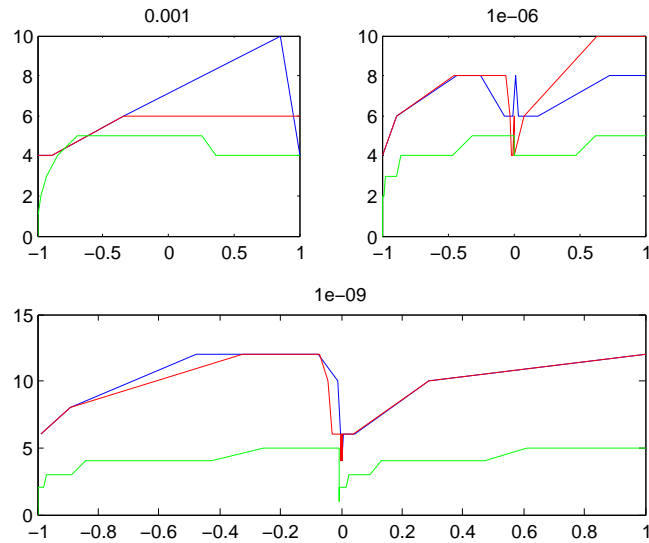


Figura 3.24: Orden local de los métodos GBS(pol.)—, GBS(rac.)—, y ode113— (tolerancias 10^{-3} , 10^{-6} , 10^{-9}).

10^{-3}	<i>Rac.</i>	<i>Pol.</i>	ode113	ode45
<i>Pasos</i>	6	5	19	12
<i>Rechazos</i>	1	0	3	0
<i>Evaluaciones</i>	20	14	42	73
10^{-6}	<i>Rac.</i>	<i>Pol.</i>	ode113	ode45
<i>Pasos</i>	12	11	45	19
<i>Rechazos</i>	5	4	17	2
<i>Evaluaciones</i>	69	68	108	127
10^{-9}	<i>Rac.</i>	<i>Pol.</i>	ode113	ode45
<i>Pasos</i>	20	13	52	23
<i>Rechazos</i>	12	5	12	5
<i>Evaluaciones</i>	140	114	117	169

Como hemos mencionado al presentar los problemas, podemos ver que en el punto donde se presenta la discontinuidad de la derivada segunda, se acumulan casi todos los puntos en la gráfica. Además el orden del método y la longitud de paso se reducen drásticamente para poder evitar el problema que genera la discontinuidad.

Como ya hemos mencionado antes al definir los algoritmos de extrapolación, la extrapolación racional no tiene ninguna ventaja frente a la polinómica, como hemos podido ver en estos ejemplos (por ejem-

plo, el número de rechazos es mayor en la extrapolación racional), y por lo tanto es evidente que en esta situación es mejor implementar la extrapolación polinómica.

También es cierto que en eficiencia respecto al número de evaluaciones, la extrapolación gana claramente tanto a `ode113` como a `ode45`, a excepción del problema 5 (ya que al ser la solución poco regular ya no nos podemos aprovechar de los órdenes altos de la extrapolación). Los métodos de extrapolación aquí mostrados tienen una desventaja clara, no podemos representar la gráfica con los valores extrapolados, ya que al generar tan pocos puntos, la gráfica no se asemejaría a la solución real.

3.9. Conclusiones

Partiendo de primeros principios hemos presentado en este trabajo todos los elementos que justifican teóricamente el buen funcionamiento de los métodos de extrapolación en la solución de problemas de valores iniciales para ecuaciones diferenciales ordinarias. La implementación de un código de paso y orden variable con métodos de extrapolación es con diferencia más simple que para los métodos lineales multipaso y ha sido estudiada en los ochenta por Deuffhard y sus colaboradores. Hemos implementado un programa Matlab que tiene en cuenta estos avances en la implementación de un código de paso y orden variable mediante métodos de extrapolación. La experimentación numérica que acabamos de presentar tiene por objeto validar el buen funcionamiento de este código al comparar los resultados con métodos alternativos, y reproducir cualitativamente experimentos numéricos encontrados en otros artículos. Naturalmente los códigos no contemplan opciones como interpolación, detección de *stiffness*, detección de ceros de la solución, etc,...

La justificación teórica ha necesitado desarrollar partiendo de primeros principios una introducción a la interpolación racional, así como la existencia de desarrollos asintóticos en el parámetro de discretización del error global para métodos explícitos de un paso. En particular, incluimos la prueba de existencia de un desarrollo asintótico en potencias pares del parámetro de discretización para la regla del punto medio modificada, nuestro método básico de integración.

Bibliografía

- [1] G. Dahlquist y Å. Björck , *Numerical Methods in Scientific Computing, Volume I*, SIAM, 2008.
- [2] P. Deuffhard, *Order and Stepsize Control in Extrapolation Methods, Numerische Mathematik 41*, Springer-Verlag, 1983, pp. 399–422.
- [3] P. Deuffhard, *Recent Progress in Extrapolation Methods for Ordinary Differential Equations*, SIAM Review 27(4), 1985, pp. 505–535.
- [4] W. B. Gragg, *On Extrapolation Algorithms for Ordinary Initial Value Problems*, SIAM J. Numer. Anal. 2, 1965, pp. 384–403.
- [5] E. Hairer y Ch. Lubich, *Asymptotic expansions of the global error of fixed-stepsize methods*, Numer. Math. 45, 1984, pp. 345-360.
- [6] E. Hairer, S. P. Nørsett y G. Wanner, *Solving Ordinary Differential Equations I, Nonstiff Problems*, Springer-Verlag, 1987.
- [7] G. Hall y J. M. Watt, *Modern Numerical Methods for Ordinary Differential Equations*, Oxford University Press, 1976.
- [8] F. B. Hildebrand, *Introduction to Numerical Analysis, Second Edition*, Dover Publications, Inc., 1987.
- [9] T. E. Hull, W. H. Enright, B. M. Fellen y A. E. Sedgwick, *Comparing Numerical Methods for Ordinary Differential Equations*, SIAM Journal on Numerical Analysis vol. 9, No 4, 1972, 603–637.
- [10] L. M. Milne-Thomson, *The Calculus of Finite Differences*, Macmillan and Co., 1933.
- [11] W. H. Press, S. A. Teukolsky, W. T. Vetterling y B. P. Flannery, *Numerical Recipes in Fortran 77: The Art of Scientific Computing*,

Second Edition, Volume 1 of Fortran Numerical Recipes, Cambridge University Press, 1992.

- [12] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery y M. Metcalf, *Numerical Recipes in Fortran 90: The Art of Parallel Scientific Computing, Second Edition, Volume 2 of Fortran Numerical Recipes*, Cambridge University Press, 1996.
- [13] J. Stoer y R. Bulirsch, *Introduction to Numerical Analysis*, Springer-Verlag, 1980.
- [14] J. Stoer y R. Bulirsch, *Introduction to Numerical Analysis, Third Edition*, Springer-Verlag, 2002.

Apéndice A

Programas de Matlab

En este apéndice incluimos los programas que hemos utilizado en Matlab que no están disponibles en su biblioteca.

odeint.m

Este código es el programa principal que recorre el intervalo del problema llamando en cada paso al método GBS.

```
1 function [xp,yp, nok, nbad,kusado,eval] = odeint(dydx
   , ystart, x1, x2, eps, h1, hmin)
2
3 format long
4 global first epsold nseq
5
6 first=true;
7 epsold=1;
8 nseq = [2, 4, 6, 8, 10, 12, 14, 16, 18];
9 eval=0;
10 kmax=200;
11 dxsav=0;
12 nvar = length(ystart);
13 MAXSTP = 10000;
14 TINY = 1.0e-30;
15 x = x1;
16 h = h1*sign(x2-x1);
17 nok = 0;
18 nbad = 0;
19 kount = 0;
```

```

20 y = ystart;
21 if (kmax > 0)
22     xsav = x - 2.0*dxsav;
23 end
24 for nstp = 1 : MAXSTP
25     yscal = abs(y) + abs(h*dydx(x,y)) + TINY;
26     if (kmax > 0)
27         if(abs(x-xsav) > abs(dxsav))
28             if (kount < (kmax-1))
29                 kount = kount+1;
30                 xp(kount) = x;
31                 yp(:,kount) = y;
32                 xsav = x;
33             end
34         end
35     end
36     if ((x+h-x2)*(x+h-x1) > 0)
37         h = x2-x;
38     end
39     [x, y, hdid, hnext, kusado(nstp), contador] =
40         BGSstep(dydx, x, y, nvar, h, eps, yscal);
41     eval=eval+contador;
42     if (hdid == h)
43         nok = nok +1;
44     else
45         nbad = nbad+1;
46     end
47     if ((x-x2)*(x2-x1) >= 0)
48         if (kmax ~= 0)
49             kount = kount+1;
50             xp(kount) = x;
51             yp(:,kount) = y;
52         end
53         return
54     end
55     if (abs(hnext) < hmin)
56         error('Longitud de paso menor que hmin.')
57     end
58     h = hnext;
59 end
error ('Demasiados pasos en la integracion para la

```

```

60     memoria.')
end

```

BGSstep.m

Se encarga de hacer cada paso del método GBS, en él podemos elegir si queremos extrapolación polinómica o extrapolación racional comentando las tres líneas correspondientes.

```

1 function [x, y, hdid, hnext, k, contador] = BGSstep(dydx
2     , x, y, nv, htry, eps, yscal)
3 global a alf epsold first kmax kopt xnew nseq
4
5 hnext=htry;
6 KMAXX = 8;
7 SAFE1 = 0.25;
8 SAFE2 = 0.7;
9 REDMAX = 1.0e-5;
10 REDMIN = 0.7;
11 TINY = 1.0e-30;
12 SCALMX = 0.1;
13 if (eps ~= epsold)
14     hnext = -1.e29;
15     xnew = -1.e29;
16     eps1 = SAFE1 * eps;
17     a(1) = nseq(1) + 1;
18     for k = 1:KMAXX
19         a(k+1) = a(k) + nseq(k+1);
20     end
21     for iq = 2 : KMAXX
22         for k = 1 : iq-1
23             alf(k,iq) = eps1^((a(k+1) - a(iq+1))/((a(
24                 iq+1)-a(1)+1.0)*(2*k+1)));
25         end
26     end
27     epsold = eps;
28     for kopt = 2 : KMAXX-1
29         if (a(kopt+1) > a(kopt)*alf(kopt-1,kopt))
30             break
31         end

```

```

31     end
32     kmax = kopt;
33 end
34 h = htry;
35 ysav = y;
36 if ((h ~ = hnext) || (x ~ = xnew))
37     first = true;
38     kopt = kmax;
39 end
40 reduct = false;
41 contador=0;
42 while true
43     T=zeros(nv,KMAXX,KMAXX);
44     %T=zeros(nv,KMAXX+1,KMAXX);
45     for k = 1 : kmax
46         xnew = x + h;
47         if (xnew == x)
48             error('Tamano de paso demasiado pequeno.')
49             ;
50         end
51         [~, yseq] = mmid(dydx, x, ysav, h, nseq(k));
52         contador=contador+1;
53         T(:,1,k)=yseq;
54         %T(:,2,k)=yseq;
55         if (k ~ = 1)
56             [y, yerr,T] = pzextr(k, T,nseq);
57             %[y, yerr,T] = rzextr(k+1, T,nseq);
58             errmax=TINY;
59             for ele=1:nv
60                 errmax = max(errmax,abs(yerr(ele)/
61                     yscal(ele)));
62             end
63             errmax =errmax/eps;
64             km = k - 1;
65             err(km) = (errmax/SAFE1)^(1.0/(2*km+1));
66         end
67         if ( (k ~ = 1) && ((k >= (kopt-1)) || first))
68             if (errmax < 1.0)
69                 accepted = true;
70                 break
71             end
72         end
73     end
74 end

```

```

70         if ( (k == kmax) || (k==kopt+1) )
71             red = SAFE2/err(km);
72             accepted = false;
73             break
74         elseif ( k == kopt )
75             if ( alf(kopt-1,kopt) < err(km))
76                 red = 1.0/err(km);
77                 accepted = false;
78                 break
79             end
80         elseif ( kopt == kmax )
81             if (alf(km,kmax-1) < err(km) )
82                 red = alf(km, kmax-1) * SAFE2 /err
                    (km);
83                 accepted = false;
84                 break
85             end
86         elseif (alf(km, kopt) < err(km))
87             red = alf(km, kopt-1)/err(km);
88             accepted = false;
89             break
90         end
91     end
92 end
93 if (accepted)
94     break
95 end
96 red = min (red, REDMIN);
97 red = max (red, REDMAX);
98 h = h * red;
99 reduct = true;
100 end
101 x = xnew;
102 hdid = h;
103 first = false;
104 wrkmin = 1.e35;
105 for kk = 1 : km
106     fact = max(err(kk), SCALMX);
107     work = fact*a(kk+1);
108     if (work < wrkmin)
109         scale = fact;

```

```

110         wrkmin = work;
111         kopt = kk + 1;
112     end
113 end
114 hnext = h/scale;
115 if ( (kopt >= k) && (kopt ~ = kmax) && (~reduct))
116     fact = max(scale/alf(kopt-1,kopt), SCALMX);
117     if (a(kopt+1)*fact <= wrkmin)
118         hnext = h/fact;
119         kopt = kopt+1;
120     end
121 end
122 end

```

mmid.m

Implementa la regla modificada del punto medio.

```

1 function [ x, yout ] = mmid(dydx, x0, y0, htot, nsteps
2 )
3 %
4 nv = size(y0,1);
5 %
6 ym = zeros(nv,1); yn = zeros(nv,1); yout = zeros(nv
7 ,1); swap = zeros(nv,1);
8 %
9 h = htot/nsteps;
10 ym = y0;
11 x = x0;
12 %
13 yout = feval(dydx, x, ym);
14 yn = y0 + h*yout;
15 %yn = y0 + h*yp0;
16 x = x + h;
17 yout = feval(dydx, x, yn);
18 h2 = 2.0*h;
19 for n=2 : nsteps
20     swap = ym + h2*yout;
21     ym = yn;
22     yn = swap;
23     x = x+h;

```

```

22     yout = feval(dydx, x, yn);
23 end
24
25 yout = 0.5*(ym + yn + h*yout);
26 return
27 end

```

pzextr.m

Genera una fila de la tabla de la extrapolación polinómica.

```

1 function [y, yerr,T] = pzextr(k, T,nseq)
2
3 for i=1:k-1
4     num = T(:,i,k) - T(:,i,k-1);
5     den = (nseq(k)/(nseq(k-i)))^2 - 1;
6     T(:,i+1,k) = T(:,i,k) + num/den;
7 end
8 y=T(:,k,k);
9 yerr=T(:,k-1,k)-T(:,k,k);
10 end

```

rzextr.m

Genera una fila de la tabla de la extrapolación racional.

```

1 function [y, yerr,T] = rzextr(k, T,nseq)
2
3 for i=1:k-2
4     num = T(:,i+1,k-1) - T(:,i+1,k-2);
5     a=1-(T(:,i+1,k-1) - T(:,i+1,k-2))./(T(:,i+1,k-1) -
6         T(:,i,k-2));
7     den = ((nseq(k-1)/(nseq(k-i-1)))^2)*a - 1;
8     T(:,i+2,k-1) = T(:,i+1,k-1) + num./den;
9 end
10 y=T(:,k,k-1);
11 yerr=T(:,k-1,k-1)-T(:,k,k-1);
12 end

```