



Universidad de Valladolid

FACULTAD DE CIENCIAS

TRABAJO DE FIN DE GRADO

Grado en Matemáticas

Métodos Runge-Kutta particionados y métodos de composición: teoría del orden y aplicaciones.

Autor: Alicia Martínez Miguel

Tutor: Luis M^a Abia Llera

Año: 2022/2023

Prefacio

La introducción por Butcher a finales de los años 60 de la teoría de árboles en el estudio de las condiciones de orden de un método Runge-Kutta general puso las bases para el posterior desarrollo de la teoría de series formales indexadas por este tipo de objetos en el análisis del orden de otros métodos numéricos. El presente proyecto considera a partir del concepto de B-serie la aplicación de estas técnicas en el análisis de la teoría del orden de los métodos Runge-Kutta particionados, Runge-Kutta particionados simplécticos y los métodos basados en composición con pasos de distintas longitudes de un método básico.

El capítulo primero parte de una revisión de la teoría de Butcher en el estudio del orden de un método Runge-Kutta general, que se estudia en la asignatura optativa “Métodos numéricos de ecuaciones diferenciales” del Grado de Matemáticas, para motivar e introducir el concepto de B-serie e ilustrar como este concepto facilita la caracterización de un método Runge-Kutta de orden p . El estudio de las condiciones de orden de métodos Runge-Kutta particionados generales introduce los árboles bicolor con raíz y la primera generalización del concepto de B-serie, que pasa a denominarse P-serie, para la caracterización del orden de estos métodos.

El capítulo segundo se concentra en las modificaciones que hay que introducir en la teoría de árboles para caracterizar el orden de métodos Runge-Kutta simplécticos y Runge-Kutta particionados simplécticos cuando se integra un sistema Hamiltoniano. Se motiva la propiedad de un sistema dinámico simpléctico como aquel cuya aplicación flujo preserva la 2-forma $dp \wedge dq$, y se pide esta misma propiedad a las aplicaciones flujo asociadas a los métodos numéricos objeto de estudio. El resultado más importante de la teoría es que un método Runge-Kutta general es simpléctico si y solo si cierta matriz M formada a partir de sus coeficientes es la matriz nula. En particular, esto permite reducir el número de condiciones de orden que se deben satisfacer para alcanzar orden p . El capítulo se cierra considerando la teoría de los métodos Runge-Kutta particionados e introduce los métodos de Ruth que luego se utilizan en el último capítulo para simular numéricamente el problema de Kepler.

El capítulo tercero considera la generalización necesaria de las B-series para abordar el estudio del orden de métodos de composición. El capítulo introduce el concepto de método adjunto de un método general porque la composición de un método con su adjunto proporciona métodos simétricos que permiten elevar el orden del método básico en una unidad. Los métodos de composición generalizan éstos al considerar composiciones múltiples del método básico y su adjunto con pasos que son fracción del paso básico. El estudio del orden de estos métodos lleva a introducir una nueva clase de árboles con raíz, en los que los nodos sucesivos tienen asociadas etiquetas dadas por números enteros, y las correspondientes series formales indexadas por estos árboles. Son los llamados árboles con raíz T_∞ y las llamas B_∞ -series. La teoría del orden que se deriva con estos objetos, desarrollada en los noventa por A. Murua y J. M. Sanz-Serna es, con diferencia, la contribución más técnica de esta memoria.

Finalmente, el último capítulo presenta las ventajas computacionales que los métodos simplécticos, en particular dos de los métodos de Ruth, tienen en la integración a

largos tiempos de sistemas Hamiltonianos. En particular, se ilustra la bondad de estas propiedades con el problema de Kepler como test.

Índice general

1. El concepto de B-serie	1
1.1. Métodos Runge-Kutta y árboles con raíz	1
1.1.1. Condiciones de orden de los métodos Runge-Kutta	5
1.2. B-series	9
1.2.1. Aplicación de las B-series sobre las condiciones de orden de los métodos Runge-Kutta	11
1.3. Árboles bicolor y P-series	12
1.4. Métodos Runge-Kutta particionados	14
1.4.1. Ejemplos de métodos Runge-Kutta particionados	16
1.4.2. Condiciones de orden de métodos Runge-Kutta particionados mediante P-series	18
2. Métodos simplécticos	21
2.1. Transformaciones y métodos simplécticos	21
2.2. Métodos Runge-Kutta simplécticos	24
2.2.1. Condiciones de orden	27
2.3. Métodos Runge-Kutta particionados simplécticos	32
2.3.1. Condiciones de orden	36
2.3.2. Ejemplos de métodos Runge-Kutta particionados simplécticos	37
3. Métodos de composición	41
3.1. Definiciones	41
3.1.1. Composición de métodos Runge-Kutta y de B-series	42
3.2. Método adjunto	44
3.3. Composición múltiple	47
3.4. Métodos de composición de orden alto	49
3.4.1. Extensión del concepto de B-serie	50
3.4.2. Composición con el adjunto: condiciones de orden	51
3.4.3. Reducción de las condiciones de orden	55
4. Experimentos numéricos	65
4.1. Resultados numéricos	65
4.1.1. Resultados obtenidos	67

Bibliografía	73
Anexos	74
A. Programas de Matlab	75

Capítulo 1

El concepto de B-serie

Comenzaremos este capítulo con la exposición de la teoría del orden de los métodos Runge-Kutta generales para la resolución numérica del problema autónomo de valores iniciales :

$$\begin{cases} \dot{y} &= f(y), & t \in [t_0, t_F], \\ y(t_0) &= y_0, \end{cases} \quad (1.1)$$

con $f : \Omega \subseteq \mathbb{R}^d \rightarrow \mathbb{R}^d$ e $y : [t_0, t_F] \rightarrow \Omega$ suficientemente diferenciables, tal y como se estudia en la asignatura de “Solución numérica de ecuaciones diferenciales” del Grado en Matemáticas.

1.1. Métodos Runge-Kutta y árboles con raíz

Un método Runge-Kutta de s etapas viene dado por las siguientes ecuaciones:

$$\begin{cases} k_i &= f\left(t_0 + c_i h, y_0 + h \sum_{j=1}^s a_{ij} k_j\right), & i = 1, \dots, s. \\ y_1 &= y_0 + h \sum_{i=1}^s b_i k_i, \end{cases} \quad (1.2)$$

siendo a_{ij} , b_i y c_i los coeficientes del método, que verifican $\sum_j a_{ij} = c_j$ y $\sum_i b_i = 1$.

Figura 1.1: Tabla de Butcher

$$\begin{array}{c|ccc} c_1 & a_{11} & \cdots & a_{1s} \\ & \vdots & & \vdots \\ & \vdots & & \vdots \\ c_s & a_{s1} & \cdots & a_{ss} \\ \hline & b_1 & \cdots & b_s \end{array} = \begin{array}{c|c} \mathbf{c} & A \\ \hline & \mathbf{b}^t \end{array}$$

Los coeficientes del método Runge-Kutta anterior de s etapas se recogen en esta tabla, denominada tabla de Butcher.

Se debe a Butcher el desarrollo de una teoría que permite describir las condiciones de orden de los métodos Runge-Kutta de forma directa asociadas a ciertos objetos llamados árboles con raíz.

La expresión general de las derivadas de orden alto $y^{(q)}(t)$ de la solución exacta de (1.1), cuando f es suficientemente diferenciable, introduce de manera natural las diferenciales elementales de una función y su vinculación a los llamados árboles con raíz (grafos conexos sin ciclos con un nodo destacado que representa la raíz del grafo). En particular, esto permite asociar una serie en potencias de h para representar el valor exacto de $y(t+h)$. Estos mismos objetos forman parte del desarrollo en potencias de h de la solución numérica y_1 dada por la fórmula (1.2) del método Runge-Kutta. Este es el punto de partida para introducir el concepto de B-serie y sus propiedades.

Definición 1.1. El conjunto de los árboles con raíz, denotado RT , se construye de manera recursiva:

1. El grafo τ con un único vértice pertenece al conjunto RT . Este vértice recibe el nombre de raíz del árbol τ .
2. Si rt_1, \dots, rt_m son elementos de RT , el grafo que se obtiene al unir las raíces de cada uno de estos árboles con un nuevo vértice mediante aristas también está en RT . Este nuevo árbol se denota $rt = [rt_1, \dots, rt_m]$ y su raíz es ese nuevo vértice. Además, se dice que rt_1, \dots, rt_m son subárboles de rt .

Observación 1.2. No vamos a distinguir el orden de la disposición de los subárboles en la notación introducida, es decir, $[rt_1, \dots, rt_m] = [rt_{\sigma(1)}, \dots, rt_{\sigma(m)}]$, donde σ es cualquier reordenación de los índices.

A continuación, veremos conceptos básicos relacionados con los árboles con raíz que nos permitirán definir B-series más adelante.

Definición 1.3. El orden de un árbol $rt \in RT$ es el número de vértices de dicho árbol y se denota $\rho(rt)$. Además, denotaremos RT_q al conjunto de árboles con raíz de orden q . Si el orden del árbol es mayor que 1, se denominan hojas del árbol a aquellos nodos del grafo de los cuales no sale ninguna rama hacia arriba. En otros términos, es un vértice distinto de la raíz que solo tiene otro vértice conectado a él. De manera recursiva:

$$\rho(\tau) = 1, \text{ siendo } \tau \in RT \text{ el árbol con un solo nodo,}$$

$$\rho(rt) = 1 + \sum_{j=1}^m \rho(rt_j), \forall rt = [rt_1, \dots, rt_m] \in RT.$$

Definición 1.4. Para un árbol $rt \in RT$, asignamos a cada vértice del árbol el número de vértices que quedan por encima del mismo más uno. La función densidad de rt , denotada $\gamma(rt)$, será el producto de todos estos valores asignados a sus vértices. De manera recursiva:

$$\begin{aligned}\gamma(\tau) &= 1, \text{ siendo } \tau \in RT \text{ el árbol con un solo nodo,} \\ \gamma(rt) &= \rho(rt)\gamma(rt_1) \cdot \dots \cdot \gamma(rt_m), \forall rt = [rt_1, \dots, rt_m] \in RT.\end{aligned}$$

Definición 1.5. Consideramos un conjunto de índices ordenados $A = \{i < j < k < l < \dots\}$. Vamos a denotar por A_q al subconjunto de A con los primeros $q \geq 1$ índices del conjunto A . Un árbol con raíz de orden q etiquetado monótonamente es una aplicación

$$rt : A_q \setminus \{i\} \longrightarrow A_q$$

que verifica $rt(z) < z$ para todo $z \in A_q \setminus \{i\}$. Denotamos por $\alpha(rt)$ al número de etiquetados monótonos posibles del árbol $rt \in RT$.

Sea $rt \in RT$ un árbol con raíz. Vamos a construir el peso elemental $\Phi(rt)$ de $rt \in RT$, que será un polinomio en los coeficientes del método Runge-Kutta (1.2). Asignamos a cada nodo del árbol una etiqueta $i < j < k < \dots$ de modo que sea un etiquetado monótono. Escribimos entonces un producto de factores en el que el primer factor es b_i . Para cada rama del árbol escribimos un factor a_{jk} , siendo j y k las etiquetas de los nodos inicio y final de la rama, suponiendo las ramas dirigidas en dirección ascendente desde la raíz. Tras escribir estos factores, hacemos su suma sobre todas las posibles elecciones de las etiquetas:

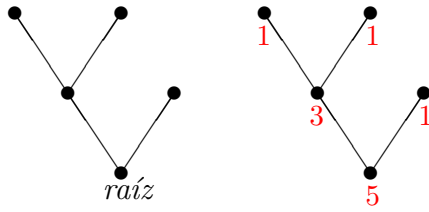
$$\Phi(rt) = \sum_{i,j,k,\dots} b_i \cdot a_{ij} \cdot a_{jk} \cdot \dots$$

Además, para las ramas que terminan en una hoja, el factor a_{jk} se transformará en $c_j := \sum_k a_{jk}$, siendo j y k los nodos inicial y final de la rama y siendo k un nodo terminal (una hoja del árbol). De hecho, se suele asignar directamente el factor c_j a la rama con nodo inicial j y con nodo final una hoja del árbol.

Definición 1.6. El peso elemental de un árbol con raíz también se puede construir de manera recursiva:

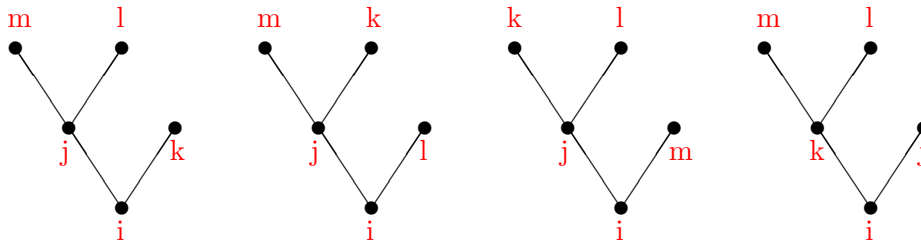
$$\begin{aligned}\Phi_i(\tau) &= \sum_j a_{ij} = c_i, \\ \Phi_i([rt_1, \dots, rt_m]) &= \sum_j a_{i,j} \Phi_j(rt_1) \cdot \dots \cdot \Phi_j(rt_m), \\ \Phi([rt_1, \dots, rt_m]) &= \sum_i b_i \Phi_i(rt_1) \cdot \dots \cdot \Phi_i(rt_m).\end{aligned}$$

Veámoslo con un ejemplo:

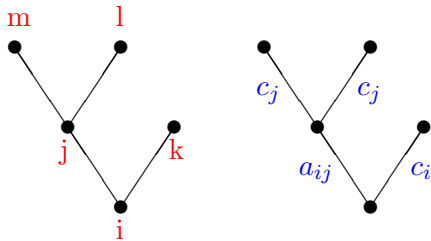


Este árbol con raíz es $rt = [\tau[\tau, \tau]]$. Tiene orden $\rho(rt) = 5$. Los valores en rojo asignados a cada vértice permiten calcular la densidad del árbol:
 $\gamma(rt) = 3 \cdot 5 = 15$.

Como tiene $\rho(rt) = 5$ nodos, consideramos los cinco índices ordenados $i < j < k < l < m$. Veamos los posibles etiquetados monótonos del árbol y elijamos uno de ellos para construir el peso elemental del árbol. Con estos índices encontramos cuatro posibles etiquetados monótonos del árbol:



Consideramos ahora el primero de los cuatro etiquetados monótonos anteriores. Construyamos el peso elemental del árbol utilizando dicho etiquetado y asignando factores como corresponda:



Multiplicando todos los coeficientes se obtiene:

$$\Phi(rt) = \sum_{i,j} b_i \cdot a_{ij} \cdot c_i \cdot c_j^2$$

Obtenemos el mismo resultado usando las fórmulas dadas en la definición:

$$\left. \begin{aligned} \Phi(rt) &= \sum b_i \cdot \Phi_i(\tau) \cdot \Phi_i([\tau, \tau]) \\ \Phi_i(\tau) &= \sum_j a_{ij} = c_i \\ \Phi_i([\tau, \tau]) &= \sum_j a_{ij} \Phi_j^2(\tau) = \sum_j a_{ij} \cdot c_j^2 \end{aligned} \right\} \implies \Phi(rt) = \sum_{i,j} b_i \cdot a_{ij} \cdot c_i \cdot c_j^2$$

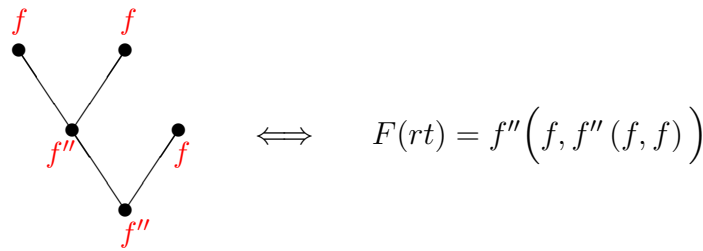
Definición 1.7. La diferencial elemental de un árbol $rt \in RT$ es una aplicación $F(rt) : \Omega \subseteq \mathbb{R}^d \longrightarrow \mathbb{R}^d$ definida recursivamente de la siguiente manera:

- $F(\tau)(y) = f(y)$.
- $F(rt)(y) = f^{(m)}(y)\left(F(rt_1)(y), \dots, F(rt_m)(y)\right)$, para $rt = [rt_1, \dots, rt_m]$, donde $f^{(m)}$ es la diferencial m -ésima de la función vectorial f actuando como aplicación m -lineal sobre la m -upla de vectores $\left(F(rt_1)(y), \dots, F(rt_m)(y)\right)$. En particular, la componente i -ésima $F^i(rt)$ de la diferencial elemental $F(rt)$ asociada al árbol $rt = [rt_1, \dots, rt_m]$ viene dada por la siguiente expresión:

$$F^i(rt) = \sum_{j_1, \dots, j_m} \frac{\partial f^i}{\partial y_{j_1} \dots \partial y_{j_m}} F^{j_1}(rt_1) \cdot \dots \cdot F^{j_m}(rt_m).$$

Se pueden construir árboles a partir de sus diferenciales elementales y viceversa.

Con el ejemplo del árbol anterior:



1.1.1. Condiciones de orden de los métodos Runge-Kutta

Presentamos ahora los resultados necesarios para determinar las condiciones de orden de los métodos Runge-Kutta de manera general. En ellos aparecen series formales, lo que más adelante denominaremos B-series, que facilitan las pruebas de los teoremas.

Con las definiciones anteriores se tiene el siguiente resultado:

Teorema 1.8. *La derivada q -ésima de la solución exacta del problema autónomo (1.2) verifica:*

$$y^{(q)}(t_0) = \sum_{rt \in LT_q} F(rt)(y_0) = \sum_{rt \in RT_q} \alpha(rt) F(rt)(y_0), \tag{1.3}$$

donde LT_q denota el conjunto de los árboles de orden q con raíz etiquetados monótonamente.

Demostración. ([3],[4]).

Vamos a calcular las derivadas de la solución exacta en t_0 .

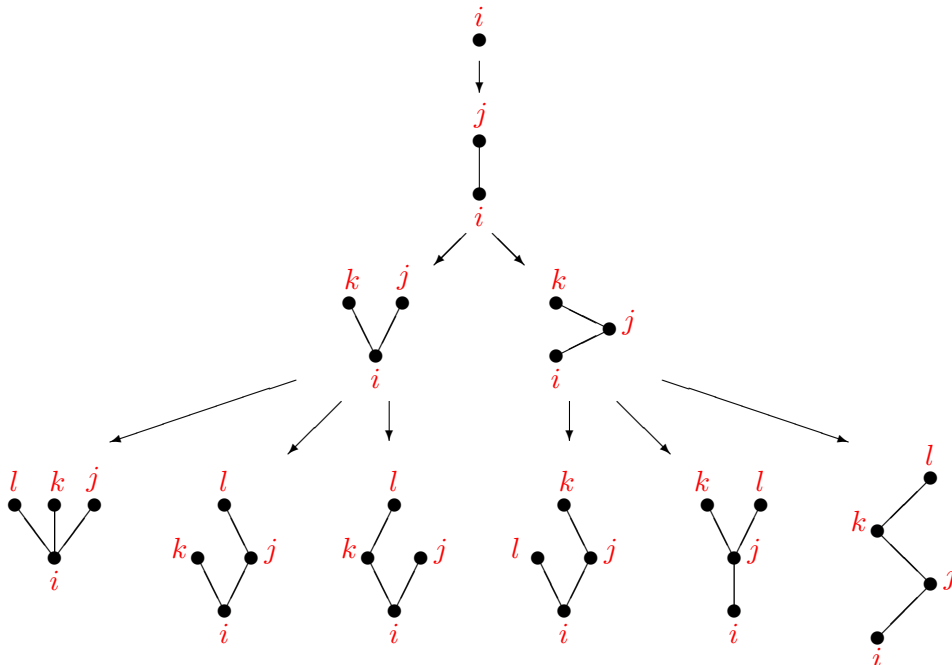
De (1.2), se tiene: $y^{(q)} = (f(y))^{q-1}$. Derivamos de manera consecutiva:

$$\begin{aligned} \dot{y} &= f(y) \\ \ddot{y} &= f'(y) \dot{y} \\ y^{(3)} &= f''(y)(\dot{y}, \dot{y}) + f'(y) \ddot{y} \\ y^{(4)} &= (f'''(y)(\dot{y}, \dot{y}, \dot{y}) + 2f''(y)(\ddot{y}, \dot{y})) + (f''(y)(\ddot{y}, \dot{y}) + f'(y) y^{(3)}) = \\ &= f'''(y)(\dot{y}, \dot{y}, \dot{y}) + 3f''(y)(\ddot{y}, \dot{y}) + f'(y) y^{(3)} \\ &\vdots \end{aligned}$$

Sustituyendo y quitando los argumentos, obtenemos:

$$\begin{aligned} \dot{y} &= f \\ \ddot{y} &= f' f \\ y^{(3)} &= f''(f, f) + f' f' f \\ y^{(4)} &= f'''(f, f, f) + 3f''(f', f) + f' f''(f, f) + f' f' f' f \\ &\vdots \end{aligned}$$

Observamos que, geoméricamente, esta diferenciación consecutiva utilizando la Regla de Leibniz e insertando las calculadas en las siguientes consiste en añadir una nueva rama con la siguiente etiqueta correspondiente en cada uno de los vértices:



Esto asegura que todos los árboles con raíz de orden q etiquetados monótonamente aparecen en la derivada q -ésima de la solución y , además, sin repetición. De este modo

queda probada la primera igualdad.

Como las etiquetas no interfieren en las diferenciales elementales del árbol, los $\alpha(t)$ etiquetados monótonos de un árbol t tienen la misma diferencial elemental. Agrupándolos se obtiene la segunda de las igualdades. \square

Hemos obtenido una expresión de las derivadas de la solución exacta como series de potencias de h . Para estudiar el orden del método podemos comparar dicha serie en potencias de h con la derivada correspondiente de la solución numérica. Por ello nos interesa obtener una expresión similar a la anterior para la solución numérica.

Teorema 1.9. *La derivada q -ésima de la solución numérica del método Runge-Kutta (1.1) verifica:*

$$y_1^{(q)}|_{h=0} = \sum_{rt \in RT_q} \gamma(rt) \Phi(rt) \alpha(rt) F(rt)(y_0). \quad (1.4)$$

Demostración. ([3],[4]).

Tomamos $g_i = hf(u_i)$ y reescribimos el método Runge-Kutta (1.1):

$$\begin{cases} g_i &= hf(u_i) \\ u_i &= y_0 + \sum_j a_{ij} g_j \\ y_1 &= y_0 + \sum_i b_i g_i \end{cases}, \quad (1.5)$$

donde g_i , u_i e y_1 son funciones de h . Aplicando la regla de Leibniz a $g_i = hf(u_i)$, se tiene:

$$g_i^{(q)} = h(f(u_i))^{(q)} + q(f(u_i))^{(q-1)}.$$

En particular, para $h = 0$, se tiene:

$$g_i^{(q)}|_{h=0} = q(f(u_i))^{(q-1)}|_{h=0}.$$

Derivando de manera consecutiva, se tiene:

$$\begin{aligned} \dot{g}_i &= 1 \cdot f(y_0) \\ \ddot{g}_i &= 2f'(y_0)\dot{u}_i \\ g_i^{(3)} &= 3\left(f''(y_0)(\dot{u}_i, \dot{u}_i) + f'(y_0)\ddot{u}_i\right) \\ g_i^{(4)} &= 4\left(f'''(y_0)(\dot{u}_i, \dot{u}_i, \dot{u}_i) + 3f''(y_0)(\ddot{u}_i, \dot{u}_i) + f'(y_0)u_i^{(3)}\right) \\ &\vdots \end{aligned}$$

Se están considerando las derivadas en $h = 0$. Observamos:

$$u_i = y_0 + \sum_j a_{ij} g_j \Rightarrow u_i^{(q)} = \sum_j a_{ij} g_j^{(q)}.$$

Por tanto, insertando cada derivada donde corresponda, se tiene:

$$\begin{aligned}
\dot{g}_i &= f \\
\ddot{g}_i &= 2f' \sum_j a_{ij} \dot{g}_j = 2f' f \sum_j a_{ij} \\
g_i^{(3)} &= 3 \left(f''(f, f) \sum_{j,k} a_{ij} a_{ik} + 2f' f' f \sum_{j,k} a_{ij} a_{jk} \right) \\
g_i^{(4)} &= 4 \left(f'''(f, f, f) \sum_{j,k,l} a_{ij} a_{ik} a_{il} + 24f''(f' f, f) \sum_{j,k,l} a_{ij} a_{jk} a_{il} + \right. \\
&\quad \left. + 12f' f''(f, f) \sum_{j,k,l} a_{ij} a_{jk} a_{jl} + 24f' f' f' f \sum_{j,k,l} a_{ij} a_{jk} a_{kl} \right) \\
&\vdots
\end{aligned}$$

Para cada árbol con raíz rt , denotamos por $\mathbf{g}_i(rt)$ al valor que verifica $\sum_i b_i \mathbf{g}_i(rt) = \Phi_i(rt)$.

Entonces se tiene que:

$$\begin{aligned}
\dot{g}_i &= f = \gamma(\tau) \alpha(\tau) \mathbf{g}_i(\tau) F(\tau)(y_0) = \sum_{rt \in RT_1} \gamma(rt) \alpha(rt) \mathbf{g}_i(rt) F(rt)(y_0) \\
\ddot{g}_i &= 2f' \sum_j a_{ij} \dot{g}_j = 2f' f \sum_j a_{ij} = \gamma([\tau]) \alpha([\tau]) \mathbf{g}_i([\tau]) F([\tau])(y_0) = \sum_{rt \in RT_2} \gamma(rt) \alpha(rt) \mathbf{g}_i(rt) F(rt)(y_0) \\
g_i^{(3)} &= 3f''(f, f) \sum_{j,k} a_{ij} a_{ik} + 6f' f' f \sum_{j,k} a_{ij} a_{jk} = \\
&= \gamma([\tau, \tau]) \alpha([\tau, \tau]) \mathbf{g}_i([\tau, \tau]) F([\tau, \tau])(y_0) + \gamma([[\tau]]) \alpha([[\tau]]) \mathbf{g}_i([[\tau]]) F([[\tau]]) (y_0) \\
g_i^{(4)} &= 4f'''(f, f, f) \sum_{j,k,l} a_{ij} a_{ik} a_{il} + 24f''(f' f, f) \sum_{j,k,l} a_{ij} a_{jk} a_{il} + \\
&\quad + 12f' f''(f, f) \sum_{j,k,l} a_{ij} a_{jk} a_{jl} + 24f' f' f' f \sum_{j,k,l} a_{ij} a_{jk} a_{kl} = \\
&= \gamma([\tau, \tau, \tau]) \alpha([\tau, \tau, \tau]) \mathbf{g}_i([\tau, \tau, \tau]) F([\tau, \tau, \tau])(y_0) + \gamma([\tau, \tau]) \alpha([\tau, \tau]) \mathbf{g}_i([\tau, \tau]) F([\tau, \tau])(y_0) + \\
&\quad + \gamma([\tau, \tau]) \alpha([\tau, \tau]) \mathbf{g}_i([\tau, \tau]) F([\tau, \tau])(y_0) + \gamma([[[\tau]]]) \alpha([[[\tau]]]) \mathbf{g}_i([[[\tau]]]) F([[[\tau]]]) (y_0) \\
&= \sum_{rt \in RT_4} \gamma(rt) \mathbf{g}_i(rt) \alpha(rt) F(rt)(y_0) \\
&\vdots
\end{aligned}$$

Finalmente, se tiene que:

$$g_i^{(q)}|_{h=0} = \sum_{rt \in RT_q} \gamma(rt) \mathbf{g}_i(rt) \alpha(rt) F(rt)(y_0). \quad (1.6)$$

En consecuencia:

$$\begin{aligned} y_1^q|_{h=0} &= \sum_i b_i g_i^q|_{h=0} = \sum_i b_i \left(\sum_{rt \in RT_q} \gamma(rt) \mathbf{g}_i(tr) \alpha(rt) F(rt)(y_0) \right) = \\ &= \sum_{rt \in RT_q} \gamma(rt) \Phi(rt) \alpha(rt) F(rt)(y_0). \end{aligned}$$

□

Gracias a las expresiones en forma de serie en potencias de h de las soluciones exacta y numérica, se llega a las condiciones generales de orden de los métodos Runge-Kutta:

Teorema 1.10. *El método Runge-Kutta tiene orden p si y solo si verifica esta condición:*

$$\Phi(rt) = \frac{1}{\gamma(rt)}, \text{ para todo árbol } rt \in RT \text{ de orden } \rho(rt) \leq p. \quad (1.7)$$

Demostración. Recordemos que un método Runge-Kutta es de orden p si el error de truncación es $O(h^{p+1})$ cuando $h \rightarrow 0$, es decir, si los correspondientes desarrollos de Taylor en potencias de h de $y(t_0 + h)$ e $y_1(h)$ coinciden hasta los términos de orden p .

Si se verifica la condición del enunciado, para cada $q \leq p$ se tiene que:

$$y_1^q|_{h=0} = \sum_{rt \in RT_q} \gamma(rt) \Phi(rt) \alpha(rt) F(rt)(y_0) = \sum_{rt \in RT_q} \alpha(rt) F(rt)(y_0) = y^q(t_0).$$

Luego, es una condición suficiente. Finalmente, teniendo en cuenta la independencia de las diferenciales elementales, se prueba el carácter necesario de la condición. □

1.2. B-series

Estudiaremos ahora las series en potencias de h con diferenciales elementales como las que han aparecido en las expresiones de las derivadas en los teoremas anteriores. Este tipo de series fueron nombradas B-series en 1974 por Hairer y Wanner. Nos centraremos en la normalización simétrica de la definición presentada por Butcher y Sanz-Serna en 1996.

El concepto de B-serie surge con el fin de obtener más información sobre el comportamiento de los métodos numéricos y extender la teoría del orden anterior a otros métodos. De hecho, permite demostrar de manera más sencilla muchos de los resultados vistos con anterioridad.

Para poder demostrar dichos resultados se requiere que las B-series sigan siendo B-series al ser introducidas en la expresión $hf(\cdot)$. Al introducir una serie como la del *teorema 1.9* en $hf(\cdot)$ surgen coeficientes pares en los distintos coeficientes de las potencias de

h . Por ello se buscó una normalización de los coeficientes teniendo así una definición simétrica de B-serie. Antes de dar la definición formal de B-serie, se requiere un concepto previo con el fin de lograr dicha normalización:

Definición 1.11. El coeficiente simétrico $\sigma(rt)$ del árbol $rt \in RT$ se define de manera recursiva:

- $\sigma(rt) = 1$ si $rt = \tau$.
- $\sigma(rt) = \sigma(rt_1) \cdot \dots \cdot \sigma(rt_m) \cdot \mu_1! \mu_2! \dots$, si $rt = [rt_1, \dots, rt_m]$, donde los enteros μ_1, μ_2, \dots cuentan el número de árboles iguales entre rt_1, \dots, rt_m .

Definición 1.12. La B-serie asociada a la aplicación $a : RT \cup \{\emptyset\} \rightarrow \mathbb{R}$ es la serie formal:

$$B(a, y) := a(\emptyset)y + \sum_{rt \in RT} \frac{h^{\rho(rt)}}{\sigma(rt)} a(rt) F(rt)(y).$$

Con esta definición de B-serie y bajo ciertas condiciones, se verifica que una B-serie introducida en $hf(\cdot)$ sigue siendo una B-serie:

Lema 1.13. Sea $a : RT \cup \{\emptyset\} \rightarrow \mathbb{R}$ una aplicación de modo que $a(\emptyset) = 1$. Entonces $hf(B(a, h))$ es una B-serie (ver [3]). De hecho:

$$hf(B(a, y)) = B(a', y)$$

donde $a'(\emptyset) = 0$, $a'(\tau) = 1$ y $a'([rt_1, \dots, rt_m]) = a(rt_1) \cdot \dots \cdot a(rt_m)$, $\forall rt_1, \dots, rt_m \in RT$.

Demostración. En primer lugar, por la propia definición de B-serie, observamos que $B(a, y) = a(\emptyset)y + O(h) = y + O(h)$. Esto permite desarrollar $hf(B(a, y))$ en serie de Taylor en potencias de h centrada en y :

$$hf(B(a, y)) = h \cdot \sum_{m \geq 0} \frac{1}{m!} f^{(m)}(y) (B(a, y) - y)^m,$$

interpretando $(B(a, y) - y)^m$ como la m -tupla de vectores

$$(B(a, y) - y, B(a, y) - y, \dots, B(a, y) - y)$$

sobre la que actúa $f^{(m)}$ como aplicación m -lineal, siendo $f^{(m)}$ la derivada m -ésima de la aplicación vectorial f . Notemos que, para cada $m \geq 0$, se tiene lo siguiente:

$$\begin{aligned} (B(a, y) - y)^m &= \left(\sum_{rt \in RT} \frac{h^{\rho(rt)}}{\sigma(rt)} a(rt) F(rt)(y), \dots, \sum_{rt \in T} \frac{h^{\rho(rt)}}{\sigma(rt)} a(rt) F(rt)(y) \right) = \\ &= \left(\sum_{rt_1 \in RT} \frac{h^{\rho(rt_1)}}{\sigma(rt_1)} a(rt_1) F(rt_1)(y), \dots, \sum_{rt_m \in RT} \frac{h^{\rho(rt_m)}}{\sigma(rt_m)} a(rt_m) F(rt_m)(y) \right) = \\ &= \sum_{rt_1 \in RT} \dots \sum_{rt_m \in RT} \frac{h^{\rho(rt_1) + \dots + \rho(rt_m)}}{\sigma(rt_1) \cdot \dots \cdot \sigma(rt_m)} a(rt_1) \cdot \dots \cdot a(rt_m) \left(F(rt_1)(y), \dots, F(rt_m)(y) \right). \end{aligned}$$

Por tanto, teniendo en cuenta las definiciones del coeficiente simétrico, el orden y la diferencial elemental de un árbol de la forma $[rt_1, \dots, rt_m] \in RT$, se tiene que:

$$\begin{aligned} hf(B(a, y)) &= h \cdot \sum_{m \geq 0} \frac{1}{m!} f^{(m)}(y) (B(a, y) - y)^m \\ &= \sum_{m \geq 0} \sum_{rt_1 \in RT} \dots \sum_{rt_m \in RT} \frac{1}{m!} \frac{h^{\rho(rt_1) + \dots + \rho(rt_m) + 1}}{\sigma(rt_1) \cdot \dots \cdot \sigma(rt_m)} a(rt_1) \cdot \dots \cdot a(rt_m) f^{(m)}(y) \left(F(rt_1)(y), \dots, F(rt_m)(y) \right) \\ &= \sum_{m \geq 0} \sum_{rt_1 \in RT} \dots \sum_{rt_m \in RT} \frac{\mu_1! \mu_2! \dots}{m!} \cdot \frac{h^{\rho([rt_1, \dots, rt_m])}}{\sigma([rt_1, \dots, rt_m])} \cdot a'([rt_1, \dots, rt_m]) \cdot F([rt_1, \dots, rt_m])(y). \end{aligned}$$

donde $a'([rt_1, \dots, rt_m]) = a(rt_1) \cdot \dots \cdot a(rt_m)$. En particular, $a'(\tau) = a(\emptyset) = 1$.

Ahora bien, el número de posibles ordenaciones de los m subárboles que dan lugar al árbol $[rt_1, \dots, rt_m] \in RT$ con algunos de ellos repetidos un número de veces prefijado (μ_1 veces uno de ellos, μ_2 veces otro de ellos,...) es:

$$\frac{m!}{\mu_1! \mu_2! \dots}$$

Por consiguiente:

$$hf(B(a, y)) = \sum_{rt \in RT} \frac{h^{\rho(rt)}}{\sigma(rt)} \cdot a'(rt) \cdot F(rt)(y) = B(a', y), \text{ siendo } a'(\emptyset) = 0.$$

□

1.2.1. Aplicación de las B-series sobre las condiciones de orden de los métodos Runge-Kutta

La finalidad de esta sección es proporcionar las condiciones de orden de los métodos Runge-Kutta generales mediante el uso de B-series.

Supongamos ahora que la solución exacta del problema es también una B-serie:

$$y(h) = B(e, y_0) = e(\emptyset)y_0 + \sum_{rt \in RT} \frac{h^{\rho(rt)}}{\sigma(rt)} e(rt)F(rt)(y_0).$$

Por un lado, aplicando el *lema 1.13* se tiene que:

$$hy' = hf(y) = hf(B(e, y_0)) = B(e', y_0) = \sum_{rt \in RT} \frac{h^{\rho(rt)}}{\sigma(rt)} e'(rt)F(rt)(y_0).$$

Por otro lado, derivando la expresión $y(h) = B(e, y_0)$ respecto de h se tiene que:

$$h\dot{y} = h \sum_{rt \in RT} \frac{\rho(rt)h^{\rho(rt)-1}}{\sigma(rt)} e(rt)F(rt)(y_0) = \sum_{rt \in RT} \frac{h^{\rho(rt)}}{\sigma(rt)} \rho(rt)e(rt)F(rt)(y_0).$$

Por tanto, se tiene la siguiente igualdad de coeficientes:

$$\left. \begin{array}{l} e'(rt) = \rho(rt)e(rt) \\ e'([rt_1, \dots, rt_m]) = e(rt_1) \cdot \dots \cdot e(rt_m) \end{array} \right\} \Rightarrow e(rt) = \frac{e(rt_1) \cdot \dots \cdot e(rt_m)}{\rho(rt)}, \forall rt = [rt_1, \dots, rt_m] \in RT.$$

Finalmente, con la definición de $\gamma(rt)$ se llega a que $e(rt) = \frac{1}{\gamma(rt)}$.

Ahora, comparando las potencias de h de las B-series $y = B(e, y_0)$ e $y_1 = B(\Phi, y_0)$, se obtiene de nuevo la condición de orden dada en el teorema 1.10.

La derivada q -ésima de la B-serie $y = B(e, y_0)$ es:

$$y^{(q)}(t_0) = \sum_{rt \in RT: \rho(rt) \geq q} \frac{(\rho(rt) \cdot \dots \cdot (\rho(rt) - q + 1)) h^{\rho(rt)-q}}{\sigma(rt) \cdot \gamma(rt)} F(rt)(y_0).$$

Igualándolo con la expresión de la derivada q -ésima de la solución exacta dada en el teorema 1.8, se obtiene una expresión para el cálculo de etiquetados monótonos de un árbol:

$$\alpha(rt) = \frac{\rho(rt)!}{\sigma(rt)\gamma(rt)}.$$

1.3. Árboles bicolor y P-series

Consideramos ahora el problema general $\dot{y} = F(y, t)$ con suficiente regularidad. Podemos tratar de integrar algunos de los componentes de y con un método y el resto con otro, incluso exigiendo que un método fuera rígido y otro no. Suponiendo $y = (p, q)$ y $F = (f, g)$ el problema se transforma en lo siguiente:

$$\frac{dp}{dt} = f(p, q, t), \quad \frac{dq}{dt} = g(p, q, t). \quad (1.8)$$

Un caso particular que resulta interesante para el estudio de los problemas hamiltonianos separables es aquel en el que las funciones solo dependen de p o de q , es decir, un sistema particionado:

$$\frac{dp}{dt} = f(q, t), \quad \frac{dq}{dt} = g(p). \quad (1.9)$$

Consideramos el sistema particionado siguiente:

$$\begin{cases} \dot{y} = f(y, z) \\ \dot{z} = g(y, z) \end{cases}$$

Vamos a construir árboles con raíz cuyos vértices podrán ser blancos o negros y que denotaremos brt . Las ramas que salen de un vértice representarán derivadas parciales: respecto de y si el vértice hijo es negro y respecto de z si el vértice hijo es blanco. Estos árboles se denominan árboles bicolor con raíz y el conjunto de los mismos se denota TP. Para indicar el color de la raíz del árbol se añade un subíndice: $[brt_1, \dots, brt_m]_y$ si tiene raíz negra y $[brt_1, \dots, brt_m]_z$ si tiene raíz blanca. Ahora hay dos árboles con un solo vértice: $\tau_y = \bullet$ y $\tau_z = \circ$. Denotaremos TP_y y TP_z al conjunto de árboles bicolor con raíz negra o blanca respectivamente.

Con esta notación se puede calcular su diferencial elemental. Veámoslo con un ejemplo:

$$[\tau_z, [\tau_y, \tau_z]_y]_y = \begin{array}{c} \begin{array}{ccc} \circ & & \bullet \\ & \diagdown & / \\ & \bullet & \\ & / & \diagdown \\ \bullet & & \circ \end{array} \\ \begin{array}{c} g \\ f \\ f_{yz} \\ g \\ f_{yz} \end{array} \end{array} \iff F(t)(y, z) = f_{yz}(f_{yz}(f, g), g)$$

Los conceptos de orden y función de densidad son los mismos que para los árboles con un color. Sin embargo, hay que tener en cuenta el color de los vértices del árbol para calcular el coeficiente simétrico, el peso elemental y el número de etiquetados monótonos, a pesar de que las definiciones son las mismas. Por ejemplo:

$$\begin{array}{ccc} \begin{array}{ccc} \bullet & \bullet & \bullet \\ & \diagdown & / \\ & \bullet & \\ & / & \diagdown \\ \bullet & & \bullet \end{array} & \begin{array}{ccc} \bullet & \bullet & \circ \\ & \diagdown & / \\ & \bullet & \\ & / & \diagdown \\ \bullet & & \bullet \end{array} \\ f_{yyy} & f_{yyy} \\ brt_1 = [\tau_y, \tau_y, \tau_y]_y & brt_2 = [\tau_y, \tau_y, \tau_z]_y \end{array}$$

En ambos casos, el orden es 4 y la función de densidad es 4.

Por un lado: $\sigma(brt_1) = 3! = 6$, $\alpha(brt_1) = 1$ y $\Phi(brt_1) = \sum_i b_i c_i^3$, donde $c_i = \sum_j a_{ij}$.

Por otro lado: $\sigma(brt_2) = 2! = 2$, $\alpha(brt_2) = 3$ y $\Phi(brt_2) = \sum_i b_i c_i^2 C_i$, donde $C_i = \sum_j A_{ij}$.

Teniendo en cuenta las dos variables y y z , obtenemos una extensión de las B-series denominada P-serie:

Definición 1.14. La P-serie asociada a la aplicación $a : TP \cup \{\emptyset_y, \emptyset_z\} \rightarrow \mathbb{R}$ es:

$$P(a, (y, z)) = \begin{pmatrix} a(\emptyset_y)y + \sum_{brt \in TP_y} \frac{h^{\rho(brt)}}{\sigma(brt)} a(brt)F(brt)(y, z) \\ a(\emptyset_z)z + \sum_{brt \in TP_z} \frac{h^{\rho(brt)}}{\sigma(brt)} a(brt)F(brt)(y, z) \end{pmatrix}. \quad (\text{Ver [5]}).$$

Los resultados dados para las B-series y los métodos Runge-Kutta generales se extienden a las P-series y los métodos Runge-Kutta particionados, dando lugar a los dos enunciados siguientes:

Lema 1.15. Sea $a : TP \cup \{\emptyset_y, \emptyset_z\} \rightarrow \mathbb{R}$ tal que $a(\emptyset_y) = a(\emptyset_z) = 1$. Entonces:

$$h \begin{pmatrix} f(P(a, (y, z))) \\ g(P(a, (y, z))) \end{pmatrix} = P(a', (y, z)),$$

donde $a'(\emptyset_y) = a'(\emptyset_z) = 0$, $a'(\tau_y) = a'(\tau_z) = 1$ y

$$a'([brt_1, \dots, brt_m]_y) = a'([brt_1, \dots, brt_m]_z) = a(brt_1) \cdot \dots \cdot a(brt_m), \quad \forall brt_1, \dots, brt_m \in TP.$$

Teorema 1.16. La solución exacta es la P-serie $P(e, (y_0, z_0)) = (y(t_0 + h), z(t_0 + h))$, donde $e(\emptyset_y) = e(\emptyset_z) = 1$ y $e(brt) = \frac{1}{\gamma(brt)}$, $\forall brt \in TP$.

Ambos resultados están recogidos en [3] y [4]. Sus demostraciones son similares a las vistas para B-series generales, por lo que no se han incluido en la memoria.

1.4. Métodos Runge-Kutta particionados

Presentamos en esta sección uno de los objetos de estudio principales de esta memoria, conocido como método Runge-Kutta particionado. Este tipo de métodos se utilizan para resolver problemas particionados, integrando parte de las componentes de la variable del problema con un método Runge-Kutta y el resto con otro. Suelen ser efectivos para la resolución de problemas hamiltonianos separables. La extensión de B-serie anteriormente presentada permite proporcionar condiciones de orden para este conjunto de métodos.

Consideramos el sistema particionado siguiente:

$$\begin{cases} \dot{y} = f(y, z) \\ \dot{z} = g(y, z) \end{cases} \quad (1.10)$$

Supongamos ahora que y y z se integran con dos métodos Runge-Kutta diferentes pero con el mismo número de etapas. Este método recibe el nombre de método Runge-Kutta particionado y viene dado por dos tableros de Butcher:

$$\begin{array}{c|c}
 a_{11} \cdot \cdot \cdot a_{1s} & A_{11} \cdot \cdot \cdot A_{1s} \\
 \vdots & \vdots \\
 \vdots & \vdots \\
 a_{s1} \cdot \cdot \cdot a_{ss} & A_{s1} \cdot \cdot \cdot A_{ss} \\
 \hline
 b_1 \cdot \cdot \cdot b_s & B_1 \cdot \cdot \cdot B_s
 \end{array}$$

Figura 1.2: Método Runge-Kutta particionado.

Este método viene dado por las siguientes expresiones:

$$\left\{ \begin{array}{l}
 k_i = f\left(y_0 + h \sum_{j=1}^s a_{ij}k_j, z_0 + h \sum_{j=1}^s A_{ij}l_j\right), \quad i = 1, \dots, s. \\
 l_i = g\left(y_0 + h \sum_{j=1}^s a_{ij}k_j, z_0 + h \sum_{j=1}^s A_{ij}l_j\right), \quad i = 1, \dots, s. \\
 (y_1, z_1) = \left(y_0 + h \sum_{i=1}^s b_i k_i, z_0 + h \sum_{i=1}^s B_i l_i\right).
 \end{array} \right. \quad (1.11)$$

Inicialmente, los métodos Runge-Kutta particionados fueron propuestos por Hofer (1976) y por Griepentrog (1978) para abordar problemas con partes rígidas y partes no rígidas.

La teoría de orden de los métodos Runge-Kutta generales se puede extender directamente a los particionados. El método Runge-Kutta particionado (1.11) se puede interpretar como un método de un paso: $(y_1, z_1) = \Psi_h(y_0, z_0)$. Entonces será de orden p si el error local es $O(h^{p+1})$, como sucede en los métodos Runge-Kutta generales. De hecho, si se considera el problema

$$\begin{cases} \dot{y} = f(y) \\ \dot{z} = g(z) \end{cases}$$

sin términos de acoplamiento, el método particionado (1.11) aplicado a este problema tiene orden, a lo sumo, el mínimo de los órdenes de los dos métodos Runge-Kutta utilizados.

Veamos las condiciones de orden que aparecen si se desea alcanzar orden 2 para el problema (1.10). Se tienen cuatro árboles bicolor que proporcionan cuatro condiciones de

orden. Todos ellos tienen densidad $\gamma = 2$. Las condiciones de orden obtenidas son:

$$\begin{aligned} [\tau_y]_y &\Rightarrow \sum_{i,j} b_i a_{ij} = \sum_i b_i c_i = \frac{1}{2} \\ [\tau_z]_z &\Rightarrow \sum_{i,j} B_i A_{ij} = \sum_i B_i C_i = \frac{1}{2} \\ [\tau_y]_z &\Rightarrow \sum_{i,j} B_i a_{ij} = \sum_i B_i c_i = \frac{1}{2} \\ [\tau_z]_y &\Rightarrow \sum_{i,j} b_i A_{ij} = \sum_i b_i C_i = \frac{1}{2} \end{aligned}$$

Las dos primeras condiciones coinciden con las condiciones de orden para orden 2 de cada uno de los métodos Runge-Kutta utilizados. Observamos que las dos restantes se verifican automáticamente por métodos particionados basados en los mismos nodos de cuadratura, es decir: $c_i = C_i, \forall i$.

Supongamos ahora que se verifica $c_i = C_i$ y veamos las condiciones de orden para orden 3. Además de las condiciones de orden de cada uno de los métodos Runge-Kutta utilizados, aparecen otras cuatro condiciones:

$$\begin{aligned} \sum_{i,j} b_i A_{ij} c_j &= \frac{1}{6}, \quad \sum_{i,j} b_i A_{ij} c_j^2 = \frac{1}{6} \\ \sum_i b_i c_i^2 &= \frac{1}{3}, \quad \sum_i B_i c_i^2 = \frac{1}{3} \end{aligned}$$

1.4.1. Ejemplos de métodos Runge-Kutta particionados

Se exponen a continuación dos ejemplos sencillos de métodos Runge-Kutta particionados. Gracias a dichos ejemplos podemos apreciar que estos métodos son adecuados para sistemas hamiltonianos separables por su carácter explícito al aplicarse a dichos problemas.

Métodos de Euler

Consideramos sistemas particionados de la forma:

$$\begin{cases} \dot{p} = f(p, q) \\ \dot{q} = g(p, q) \end{cases}$$

Uno de los ejemplos más básicos de métodos Runge-Kutta particionados son aquellos en los que se utiliza el método de Euler tanto implícito como explícito. Por ejemplo:

$$\begin{cases} p_{n+1} = p_n + hf(p_{n+1}, q_n) & \text{(implícito)} \\ q_{n+1} = q_n + hg(p_{n+1}, q_n) & \text{(explícito)} \end{cases}$$

Es un método Runge-Kutta particionado y viene dado por las siguientes tablas de Butcher:

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array} \quad \begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

Ahora bien, si se aplica a un sistema hamiltoniano separable de la forma

$$\begin{cases} \dot{p} = f(q) \\ \dot{q} = g(p) \end{cases}$$

se convierte en un método explícito dado por las siguientes ecuaciones:

$$\begin{cases} p_{n+1} = p_n + hf(q_n) \\ q_{n+1} = q_n + hg(p_{n+1}) \end{cases}$$

Método Störmer-Verlet

Consideramos problemas Hamiltonianos separables de la forma siguiente:

$$\begin{cases} \dot{p} = f(q) \\ \dot{q} = p \end{cases} \quad \text{o, equivalentemente, } \ddot{q} = f(q).$$

Una discretización natural de este problema se obtiene reemplazando la derivada segunda por la diferencia central de segundo orden:

$$q_{n+1} - 2q_n + q_{n-1} = h^2 f(q_n).$$

Este método básico se conoce como el método Störmer-Verlet. Las aproximaciones a la derivada primera de q se obtienen discretizando $\dot{q} = p$ mediante la diferencia central que aproxima la primera derivada:

$$p_n = \frac{q_{n+1} - q_{n-1}}{2h}, \quad p_{n+1/2} = \frac{q_{n+1} - q_n}{h},$$

obteniendo la segunda igualdad a partir de la primera al dar un paso de tamaño $h/2$. Veamos que este método admite una formulación de un paso. Por un lado, discretizamos $\dot{p} = f(q)$ mediante la diferencia central de primer orden:

$$2hf(q_n) = p_{n+1} - p_{n-1} \Rightarrow hf(q_n) = p_{n+1/2} - p_{n-1/2}.$$

Por otro lado:

$$p_{n+1/2} + p_{n-1/2} = \frac{q_{n+1} - q_n}{h} + \frac{q_n - q_{n-1}}{h} = \frac{q_{n+1} - q_{n-1}}{h} = 2p_n.$$

Sumando estas dos expresiones obtenemos:

$$p_{n+1/2} = p_n + \frac{h}{2}f(q_n).$$

Restando las expresiones anteriores para $n + 1$, se tiene:

$$\left. \begin{array}{l} p_{n+3/2} + p_{n+1/2} = 2p_{n+1} \\ p_{n+3/2} - p_{n+1/2} = hf(q_{n+1}) \end{array} \right\} \Rightarrow p_{n+1} = p_{n+1/2} + \frac{h}{2}f(q_{n+1}).$$

Además:

$$p_{n+1/2} = \frac{q_{n+1} - q_n}{h} \Rightarrow q_{n+1} = q_n + hp_{n+1/2}.$$

Finalmente, la formulación de un paso del método Störmer-Verlet viene dada por las siguientes expresiones:

$$\left\{ \begin{array}{l} p_{n+1/2} = p_n + \frac{h}{2}f(q_n) \\ q_{n+1} = q_n + hp_{n+1/2} \\ p_{n+1} = p_{n+1/2} + \frac{h}{2}f(q_{n+1}) \end{array} \right.$$

No es más que un método explícito $\Psi_h : (q_n, p_n) \mapsto (q_{n+1}, p_{n+1})$ de un paso. Este es un método Runge-Kutta particionado con coeficientes:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1/2 & 1/2 \\ \hline & 1/2 & 1/2 \end{array} \quad \begin{array}{c|cc} 1/2 & 1/2 & 0 \\ 1/2 & 1/2 & 0 \\ \hline & 1/2 & 1/2 \end{array}$$

Un problema de la forma $\ddot{q} = f(q)$ es el problema del péndulo simple, cuyas ecuaciones del movimiento vienen dadas por:

$$\left\{ \begin{array}{l} \dot{p} = -\sin(q) \\ \dot{q} = p \end{array} \right.$$

1.4.2. Condiciones de orden de métodos Runge-Kutta particionados mediante P-series

Vimos que la solución numérica de todo método Runge-Kutta era una P-serie de la forma $P(\Phi, y)$, donde Φ es la función de los pesos elementales de los árboles. Lo mismo sucede en el caso de los métodos Runge-Kutta particionados, teniendo en cuenta que el peso elemental de un árbol dependerá de su raíz. Los siguientes resultados están formulados en [3].

Teorema 1.17. *La solución numérica del método Runge-Kutta particionado es la P-serie $P(\Phi, (y_0, z_0)) = (y_1, z_1)$, donde $\Phi(\emptyset_y) = \Phi(\emptyset_z) = 1$ y*

$$\Phi(brt) = \begin{cases} \sum_{i=1}^s b_i \Phi_i(brt), & \text{si } brt \in TP_y \\ \sum_{i=1}^s B_i \Phi_i(brt), & \text{si } brt \in TP_z \end{cases},$$

$$\text{con } \Phi_i(\tau_y) = \Phi_i(\tau_z) = 1 \text{ y } \Phi([brt_1, \dots, brt_m]_y) = \Phi([brt_1, \dots, brt_m]_z) = \prod_{j=1}^m \psi_i(brt_j),$$

$$\text{siendo } \psi_i(brt_j) = \begin{cases} \sum_{k=1}^s a_{ik} \Phi_k(brt_j), & \text{si } brt_j \in TP_y \\ \sum_{k=1}^s A_{ik} \Phi_k(brt_j), & \text{si } brt_j \in TP_z \end{cases},$$

Recordemos que un método Runge-Kutta es de orden r si verifica que $y_1 - y(t_0 + h) = O(h^{r+1})$. En particular, un método Runge-Kutta particionado es de orden r si verifica:

$$\begin{cases} y_1 - y(t_0 + h) = O(h^{r+1}) \\ z_1 - z(t_0 + h) = O(h^{r+1}) \end{cases}$$

Del mismo modo que en el caso general, se obtiene la condición de orden siguiente:

Teorema 1.18. *Un método Runge-Kutta particionado es de orden r si y solo si verifica esta condición:*

$$\Phi(brt) = \frac{1}{\gamma(brt)}, \text{ para todo árbol } brt \in TP \text{ de orden } \rho(brt) \leq r.$$

Observación 1.19. Nótese que, en el caso de sistemas hamiltonianos separables (1.9), los árboles bicolor que aparecen no tienen vértices consecutivos del mismo color por la forma de las ecuaciones del sistema. Este subconjunto del conjunto de árboles bicolor constituido por aquellos con vértices consecutivos de distinto color será el utilizado para estudiar las condiciones de orden de los métodos Runge-Kutta simpléticos en el siguiente capítulo.

Capítulo 2

Métodos simplécticos

En este capítulo trataremos el concepto de métodos simplécticos. En particular, veremos las condiciones de orden de los métodos Runge-Kutta simplécticos y Runge-Kutta particionados simplécticos, así como las ventajas que presentan respecto al cálculo del orden.

2.1. Transformaciones y métodos simplécticos

Sea Ω un dominio en \mathbb{R}^{2d} y sea I un intervalo real. Consideraremos la variable temporal $t \in I$ y las variables $(p, q) = (p_1, \dots, p_d, q_1, \dots, q_d) \in \Omega$. Sea $H = H(p, q, t)$ una función real definida en $\Omega \times I$ suficientemente regular. Entonces el sistema hamiltoniano de ecuaciones diferenciales con hamiltoniano H viene dado por:

$$\begin{cases} \frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i}, & i = 1, \dots, d, \\ \frac{dq_i}{dt} = \frac{\partial H}{\partial p_i}, & i = 1, \dots, d. \end{cases} \quad (2.1)$$

El espacio Ω recibe el nombre de espacio de fase y el valor d denota los grados de libertad del sistema.

En muchos casos, el hamiltoniano H corresponde a la energía mecánica total del sistema. Pero esto no es siempre así:

Ejemplo 2.1. Consideramos el péndulo simple con una masa m sujeta a una cuerda de longitud inicial l_0 y, cuando el péndulo se pone en movimiento, la cuerda se acorta a velocidad constante $-\alpha$.

Luego, la longitud de la cuerda en tiempo t es: $l(t) = l_0 - \alpha t$. Además, se tiene que:

$$V = mg(l_0 - l \cos(\theta)), \quad T = \frac{1}{2}m(l^2\dot{\theta}^2 + \alpha^2),$$

siendo θ la coordenada generalizada que mide el ángulo que forma la cuerda con la posición de equilibrio vertical. El Lagrangiano es $L = T - V$. La ecuación de Lagrange nos da la ecuación del movimiento:

$$0 = L_\theta - \frac{d}{dt}L_{\dot{\theta}} \Rightarrow g \sin(\theta)l\ddot{\theta} - 2l\alpha\dot{\theta} = 0.$$

Entonces el Hamiltoniano del sistema es:

$$H = \dot{\theta}L_{\dot{\theta}} - L = (T + V) - m\alpha^2.$$

En este ejemplo, el Hamiltoniano no coincide con la energía total del sistema.

Entre los sistemas hamiltonianos destacan los sistemas autónomos, pues en ellos H no depende del tiempo t ($\frac{dH}{dt} = 0$), es decir, H se conserva a lo largo de las soluciones del sistema. En la práctica, se corresponde con la conservación de la energía.

Algunos de los ejemplos más conocidos de sistemas hamiltonianos son el oscilador armónico, el péndulo y el problema de Kepler.

El concepto de sistema simpléctico que veremos más adelante caracteriza a los sistemas hamiltonianos por su solución más que por la forma de las ecuaciones diferenciales que lo conforman. Para dar dicho concepto debemos presentar un operador: el operador solución.

Definición 2.2. El operador solución o flujo del sistema, denotado φ_t , es la aplicación de Ω en Ω que asigna a cada punto $y_0 \in \Omega$ del espacio de fase el valor $y(t)$ de la solución del sistema hamiltoniano (2.1) con valor inicial $y(0) = y_0$, es decir:

$$\varphi_t(y_0) := y(t), \text{ si } y(0) = y_0.$$

En general, se considera que el subíndice t indica el tiempo transcurrido desde la condición inicial, es decir:

$$\varphi_t(y_0) := y(t + t_0), \text{ si } y(t_0) = y_0.$$

También se suele denotar por $\Phi_H(t_0 + t, t_0)$, para indicar el tiempo de la condición inicial y el sistema hamiltoniano H al que está asociado.

Consideramos ahora una superficie D orientada dos-dimensional en Ω y sus correspondientes proyecciones D_i , $i = 1, \dots, d$, en los planos correspondientes a las variables (p_i, q_i) . Definimos $m(D)$ como la suma de las áreas orientadas de las proyecciones D_i . El flujo del sistema hamiltoniano (2.1) conserva este valor, es decir:

$$m(\varphi_t(D)) = m(D), \text{ si } D \text{ está contenido en el dominio del flujo del sistema.}$$

Una transformación con esta propiedad se denomina simpléctica o canónica. Formalmente (ver [6]):

Definición 2.3. Una aplicación lineal $a : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ se dice que es simpléctica si $a^t J a = J$, siendo $J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$ e I la matriz identidad de tamaño d .

Teorema 2.4. *El operador solución o flujo de un sistema hamiltoniano es una transformación simpléctica. (Ver [8]).*

Los métodos numéricos buscan cambiar este operador solución por una función suficientemente regular que aproxime la solución del problema.

Definición 2.5. Un método numérico proporciona una función continua $\Psi_F(t_{n+1}, t_n)(y)$ que aproxima el flujo φ_t del sistema $\frac{dy}{dt} = F(y, t)$ de modo que $y^{n+1} = \Psi_F(t_{n+1}, t_n)(y^n)$, siendo y^n una aproximación en tiempo t_n a la solución de $\frac{dy}{dt} = F(y, t)$

Buscaremos que el operador Ψ sea simpléctico, al igual que el operador solución. En general, los métodos numéricos no utilizan transformaciones simplécticas. Obtengamos una caracterización de las transformaciones simplécticas en función de formas diferenciales y del producto exterior.

Comencemos por un caso más sencillo y de menor dimensión. Consideremos una transformación $\Psi(p, q) = (p^*, q^*)$ definida en el dominio Ω . Vemos que, por el cambio de variable integral, Ψ conserva el área y la orientación si y solo si el determinante de su jacobiano es 1, es decir:

$$\frac{\partial p^*}{\partial p} \frac{\partial q^*}{\partial q} - \frac{\partial p^*}{\partial q} \frac{\partial q^*}{\partial p} = 1, \forall (p, q) \in \Omega.$$

Dadas dos 1-formas diferenciales ω y ω' , su producto externo $\omega \wedge \omega'$ es una 2-forma diferencial. Las diferenciales de las componentes de la transformación Ψ son las 1-formas diferenciales siguientes:

$$dp^* = \frac{\partial p^*}{\partial p} dp + \frac{\partial p^*}{\partial q} dq, \quad dq^* = \frac{\partial q^*}{\partial p} dp + \frac{\partial q^*}{\partial q} dq$$

El producto exterior es bilineal, antisimétrico y distributivo:

$$\begin{cases} dp^* \wedge dq^* &= \frac{\partial p^*}{\partial p} \frac{\partial q^*}{\partial p} dp \wedge dp + \frac{\partial p^*}{\partial p} \frac{\partial q^*}{\partial q} dp \wedge dq + \frac{\partial p^*}{\partial q} \frac{\partial q^*}{\partial p} dq \wedge dp + \frac{\partial p^*}{\partial q} \frac{\partial q^*}{\partial q} dq \wedge dq, \\ dp \wedge dp &= dq \wedge dq = 0, \\ dp \wedge dq &= -dq \wedge dp. \end{cases}$$

Por tanto:

$$dp^* \wedge dq^* = \left(\frac{\partial p^*}{\partial p} \frac{\partial q^*}{\partial q} - \frac{\partial p^*}{\partial q} \frac{\partial q^*}{\partial p} \right) dp \wedge dq.$$

De este modo se tiene que Ψ conserva el área si y solo si $dp^* \wedge dq^* = dp \wedge dq$.

Con un razonamiento análogo y aumentando la dimensión, se puede obtener una condición de simpléctico en función de 1-formas diferenciales:

Teorema 2.6. *La transformación $\Psi(\mathbf{p}, \mathbf{q}) = (\mathbf{p}^*, \mathbf{q}^*)$ es una transformación simpléctica si y solo si $d\mathbf{p}^* \wedge d\mathbf{q}^* = d\mathbf{p} \wedge d\mathbf{q}$. (Ver [8]).*

Definición 2.7. Los métodos numéricos que generan aplicaciones simplécticas al aplicarse sobre sistemas hamiltonianos reciben el nombre de métodos simplécticos o canónicos.

2.2. Métodos Runge-Kutta simplécticos

Algunos métodos Runge-Kutta son simplécticos. El método Runge-Kutta (1.1) aplicado al sistema Hamiltoniano (2.1) proporciona las siguientes relaciones:

$$\left\{ \begin{array}{l} \mathbf{P}_i = \mathbf{p}^n + h_{n+1} \sum_{j=1}^s a_{ij} \mathbf{f}(\mathbf{P}_j, \mathbf{Q}_j, t_n + c_j h_{n+1}) \\ \mathbf{Q}_i = \mathbf{q}^n + h_{n+1} \sum_{j=1}^s a_{ij} \mathbf{g}(\mathbf{P}_j, \mathbf{Q}_j, t_n + c_j h_{n+1}) \\ \mathbf{p}^{n+1} = \mathbf{p}^n + h_{n+1} \sum_{i=1}^s b_i \mathbf{f}(\mathbf{P}_i, \mathbf{Q}_i, t_n + c_i h_{n+1}) \\ \mathbf{q}^{n+1} = \mathbf{q}^n + h_{n+1} \sum_{i=1}^s b_i \mathbf{g}(\mathbf{P}_i, \mathbf{Q}_i, t_n + c_i h_{n+1}) \end{array} \right. \quad (2.2)$$

donde $\mathbf{f} = \left(-\frac{\partial H}{\partial q_i} \right)_{i=1}^s$ y $\mathbf{g} = \left(\frac{\partial H}{\partial p_i} \right)_{i=1}^s$, siendo \mathbf{P}_i y \mathbf{Q}_i pasos intermedios.

Teorema 2.8. *Supongamos que los coeficientes del método Runge-Kutta (1.1) satisfacen la siguiente relación:*

$$b_i a_{ij} + b_j a_{ji} - b_i b_j = 0, \text{ para } i, j = 1, \dots, s.$$

Entonces el método es simpléctico. De hecho, por simetría, basta que se satisfaga la relación para $i \leq j$, con $j = 1, \dots, s$. El recíproco también es cierto. (Ver [8] y [2]).

Demostración. Por lo visto en la sección anterior, bastará probar:

$$d\mathbf{p}^{n+1} \wedge d\mathbf{q}^{n+1} - d\mathbf{p}^n \wedge d\mathbf{q}^n = 0$$

Denotaremos:

$$\left\{ \begin{array}{l} \mathbf{k}_i = \mathbf{f}(\mathbf{P}_i, \mathbf{Q}_i, t_n + c_i h_{n+1}), \\ \mathbf{l}_i = \mathbf{g}(\mathbf{P}_i, \mathbf{Q}_i, t_n + c_i h_{n+1}). \end{array} \right.$$

Diferenciando y aplicando el producto exterior a las dos últimas expresiones de (2.2), se tiene que:

$$\begin{aligned}
d\mathbf{p}^{n+1} \wedge d\mathbf{q}^{n+1} - d\mathbf{p}^n \wedge d\mathbf{q}^n &= \\
&= h_{n+1} \sum_{i=1}^s b_i d\mathbf{k}_i \wedge d\mathbf{q}^n + h_{n+1} \sum_{j=1}^s b_j d\mathbf{p}^n \wedge d\mathbf{l}_j + h_{n+1}^2 \sum_{i,j=1}^s b_i b_j d\mathbf{k}_i \wedge d\mathbf{l}_j = \\
&= h_{n+1} \sum_{i=1}^s b_i (d\mathbf{k}_i \wedge d\mathbf{q}^n + d\mathbf{p}^n \wedge d\mathbf{l}_i) + h_{n+1}^2 \sum_{i,j=1}^s b_i b_j d\mathbf{k}_i \wedge d\mathbf{l}_j.
\end{aligned}$$

Diferenciando ahora en las dos expresiones restantes de (2.2) se tiene que:

$$\begin{aligned}
d\mathbf{p}^n \wedge d\mathbf{l}_i &= d\mathbf{P}_i \wedge d\mathbf{l}_i - h_{n+1} \sum_{j=1}^s a_{ij} d\mathbf{k}_j \wedge d\mathbf{l}_i, \\
d\mathbf{k}_i \wedge d\mathbf{q}_n &= d\mathbf{k}_i \wedge d\mathbf{Q}_i - h_{n+1} \sum_{j=1}^s a_{ij} d\mathbf{k}_i \wedge d\mathbf{l}_j.
\end{aligned}$$

Sustituyendo estas expresiones en la igualdad anterior:

$$\begin{aligned}
d\mathbf{p}^{n+1} \wedge d\mathbf{q}^{n+1} - d\mathbf{p}^n \wedge d\mathbf{q}^n &= \\
&= h_{n+1} \sum_{i=1}^s b_i (d\mathbf{k}_i \wedge d\mathbf{Q}_i + d\mathbf{P}_i \wedge d\mathbf{l}_i) - h_{n+1}^2 \sum_{i,j=1}^s (b_i a_{ij} + b_j a_{ji} - b_i b_j) d\mathbf{k}_i \wedge d\mathbf{l}_j = \\
&= h_{n+1} \sum_{i=1}^s b_i (d\mathbf{k}_i \wedge d\mathbf{Q}_i + d\mathbf{P}_i \wedge d\mathbf{l}_i),
\end{aligned}$$

utilizando en la última igualdad la hipótesis del enunciado.

Para finalizar basta ver que, para cada $i = 1, \dots, s$, se tiene: $d\mathbf{k}_i \wedge d\mathbf{Q}_i + d\mathbf{P}_i \wedge d\mathbf{l}_i = 0$.

Observamos:

$$d\mathbf{k} \wedge d\mathbf{Q} + d\mathbf{P} \wedge d\mathbf{l} = \sum_{\alpha=1}^d \left(dk_{\alpha} \wedge dQ_{\alpha} + dP_{\alpha} \wedge dl_{\alpha} \right)$$

Teniendo en cuenta que:

$$dk_{\alpha} = \frac{\partial f_{\alpha}}{\partial p_{\beta}} dP_{\beta} + \frac{\partial f_{\alpha}}{\partial q_{\beta}} dQ_{\beta} \text{ y } dl_{\alpha} = \frac{\partial g_{\alpha}}{\partial p_{\beta}} dP_{\beta} + \frac{\partial g_{\alpha}}{\partial q_{\beta}} dQ_{\beta},$$

se tiene que:

$$\begin{aligned}
d\mathbf{k} \wedge d\mathbf{Q} + d\mathbf{P} \wedge d\mathbf{l} &= \sum_{\alpha,\beta=1}^d \left[\left(\frac{\partial f_{\alpha}}{\partial p_{\beta}} dP_{\beta} \wedge dQ_{\alpha} + \frac{\partial f_{\alpha}}{\partial q_{\beta}} dQ_{\beta} \wedge dQ_{\alpha} \right) + \right. \\
&\quad \left. + \left(\frac{\partial g_{\alpha}}{\partial p_{\beta}} dP_{\alpha} \wedge dP_{\beta} + \frac{\partial g_{\alpha}}{\partial q_{\beta}} dP_{\alpha} \wedge dQ_{\beta} \right) \right]
\end{aligned}$$

Como $\mathbf{f} = \left(-\frac{\partial H}{\partial q_i}\right)_{i=1}^s$ y $\mathbf{g} = \left(\frac{\partial H}{\partial p_i}\right)_{i=1}^s$, entonces:

$$\begin{aligned}\frac{\partial f_\alpha}{\partial p_\beta} &= -\frac{\partial^2 H}{\partial q_\alpha \partial p_\beta}, & \frac{\partial f_\alpha}{\partial q_\beta} &= -\frac{\partial^2 H}{\partial q_\alpha \partial q_\beta}, \\ \frac{\partial g_\alpha}{\partial p_\beta} &= \frac{\partial^2 H}{\partial p_\alpha \partial p_\beta}, & \frac{\partial g_\alpha}{\partial q_\beta} &= \frac{\partial^2 H}{\partial p_\alpha \partial q_\beta}.\end{aligned}$$

Por la antisimetría del producto exterior, obtenemos:

$$\begin{aligned}& \left(\frac{\partial f_\alpha}{\partial p_\beta} dP_\beta \wedge dQ_\alpha + \frac{\partial g_\alpha}{\partial q_\beta} dP_\alpha \wedge dQ_\beta\right) + \left(\frac{\partial f_\beta}{\partial p_\alpha} dP_\alpha \wedge dQ_\beta + \frac{\partial g_\beta}{\partial q_\alpha} dP_\beta \wedge dQ_\alpha\right) = \\ & \left(-\frac{\partial^2 H}{\partial p_\beta \partial q_\alpha} dP_\beta \wedge dQ_\alpha + \frac{\partial^2 H}{\partial p_\alpha \partial q_\beta} dP_\alpha \wedge dQ_\beta\right) + \left(-\frac{\partial^2 H}{\partial p_\alpha \partial q_\beta} dP_\alpha \wedge dQ_\beta + \frac{\partial^2 H}{\partial p_\beta \partial q_\alpha} dP_\beta \wedge dQ_\alpha\right) = 0 \\ & \Rightarrow \sum_{\alpha, \beta=1}^d \left(\frac{\partial f_\alpha}{\partial p_\beta} dP_\beta \wedge dQ_\alpha + \frac{\partial g_\alpha}{\partial q_\beta} dP_\alpha \wedge dQ_\beta\right) = 0.\end{aligned}$$

$$\begin{aligned}\frac{\partial f_\alpha}{\partial q_\beta} dQ_\beta \wedge dQ_\alpha + \frac{\partial f_\beta}{\partial q_\alpha} dQ_\alpha \wedge dQ_\beta &= \frac{-\partial^2 H}{\partial q_\alpha \partial q_\beta} dQ_\beta \wedge dQ_\alpha + \frac{-\partial^2 H}{\partial q_\alpha \partial q_\beta} dQ_\alpha \wedge dQ_\beta = 0 \\ &\Rightarrow \sum_{\alpha, \beta=1}^d \frac{\partial f_\alpha}{\partial q_\beta} dQ_\beta \wedge dQ_\alpha = 0.\end{aligned}$$

$$\begin{aligned}\frac{\partial g_\alpha}{\partial p_\beta} dP_\beta \wedge dP_\alpha + \frac{\partial g_\beta}{\partial p_\alpha} dP_\alpha \wedge dP_\beta &= \frac{\partial^2 H}{\partial p_\alpha \partial p_\beta} dP_\beta \wedge dP_\alpha + \frac{\partial^2 H}{\partial p_\alpha \partial p_\beta} dP_\alpha \wedge dP_\beta = 0 \\ &\Rightarrow \sum_{\alpha, \beta=1}^d \frac{\partial g_\alpha}{\partial p_\beta} dP_\alpha \wedge dP_\beta = 0.\end{aligned}$$

En consecuencia, $d\mathbf{k}_i \wedge d\mathbf{Q}_i + d\mathbf{P}_i \wedge d\mathbf{l}_i = 0$ y, por tanto, el método es simpléctico. Supongamos ahora que el método es simpléctico, es decir:

$$\begin{aligned}0 &= d\mathbf{p}^{n+1} \wedge d\mathbf{q}^{n+1} - d\mathbf{p}^n \wedge d\mathbf{q}^n = \\ &= h_{n+1} \sum_{i=1}^s b_i (d\mathbf{k}_i \wedge d\mathbf{Q}_i + d\mathbf{P}_i \wedge d\mathbf{l}_i) - h_{n+1}^2 \sum_{i, j=1}^s (b_i a_{ij} + b_j a_{ji} - b_i b_j) d\mathbf{k}_i \wedge d\mathbf{l}_j.\end{aligned}$$

Como antes, se obtiene que $d\mathbf{k}_i \wedge d\mathbf{Q}_i + d\mathbf{P}_i \wedge d\mathbf{l}_i = 0$, $\forall i = 1, \dots, s$, y por tanto:

$$h_{n+1}^2 \sum_{i, j=1}^s (b_i a_{ij} + b_j a_{ji} - b_i b_j) d\mathbf{k}_i \wedge d\mathbf{l}_j = 0.$$

Como la igualdad anterior se debe verificar para cualesquiera funciones f y g (es decir, independientemente de k_i y l_j), debe ser:

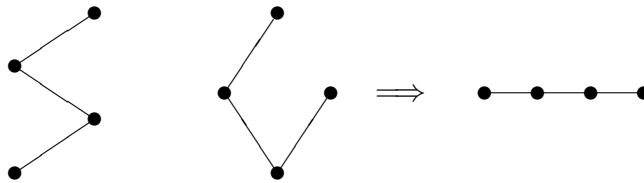
$$b_i a_{ij} + b_j a_{ji} - b_i b_j = 0, \text{ para } i, j = 1, \dots, s,$$

que es la condición del enunciado. \square

2.2.1. Condiciones de orden

Ya hemos visto las condiciones de orden de los métodos Runge-Kutta. Estas condiciones se imponen sobre el número de etapas y los coeficientes del método y, además, son independientes las unas de las otras, es decir, la verificación de la condición de orden para un árbol con raíz no es consecuencia de la verificación de la condición de orden para otros árboles. En el caso de los métodos Runge-Kutta simplécticos, los coeficientes del método están sujetos a la condición de simpléctico dada en el *teorema 2.8*. De hecho, algunas condiciones de orden dependerán de otras.

Observamos que, sin tener en cuenta qué vértice es la raíz, algunos árboles con raíz dan lugar al mismo árbol:



Un árbol sin raíz recibe el nombre de árbol libre o, directamente, árbol.

Sea t un árbol libre. Vamos a considerar para cada par de vértices adyacentes, i y j , los siguientes árboles con raíz, explicados con detalle en [1].

- El árbol con raíz rt_i , obtenido al fijar el vértice i como raíz en el árbol t . De manera análoga, consideramos el árbol con raíz rt_j , obtenido al fijar el vértice j como raíz en el árbol t .
- El árbol con raíz rt_I , obtenido al fijar el vértice i como raíz en el árbol t y eliminando el vértice j y las ramas correspondientes. De manera análoga, consideramos el árbol con raíz rt_J , obtenido al fijar el vértice j como raíz en el árbol t y eliminando el vértice i y las ramas correspondientes.

Con esta notación se tiene el siguiente resultado:

Lema 2.9. (Ver [2]).

$$\frac{1}{\gamma(rt_i)} + \frac{1}{\gamma(rt_j)} = \frac{1}{\gamma(rt_I)\gamma(rt_J)}$$

$$\Phi(rt_i) + \Phi(rt_j) = \Phi(rt_I)\Phi(rt_J).$$

Demostración. En primer lugar, recordemos la definición de la densidad de un árbol con raíz de manera recursiva (*definición 1.4*):

$$\gamma(\tau) = 1 \text{ y } \gamma([rt_1, \dots, rt_m]) = \rho([rt_1, \dots, rt_m])\gamma(rt_1) \cdot \dots \cdot \gamma(rt_m).$$

Por un lado:

$$\gamma(rt_i) = \rho(t)\gamma(rt_j) \prod_{rt \text{ subárbol de } rt_I} \gamma(rt).$$

Por otro lado:

$$\gamma(rt_I) = \rho(rt_I) \prod_{rt \text{ subárbol de } rt_I} \gamma(rt).$$

Luego:

$$\gamma(rt_i) = \rho(t)\gamma(rt_j) \frac{\gamma(rt_I)}{\rho(rt_I)} \Rightarrow \frac{1}{\gamma(rt_i)} = \frac{\rho(rt_I)}{\gamma(rt_I)\gamma(rt_j)\rho(t)}.$$

De igual modo:

$$\frac{1}{\gamma(rt_j)} = \frac{\rho(rt_I)}{\gamma(rt_I)\gamma(rt_j)\rho(t)}.$$

Luego, como $\rho(t) = \rho(rt_I) + \rho(rt_j)$, se tiene que:

$$\frac{1}{\gamma(rt_i)} + \frac{1}{\gamma(rt_j)} = \frac{\rho(rt_I) + \rho(rt_j)}{\gamma(rt_I)\gamma(rt_j)\rho(t)} = \frac{1}{\gamma(rt_I)\gamma(rt_j)}.$$

Veamos la otra igualdad. Recordemos la condición de simpléctico sobre los coeficientes de los métodos Runge-Kutta:

$$b_i a_{ij} + b_j a_{ji} - b_i b_j = 0 \text{ para } i, j = 1, \dots, s \text{ (Teorema 2.8)}.$$

Observamos que en $\Phi(rt_i)$ y en $\Phi(rt_j)$ aparecen ciertos factores iguales:

$$\Phi(rt_i) = \sum_{i,j,\dots} b_i a_{ij} \Pi \text{ y } \Phi(rt_j) = \sum_{i,j,\dots} b_j a_{ji} \Pi,$$

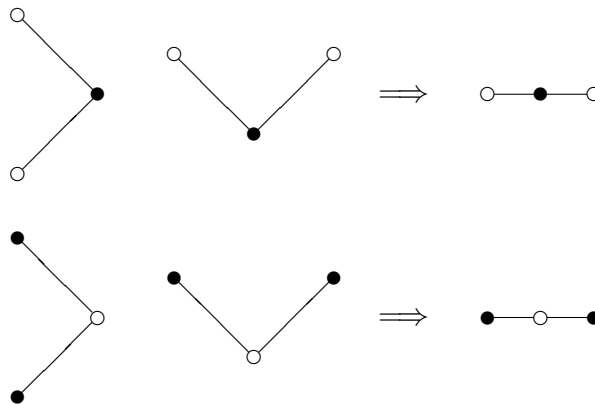
donde Π denota el producto de $\rho(t) - 2$ factores de la forma a_{mn} . Estos productos coinciden en ambas sumas ya que rt_i y rt_j solo difieren en los vértices i y j y, por tanto, sus pesos solo difieren en los coeficientes que involucran ambos vértices a la vez. Por tanto:

$$\begin{aligned} \Phi(rt_i) + \Phi(rt_j) &= \sum_{i,j,\dots} (b_i a_{ij} + b_j a_{ji}) \Pi = \sum_{i,j,\dots} b_i b_j \Pi = \sum_{i,j,\dots} (b_i \Pi_I) (b_j \Pi_J) \\ &= \left(\sum_{i,k,\dots} b_i \Pi_I \right) \left(\sum_{j,k,\dots} b_j \Pi_J \right) = \Phi(rt_I) \Phi(rt_J), \end{aligned}$$

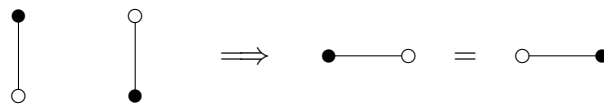
donde Π_I y Π_J denota el producto de los factores de la forma a_{mn} correspondientes a los árboles rt_I (es decir, los factores correspondientes al vértice i y a sus hijos, quitando el vértice j y sus hijos) y rt_J respectivamente. □

Al igual que para los árboles con raíz monocolor, se pueden obtener árboles bicolor sin raíz a partir de árboles bicolor con raíz. Estos árboles sin raíz reciben el nombre de árboles bicolor libres o, simplemente, árboles bicolor.

Puesto que estamos estudiando los métodos Runge-Kutta particionados aplicados a sistemas hamiltonianos separables, los únicos árboles bicolor que vamos a considerar son aquellos cuyos vértices consecutivos son siempre de distinto color.



Observamos que el árbol libre de orden 3 puede ser coloreado de dos formas distintas, pero hay otros que solo pueden ser coloreados de una manera:



Este árbol solo puede colorearse de una manera si se considera que vértices consecutivos no pueden ser del mismo color, dando lugar a una sola clase de árboles con raíz.

Teniendo en cuenta esta coloración surge un nuevo concepto:

Definición 2.10. Los árboles libres que solo pueden ser coloreados de una manera, estableciendo que dos vértices consecutivos son siempre de distinto color, se denominan árboles supérfluos. Además, se construyen árboles superfluos mediante la yuxtaposición de dos copias de un árbol con raíz, uniendo sus raíces mediante una nueva rama.

A consecuencia del lema anterior y teniendo en cuenta este nuevo concepto, podemos probar el siguiente teorema:

Teorema 2.11. *Sea $r \geq 2$. Consideramos un método Runge-Kutta simpléctico que satisface las condiciones de orden hasta $r - 1$. Para asegurar que tiene orden, al menos, r es suficiente ver que para cada árbol libre t no superfluo de r vértices existe un árbol rt con raíz asociado a t verificando la condición de orden. (Ver [8]).*

Demostración. Consideramos un método Runge-Kutta simpléctico que satisface las condiciones de orden hasta $r - 1$. Sea t un árbol libre. Sea rt_i el árbol con raíz obtenido fijando en t el vértice i como raíz que verifica la condición de orden, es decir:

$$\Phi(rt_i) = \frac{1}{\gamma(rt_i)}.$$

Queremos ver: $\Phi(rt) = \frac{1}{\gamma(rt)}$, para todo árbol con raíz rt obtenido al fijar en t un vértice como raíz. Sea j un vértice de t adyacente al vértice i . Utilizando la notación vista en el lema anterior, los árboles rt_I y rt_J tienen orden menor estrictamente que r y, por tanto, verifican la condición de orden:

$$\Phi(rt_I) = \frac{1}{\gamma(rt_I)} \text{ y } \Phi(rt_J) = \frac{1}{\gamma(rt_J)}.$$

Aplicando el lema anterior:

$$\Phi(rt_j) = \Phi(rt_I)\Phi(rt_J) - \Phi(rt_i) = \frac{1}{\gamma(rt_I)\gamma(rt_J)} - \frac{1}{\gamma(rt_i)} = \frac{1}{\gamma(rt_j)}.$$

De manera recursiva, se llega a que todo árbol con raíz obtenido a partir de t al fijar una raíz verifica la condición de orden, que es lo que se quería probar. \square

Este resultado permite reducir considerablemente las comprobaciones que hay que realizar para determinar el orden de un método Runge-Kutta simpléctico en comparación con un método Runge-Kutta no simpléctico. En general, para los métodos simplécticos hay una ecuación de orden por cada árbol libre no superfluo en vez de una por cada árbol con raíz.

Veamos como se obtienen las condiciones para cada posible orden r . Denotaremos por n y m al número de condiciones de orden de los métodos Runge-Kutta generales y al número de condiciones de orden de los métodos Runge-Kutta simplécticos respectivamente.

- Para $r = 1$, es claro que se tiene una condición tanto para simplécticos como para no simplécticos: $n = m = 1$. La condición que proporciona es la de consistencia:

$$\sum_i b_i = 1.$$

- Para $r = 2$, se requieren los árboles con raíz de dos nodos. Solo hay un árbol con raíz de orden 2 y proporciona una condición de orden para los métodos Runge-Kutta generales:

$$\sum_{i,j} b_i a_{ij} = \frac{1}{2}, \quad n = n + 1 = 2.$$

Este árbol con raíz da lugar a un árbol libre monocolor de orden 2: $\bullet\text{---}\bullet$

Ya vimos que este árbol libre es superfluo pues admite una única coloración. Por tanto, $m = m + 0 = 1$.

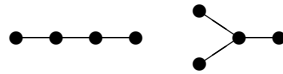
- Para $r = 3$, encontramos dos árboles con raíz de orden 3 que dan lugar a un mismo árbol libre: $\bullet\text{---}\bullet\text{---}\bullet$

Los dos árboles con raíz proporcionan dos condiciones de orden para los métodos Runge-Kutta generales:

$$\sum_{i,j,k} b_i a_{ij} a_{ik} = \frac{1}{3}, \quad \sum_{i,j,k} b_i a_{ij} a_{jk} = \frac{1}{6}, \quad n = n + 2 = 4.$$

El árbol libre que se obtiene a partir de los dos árboles de orden 3 posibles no es superfluo pues admite dos coloraciones. Por tanto, proporciona una condición de orden para los métodos Runge-Kutta simplécticos ($m = m + 1 = 2$), que se puede tomar como una cualquiera de las dos anteriores por el *teorema 2.11*.

- Para $r = 4$, encontramos cuatro árboles con raíz de orden 4. Por tanto, se añaden cuatro condiciones de orden más para los métodos Runge-Kutta generales: $n = n + 4 = 8$. De estos cuatro árboles con raíz se obtienen únicamente dos árboles libres monocolor distintos:



El primero de ellos es superfluo. El segundo no es superfluo, por lo que proporciona una condición de orden para los métodos Runge-Kutta simplécticos: $m = m + 1 = 3$. Están dibujados a continuación:

El número de condiciones que hay que comprobar para establecer que el método Runge-Kutta es, al menos, de orden r queda recogido en esta tabla para los primeros órdenes (ver [8]):

ORDEN	RK GENERAL	RK SIMPLÉCTICO
1	1	1
2	2	1
3	4	2
4	8	3
5	17	6
6	37	10
7	85	21
8	200	40

2.3. Métodos Runge-Kutta particionados simplécticos

Algunos sistemas hamiltonianos presentan un hamiltoniano con un aspecto de esta forma:

$$H(p, q, t) = T(p) + V(q, t),$$

donde T y V denotan las energías cinética y potencial respectivamente. Este tipo de sistemas hamiltonianos recibe el nombre de sistemas hamiltonianos separables (1.9).

En este caso, las ecuaciones de Hamilton que se obtienen son las siguientes:

$$\frac{dp}{dt} = f(q, t), \quad \frac{dq}{dt} = g(p),$$

donde $f = -\nabla_q V$ y $g = \nabla_p T$.

Ejemplo 2.12. El problema de Kepler

El problema de Kepler consiste en determinar el movimiento de dos partículas que interactúan entre sí por medio de una fuerza central que varía según la distancia entre ambas.

La energía cinética es: $T = \frac{1}{2}mv^2$, donde v denota el módulo de la velocidad.

La energía potencial es: $V = \frac{-k}{d}$, donde k es la interacción gravitatoria y d es la distancia entre los dos cuerpos.

Por simplificar consideraremos $m = 1$ y $k = 1$. Además, tomaremos la masa central en la posición $(0, 0)$. Bajo estas condiciones, la solución del problema describe el movimiento en el plano de una partícula atraída hacia el origen con una fuerza inversamente proporcional al cuadrado de la distancia. Consideramos $\mathbf{y} = (\mathbf{p}, \mathbf{q}) \in \mathbb{R}^4$, donde $\mathbf{q} = (q_1, q_2)$ denota la posición y $\mathbf{p} = (p_1, p_2)$ la velocidad de una partícula. Entonces: $p_i = \dot{q}_i$, para $i = 1, 2$, y

$$T = \frac{1}{2}(p_1^2 + p_2^2), \quad V = \frac{-1}{(q_1^2 + q_2^2)^{1/2}}.$$

El Lagrangiano es $L = T - V$. Los estados del sistema deben satisfacer las ecuaciones de Lagrange siguientes:

$$0 = L_{q_i} - \frac{d}{dt}L_{p_i} = \frac{-q_i}{(q_1^2 + q_2^2)^{3/2}} - \dot{p}_i, \quad \text{para } i = 1, 2.$$

De este modo se obtienen las ecuaciones del movimiento:

$$\begin{cases} \dot{q}_i = p_i, \text{ para } i = 1, 2. \\ \dot{p}_i = \frac{-q_i}{(q_1^2 + q_2^2)^{3/2}}, \text{ para } i = 1, 2. \end{cases}$$

El hamiltoniano del sistema es:

$$H = p_1 L_{p_1} + p_2 L_{p_2} - L = \frac{1}{2} (p_1^2 + p_2^2) + \frac{(-1)}{(q_1^2 + q_2^2)^{1/2}} = T + V.$$

Esta formulación del problema proporciona un sistema hamiltoniano separable. Además, el hamiltoniano coincide con la energía total del sistema. Debido al carácter central de la fuerza, también se conserva el momento angular: $M = p_1 q_2 - q_1 p_2$.

Para analizar el sistema se suelen utilizar coordenadas polares (r, θ) en el plano (p_1, p_2) :

$$r^2 = p_1^2 + p_2^2, \quad p_1 = r \cos(\theta), \quad p_2 = r \sin(\theta).$$

Luego: $q_1^2 + q_2^2 = \dot{r}^2 + r^2 \dot{\theta}^2$. En estas coordenadas el hamiltoniano es:

$$H = \frac{1}{2} (\dot{r}^2 + r^2 \dot{\theta}^2) + \frac{(-1)}{r}.$$

Los momentos son:

$$p_r = \frac{\partial L}{\partial \dot{r}} = \dot{r}, \quad p_\theta = \frac{\partial L}{\partial \dot{\theta}} = r^2 \dot{\theta}$$

Por tanto, las ecuaciones del sistema son:

$$\begin{cases} \dot{r} = p_r \\ \dot{\theta} = \frac{p_\theta}{r^2} \\ \dot{p}_r = -H_r = \frac{p_\theta^2}{r^3} - V'(r) \\ \dot{p}_\theta = -H_\theta = 0 \end{cases} \quad (2.3)$$

Además, observamos:

$$\theta = \arctg\left(\frac{p_2}{p_1}\right) \Rightarrow \dot{\theta} = \frac{M}{r^2} \Rightarrow p_\theta = M \text{ es constante}$$

Reescribimos el hamiltoniano:

$$H = \frac{1}{2} \left(\dot{r}^2 + \frac{M^2}{r^2} \right) + V(r).$$

Como H es constante a lo largo del tiempo, podemos despejar \dot{r} :

$$\dot{r} = \sqrt{\left(H - \frac{M^2}{2r^2} - V(r)\right)} \Rightarrow dt = \frac{dr}{\sqrt{\left(H - \frac{M^2}{2r^2} - V(r)\right)}}$$

donde H denota el valor constante del hamiltoniano. Integrando la expresión anterior se obtiene el periodo de r .

Estos sistemas hamiltonianos pueden integrarse con los métodos Runge-Kutta particionados que hemos visto anteriormente.

El método Runge-Kutta particionado dado por el tablero de Butcher de la *figura (1.2)* aplicado al sistema Hamiltoniano separable (1.9) proporciona las siguientes relaciones:

$$\left\{ \begin{array}{l} \mathbf{P}_i \\ \mathbf{Q}_i \\ \mathbf{p}^{n+1} \\ \mathbf{q}^{n+1} \end{array} \right. = \left\{ \begin{array}{l} \mathbf{p}^n + h_{n+1} \sum_{j=1}^s a_{ij} \mathbf{f}(\mathbf{Q}_j, t_n + C_j h_{n+1}) \\ \mathbf{q}^n + h_{n+1} \sum_{j=1}^s A_{ij} \mathbf{g}(\mathbf{P}_j) \\ \mathbf{p}^n + h_{n+1} \sum_{i=1}^s b_i \mathbf{f}(\mathbf{Q}_i, t_n + C_i h_{n+1}) \\ \mathbf{q}^n + h_{n+1} \sum_{i=1}^s B_i \mathbf{g}(\mathbf{P}_i) \end{array} \right. \quad (2.4)$$

donde $\mathbf{f} = \left(-\frac{\partial H}{\partial q_i}\right)_{i=1}^s = -\nabla_{\mathbf{q}} V$ y $\mathbf{g} = \left(\frac{\partial H}{\partial p_i}\right)_{i=1}^s = \nabla_{\mathbf{p}} T$, siendo \mathbf{P}_i y \mathbf{Q}_i pasos intermedios, con $C_i = \sum_{j=1}^s A_{ij}$.

Un resultado similar al *teorema 2.8* se tiene para los métodos Runge-Kutta particionados, dotando dichos métodos de carácter simpléctico o no simpléctico:

Teorema 2.13. *Supongamos que los coeficientes del método Runge-Kutta particionado satisfacen la siguiente relación:*

$$b_i A_{ij} + B_j a_{ji} - b_i B_j = 0, \text{ para } i, j = 1, \dots, s.$$

Entonces el método es simpléctico cuando se aplica a sistemas hamiltonianos separables. El recíproco también es cierto. (Ver [8]).

Demostración. Como en el *teorema 2.8*, suponiendo que se verifica la condición del enunciado, bastará probar:

$$d\mathbf{p}^{n+1} \wedge d\mathbf{q}^{n+1} - d\mathbf{p}^n \wedge d\mathbf{q}^n = 0,$$

para ver que el método es simpléctico. Denotaremos:

$$\begin{cases} \mathbf{k}_i &= \mathbf{f}(\mathbf{Q}_i, t_n + c_i h_{n+1}), \\ \mathbf{l}_i &= \mathbf{g}(\mathbf{P}_i). \end{cases}$$

Razonando de manera análoga a la demostración del *teorema 2.8* y aplicando la hipótesis del enunciado del teorema, se tiene:

$$\begin{aligned} d\mathbf{p}^{n+1} \wedge d\mathbf{q}^{n+1} - d\mathbf{p}^n \wedge d\mathbf{q}^n &= \\ &= h_{n+1} \sum_{i=1}^s b_i d\mathbf{k}_i \wedge d\mathbf{q}^n + h_{n+1} \sum_{j=1}^s B_j d\mathbf{p}^n \wedge d\mathbf{l}_j + h_{n+1}^2 \sum_{i,j=1}^s b_i B_j d\mathbf{k}_i \wedge d\mathbf{l}_j = \\ &= h_{n+1} \sum_{i=1}^s (b_i d\mathbf{k}_i \wedge d\mathbf{Q}_i + B_i d\mathbf{P}_i \wedge d\mathbf{l}_i) - h_{n+1}^2 \sum_{i,j=1}^s (b_i A_{ij} + B_j a_{ji} - b_i B_j) d\mathbf{k}_i \wedge d\mathbf{l}_j = \\ &= h_{n+1} \sum_{i=1}^s (b_i d\mathbf{k}_i \wedge d\mathbf{Q}_i + B_i d\mathbf{P}_i \wedge d\mathbf{l}_i) \end{aligned}$$

Para finalizar basta ver que, para cada $i = 1, \dots, s$, se tiene: $d\mathbf{k}_i \wedge d\mathbf{Q}_i + d\mathbf{P}_i \wedge d\mathbf{l}_i = 0$. Teniendo en cuenta que:

$$dk_\alpha = \frac{\partial f_\alpha}{\partial q_\beta} dQ_\beta,$$

se tiene que:

$$\left(\frac{\partial f_\alpha}{\partial q_\beta} \right) dQ_\beta \wedge dQ_\alpha + \left(\frac{\partial f_\beta}{\partial q_\alpha} \right) dQ_\alpha \wedge dQ_\beta = \left(-\frac{\partial^2 H}{\partial q_\alpha \partial q_\beta} \right) dQ_\beta \wedge dQ_\alpha + \left(-\frac{\partial^2 H}{\partial q_\alpha \partial q_\beta} \right) dQ_\alpha \wedge dQ_\beta = 0,$$

y, por tanto:

$$d\mathbf{k} \wedge d\mathbf{Q} = \sum_{\alpha=1}^d dk_\alpha \wedge dQ_\alpha = \sum_{\alpha=1}^d \left(\frac{\partial f_\alpha}{\partial q_\beta} \right) dQ_\beta \wedge dQ_\alpha = 0.$$

De manera análoga se prueba que $d\mathbf{P} \wedge d\mathbf{l} = 0$. En consecuencia, $d\mathbf{p}^{n+1} \wedge d\mathbf{q}^{n+1} - d\mathbf{p}^n \wedge d\mathbf{q}^n = 0$ y, por tanto, el método es simpléctico.

Supongamos ahora que el método es simpléctico, es decir:

$$\begin{aligned} 0 &= d\mathbf{p}^{n+1} \wedge d\mathbf{q}^{n+1} - d\mathbf{p}^n \wedge d\mathbf{q}^n = \\ &= h_{n+1} \sum_{i=1}^s (b_i d\mathbf{k}_i \wedge d\mathbf{Q}_i + B_i d\mathbf{P}_i \wedge d\mathbf{l}_i) - h_{n+1}^2 \sum_{i,j=1}^s (b_i A_{ij} + B_j a_{ji} - b_i B_j) d\mathbf{k}_i \wedge d\mathbf{l}_j \end{aligned}$$

Hemos visto que, para cada $i = 1, \dots, s$, se tiene:

$$d\mathbf{k}_i \wedge d\mathbf{Q}_i + d\mathbf{P}_i \wedge d\mathbf{l}_i = 0.$$

Por tanto:

$$h_{n+1}^2 \sum_{i,j=1}^s (b_i A_{ij} + B_j a_{ji} - b_i B_j) d\mathbf{k}_i \wedge d\mathbf{l}_j = 0.$$

Como la igualdad anterior se debe verificar para cuales quiera funciones f y g (que solo involucren a p o a q respectivamente), entonces debe ser:

$$b_i A_{ij} + B_j a_{ji} - b_i B_j = 0, \text{ para } i, j = 1, \dots, s,$$

que es la condición del enunciado. □

2.3.1. Condiciones de orden

Se tiene un resultado similar al visto para métodos Runge-Kutta simplécticos generales pero para métodos particionados:

Teorema 2.14. *Sea $r \geq 2$. Consideramos un método Runge-Kutta simpléctico particionado que satisface las condiciones de orden hasta $r - 1$. Para asegurar que tiene orden, al menos, r es suficiente ver que para cada árbol libre bt no de r vértices existe un árbol bicolor brt con raíz asociado a bt verificando la condición de orden. (Ver [8]).*

La demostración es análoga a la del *teorema 2.11*. Al igual que dicho teorema, reduce notablemente el número de condiciones de orden.

En el caso de los no particionados los árboles superfluos no proporcionan ninguna condición de orden mientras que en los Runge-Kutta particionados los árboles bicolor resultantes de colorear árboles superfluos si que proporcionan una condición de orden.

Siguiendo un razonamiento similar al visto para métodos Runge-Kutta, se puede calcular el número de condiciones de orden de métodos Runge-Kutta particionados (n) y de métodos Runge-Kutta particionados simplécticos (m):

- Para obtener orden $r = 1$, se requieren dos condiciones de orden para ambos casos pues hay dos árboles bicolor distintos: $n = m = 2$.
- Para $r = 2$, encontramos dos árboles bicolor con raíz distintos que proporcionan dos condiciones de orden para los métodos Runge-Kutta particionados:

$$\sum_{i,j} b_i A_{i,j} = \frac{1}{2}, \quad \sum_{i,j} B_i a_{i,j} = \frac{1}{2}, \quad n = n + 2 = 4.$$

Estos dos árboles dan lugar a un mismo árbol libre bicolor que proporciona una sola condición de orden ($m = m + 1 = 3$). Dicha condición se puede elegir entre las dos anteriores.

$$\begin{array}{l}
 \begin{array}{c} \bullet \\ \diagdown \\ \bullet \end{array} \Rightarrow \sum_{i,j,k} b_i A_{i,j} a_{j,k} = \frac{1}{6}. \\
 \begin{array}{c} \bullet \quad \bullet \\ \diagdown \quad \diagup \\ \circ \end{array} \Rightarrow \sum_{i,j,k} B_i a_{i,j} a_{i,k} = \frac{1}{3}. \\
 \begin{array}{c} \circ \\ \diagdown \\ \bullet \end{array} \Rightarrow \sum_{i,j,k} b_i A_{i,j} a_{j,k} = \frac{1}{6}. \\
 \begin{array}{c} \circ \quad \circ \\ \diagdown \quad \diagup \\ \bullet \end{array} \Rightarrow \sum_{i,j,k} b_i A_{i,j} A_{i,k} = \frac{1}{3}.
 \end{array}$$

- Para $r = 3$, encontramos cuatro árboles bicolor que proporcionan una condición de orden cada uno para los métodos Runge-Kutta particionados generales:

De estos árboles con raíz se obtienen dos árboles libre bicolor, que proporcionan dos condiciones de orden. Los dos primeros árboles bicolor con raíz dan lugar al mismo libre por lo que para el caso simpléctico se impone una de las dos condiciones primeras anteriormente formuladas. Del mismo modo, se impone una de las dos últimas.

El número de condiciones que hay que comprobar para establecer que el método Runge-Kutta particionado (PRK) es, al menos, de orden r queda recogido en esta tabla para los primeros órdenes (ver [8]):

ORDEN	PRK GENERAL	PRK SIMPLÉCTICO
1	2	2
2	4	3
3	8	5
4	16	8
5	34	14
6	74	24
7	170	46
8	400	88

2.3.2. Ejemplos de métodos Runge-Kutta particionados simplécticos

A pesar de que los métodos Runge-Kutta particionados solo se pueden aplicar a sistemas hamiltonianos separables, son muy interesantes. Sobre todo desde el punto de vista

práctico debido a que son métodos explícitos.

Métodos de Euler simplécticos

Consideramos un sistema hamiltoniano separable:

$$\begin{cases} \dot{p} = f(q) \\ \dot{q} = g(p) \end{cases}$$

El método Runge-Kutta particionado de Euler visto en los ejemplos de la sección 1.4.1 aplicado sobre el sistema anterior, viene dado por las siguientes expresiones:

$$\begin{cases} p_{n+1} = p_n + hf(q_n) \\ q_{n+1} = q_n + hf(p_{n+1}) \end{cases}$$

Los coeficientes de este método Runge-Kutta particionado son: $a_1 = b_1 = B_1 = 1$ y $A_1 = 0$. Es claro que verifican la condición de simpléctico del *teorema 2.13* y, por consiguiente, es un método Runge-Kutta particionado simpléctico al aplicarse sobre métodos Hamiltonianos separables.

Métodos de Ruth

Vamos a considerar la siguiente familia de métodos, denominados métodos de Ruth:

$$\left| \begin{array}{cccc} 0 & 0 & 0 & \dots & 0 \\ b_1 & 0 & 0 & \dots & 0 \\ b_1 & b_2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_1 & b_2 & b_3 & \dots & 0 \end{array} \right| \quad \left| \begin{array}{cccc} B_1 & 0 & 0 & \dots & 0 \\ B_1 & B_2 & 0 & \dots & 0 \\ B_1 & B_2 & B_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ B_1 & B_2 & B_3 & \dots & B_s \end{array} \right|$$

$$\left| \begin{array}{cccc} b_1 & b_2 & b_3 & \dots & b_s \end{array} \right| \quad \left| \begin{array}{cccc} B_1 & B_2 & B_3 & \dots & B_s \end{array} \right|$$

Figura 2.1: Método de Ruth

Se representa de esta manera: $(b_1, \dots, b_s) [B_1, \dots, B_s]$. En primer lugar, se escriben los b_i y después los B_i , poniendo entre paréntesis los coeficientes correspondientes a un método con diagonal nula y entre corchetes el restante.

La implementación de un paso de este método es como sigue:

$$\begin{aligned}
 Q_0 &= q^n \\
 P_1 &= p^n \\
 \text{Para } i &= 1, \dots, s \\
 Q_i &= Q_{i-1} + h_{n+1} B_i g(P_i) \\
 P_{i+1} &= P_i + h_{n+1} b_i f(Q_i, t_n + C_i h_{n+1}) \\
 q^{n+1} &= Q_s \\
 p^{n+1} &= P_{s+1}
 \end{aligned}$$

Esta notación da lugar a una forma alternativa:

$$\begin{array}{c|cccc}
 & b_1 & 0 & 0 & \dots & 0 \\
 & b_1 & b_2 & 0 & \dots & 0 \\
 & b_1 & b_2 & b_3 & \dots & 0 \\
 & \cdot & \cdot & \cdot & \cdot & \cdot \\
 & \cdot & \cdot & \cdot & \cdot & \cdot \\
 & b_1 & b_2 & b_3 & \dots & b_s \\
 \hline
 & b_1 & b_2 & b_3 & \dots & b_s
 \end{array}
 \quad
 \begin{array}{c|cccc}
 & 0 & 0 & 0 & \dots & 0 \\
 & B_1 & 0 & 0 & \dots & 0 \\
 & B_1 & B_2 & 0 & \dots & 0 \\
 & \cdot & \cdot & \cdot & \cdot & \cdot \\
 & \cdot & \cdot & \cdot & \cdot & \cdot \\
 & B_1 & B_2 & B_3 & \dots & 0 \\
 \hline
 & B_1 & B_2 & B_3 & \dots & B_s
 \end{array}$$

Figura 2.2: $[b_1, \dots, b_s] (B_1, \dots, B_s)$

La implementación de un paso del método en esta forma alternativa es como sigue:

$$\begin{aligned}
 P_0 &= p^n \\
 Q_1 &= q^n \\
 \text{Para } i &= 1, \dots, s \\
 P_i &= P_{i-1} + h_{n+1} b_i f(Q_i, t_n + C_i h_{n+1}) \\
 Q_{i+1} &= Q_i + h_{n+1} B_i g(P_i) \\
 p^{n+1} &= P_s \\
 q^{n+1} &= Q_{s+1}
 \end{aligned}$$

En la forma habitual (2.1), se tiene:

$$a_{ij} = \begin{cases} 0, & i \leq j \\ b_j, & i > j \end{cases}, \quad A_{ij} = \begin{cases} 0, & i < j \\ B_j, & i \geq j \end{cases}$$

Con estos coeficientes, observamos que se verifica la condición de método simpléctico:

$$\begin{aligned}
 b_i A_{ij} + B_j a_{ji} - b_i B_j &= B_j a_{ji} - b_i B_j = B_j b_i - b_i B_j = 0, \text{ si } i < j \\
 b_i A_{ij} + B_j a_{ji} - b_i B_j &= b_i A_{ij} - b_i B_j = b_i B_j - b_i B_j = 0, \text{ si } i \geq j
 \end{aligned}$$

Algunos de los métodos de Ruth con coeficientes particulares serán presentados más adelante en los experimentos numéricos.

Método Störmer-Verlet

Retomamos ahora otro ejemplo visto en la sección 1.4.1: el método Störmer-Verlet. Este método se aplica a sistemas Hamiltonianos separables de la forma $\{\dot{p} = f(q), \dot{q} = p\}$. Vimos que la formulación del método Störmer-Verlet de un paso viene dada por las siguientes ecuaciones:

$$\begin{cases} p_{n+1/2} = p_n + \frac{h}{2}f(q_n) \\ q_{n+1} = q_n + hp_{n+1/2} \\ p_{n+1} = p_{n+1/2} + \frac{h}{2}f(q_{n+1}) \end{cases}$$

Es un método Runge-Kutta particionado con coeficientes:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1/2 & 1/2 \\ \hline & 1/2 & 1/2 \end{array} \quad \begin{array}{c|cc} 1/2 & 1/2 & 0 \\ 1/2 & 1/2 & 0 \\ \hline & 1/2 & 1/2 \end{array}$$

Veamos que se verifica la condición de simpléctico al aplicarse sobre un sistema Hamiltoniano separable. Observamos: $1/2 = a_{ji} \neq 0 \Leftrightarrow j = 2 \Leftrightarrow A_{ij} = 0$. Además, $b_i = B_i = 1/2$, para $i = 1, 2$. Por tanto:

$$b_i A_{ij} + B_j a_{ji} - b_i B_j = b_i B_j - b_i B_j = 0, \forall i = 1, 2$$

Capítulo 3

Métodos de composición

En este capítulo se presenta el segundo objeto de estudio principal de esta memoria: los métodos de composición. A pesar de que no se puede establecer una teoría general de orden de métodos de composición, se utilizarán, una vez más, los conceptos de árbol con raíz y B-serie para determinar condiciones de orden para ciertos métodos de composición.

Se introduce el concepto de adjunto de un método. La composición de un método con su adjunto resulta en métodos para los que, bajo ciertas condiciones, se puede determinar el orden. De hecho, permite construir métodos de orden tan alto como se desee mediante la composición.

Finalmente, se desarrolla la teoría necesaria para determinar el orden de este tipo de métodos y reducir lo máximo posible el número de condiciones de orden requeridas.

3.1. Definiciones

Consideramos dos métodos numéricos $\Psi_F^1(t_{n+1}, t_n)$ y $\Psi_F^2(t_{n+1}, t_n)$ y un parámetro real θ . La concatenación de un paso de longitud θh_{n+1} en el primer método con un paso de longitud $(1 - \theta)h_{n+1}$ en el segundo método (es decir, la aplicación $\Psi_F^2(t_{n+1}, \delta) \circ \Psi_F^1(\delta, t_n)$ con $\delta = \theta t_{n+1} + (1 - \theta)t_n$) define un método numérico $\Psi_F(t_{n+1}, t_n)$ denominado método de composición de los métodos Ψ^1 y Ψ^2 .

Se parte de un valor inicial y_0 . Se obtiene y_1 aplicando el primer método Ψ^1 con el paso correspondiente y, finalmente, se calcula y_2 aplicando a y_1 el segundo método Ψ^2 con el paso adecuado. Este valor y_2 es la aproximación a y_0 en un paso mediante el método de composición de Ψ^1 y Ψ^2 .

El concepto de composición de dos métodos se puede extender a la composición de una cantidad finita de métodos.

3.1.1. Composición de métodos Runge-Kutta y de B-series

Como hemos comentado anteriormente, no existe una teoría de orden que generalice cualquier composición de métodos. En el caso de la composición de métodos Runge-Kutta, veremos que el método resultante es un método conocido y, por ello, no es necesario presentar una nueva teoría de orden.

Sean Ψ^1 y Ψ^2 dos métodos Runge-Kutta de s y t etapas con coeficientes a_{ij}^1, b_i^1 y a_{ij}^2, b_i^2 respectivamente.

La composición de estos dos métodos Runge-Kutta da lugar a un nuevo método Runge-Kutta de $s + t$ etapas. Es por ello que las condiciones de orden para estos métodos se derivan de forma directa de las condiciones de orden de métodos Runge-Kutta generales. El siguiente tablero de Butcher representa a este nuevo método Runge-Kutta de $s + t$ etapas:

$$\begin{array}{c|ccc|ccc}
 \theta a_{11}^1 & \cdots & \theta a_{1s}^1 & & 0 & & 0 \\
 \vdots & & \vdots & & & & \\
 \vdots & & \vdots & & & & \\
 \theta a_{s1}^1 & \cdots & \theta a_{ss}^1 & & 0 & & 0 \\
 \hline
 \theta b_1^1 & \cdots & \theta b_s^1 & (1-\theta)a_{11}^2 & \cdots & (1-\theta)a_{1t}^2 & \\
 \vdots & & \vdots & \vdots & \ddots & \vdots & \\
 \vdots & & \vdots & \vdots & \ddots & \vdots & \\
 \theta b_1^1 & \cdots & \theta b_s^1 & (1-\theta)a_{t1}^2 & \cdots & (1-\theta)a_{tt}^2 & \\
 \hline
 \theta b_1^1 & \cdots & \theta b_s^1 & (1-\theta)b_1^2 & \cdots & (1-\theta)b_t^2 &
 \end{array}$$

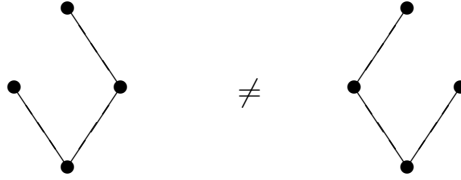
Los pesos elementales de este método Runge-Kutta obtenido como composición se pueden calcular a partir de los pesos elementales asociados a los dos métodos Runge-Kutta elementales mediante una fórmula conocida como “*Regla general de composición*”. Como veremos más adelante, esta fórmula se puede utilizar a su vez para expresar los coeficientes de la B-serie obtenida al componer dos B-series en función de los coeficientes de las mismas. Para ello, se requiere un nuevo conjunto de árboles: los árboles ordenados. La teoría que se desarrolla a continuación se encuentra recogida en [3], [4] y [5].

Definición 3.1. El conjunto de árboles (con raíz) ordenados, denotado por OT , se construye de forma recursiva:

1. El árbol τ de un único nodo pertenece al conjunto de árboles ordenados: $\tau \in OT$.
2. Si $\omega_1, \dots, \omega_m \in OT$, entonces la m -tupla ordenada $(\omega_1, \dots, \omega_m)$ también es un árbol ordenado.

Los árboles ordenados no son más que árboles con raíz en los que el orden de las ramas salientes no puede ser permutado.

Por ejemplo, en este conjunto OT los árboles siguientes son árboles distintos:



Son dos ordenaciones de un árbol con raíz del conjunto RT . Por ello, los árboles de RT se pueden considerar como clases de equivalencia de árboles ordenados, es decir:

$$rt = \bar{\omega}, \text{ con } rt \in RT \text{ y } \omega \in OT.$$

El número de posibles ordenaciones de un árbol $rt \in RT$ se denota $\nu(rt)$. No es más que el número de árboles ordenados que pertenecen a la clase de equivalencia rt . Este valor se puede obtener de manera recursiva:

$$\begin{aligned} \nu(\tau) &= 1 \\ \nu(rt) &= \frac{m!}{\mu_1! \mu_2! \dots} \nu(rt_1) \cdot \dots \cdot \nu(rt_m), \text{ si } rt = [rt_1, \dots, rt_m] \in RT, \end{aligned}$$

donde μ_1, μ_2, \dots son el número de árboles iguales entre $rt_1, \dots, rt_m \in RT$.

Este concepto está estrechamente relacionado con el coeficiente simétrico (*Definición 1.11*). De hecho, se satisface la siguiente relación:

$$\kappa(rt) = \sigma(rt) \cdot \nu(rt), \forall rt \in RT,$$

donde $\kappa(\tau) = 1$ y $\kappa(rt) = m! \kappa(rt_1) \cdot \dots \cdot \kappa(rt_m)$, si $rt = [rt_1, \dots, rt_m] \in RT$.

Es necesario que introduzcamos el conjunto de los subárboles ordenados de un árbol $\omega \in OT$, denotado $OST(\omega)$. Este conjunto se construye de la siguiente manera:

1. El conjunto de subárboles ordenados del árbol ordenado de un único nodo es:

$$OST(\tau) = \{\emptyset, \tau\}.$$

2. El conjunto de subárboles ordenados del árbol ordenado $\omega = (\omega_1, \dots, \omega_m) \in OT$ es:

$$OST(\omega) = \{\emptyset\} \cup \{(\theta_1, \dots, \theta_m) : \theta_i \in OST(\omega_i)\}.$$

Cada subárbol $\theta \in OST(\omega)$ está asociado de manera natural con un árbol $\bar{\theta} \in RT$ obtenido al no tener en cuenta el orden de sus ramas. Para cada árbol $rt \in RT$ elegimos un orden que denotaremos $\omega(rt)$ y vamos a considerar: $OST(rt) := OST(\omega(rt))$. Para cada $\theta \in OST(\omega)$, denotamos por $\omega \setminus \theta$ al "bosque" formado por los árboles resultantes al eliminar el subárbol θ del árbol ordenado $\omega \in OT$. Se va a considerar: $rt \setminus \theta := \omega(rt) \setminus \theta$. Mediante los conceptos que acabamos de describir, se enuncian los siguientes resultados, recogidos en [3].

Teorema 3.2. *Sean Φ_1 y Φ_2 los pesos elementales asociados a los dos métodos Runge-Kutta que se desea componer. Bajo las convenciones $\Phi_2(\theta) = \Phi_2(\bar{\theta})$ y $\Phi_2(\emptyset) = 1$, se tiene una regla general para determinar los pesos elementales asociados a la composición de dos métodos Runge-Kutta:*

$$\widehat{\Phi}(rt) = \sum_{\theta \in OST(rt)} \left(\Phi_2(\theta) \cdot \prod_{\delta \in rt \setminus \theta} \Phi_1(\delta) \right).$$

Esta composición se puede extender a la composición de B-series, es decir, introducir una B-serie en otra. Esto nos permite generalizar el lema (1.13).

Teorema 3.3. *Sean $a : RT \cup \{\emptyset\} \rightarrow \mathbb{R}$ con $a(\emptyset) = 1$ y $b : RT \cup \{\emptyset\} \rightarrow \mathbb{R}$ dos aplicaciones. Entonces la B-serie $B(a, y)$ insertada en $B(b, \cdot)$ es de nuevo una B-serie:*

$$B(b, B(a, y)) = B(ab, y),$$

donde $ab : RT \cup \{\emptyset\} \rightarrow \mathbb{R}$ se obtiene mediante la regla general anterior (3.2). Es decir;

$$ab(rt) = \sum_{\theta \in OST(rt)} b(\theta) \cdot a(rt \setminus \theta), \text{ con } a(rt \setminus \theta) = \prod_{\delta \in rt \setminus \theta} a(\delta).$$

3.2. Método adjunto

Definición 3.4. Dado un método Ψ , su método adjunto será un método $\bar{\Psi}$ tal que $\bar{\Psi}(t_n, t_{n+1}) \circ \Psi(t_{n+1}, t_n)$ es la identidad. Formalmente, el método adjunto de un método es la aplicación inversa del método original con paso opuesto, es decir: $\bar{\Psi}_h := \Psi_{-h}^{-1}$.

Observación 3.5. Avanzar un paso del método original equivale a dar un paso hacia atrás en el adjunto.

La consideración de los métodos adjuntos apareció de manera independiente en el estudio de métodos simétricos (Stetter (1973), p. 125, Wanner (1973)) y en los trabajos realizados para construir métodos para problemas rígidos a partir de métodos explícitos.

Proposición 3.6. *Sean Ψ_h y Φ_h dos métodos numéricos. Se tienen las siguientes propiedades:*

- $\overline{\overline{\Psi}}_h = \Psi_h$, es decir, el adjunto del adjunto es el método original.
- $\overline{\Psi}_h \circ \overline{\Phi}_h = \overline{\Phi}_h \circ \overline{\Psi}_h$, es decir, el adjunto de un método de composición se obtiene componiendo los métodos adjuntos de los factores de la composición en orden inverso. (Ver [8]).

La segunda propiedad dada no es más que la aplicación de la regla general para calcular el inverso de la composición de aplicaciones.

Definición 3.7. Si el método adjunto es el propio método, se dice que el método original es simétrico.

Observación 3.8. De esta definición se sigue que la composición de un paso de longitud $h/2$ de un método Ψ con un paso de longitud $h/2$ de su adjunto $\overline{\Psi}$ proporciona un método simétrico.

Ejemplo 3.9. Veamos que el método implícito de Euler es el adjunto del método explícito de Euler. Aplicamos el método explícito de Euler de paso $h = t_{n+1} - t_n$:

$$y_{n+1} = y_n + hf(y_n, t_n) \Rightarrow y_n = y_{n+1} + (-h)f(y_n, t_{n+1} - h).$$

Con la segunda igualdad vemos que y_n se obtiene a partir de y_{n+1} aplicando el método de Euler implícito con paso $-h = t_n - t_{n+1}$. En consecuencia, el método de Euler implícito es el adjunto del método de Euler explícito y viceversa.

Ejemplo 3.10. Veamos como obtener el adjunto de los métodos Runge-Kutta. Consideramos un método Runge-Kutta de s etapas y con coeficientes a_{ij} , b_i . Su método adjunto será un método Runge-Kutta de s etapas con coeficientes \overline{a}_{ij} , \overline{b}_i dados por

$$\overline{a}_{ij} = b_{s+1-j} - a_{s+1-i, s+1-j}, \quad \overline{b}_i = b_{s+1-j} \quad (\text{ver [8]}).$$

Además, si $a_{s+1-i, s+1-j} + a_{ij} = b_j$, entonces el método Runge-Kutta será simétrico ya que esta condición implica: $\overline{a}_{ij} = a_{ij}$ y $\overline{b}_i = b_i$, $\forall i, j$. De manera análoga, el método adjunto de un método Runge-Kutta particionado de s etapas será también un método Runge-Kutta particionado de s etapas y sus coeficientes se obtendrán con la fórmula anterior (ver [8]):

$$\begin{aligned} \overline{a}_{ij} &= b_{s+1-j} - a_{s+1-i, s+1-j}, & \overline{b}_i &= b_{s+1-j} \\ \overline{A}_{ij} &= B_{s+1-j} - A_{s+1-i, s+1-j}, & \overline{B}_i &= B_{s+1-j} \end{aligned}$$

El siguiente resultado muestra que el método adjunto tiene el mismo orden que el método original y, salvo un posible cambio de signo, el mismo término líder en el error.

Teorema 3.11. *Consideramos el sistema diferencial autónomo*

$$\begin{cases} \dot{y} = f(y) \\ y(t_0) = y_0 \end{cases}$$

Denotamos por φ_t al operador solución o flujo del sistema hamiltoniano anterior que proporciona el valor de la solución del sistema al transcurrir un tiempo t desde t_0 , es decir: $\varphi_t(y_0) = y(t + t_0)$. Sea Ψ_h un método de un paso de orden p que satisface:

$$\Psi_h(y_0) = \varphi_h(y_0) + C(y_0)h^{p+1} + O(h^{p+2})$$

Entonces su método adjunto $\overline{\Psi}_h$ tiene orden p y satisface:

$$\overline{\Psi}_h(y_0) = \varphi_h(y_0) + (-1)^p C(y_0)h^{p+1} + O(h^{p+2}).$$

Además, si el método Ψ_h es simétrico su orden debe ser par. (Ver [3] y [4].)

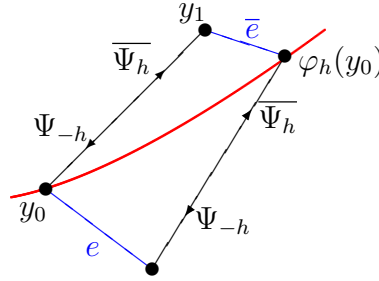


Figura 3.1: Ver [3].

Demostración. Partimos de un valor inicial y_0 . Computamos $\varphi_h(y_0)$ e $y_1 = \overline{\Psi}_h(y_0)$. La diferencia entre estos dos valores es el error local \bar{e} del método $\overline{\Psi}_h$: $\bar{e} = y_1 - \varphi_h(y_0)$. Ahora aplicamos Ψ_{-h} : $\Psi_{-h}(\bar{e}) = e$, siendo e el error local de Ψ_h . Observamos que $-e$ es el error local del método Ψ_{-h} . Por tanto, aplicando la hipótesis del enunciado, se tiene que:

$$\begin{aligned} -e &= \Psi_{-h}(\varphi_h(y_0)) - y_0 = \\ &= (y_0 + C(\varphi_h(y_0))(-h)^{p+1} + O(h^{p+2})) - y_0 = \\ &= (-1)^{p+1}C(\varphi_h(y_0))h^{p+1} + O(h^{p+2}) \\ &\Rightarrow e = (-1)^p C(\varphi_h(y_0))h^{p+1} + O(h^{p+2}) \end{aligned}$$

Puesto que $\varphi_h(y_0) = y_0 + O(h)$ y $e = (I + O(h))\bar{e}$, se sigue que:

$$\begin{aligned} (I + O(h))\bar{e} &= e = (-1)^p C(\varphi_h(y_0))h^{p+1} + O(h^{p+2}) = \\ &= (-1)^p C(y_0 + O(h))h^{p+1} + O(h^{p+2}) = \\ &= (I + O(h)) \left((-1)^p C(y_0)h^{p+1} + O(h^{p+2}) \right) \\ &\Rightarrow \bar{e} = (-1)^p C(y_0)h^{p+1} + O(h^{p+2}) \end{aligned}$$

Por tanto, teniendo en cuenta que \bar{e} es el error local del método $\bar{\Psi}_h$, se concluye que:

$$\bar{\Psi}_h(y_0) = \bar{e} + \varphi_h(y_0)\varphi_h(y_0) + (-1)^p C(y_0)h^{p+1} + O(h^{p+2}).$$

que es lo que se quería probar.

Supongamos ahora que el método Ψ_h es simétrico, es decir, $\bar{\Psi}_h = \Psi_h$. En particular, $\bar{\Psi}_h(y_0) = \Psi_h(y_0)$. Aplicando la hipótesis y la igualdad que acabamos de probar, se obtiene que $C(y_0) = (-1)^p C(y_0)$. Como $C(y_0)$ puede ser no nulo, debe ser $(-1)^p = 1$, es decir, p debe ser par. En consecuencia, el orden de un método simétrico debe ser par. □

3.3. Composición de un método con distintas longitudes de paso

En esta sección vamos a considerar la composición de un método de un paso con distintas longitudes de paso. Este caso particular de método de composición permite aumentar en, al menos, una unidad el orden de un método dado. De hecho, va a permitir construir métodos de composición de orden tan alto como se desee. Consideramos la composición de un método de un paso con distintas longitudes de paso. Sea Ψ_h un método básico de un paso y sean $\gamma_1, \dots, \gamma_s \in \mathbb{R}$ números reales. Construimos el siguiente método de composición: $\Psi_{\gamma_s h} \circ \dots \circ \Psi_{\gamma_1 h}$.

Teorema 3.12. *Sea Ψ_h un método de un paso de orden p . Supongamos que se verifica:*

$$\sum_{i=1}^s \gamma_i = 1 \text{ y } \sum_{i=1}^s \gamma_i^{p+1} = 0.$$

Entonces el método de composición $\Psi_{\gamma_s h} \circ \dots \circ \Psi_{\gamma_1 h}$ es, al menos, de orden $p+1$ (ver [3]).

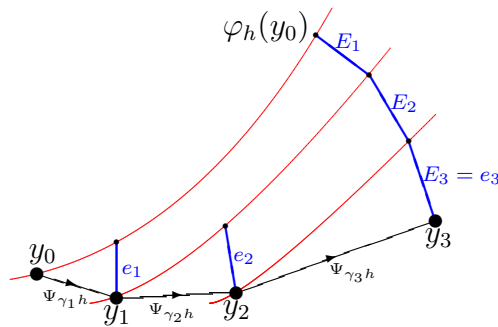


Figura 3.2: Ver [3].

Demostración. La demostración de este resultado es similar a la del *teorema 3.11*.

Por un lado, para cada $i \in \{1, \dots, s\}$ e_i , consideramos el error local e_i de $\Psi_{\gamma_i h}$, es decir, $e_i = \Psi_{\gamma_i h}(y_{i-1}) - \varphi_{\gamma_i h}(y_{i-1})$. Por hipótesis, el método original es de orden p con lo que:

$$e_i = C(y_0)\gamma_i^{p+1}h^{p+1} + O(h^{p+2}), \text{ para } 1 \leq i \leq s.$$

Por otro lado, consideramos:

$$\begin{cases} E_j = \varphi_{\sum_{i \geq j} \gamma_i h}(y_{j-1}) - \varphi_{\sum_{i > j} \gamma_i h}(y_j), \text{ para } 1 \leq j \leq s-1 \\ E_s = \varphi_{\gamma_s h}(y_{s-1}) - y_s = e_s. \end{cases}$$

Cada E_j es la diferencia entre avanzar la solución desde y_{j-1} un paso de tamaño $\sum_{i \geq j} \gamma_i h$ y avanzar la solución desde la aproximación $\Psi_{\gamma_j h}$ un paso de tamaño $\sum_{i > j} \gamma_i h$. Aplicando la primera hipótesis del enunciado, observamos que:

$$\varphi_{\sum_{i=1}^s \gamma_i h} - \left(\Psi_{\gamma_s h} \circ \dots \circ \Psi_{\gamma_1 h} \right)(y_0) = \varphi_h(y_0) - y_s = \sum_{i=1}^s E_i,$$

Teniendo en cuenta que $y_i = y_0 + O(h)$ y $E_i = (I + O(h))e_i$ para $1 \leq i \leq s$, se tiene que:

$$\begin{aligned} \varphi_h(y_0) - y_s &= \sum_{i=1}^s E_i = \sum_{i=1}^s (I + O(h))e_i = \sum_{i=1}^s (I + O(h)) \left(C(y_0)\gamma_i^{p+1}h^{p+1} + O(h^{p+2}) \right) = \\ &= (I + O(h)) \left(C(y_0)h^{p+1} \right) \left(\sum_{i=1}^s \gamma_i^{p+1} \right) + O(h^{p+2}) = O(h^{p+2}), \end{aligned}$$

aplicando la segunda hipótesis del enunciado.

En consecuencia, el método de composición $\Psi_{\gamma_s h} \circ \dots \circ \Psi_{\gamma_1 h}$ es, al menos, de orden $p+1$. \square

Observación 3.13. Es claro que si $\gamma_i \neq 0$, para todo $i = 1, \dots, s$, entonces $\sum_{i=1}^s \gamma_i^{p+1} = 0$ no tiene solución para p impar. Luego, si se desea obtener un método de orden $p+1$ mediante la composición consecutiva de un método básico de orden p , será necesario que p sea un número par.

Observación 3.14. Para que se pueda aplicar este teorema, es necesario que se componga tres veces ($s \geq 3$). Para $s = 2$, se tiene lo siguiente:

$$\left. \begin{aligned} \gamma_1 + \gamma_2 = 1 &\Rightarrow \gamma_2 = 1 - \gamma_1 \\ \gamma_1^{p+1} + \gamma_2^{p+1} = 0 &\Rightarrow \gamma_1 < 0 \text{ y } \gamma_2 > 0, \text{ o viceversa} \\ \gamma_1^{p+1} + \gamma_2^{p+1} = 0 &\Rightarrow \gamma_2^{p+1} = -\gamma_1^{p+1} = (-\gamma_1)^{p+1} \end{aligned} \right\} \Rightarrow \ln(1 - \gamma_1) = \ln(\gamma_2) = \ln(-\gamma_1)$$

Luego: $1 - \gamma_1 = -\gamma_1$, lo cual es absurdo. Por tanto, s debe ser, al menos, 3.

Ejemplo 3.15. Para $s = 3$ se tiene cierta libertad para elegir los coeficientes γ_1 , γ_2 y γ_3 . Para un orden p dado, tomamos:

$$\gamma_1 = \gamma_3 = \frac{1}{2 - 2^{1/(p+1)}}, \quad \gamma_2 = \frac{-2^{1/(p+1)}}{2 - 2^{1/(p+1)}}$$

Estos coeficientes verifican las dos condiciones del *teorema (3.12)*. Si consideramos un método simétrico Ψ de orden $p = 2$ y lo componemos como en el *teorema (3.12)* utilizando estos coeficientes γ_1 , γ_2 y $\gamma_3 = \gamma_1$, obtendremos un método de orden, al menos, $p + 1 = 3$. Además, es simétrico:

$$\overline{\Psi_{\gamma_3 h} \circ \Psi_{\gamma_2 h} \circ \Psi_{\gamma_1 h}} = \overline{\Psi_{\gamma_1 h} \circ \Psi_{\gamma_2 h} \circ \Psi_{\gamma_3 h}} = \Psi_{\gamma_1 h} \circ \Psi_{\gamma_2 h} \circ \Psi_{\gamma_3 h} = \Psi_{\gamma_3 h} \circ \Psi_{\gamma_2 h} \circ \Psi_{\gamma_1 h}.$$

Luego, por el *teorema (3.11)*, tendrá orden par. Es decir, hemos construido un método de orden, al menos, cuatro. Este proceso se puede repetir: componemos tres veces este nuevo método usando los mismos γ_1 , γ_2 y γ_3 y obtenemos un nuevo método simétrico de orden, al menos, 6. De este modo, se pueden construir métodos de composición de orden tan alto como se quiera. Si se desean pasos más pequeños, basta con aumentar s y modificar de manera adecuada los coeficientes γ_1 , γ_2 y γ_3 . La repetición de este algoritmo da lugar a los métodos fractales de Suzuki.

3.4. Métodos de composición de orden más alto: composición de un método y su adjunto

En esta sección se define otra extensión del concepto de B-serie, requerida para la obtención de métodos de composición de orden alto y, además, óptimos.

Consideramos la expresión del método Ψ_h en desarrollo de Taylor:

$$\Psi_h(y) = y + hd_1(y) + h^2d_2(y) + \dots$$

Requerimos que el método sea consistente, es decir, $d_1(y) = f(y)$. Consideramos la composición de los métodos $\Psi_{\gamma_1 h}$ y $\Psi_{\gamma_2 h}$:

$$\begin{aligned} y_1 &= \Psi_{\gamma_1 h}(y_0) = y_0 + \gamma_1 h d_1(y_0) + \gamma_1^2 h^2 d_2(y_0) + \dots \\ y_2 &= \Psi_{\gamma_2 h}(y_1) = y_1 + \gamma_2 h d_1(y_1) + \gamma_2^2 h^2 d_2(y_1) + \dots \end{aligned}$$

Introduciendo la primera expresión en la segunda, llegamos a lo que se conoce como B-serie generalizada, donde las diferenciales elementales están compuestas por infinitas funciones (que son los coeficientes que acompañan a las distintas potencias de h en el desarrollo de Taylor del método básico $\Psi_h(y)$ y sus derivadas.

3.4.1. Extensión del concepto de B-serie

Para poder estudiar las condiciones de orden de este tipo de métodos de composición utilizaremos, una vez más, el concepto de B-serie. En este caso, se requieren unos árboles con raíz que difieren de los utilizados anteriormente: hay un entero positivo asociado a cada vértice de estos nuevos árboles. Esta etiqueta numérica asociada a cada vértice se llama *tipo* del vértice.

Definición 3.16. El conjunto de todos los árboles con raíz donde cada vértice tiene asignada una etiqueta numérica, que es un entero positivo sin más restricciones, se denota T_∞ .

Los árboles con un único vértice son: $(1), (2), (3), \dots$

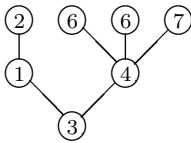
El árbol construido al unir $t_1, \dots, t_m \in T_\infty$ a un vértice (i) se denota $[t_1, \dots, t_m]_i \in T_\infty$ y es un nuevo árbol de T_∞ con raíz (i) . El orden y el coeficiente simétrico se definen de la manera habitual. Las diferenciales elementales de este tipo de árboles se construyen de la siguiente forma:

$$\begin{aligned} F((i))(y) &= d_i(y), \\ F([t_1, \dots, t_m]_i)(y) &= d_i^{(m)}(y) \left(F(t_1)(y), \dots, F(t_m)(y) \right). \end{aligned}$$

Con $i(t) = j$ indicamos que la etiqueta numérica de la raíz del árbol $t \in T_\infty$ es j . La suma de las etiquetas numéricas de un árbol $t \in T_\infty$ se denota por $\|t\|$ y se calcula de la siguiente manera:

$$\begin{aligned} \|(i)\| &= i((i)) = i, \\ \|[t_1, \dots, t_m]_i\| &= i + \sum_{i=1}^m \|t_i\|. \end{aligned}$$

Ejemplo 3.17. Consideramos el siguiente árbol de T_∞ :



El árbol representado es $t = [t_1, t_2]_3$, donde se están considerando los árboles $t_1 = (1)$ y $t_2 = [(6), (6), (7)]_4$.

El orden del árbol es $\rho(t) = 7$, la suma de las etiquetas del árbol es $\|t\| = 29$, la etiqueta de la raíz del árbol es $i(t) = 3$ y el coeficiente simétrico del árbol es $\sigma(t) = 2$. La diferencial elemental asociada al árbol es:

$$F(t)(y) = d_3''(y) \left(F(t_1)(y), F(t_2)(y) \right) = d_3''(y) \left(d_1'(y) d_2(y), d_4'''(y) (d_6(y), d_6(y), d_7(y)) \right).$$

Con este nuevo conjunto de árboles se extiende el concepto de B-serie a B_∞ -serie.

Definición 3.18. La B_∞ -serie asociada a la aplicación $a : T_\infty \rightarrow \mathbb{R}$ es la serie formal:

$$B_\infty(a, y) := a(\emptyset) y + \sum_{t \in T_\infty} \frac{h^{\|t\|}}{\sigma(t)} a(t) F(t)(y).$$

Un lema similar al *lema 1.13* visto para B-series se tiene para esta nueva extensión del concepto. La demostración es análoga a la vista en dicho lema por lo que no se va a desarrollar.

Lema 3.19. *Sea $a : T_\infty \rightarrow \mathbb{R}$ una aplicación tal que $a(\emptyset) = 1$. Entonces:*

$$h^i d_i(B_\infty(a, y)) = \sum_{t \in T_\infty, i(t)=1} \frac{h^{\|t\|}}{\sigma(t)} a'(t) F(t)(y),$$

donde $a'(\odot) = 1$ y $a'([t_1, \dots, t_m]_i) = a(t_1) \cdot \dots \cdot a(t_m)$. (Ver [3]).

3.4.2. Composición con el adjunto: condiciones de orden

Los resultados de esta subsección pueden encontrarse en su mayoría en [3].

Consideremos ahora la siguiente composición:

$$\Psi_{\alpha_k h} \circ \bar{\Psi}_{\beta_k h} \circ \dots \circ \Psi_{\alpha_1 h} \circ \bar{\Psi}_{\beta_1 h}. \quad (3.1)$$

Veamos las condiciones de orden de este método de composición.

En primer lugar, nos encontramos con un teorema similar al *teorema 3.12*, que proporciona condiciones sobre el tamaño de los pasos para poder aumentar el orden en una unidad.

Teorema 3.20. *Sea Ψ_h un método de un paso de orden p . Supongamos que se verifica:*

$$\sum_{i=1}^s (\alpha_i + \beta_i) = 1 \text{ y } \sum_{i=1}^s (\alpha_i^{p+1} + (-1)^p \beta_i^{p+1}) = 0. \quad (3.2)$$

Entonces el método de composición (3.1) es, al menos, de orden $p + 1$.

A diferencia del *teorema 3.12*, este resultado también es válido para p impar. Tampoco hay restricciones sobre s .

Ejemplo 3.21. Si consideramos $p = s = 1$, una solución posible de (3.2) es $\alpha_1 = \beta_1 = 1$. Esto permite obtener un método simétrico de orden dos a partir de un método consistente de un paso de orden uno. Tomamos como método básico de orden uno el método de Euler explícito. Ya vimos que su adjunto es el método de Euler implícito. La composición de estos dos métodos da lugar a la regla del punto medio, probando así que es de orden 2 y simétrico. Si se toma como método básico el implícito, el resultado de la composición será la regla trapezoidal.

Si vemos el método como una B_∞ -serie, podremos establecer condiciones de orden de estos métodos de composición utilizando, una vez más, el concepto de B-serie.

Lema 3.22. *Se tienen las siguientes B_∞ -series:*

$$\begin{aligned} \left(\overline{\Psi}_{\beta_k h} \circ \dots \circ \Psi_{\alpha_1 h} \circ \overline{\Psi}_{\beta_1 h} \right) (y) &= B_\infty(b_k, y), \\ \left(\Psi_{\alpha_k h} \circ \overline{\Psi}_{\beta_k h} \circ \dots \circ \Psi_{\alpha_1 h} \circ \overline{\Psi}_{\beta_1 h} \right) (y) &= B_\infty(a_k, y), \end{aligned}$$

donde los coeficientes se construyen de manera recursiva: $a_k(\emptyset) = b_k(\emptyset) = 1$, $a_0(t) = 0$ para todo $t \in T_\infty$, y

$$\begin{aligned} b_k(t) &= a_{k-1}(t) - (-\beta_k)^{i(t)} b'_k(t), \\ a_k(t) &= b_k(t) + \alpha_k^{i(t)} b'_k(t). \end{aligned}$$

Demostración. (Ver [3]). En primer lugar, vemos que $y = B_\infty(a_0, y)$. Por tanto, $a_0(t) = 0$ para todo $t \in T_\infty$.

Observamos:

$$B_\infty(a_k, y) = \Psi_{\alpha_k h} \left(B_\infty(b_k, y) \right).$$

Como $\Psi_{\alpha_k h} = \sum_{i \geq 0} \alpha_k^i h^i d_i(y) = y + \sum_{i \geq 1} \alpha_k^i h^i d_i(y)$, entonces:

$$B_\infty(a_k, y) = \Psi_{\alpha_k h} \left(B_\infty(b_k, y) \right) = B_\infty(b_k, y) + \sum_{i \geq 1} \alpha_k^i h^i d_i \left(B_\infty(b_k, y) \right).$$

Puesto que $a(\emptyset) = 1$, podemos aplicar el lema anterior:

$$h^i d_i(B_\infty(b_k, y)) = \sum_{t \in T_\infty, i(t)=1} \frac{h^{\|t\|}}{\sigma(t)} b'_k(t) F(t)(y)$$

Por tanto:

$$\begin{aligned} B_\infty(a_k, y) &= B_\infty(b_k, y) + \sum_{i \geq 1} \alpha_k^i h^i d_i \left(B_\infty(b_k, y) \right) = \\ &= B_\infty(b_k, y) + \sum_{i \geq 1} \alpha_k^i \left(\sum_{t \in T_\infty, i(t)=1} \frac{h^{\|t\|}}{\sigma(t)} b'_k(t) F(t)(y) \right) = \\ &= B_\infty(b_k, y) + \left(\sum_{t \in T_\infty} \frac{h^{\|t\|}}{\sigma(t)} \alpha_k^{i(t)} b'_k(t) F(t)(y) \right). \end{aligned}$$

Igualando coeficientes se obtiene la segunda fórmula de coeficientes del enunciado:

$$a_k(t) = b_k(t) + \alpha_k^{i(t)} b'_k(t).$$

Veamos ahora como obtener la primera fórmula del enunciado. Teniendo en cuenta que $\overline{\Psi}_h := \Psi_{-h}^{-1}$ para cualquier paso h , observamos:

$$B_\infty(a_{k-1}, y) = \Psi_{-\beta_k h}(B_\infty(b_k, y))$$

Razonando como antes, se tiene:

$$\begin{aligned} B_\infty(a_{k-1}, y) &= \Psi_{-\beta_k h}(B_\infty(b_k, y)) = B_\infty(b_k, y) + \sum_{i \geq 1} (-\beta_k)^i h^i d_i(B_\infty(b_k, y)) = \\ &= B_\infty(b_k, y) + \sum_{i \geq 1} (-\beta_k)^i \left(\sum_{t \in T_\infty, i(t)=1} \frac{h^{\|t\|}}{\sigma(t)} b'_k(t) F(t)(y) \right) = \\ &= B_\infty(b_k, y) + \left(\sum_{t \in T_\infty} \frac{h^{\|t\|}}{\sigma(t)} (-\beta_k)^{i(t)} b'_k(t) F(t)(y) \right). \end{aligned}$$

Por tanto:

$$B_\infty(b_k, y) = B_\infty(a_{k-1}, y) - \left(\sum_{t \in T_\infty} \frac{h^{\|t\|}}{\sigma(t)} (-\beta_k)^{i(t)} b'_k(t) F(t)(y) \right).$$

Igualando coeficientes en la expresión anterior se obtiene la primera fórmula de coeficientes del enunciado:

$$b_k(t) = a_{k-1}(t) - (-\beta_k)^{i(t)} b'_k(t).$$

□

Observación 3.23. Sumando las dos expresiones de coeficientes del lema anterior, obtenemos:

$$a_k(t) = a_{k-1}(t) + \left(\alpha_k^{i(t)} - (-\beta_k)^{i(t)} \right) b'_k(t) \Rightarrow a_k(t) = \sum_{j=1}^k \left(\alpha_j^{i(t)} - (-\beta_j)^{i(t)} \right) b'_j(t).$$

Sustituyendo esta expresión en la definición del coeficiente b_k , se tiene:

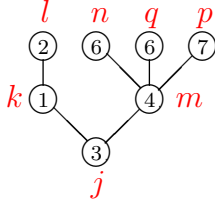
$$\begin{aligned} b_k(t) &= \sum_{j=1}^{k-1} \left(\alpha_j^{i(t)} - (-\beta_j)^{i(t)} \right) b'_j(t) - (-\beta_k)^{i(t)} b'_k(t) = \\ &= \sum_{j=1}^{k-1} \alpha_j^{i(t)} b'_j(t) - \sum_{j=1}^k (-\beta_j)^{i(t)} b'_j(t) = \sum_{j=1}^{k-1} \left(\alpha_j^{i(t)} - (-\beta_j)^{i(t)} \right) b'_j(t). \end{aligned}$$

En particular, teniendo en cuenta que $b'_k(\odot) = a_k(\odot) = 1$, observamos lo siguiente:

$$\begin{aligned} a_k(\odot) &= \sum_{j=1}^k \left(\alpha_j^{i(t)} - (-\beta_j)^i \right), \\ b_k(\odot) &= \sum_{j=1}^{k-1} \alpha_j^i - \sum_{j=1}^k (-\beta_j)^i = \sum_{j=1}^{k-1} \left(\alpha_j^i - (-\beta_j)^i \right) \end{aligned}$$

Estas expresiones facilitan el cálculo de los correspondientes coeficientes de un árbol.

Ejemplo 3.24. Consideramos de nuevo el árbol siguiente:



El árbol representado es $t = [t_1, t_2]_3$, considerando los árboles $t_1 = \textcircled{1}$ y $t_2 = [\textcircled{6}, \textcircled{6}, \textcircled{7}]_4$.

Observamos: $i(t) = 3$, $i(t_1) = 1$ y $i(t_2) = 4$.

$$a_s(t) = \sum_{j=1}^s \left(\alpha_j^3 - (-\beta_j)^3 \right) b'_j(t) = \sum_{j=1}^s \left(\alpha_j^3 + \beta_j^3 \right) b'_j(t)$$

Por la definición dada en el *lema 3.19* de los coeficientes $b'_j(t)$, se tiene:

$$b'_j(t) = b_j(t_1) \cdot b_j(t_2).$$

Aplicando ahora la fórmula vista en la observación anterior para los coeficientes b_j se tiene:

$$b_j(t_1) = \sum_{k=1}^{j'} \left(\alpha_k^1 - (-\beta_k)^1 \right) b'_k(t_1) = \sum_{k=1}^{j'} \left(\alpha_k + \beta_k \right) b'_k(t_1),$$

$$b_j(t_2) = \sum_{m=1}^{j'} \left(\alpha_m^4 - (-\beta_m)^4 \right) b'_m(t_2) = \sum_{m=1}^{j'} \left(\alpha_m^4 - \beta_m^4 \right) b'_m(t_2),$$

Además:

$$b'_k(t_1) = b_k(\textcircled{2}) = \sum_{l=1}^{k'} \left(\alpha_l^2 - \beta_l^2 \right)$$

$$b'_m(t_2) = b_m^2(\textcircled{6}) \cdot b_m(\textcircled{7}) = \left(\sum_{n=1}^{m'} \left(\alpha_n^6 - \beta_n^6 \right) \right)^2 \cdot \sum_{p=1}^{m'} \left(\alpha_p^7 + \beta_p^7 \right)$$

En consecuencia:

$$a_s(t) = \sum_{j=1}^s \left(\alpha_j^3 + \beta_j^3 \right) \cdot \left[\sum_{k=1}^{j'} \left(\alpha_k + \beta_k \right) \cdot \left(\sum_{l=1}^{k'} \left(\alpha_l^2 - \beta_l^2 \right) \right) \right] \cdot \left[\sum_{m=1}^{j'} \left(\alpha_m^4 - \beta_m^4 \right) \cdot \left(\sum_{n=1}^{m'} \left(\alpha_n^6 - \beta_n^6 \right) \right)^2 \cdot \left(\sum_{p=1}^{m'} \left(\alpha_p^7 + \beta_p^7 \right) \right) \right]$$

La solución exacta de $\dot{y} = f(y)$ es una B-serie: $y(t_0 + h) = B(\mathbf{e}, y_0)$. En particular, es una B_∞ -serie si se consideran coeficientes nulos para todo árbol de T_∞ con, al menos, una etiqueta distinta de 1. Luego, $y(t_0 + h) = B_\infty(\mathbf{e}, y_0)$, con $e(t) = 0$ para todo $t \in T_\infty$ con $i(t) > 1$ y $e(\textcircled{1}) = 1$. Además:

$$e(t) = \frac{1}{\rho(t)} e(t_1) \cdot \dots \cdot e(t_m), \text{ para } t = [t_1, \dots, t_m]_i \in T_\infty.$$

Teorema 3.25. *El método de composición $\Psi_{\alpha_k h} \circ \overline{\Psi}_{\beta_k h} \circ \dots \circ \Psi_{\alpha_1 h} \circ \overline{\Psi}_{\beta_1 h}$ tiene orden p si y solo si se verifica la siguiente condición:*

$$a_k(t) = e(t), \text{ para todo árbol } t \in T_\infty \text{ con } ||t|| \leq p.$$

Demostración. (Ver [3]). Recordemos que un método numérico es de orden p si el error de truncación es $O(h^{p+1})$ cuando $h \rightarrow 0$, es decir, si las correspondientes series coinciden hasta los términos de orden p . Sabemos que $\Psi_{\alpha_k h} \circ \overline{\Psi}_{\beta_k h} \circ \dots \circ \Psi_{\alpha_1 h} \circ \overline{\Psi}_{\beta_1 h}(y) = B_\infty(a_k, y)$. Comparando esta B_∞ -serie con la correspondiente B_∞ -serie de la solución exacta se prueba la suficiencia de la condición. La independencia de las diferenciales elementales proporciona el carácter necesario de la condición. \square

3.4.3. Reducción de las condiciones de orden

El teorema anterior proporciona una condición de orden válida para el método de composición $\Psi_{\alpha_k h} \circ \overline{\Psi}_{\beta_k h} \circ \dots \circ \Psi_{\alpha_1 h} \circ \overline{\Psi}_{\beta_1 h}$. Sin embargo, la cantidad de árboles en T_∞ hace que sea difícil de aplicar. Veremos que algunas de las condiciones que aparecen en el teorema son equivalentes, facilitando su uso en la práctica.

Definición 3.26. Sean $u = [u_1, \dots, u_m]_i, v = [v_1, \dots, v_n]_j \in T_\infty$.

- Se define el producto de Butcher de los árboles u y v por: $u \circ v = [u_1, \dots, u_m, v]_i$. Esta operación no es asociativa ni conmutativa.
- Se define el producto de fusión de los árboles u y v por: $u \times v = [u_1, \dots, u_m, v_1, \dots, v_n]_{i+1}$. Esta operación es asociativa y conmutativa.

Con el fin de evitar escribir más fórmulas de las necesarias, denotaremos por c a los coeficientes a_k y b_k .

Lema 3.27. (Ver [7]). *Los coeficientes a_k y b_k verificando las relaciones de recurrencia del lema 3.22 verifican la siguiente igualdad:*

$$c(u \circ v) + c(v \circ u) = c(u) \cdot c(v) - c(u \times v), \forall u, v \in T_\infty.$$

Demostración. Denotaremos por a y b a los coeficientes a_k y b_k respectivamente. Por hipótesis verifican: $a(t) = b(t) + \alpha^{i(t)} b'(t)$. Sean $u = [u_1, \dots, u_m]_i, v = [v_1, \dots, v_n]_j \in T_\infty$. Entonces:

$$\begin{aligned} a(u \circ v) &= b(u \circ v) + \alpha^{i(u \circ v)} b'(u \circ v) = b(u \circ v) + \alpha^{i(u)} b'(u \circ v) \\ a(v \circ u) &= b(v \circ u) + \alpha^{i(v \circ u)} b'(v \circ u) = b(v \circ u) + \alpha^{i(v)} b'(v \circ u) \\ a(u \times v) &= b(u \times v) + \alpha^{i(u \times v)} b'(u \times v) = b(u \times v) + \alpha^{i(u)+i(v)} b'(u \times v) \end{aligned}$$

Por tanto:

$$\begin{aligned} a(u \circ v) + a(v \circ u) + a(u \times v) - a(u) \cdot a(v) &= (b(u \circ v) + \alpha^{i(u)} b'(u \circ v)) + \\ &+ (b(v \circ u) + \alpha^{i(v)} b'(v \circ u)) + (b(u \times v) + \alpha^{i(u)+i(v)} b'(u \times v)) - \\ &- (b(u) + \alpha^{i(u)} b'(u)) \cdot (b(v) + \alpha^{i(v)} b'(v)) \end{aligned}$$

Observamos:

$$\begin{aligned} b'(u \circ v) &= b'([u_1, \dots, u_m, v]_i) = b(u_1) \cdot \dots \cdot b(u_m) \cdot b(v) = b'(u) \cdot b(v), \\ b'(v \circ u) &= b'([v_1, \dots, v_n, u]_j) = b(v_1) \cdot \dots \cdot b(v_n) \cdot b(u) = b'(v) \cdot b(u), \\ b'(u \times v) &= b'([u_1, \dots, u_m, v_1, \dots, v_n]_{i+j}) = b'(u) \cdot b'(v). \end{aligned}$$

En consecuencia:

$$a(u \circ v) + a(v \circ u) + a(u \times v) - a(u) \cdot a(v) = b(u \circ v) + b(v \circ u) + b(u \times v) - b(u) \cdot b(v)$$

Luego, a_k satisface la igualdad si y solo si b_k satisface la igualdad. Es claro que a_0 satisface la igualdad ya que $a_0(t) = 0$ para todo $t \in T_\infty$. Luego, también debe satisfacerla b_0 . Bajo estas condiciones y teniendo en cuenta que $b_1(t) = a_0(t) - (-\beta_1)^{i(t)} b'_1(t) = -(-\beta_1)^{i(t)} b'_1(t)$, obtenemos:

$$\begin{aligned} b_1(u \circ v) + b_1(v \circ u) &= -(-\beta_1)^{i(u \circ v)} b'_1(u \circ v) - (-\beta_1)^{i(v \circ u)} b'_1(v \circ u) = \\ &= -(-\beta_1)^{i(u)} b'_1(u) b_1(v) - (-\beta_1)^{i(v)} b'_1(v) b_1(u) = \\ &= (-\beta_1)^{i(u)+i(v)} b'_1(u) b'_1(v) + (-\beta_1)^{i(v)+i(u)} b'_1(v) b'_1(u) = \\ &= (-(-\beta_1)^{i(u)} b'_1(u)) \cdot (-(-\beta_1)^{i(v)} b'_1(v)) - (-(-\beta_1)^{i(v \times u)} b'_1(u \times v)) = \\ &= b_1(u) \cdot b_1(v) - b_1(u \times v). \end{aligned}$$

Queda probado para b_1 y, por tanto, para a_1 . Ahora se probaría para b_2 usando que a_1 lo verifica. De manera recursiva se prueba para todo a_k y b_k . □

Observación 3.28. El lema anterior reduce considerablemente el número de comprobaciones que hay que realizar para determinar el orden del método. Es sencillo ver que las condiciones de orden de $u \circ v$ y $v \circ u$ son equivalentes, ya que la operación $u \circ v \mapsto v \circ u$ que permuta los factores consiste simplemente en intercambia la raíz con un vértice adyacente concreto. Repitiendo este argumento, todos los árboles obtenidos al cambiar la raíz con un vértice adyacente cualquiera tantas veces como se quiera tienen condiciones de orden equivalentes.

Lema 3.29. (Ver [3]). Sean a_k y b_k los coeficientes que verifican las relaciones de recurrencia del lema 3.22. Supongamos que todos los coeficientes b_k verifican

$$\sum_{i=1}^N A_i \prod_{j=1}^{n_i} c(u_{ij}) = 0, \text{ con } n_i > 0.$$

Entonces se satisface la siguiente igualdad:

$$\sum_{i=1}^N A_i c(w \circ u_{i1} \circ \dots \circ u_{i,n_i}) = 0, \text{ para todo árbol } w \in T_\infty.$$

donde $w \circ u_{i1} \circ \dots \circ u_{i,n_i}$ denota $(\dots((w \circ u_{i1}) \circ u_{i2}) \circ \dots) \circ u_{i,n_i}$.

Demostración. Denotaremos por a y b a los coeficientes a_k y b_k respectivamente. Por hipótesis verifican: $a(t) = b(t) + \alpha^{i(t)}b'(t)$. Entonces:

$$\begin{aligned} a(w \circ u_{i1} \circ \dots \circ u_{i,n_i}) &= b(w \circ u_{i1} \circ \dots \circ u_{i,n_i}) + \alpha^{i(w \circ u_{i1} \circ \dots \circ u_{i,n_i})}b'(w \circ u_{i1} \circ \dots \circ u_{i,n_i}) = \\ &= b(w \circ u_{i1} \circ \dots \circ u_{i,n_i}) + \alpha^{i(w)}b'(w) \cdot b(u_{i1}) \cdot \dots \cdot b(u_{i,n_i}) \end{aligned}$$

Por tanto:

$$\begin{aligned} \sum_{i=1}^N A_i a(w \circ u_{i1} \circ \dots \circ u_{i,n_i}) &= \sum_{i=1}^N A_i \left(b(w \circ u_{i1} \circ \dots \circ u_{i,n_i}) + \alpha^{i(w)}b'(w) \cdot b(u_{i1}) \cdot \dots \cdot b(u_{i,n_i}) \right) \\ &= \sum_{i=1}^N A_i b(w \circ u_{i1} \circ \dots \circ u_{i,n_i}) + \alpha^{i(w)}b'(w) \sum_{i=1}^N A_i \prod_{j=1}^{n_i} b(u_{ij}). \end{aligned}$$

En consecuencia, aplicando la hipótesis del enunciado, se tiene:

$$\sum_{i=1}^N A_i a(w \circ u_{i1} \circ \dots \circ u_{i,n_i}) = \sum_{i=1}^N A_i b(w \circ u_{i1} \circ \dots \circ u_{i,n_i}).$$

Es decir, se verifica la igualdad para a_k si y solo si se verifica para b_k . De nuevo, es evidente que se verifica para a_0 y, por tanto, se verifica para b_0 . Veámoslo para b_1 . Recordemos que $b_1(t) = -(-\beta_1)^{i(t)}b'_1(t)$. Entonces:

$$\begin{aligned} \sum_{i=1}^N A_i b(w \circ u_{i1} \circ \dots \circ u_{i,n_i}) &= - \sum_{i=1}^N A_i (-\beta_1)^{i(w \circ u_{i1} \circ \dots \circ u_{i,n_i})} b'_1(w \circ u_{i1} \circ \dots \circ u_{i,n_i}) = \\ &= -(-\beta_1)^{i(w)} \sum_{i=1}^N A_i b'_1(w \circ u_{i1} \circ \dots \circ u_{i,n_i-1}) b_1(u_{i,n_i}) \\ &= -(-\beta_1)^{i(w)} \sum_{i=1}^N A_i b'_1(w \circ u_{i1} \circ \dots \circ u_{i,n_i-2}) b_1(u_{i,n_i-1}) b_1(u_{i,n_i}) = \\ &= \dots = -(-\beta_1)^{i(w)} b'_1(w) \left(\sum_{i=1}^N A_i \prod_{j=1}^{n_i} b(u_{ij}) \right) = 0. \end{aligned}$$

De este modo quedaría probado para b_1 y, por tanto, también para a_1 . Lo siguiente sería probarlo para b_2 usando que a_1 lo verifica. De manera recursiva se prueba para todos los coeficientes a_k y b_k . □

Lema 3.30. (Ver [3]). Sean a_k y b_k coeficientes que verifican las relaciones de recurrencia del lema 3.22. Entonces se satisface la siguiente relación:

$$c(u \circ v \circ w) + c(v \circ u \circ w) + c(w \circ u \circ v) = c(u)c(v)c(w) + \dots, \text{ para todos } u, v, w \in T_\infty,$$

donde los puntos suspensivos denotan combinaciones lineales de productos de la forma $\prod_j c(v_j)$ con $\sum_j \rho(v_j) < \rho(u) + \rho(v) + \rho(w)$ y, para cada uno de estos productos, al menos un v_j tiene una etiqueta mayor que 1. Esta fórmula se puede generalizar con los productos de Butcher de m árboles.

Demostración. Sean $u, v, w \in T_\infty$. Por el lema 3.27, se verifica lo siguiente:

$$c(u \circ v) + c(v \circ u) + c(u \times v) - c(u) \cdot c(v) = 0.$$

Es decir:

$$\sum_{i=1}^4 A_i \prod_{j=1}^{n_i} c(u_{ij}) = 0$$

siendo $n_1 = n_2 = n_3 = 1$, $n_4 = 2$, $A_1 = A_2 = A_3 = 1 = -A_4$, $u_{11} = u \circ v$, $u_{21} = v \circ u$, $u_{31} = u \times v$, $u_{41} = u$ y $u_{42} = v$. Bajo estas condiciones podemos aplicar el lema 3.29 para el árbol w :

$$0 = \sum_{i=1}^4 A_i c(w \circ u_{i1} \circ u_{i2}) = c(w \circ (u \circ v)) + c(w \circ (v \circ u)) + c(w \circ (u \times v)) - c(w \circ u \circ v).$$

De nuevo, por el lema 3.27:

$$\begin{aligned} c(w \circ (u \circ v)) + c((u \circ v) \circ w) &= c(w)c(u \circ v) - c(w \times (u \circ v)) \\ c(w \circ (v \circ u)) + c((v \circ u) \circ w) &= c(w)c(v \circ u) - c(w \times (v \circ u)) \end{aligned}$$

Sustituyendo estas expresiones en la igualdad anterior:

$$\begin{aligned} &c(u \circ v \circ w) + c(v \circ u \circ w) + c(w \circ u \circ v) = \\ &= c(w) \left(c(u \circ v) + c(v \circ u) \right) - c(w \times (u \circ v)) - c(w \times (v \circ u)) + c(w \circ (u \times v)) = \\ &= c(w) \left(c(u)c(v) - c(u \times v) \right) - c(w \times (u \circ v)) - c(w \times (v \circ u)) + c(w \circ (u \times v)) = \\ &= c(w)c(u)c(v) + \left(-c(w)c(u \times v) - c(w \times (u \circ v)) - c(w \times (v \circ u)) + c(w \circ (u \times v)) \right) \end{aligned}$$

Falta comprobar las condiciones respecto al orden y las etiquetas de productos que aparecen dentro del paréntesis anterior. Teniendo en cuenta que $\rho(t_1 \times t_2) = \rho(t_1) + \rho(t_2) - 1$ y que $\rho(t_1 \circ t_2) = \rho(t_2 \circ t_1) = \rho(t_1) + \rho(t_2)$, se tiene:

$$\begin{aligned} \rho(w) + \rho(u \times v) &= \rho(w) + \rho(u) + \rho(v) - 1 < \rho(w) + \rho(u) + \rho(v) \\ \rho(w \times (u \circ v)) &= \rho(w) + \rho(u \circ v) - 1 = \rho(w) + \rho(u) + \rho(v) - 1 < \rho(w) + \rho(u) + \rho(v) \\ \rho(w \circ (u \times v)) &= \rho(w) + \rho(u \times v) = \rho(w) + \rho(u) + \rho(v) - 1 < \rho(w) + \rho(u) + \rho(v). \end{aligned}$$

Luego, en todos los productos la suma de los órdenes de los árboles es menor estrictamente que $\rho(w) + \rho(u) + \rho(v)$. Además, como en todos los productos aparece un producto de fusión y la etiqueta de la raíz de un producto de fusión es siempre mayor o igual a 2, podemos afirmar que en cada uno de estos productos aparece un árbol con alguna etiqueta mayor que 1. \square

Con el fin de que las simplificaciones que se han ido introduciendo sigan algún orden, fijamos una relación de orden total “<” en T_∞ que sea compatible con el número de vértices, es decir:

- Para $u, v \in T_\infty$ con $u \neq v$, se tiene $u < v$ o $v < u$.
- El orden respeta el número de vértices: $\rho(u) < \rho(v) \Rightarrow u < v$.

Definición 3.31. Un subconjunto $\mathcal{H} \subseteq T_\infty$ (ver [7]) se dice que es un conjunto de Hall si existe una relación de orden “<” en \mathcal{H} verificando las siguientes propiedades:

1. El orden “<” es total en \mathcal{H} : para $u, v \in \mathcal{H}$ con $u \neq v$, se tiene $u < v$ o $v < u$.
2. El orden “<” es compatible con el número de vértices: $\rho(u) < \rho(v) \Rightarrow u < v$.
3. Si $t \in T_\infty$ con $\rho(t) = 1$, entonces $t \in \mathcal{H}$: $\circledast i \in \mathcal{H}, \forall i$.
4. Si $t \in T_\infty$ con $\rho(t) > 1$, entonces: $t \in \mathcal{H} \Leftrightarrow \exists u, v \in \mathcal{H}$ con $u > v$ tales que $t = u \circ v$.

Fijada la relación de orden total “<” en T_∞ que es compatible con el número de vértices, encontramos un conjunto de Hall $\mathcal{H} \subseteq T_\infty$. Veamos algunos ejemplos de elementos de este conjunto.

Ejemplo 3.32.

$$t_1 = \begin{array}{c} \circledast 1 \quad \circledast 2 \\ \diagdown \quad / \\ \circledast 3 \end{array} = [\circledast 1, \circledast 2]_3 \in \mathcal{H}, \text{ ya que } t_1 = u \circ v \text{ con } u > v \text{ siendo } u = \begin{array}{c} \circledast 2 \\ | \\ \circledast 3 \end{array} = [\circledast 2]_3 \text{ y } v = \circledast 1.$$

$$t_2 = \begin{array}{c} \circledast 2 \\ | \\ \circledast 1 \\ / \quad \backslash \\ \circledast 3 \end{array} = [[\circledast 2]_1]_3 \notin \mathcal{H}, \text{ ya que los únicos árboles de } T_\infty \text{ cuyo producto de Butcher } u \circ v \text{ es } t_2 \text{ son } u = \circledast 3 \text{ y } v = \begin{array}{c} \circledast 1 \\ | \\ \circledast 2 \end{array} = [\circledast 1]_2 \text{ pero } u \not> v, \text{ ya que } \rho(u) < \rho(v).$$

Teorema 3.33. Para cada $t \in T_\infty$ existen constantes A_i , enteros n_i y árboles $u_{ij} \in \mathcal{H}$ tales que para todo coeficiente a_k y b_k que verifican las relaciones de recurrencia del lema 3.22 se satisface:

$$c(t) = \sum_{i=1}^N A_i \prod_{j=1}^{n_i} c(u_{ij}), \text{ con } \sum_{j=1}^{n_i} \rho(u_{ij}) \leq \rho(t). \quad (\text{Ver [3] y [7]}).$$

Demostración. Razonamos por inducción sobre el orden de los árboles.

Para $\rho(t) = 1$ es trivial, ya que debe ser $t = \circledast i$.

Supongámoslo cierto para $m \leq n - 1$ y veámoslo para n .

Sea $t = u \circ v \in T_\infty$ con $\rho(t) = \rho(u) + \rho(v) = n$ y u y v árboles de T_∞ no vacíos. Entonces $\rho(v) < \rho(t) = n$ y podemos aplicar la hipótesis de inducción:

$$c(v) = \sum_{i=1}^M B_i \prod_{j=1}^{m_i} c(v_{ij}), \text{ con } v_{ij} \in \mathcal{H} \text{ y } \sum_{j=1}^{m_i} \rho(v_{ij}) \leq \rho(v).$$

Luego, considerando $B_{M+1} = -1$, $m_{M+1} = 1$ y $v_{M+1,1} = v$, se tiene:

$$\sum_{i=1}^{M+1} B_i \prod_{j=1}^{m_i} c(v_{ij}) = 0.$$

Bajo estas condiciones podemos aplicar el *lema 3.29* y obtenemos:

$$\begin{aligned} 0 &= \sum_{i=1}^{M+1} B_i c(u \circ v_{i1} \circ \dots \circ v_{i,n_i}) = \sum_{i=1}^M B_i c(u \circ v_{i1} \circ \dots \circ v_{i,n_i}) - c(u \circ v) \\ &\Rightarrow c(t) = c(u \circ v) = \sum_{i=1}^M B_i c(u \circ v_{i1} \circ \dots \circ v_{i,n_i}). \end{aligned}$$

Por el *lema 3.27* se tiene que:

$$\begin{aligned} c(u \circ v_{i1} \circ \dots \circ v_{i,n_i}) &= -c(v_{i,n_i} \circ (u \circ v_{i1} \circ \dots \circ v_{i,n_i-1})) + c(v_{i,n_i})c(u \circ v_{i1} \circ \dots \circ v_{i,n_i-1}) \\ &\quad - c((u \circ v_{i1} \circ \dots \circ v_{i,n_i-1}) \times v_{i,n_i}) \end{aligned}$$

Por tanto:

$$\begin{aligned} c(t) &= - \sum_{i=1}^M B_i c(v_{i,n_i} \circ (u \circ v_{i1} \circ \dots \circ v_{i,n_i-1})) + \\ &\quad + \sum_{i=1}^M B_i \left(c(v_{i,n_i})c(u \circ v_{i1} \circ \dots \circ v_{i,n_i-1}) - c((u \circ v_{i1} \circ \dots \circ v_{i,n_i-1}) \times v_{i,n_i}) \right). \end{aligned}$$

Todos los árboles que aparecen en el segundo sumatorio son de orden estrictamente menores que $\rho(t) = n$. Sobre dichos árboles se puede aplicar la hipótesis de inducción hasta obtener una expresión similar a la del enunciado eliminando todos los productos de Butcher que aparecen.

Faltaría eliminar los productos de Butcher del primer sumatorio. Aplicamos hipótesis de inducción sobre el árbol $u \circ v_{i1} \circ \dots \circ v_{i,n_i-1}$:

$$c(u \circ v_{i1} \circ \dots \circ v_{i,n_i-1}) = \sum_{i=1}^L C_i \prod_{j=1}^{l_i} c(u_{ij}), \text{ con } u_{ij} \in \mathcal{H} \text{ y } \sum_{j=1}^{l_i} \rho(u_{ij}) \leq \rho(u \circ v_{i1} \circ \dots \circ v_{i,n_i-1})$$

Tomando $C_{L+1} = -1$, $l_{L+1} = 1$ y $u_{L+1,1} = u \circ v_{i1} \circ \dots \circ v_{i,n_i-1}$, se tiene:

$$\sum_{i=1}^{L+1} C_i \prod_{j=1}^{l_i} c(u_{ij}) = 0.$$

Bajo estas condiciones volvemos a aplicar el *lema 3.29* y obtenemos:

$$\begin{aligned}
0 &= \sum_{i=1}^{L+1} C_i c(v_{i,n_i} \circ u_{i1} \circ \dots \circ u_{i,l_i}) = \sum_{i=1}^L C_i c(v_{i,n_i} \circ u_{i1} \circ \dots \circ u_{i,l_i}) - c(v_{i,n_i} \circ (u \circ v_{i1} \circ \dots \circ v_{i,n_i-1})) \\
&\Rightarrow \sum_{i=1}^M B_i c(v_{i,n_i} \circ (u \circ v_{i1} \circ \dots \circ v_{i,n_i-1})) = \sum_{i=1}^M B_i \left(\sum_{i=1}^L C_i c(v_{i,n_i} \circ u_{i1} \circ \dots \circ u_{i,l_i}) \right)
\end{aligned}$$

Por tanto, el sumatorio $\Rightarrow \sum_{i=1}^M B_i c(v_{i,n_i} \circ (u \circ v_{i1} \circ \dots \circ v_{i,n_i-1}))$ es una combinación lineal de términos de la forma $c(u_1 \circ \dots \circ u_m)$, con $u_i \in \mathcal{H}$ por construcción. Finalmente, debemos asegurar que $u_1 \circ \dots \circ u_m \in \mathcal{H}$. Para ello debemos reordenar los u_i . Suponemos $u_2 \leq u_3 \leq \dots \leq u_m$:

1. Si $u_1 > u_2$, entonces $u_1 \circ u_2 = w \in \mathcal{H}$ y sustituimos: $u_1 \circ u_2 \circ \dots \circ u_m = w \circ u_3 \circ \dots \circ u_m$.
2. Si $u_1 < u_2$, aplicamos el *lema 3.30*:

$$c(u_1 \circ \dots \circ u_m) = - \sum_{i=2}^m c(u_i \circ u_1 \circ \dots) + \prod_{i=1}^m c(u_i) + \dots$$

El primer sumando nos vuelve a llevar al caso anterior, el segundo sumando es un producto de la misma forma que el del enunciado y el resto de términos tienen orden menor que n y se puede aplicar hipótesis de inducción.

3. Si $u_1 = u_2$, razonamos como en el caso anterior:

$$c(u_1 \circ \dots \circ u_m) = -\frac{1}{2} \sum_{i=3}^m c(u_i \circ u_1 \circ u_1 \circ \dots) + \prod_{i=1}^m c(u_i) + \dots$$

con las mismas conclusiones que en el caso anterior.

Repitiendo este proceso se hacen desaparecer todos los productos de Butcher necesarios dando lugar a una combinación lineal de productos como los del enunciado y se concluye la prueba. □

Antes de abordar el teorema principal de esta sección, introducimos una relación de precedencia entre árboles de T_∞ que no debe confundirse con la relación de orden total del conjunto de Hall \mathcal{H} . Decimos que $u \in T_\infty$ precede a $v \in T_\infty$ si, o bien $\|u\| < \|v\|$, o bien si cuando $\|u\| = \|v\|$ entonces $\rho(u) < \rho(v)$. (Ver [7]).

El teorema principal de esta sección se formula entonces en los siguientes términos:

Teorema 3.34. *El método de composición $\Psi_{\alpha_k h} \circ \bar{\Psi}_{\beta_k h} \circ \dots \circ \Psi_{\alpha_1 h} \circ \bar{\Psi}_{\beta_1 h} = B_\infty(a_k, y)$ tiene orden, al menos, p si y solo si se verifica la siguiente condición:*

$$a_k(t) = e(t), \text{ para todo árbol } t \in \mathcal{H} \text{ con } \|t\| \leq p.$$

siendo e los coeficientes de la B -serie de la solución exacta, los mismos coeficientes que en el teorema 3.25. (Ver [3]).

Demostración. En primer lugar, supongamos que el método (3.1) dado por

$$\Psi_{\alpha_k h} \circ \bar{\Psi}_{\beta_k h} \circ \dots \circ \Psi_{\alpha_1 h} \circ \bar{\Psi}_{\beta_1 h} = B_\infty(a_k, y)$$

es de orden p . Por el teorema 3.25, se tiene:

$$a_k(t) = e(t), \quad \forall t \in T_\infty \text{ con } \|t\| \leq p.$$

En particular, como \mathcal{H} está contenido en T_∞ , se tiene la anterior igualdad para todo $t \in \mathcal{H}$ con $\|t\| \leq p$.

Supongamos ahora que se verifica la condición del enunciado. Sea $t \in T_\infty$ con $\|t\| \leq p$. Bajo estas condiciones, por el lema 3.29, existen constantes A_i , enteros n_i y árboles $u_{ij} \in \mathcal{H}$ tales que:

$$c(t) = \sum_{i=1}^N A_i \prod_{j=1}^{n_i} c(u_{ij}), \text{ con } \sum_{j=1}^{n_i} \rho(u_{ij}) \leq \rho(t),$$

para cualquier aplicación $c : T_\infty \rightarrow \mathbb{R}$.

Nótese que en la demostración del lema 3.29 no se utiliza el hecho de que c es a_k o b_k . De hecho, las constantes A_i , los enteros n_i y los árboles $u_{ij} \in \mathcal{H}$ dependen únicamente del árbol t y no de la aplicación $c : T_\infty \rightarrow \mathbb{R}$.

En particular:

$$a_k(t) = \sum_{i=1}^N A_i \prod_{j=1}^{n_i} a_k(u_{ij}), \text{ con } \sum_{j=1}^{n_i} \rho(u_{ij}) \leq \rho(t),$$

$$e(t) = \sum_{i=1}^N A_i \prod_{j=1}^{n_i} e(u_{ij}), \text{ con } \sum_{j=1}^{n_i} \rho(u_{ij}) \leq \rho(t).$$

Para finalizar recurrimos a un resultado técnico adicional que formularemos sin prueba. La prueba de este resultado puede encontrarse en [7].

Lema 3.35. *Para cualquier $u \in T_\infty$, $c(u)$ es equivalente a una combinación lineal de términos de la forma $c(v)$, con $v \in T_\infty$, $\rho(u) = \rho(v)$ y $\|u\| = \|v\|$.*

Por expresiones equivalentes entendemos que la diferencia entre ambas es un polinomio en los coeficientes de árboles de T_∞ que preceden al árbol u y a cada uno de los árboles

v de la combinación lineal.

Tras este lema continuamos con la demostración del teorema (ver [7]). Tenemos que demostrar que, para cada $u \in T_\infty$ con $\|u\| \leq p$, se tiene $c(u) = e(v)$. Para ello vamos a hacer una inducción sobre $\|u\|$. Para $\|u\| = 1$, es trivial que u está en \mathcal{H} . Para pasar de suma de etiquetas $\nu - 1$ a $\nu > 2$ usamos, de nuevo, una inducción sobre el número de nodos. Si $\rho(u) = 1$, entonces no hay nada que probar porque \mathcal{H} contiene a todos los árboles de T_∞ con un solo nodo. Para pasar de orden $\rho - 1$ a orden $\rho > 2$ notemos que el *lema 3.35* nos da $c(u)$ como una función polinómica $F(c(u_1^*), \dots, c(u_l^*), c(u_1^{**}), \dots, c(u_m^{**}))$, donde los $u_i^* \in \mathcal{H}$ tienen $\|u_i^*\| = \|u\|$ y $\rho(u_i^*) = \rho(u)$, y los u_j^{**} tienen o $\|u_j^{**}\| < \|u\|$, o si $\|u_j^{**}\| = \|u\|$ entonces $\rho(u_j^{**}) < \rho(u)$. De igual modo, $e(u) = F(e(u_1^*), \dots, e(u_l^*), e(u_1^{**}), \dots, e(u_m^{**}))$. Se tiene $c(u_j^*) = e(u_j^*)$ por las hipótesis del teorema y $c(u_j^{**}) = e(u_j^{**})$ por la hipótesis de inducción. En consecuencia, $c(u) = e(u)$. □

Capítulo 4

Experimentos numéricos

4.1. Comparación entre métodos Runge-Kutta particionados simplécticos y métodos Runge-Kutta convencionales

En la teoría los métodos Runge-Kutta simplécticos presentan una ventaja importante a la hora de determinar su orden. El objetivo de esta sección es ilustrar de forma práctica las bondades de los métodos simplécticos cuando se aplican a problemas Hamiltonianos.

Se van a comparar dos métodos Runge-Kutta particionados simplécticos con uno no simpléctico. El método no simpléctico elegido ha sido el correspondiente a la función ode45 de Matlab, que corresponde a un método Runge-Kutta de orden 5 con paso variable. En la subsección 2.3.2, vimos que los métodos de Ruth son un tipo de métodos Runge-Kutta particionados simplécticos. Los métodos elegidos para el experimento son los siguientes:

Figura 4.1: Primer método

Método de Ruth de orden 3: $[7/24, 3/4, -1/24](2/3, -2/3, 1)$

7/24	0	0	0	0	0
7/24	3/4	0	2/3	0	0
7/24	3/4	-1/24	2/3	-2/3	0
7/24	3/4	-1/24	2/3	-2/3	1

A partir de este método de orden 3 se puede obtener uno de orden 4 de 6 etapas:

$$[b_1, b_2, b_3](B_1, B_2, B_3) \Rightarrow \left(\frac{b_3}{2}, \frac{b_2}{2}, b_1, \frac{b_2}{2}, \frac{b_3}{2}, 0\right) \left[\frac{B_3}{2}, \frac{B_2}{2}, \frac{B_1}{2}, \frac{B_1}{2}, \frac{B_2}{2}, \frac{B_3}{2}\right]$$

Es decir:

$$[7/24, 3/4, -1/24](2/3, -2/3, 1) \Rightarrow (-1/48, 3/8, 7/24, 3/8, -1/48, 0)[1/2, -2/6, 2/6, 2/6, -2/6, 1/2]$$

En [2] se explica que este método proviene de componer un método de Ruth de orden 3 de s etapas con paso $h/2$ con su adjunto también con paso $h/2$. Esta composición da lugar a un método simétrico de $2s$ etapas. Por la composición es, al menos, de orden 3. Por ser simétrico, tendrá orden par y, por tanto, será de orden, al menos 4.

Figura 4.2: Segundo método

Método de Ruth de orden 4: $(-1/48, 3/8, 7/24, 3/8, -1/48, 0)[1/2, -2/6, 2/6, 2/6, -2/6, 1/2]$

0	0	0	0	0	0
-1/48	0	0	0	0	0
-1/48	3/8	0	0	0	0
-1/48	3/8	7/24	0	0	0
-1/48	3/8	7/24	3/8	0	0
-1/48	3/8	7/24	3/8	-1/48	0
-1/48	3/8	7/48	3/8	-1/48	0
1/2	0	0	0	0	0
1/2	-2/6	0	0	0	0
1/2	-2/6	2/6	0	0	0
1/2	-2/6	2/6	2/6	0	0
1/2	-2/6	2/6	2/6	-2/6	0
1/2	-2/6	2/6	2/6	-2/6	1/2
1/2	-2/6	2/6	2/6	-2/6	1/2

En estas comparaciones hay que tener presente que los métodos de Ruth se implementan con un paso fijo, en contraste con la implementación del método ode45 que es de paso variable.

El problema Hamiltoniano que se va a utilizar para dicho experimento es el problema de Kepler. Este problema ya ha sido estudiado en capítulos anteriores.

Las ecuaciones del movimiento del problema descrito como un sistema diferencial de

primer orden:

$$\dot{y}(t) = f(y(t)) = \left[\frac{-y_3(t)}{(y_3(t)^2 + y_4(t)^2)^{3/2}}, \frac{-y_4(t)}{(y_3(t)^2 + y_4(t)^2)^{3/2}}, y_1(t), y_2(t) \right]$$

También vimos que el Hamiltoniano de este sistema es una magnitud que se conserva a lo largo del tiempo y, además, coincide con la energía total del sistema:

$$H = p_1 L_{p_1} + p_2 L_{p_2} - L = \frac{1}{2} (p_1^2 + p_2^2) + \frac{(-1)}{(q_1^2 + q_2^2)^{1/2}} = T + V.$$

Se van a considerar las condiciones iniciales siguientes:

$$q_1(0) = 1 - e, \quad q_2(0) = 0, \quad p_1(0) = 0, \quad p_2(0) = \sqrt{\frac{1+e}{1-e}},$$

siendo $0 < e < 1$ un parámetro real. La solución de este sistema bajo estas condiciones iniciales es 2π -periódica y en el plano (q_1, q_2) es una elipse de excentricidad e . En este caso, el valor constante del Hamiltoniano es:

$$H = \frac{1}{2} (p_1^2(0) + p_2^2(0)) + \frac{(-1)}{(q_1^2(0) + q_2^2(0))^{1/2}} = \frac{1}{2} \left(\frac{1+e}{1-e} \right) + \frac{-1}{1-e} = -\frac{1}{2}.$$

4.1.1. Resultados obtenidos

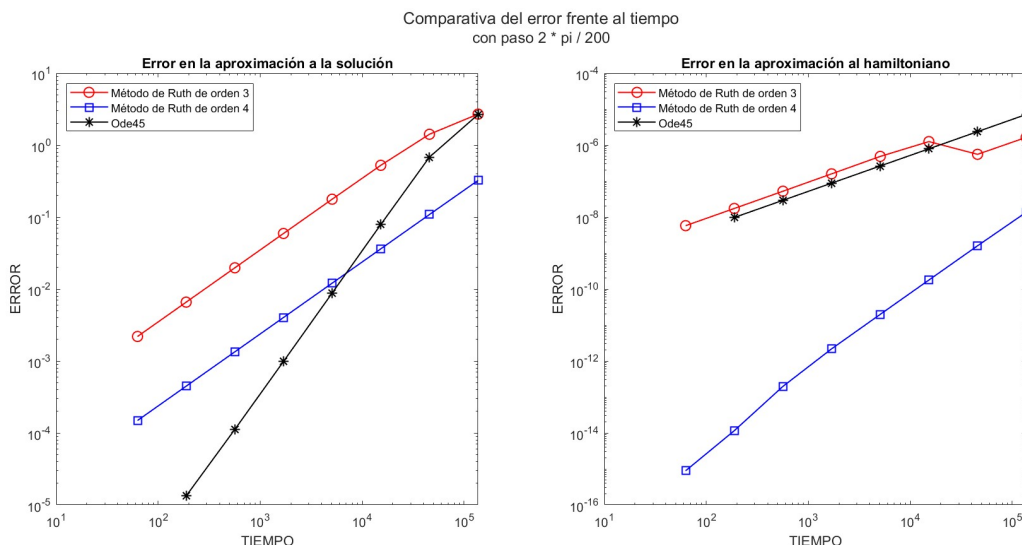
Presentamos resultados de algunas simulaciones numéricas para comparar la eficiencia relativa a la implementación de los métodos Runge-Kutta particionados simplécticos (métodos de Ruth de orden 3 y de orden 4) con los resultados de un software general de paso variable para la integración del problema de valores iniciales representado en estos experimentos por la función ode45 de Matlab (par encajado de métodos Runge-Kutta de orden 5).

Se ha realizado la integración tomando $e = 0,5$, con las anteriores condiciones iniciales. Se ha considerado un tiempo inicial $t_0 = 0$ y un tiempo final $t_F = 21870 \times 2\pi$, correspondiente a dar $21870 = 10 \times 3^7$ vueltas completas al periodo $2 * \pi$ de la solución. Se han obtenido aproximaciones en los tiempos intermedios $10 \times 3^k \times 2\pi$, $0 \leq k \leq 7$, y se han calculado los errores correspondientes a la aproximación de la solución y al Hamiltoniano (es decir, a la energía total del sistema).

Las primeras gráficas que mostraremos representan el error global de los métodos conforme aumenta la integración temporal en múltiplos del periodo de la solución teórica (correspondientes a los ocho tiempos intermedios considerados $10 \times 3^k \times 2\pi$, $0 \leq k \leq 7$), así como el error global en la cantidad conservada dada por el Hamiltoniano del problema.

Los resultados para $h = 2\pi/200$ corresponden a la elección del paso fijo en los métodos de Ruth para los que estos métodos proporcionan el mismo nivel de error en la solución que el método de paso variable ode45 en el tiempo final de la integración. Mostramos este hecho en la siguiente gráfica, a pesar de que para este tamaño de paso el método Runge-Kutta particionado simpléctico no presenta ventajas:

Figura 4.3



Las siguientes gráficas comparan el error en la solución y en el Hamiltoniano cuando el paso fijo de la interpolación de los métodos de Ruth es $2\pi/400$, $2\pi/800$ y $2\pi/1600$.

Figura 4.4

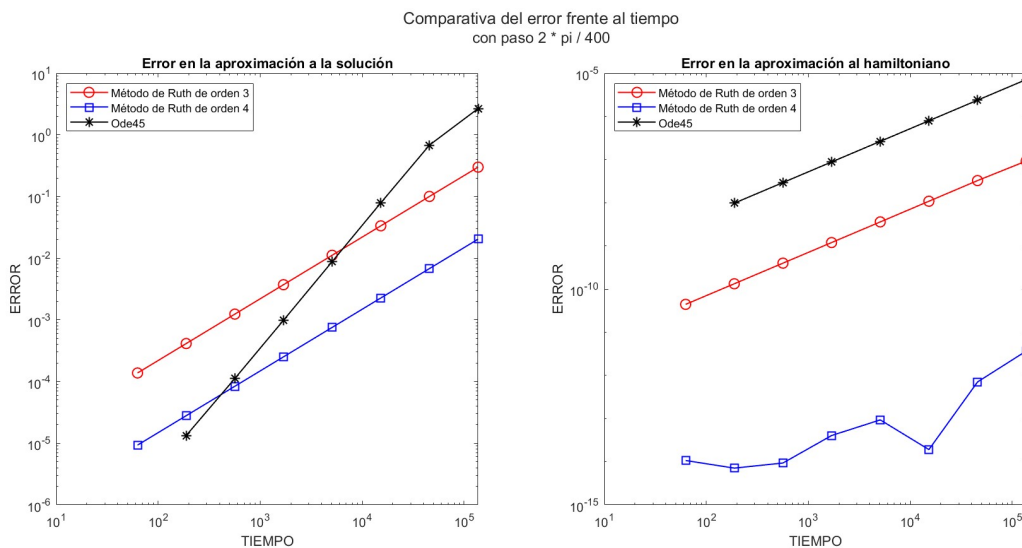


Figura 4.5

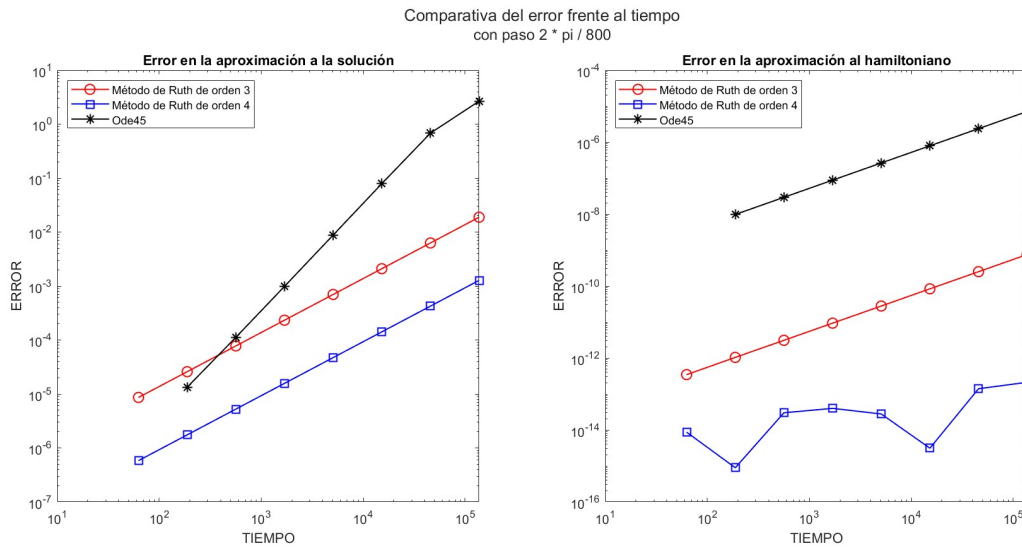
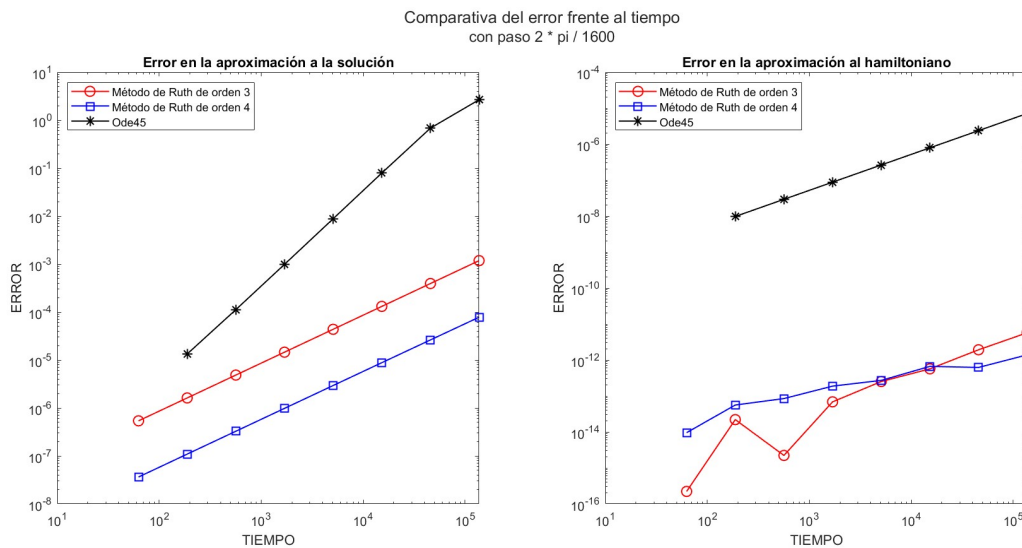


Figura 4.6



En relación con el comportamiento del error de la solución, conforme disminuye el tamaño del paso fijo h , el número de vueltas completas del periodo a partir del cual los métodos de Ruth son más ventajosos que el método ode45 disminuye, manifestándose las ventajas de los métodos simplécticos mucho antes. En el caso de $h = 2 * \pi / 200$, ya hemos visto que esto no se verificaba. Sin embargo, en cuanto al error del hamiltoniano, los métodos de Ruth se muestran mejores que el método ode45 en todo el intervalo temporal de integración para pasos $2\pi/400$, $2\pi/800$ y $2\pi/1600$, siento esta ventaja más efectiva cuanto menor es el paso de integración h . Debe observarse, que el nivel de error en el hamiltoniano es, para todos los métodos, varios órdenes de magnitud menor que el nivel

de error obtenido para la solución numérica.

Para una comparativa de la eficiencia de los diferentes métodos presentamos gráficas del error en la solución y en el hamiltoniano frente al número de evaluaciones de la función f que define el problema como sistema diferencial de primer orden. Como el coste computacional en ode45 viene dado en función de evaluaciones de la función f del sistema, computamos para los métodos de Ruth dos evaluaciones de f por etapa obviando que el cálculo de las etapas intermedias de $q = (y(3), y(4))$ no requiere de facto una evaluación de función.

Figura 4.7

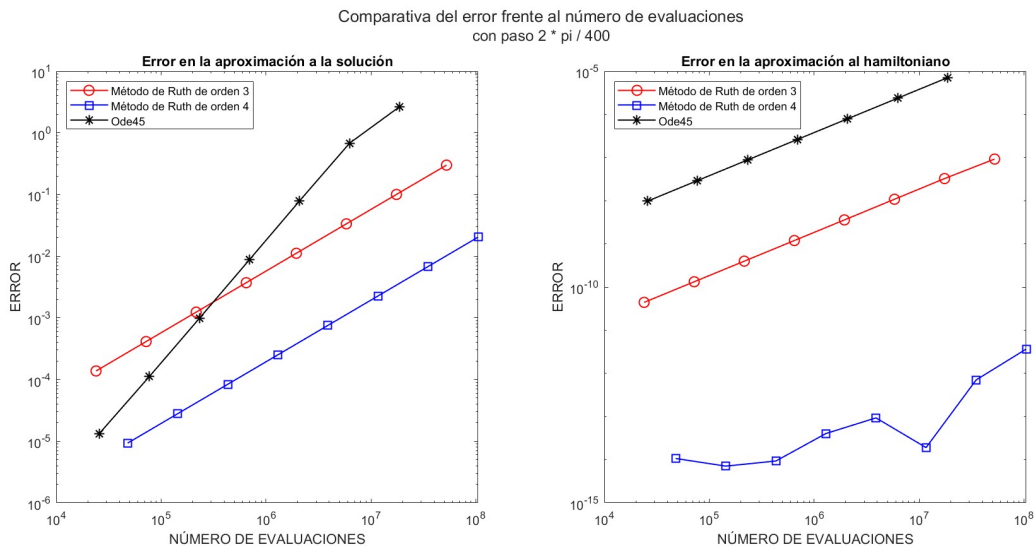


Figura 4.8

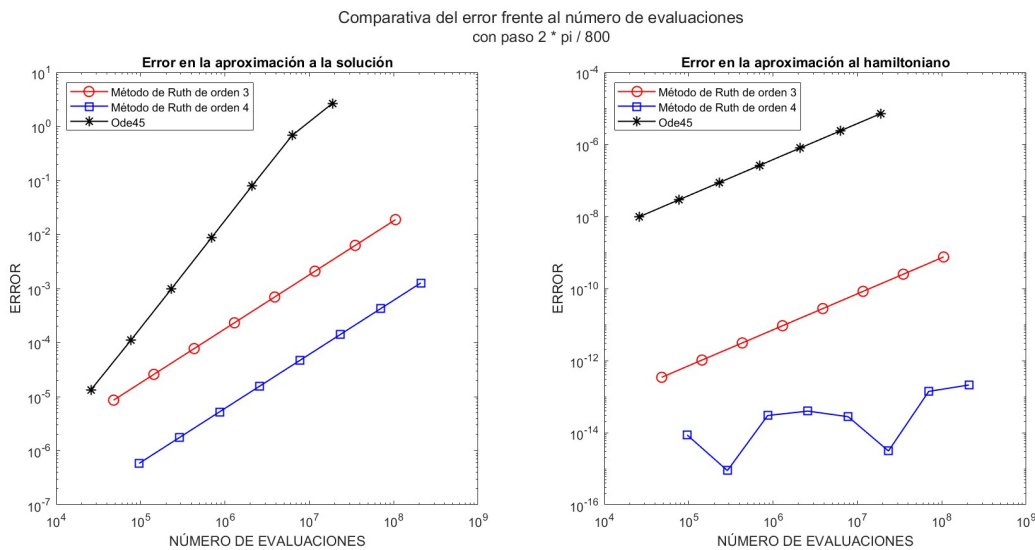
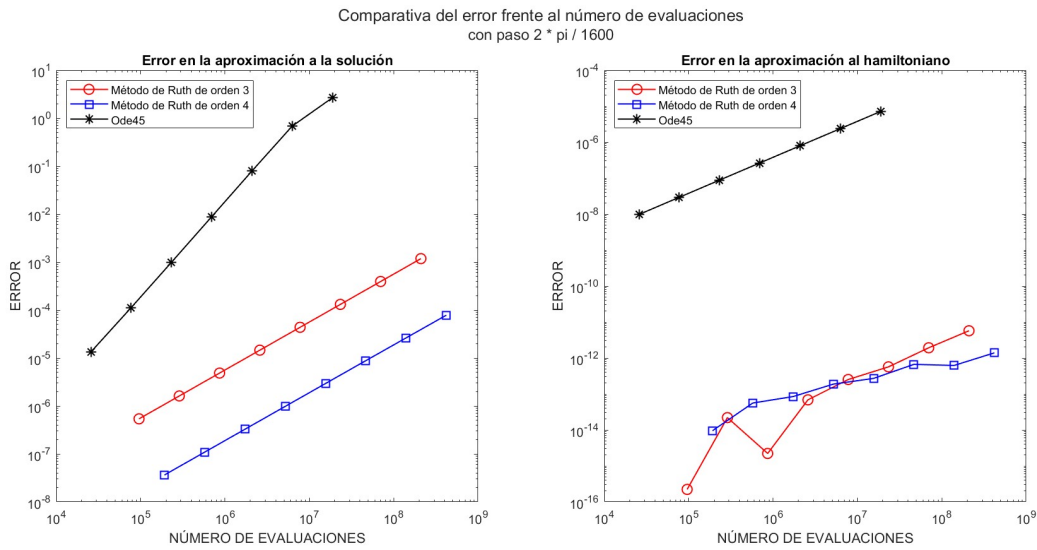


Figura 4.9



Presentamos también gráficas de eficiencia del error en la solución y en el hamiltoniano frente al tiempo de CPU (en segundos) del procesador en computación. Ambos tipos de gráficas de error (error frente a número de evaluaciones y error frente a tiempo de CPU) presentan resultados cualitativamente semejantes: ilustran de nuevo que para un mismo orden de tiempo de CPU o de número de evaluaciones de f , los métodos de Ruth son más eficientes y esta eficiencia se manifiesta antes en el tiempo de integración cuanto más pequeño es el paso de integración.

Figura 4.10

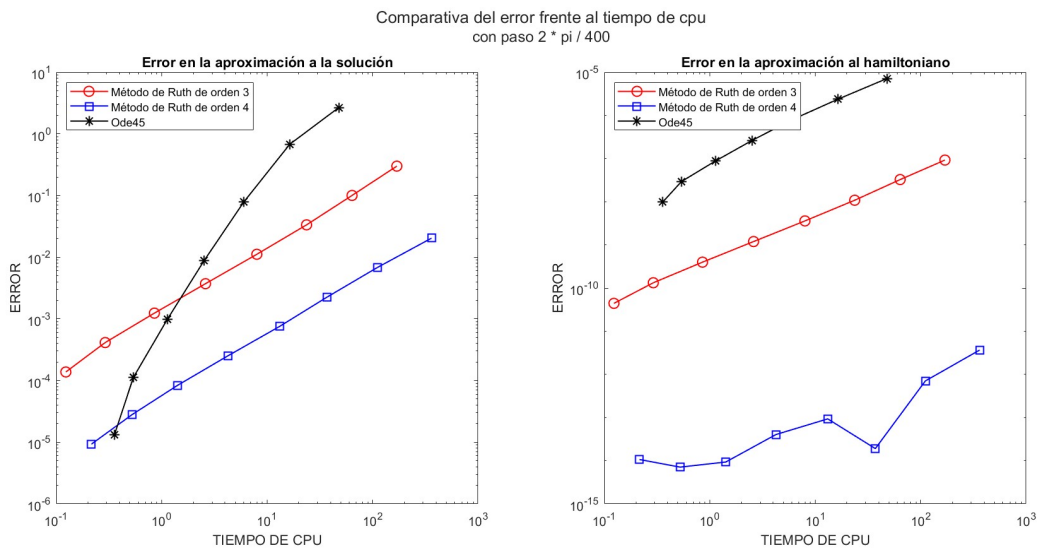


Figura 4.11

Comparativa del error frente al tiempo de cpu
con paso $2 * \pi / 800$

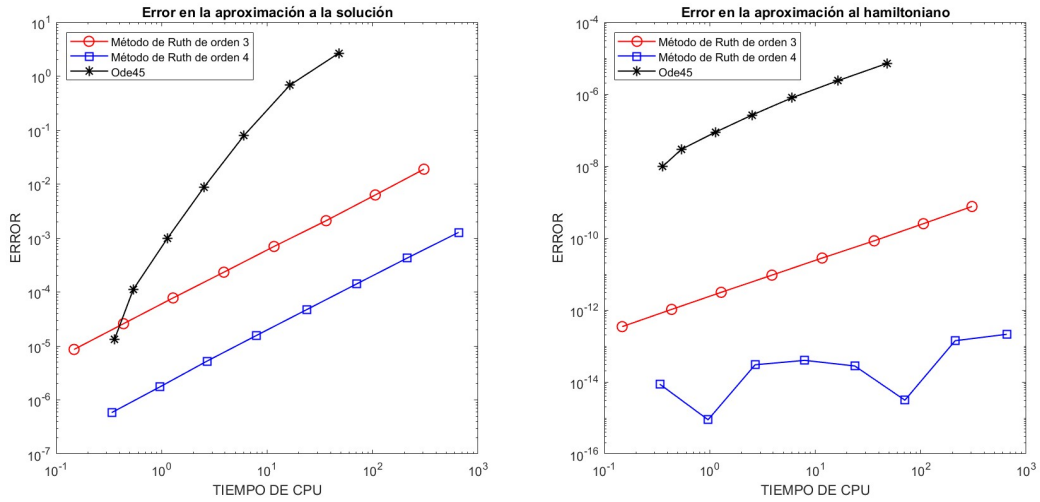
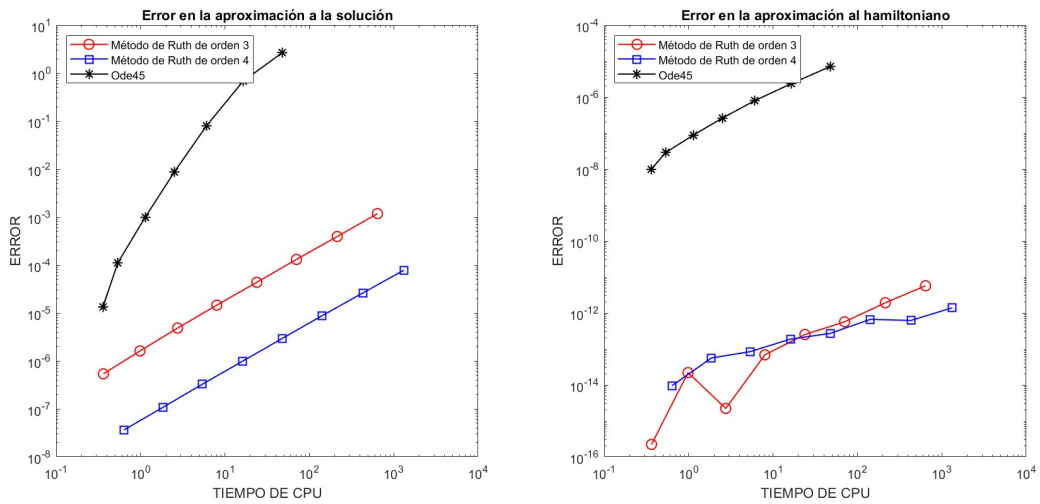


Figura 4.12

Comparativa del error frente al tiempo de cpu
con paso $2 * \pi / 1600$



Bibliografía

- [1] Abia, L. & Sanz-Serna, J.M. (1991). Order conditions for canonical Runge-Kutta schemes. *SIAM Journal on Numerical Analysis*, 28 (4), 1081-1096.
- [2] Abia, L. & Sanz-Serna, J.M. (1993). Partitioned Runge-Kutta methods for separable hamiltonian problems. *SIAM Journal on Numerical Analysis*, 60 (202), 617-634.
- [3] Hairer, E., Lubich, C. & Wanner, G. (2006). *Geometric Numerical Integration. Second edition*. Springer.
- [4] Hairer, E., Nørsett, S.P. & Wanner, G. (1987). *Solving Ordinary Differential Equations I. Nonstiff Problems*. Springer.
- [5] Murua, A. (1997). On order conditions for partitioned symplectic methods. *SIAM Journal on Numerical Analysis*, 34 (6), 2204-2211.
- [6] Murua, A. (1999). Formal series and numerical integrations, Part I: systems of ODEs and symplectic integrators. *Applied Numerical Mathematics*, 29, 221-251.
- [7] Murua, A. & Sanz-Serna, J.M. (1999). Geometric Integration: Numerical Solution of Differential Equations on Manifolds. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 357 (1754), 1079-1100.
- [8] Sanz-Serna, J.M & Calvo, M.P. (1994). *Numerical Hamiltonian Problems*. Chapman & Hall.

Apéndice A

Programas de Matlab

La función denominada *practica* es la función principal del programa. En ella se ejecutan los algoritmos correspondientes a cada uno de los métodos utilizados y se realizan las gráficas presentadas en la *sección 4.1.1*.

```
function [] = practica()
    % Datos iniciales
    exc = 0.5; % excentricidad
    t0 = 0; % tiempo inicial
    vueltas = 10 * 3^7;
    tF = 2 * pi * vueltas; % tiempo final
    times = 2 * pi * 10 * 3.^(0:7)'; % tiempos intermedios
    % condicion inicial:
    x0 = [ 0, sqrt( (1 + exc) / (1 - exc) ), 1 - exc, 0]';
    N0 = 10^3 * 3^7; % numero de pasos de la primera iteracion
    npasos = N0 * 2.^(1:4); % pasos utilizados
    % El paso es h = (tF - t0) ./ npasos
    % Es decir: h = 2 * pi ./ (100*2.^(1:4))

    ham = hamiltoniano(x0); % valor cte del hamiltoniano

    % Matrices para almacenar los errores
    % Las cuatro primeras columnas son para el de Ruth de orden 3,
    % las cuatro siguientes son para el de orden 4.
    errorsx      = zeros(8,8);
    errorsham    = zeros(8,8);
    errorsxode   = zeros(8,1);
    errorshamode = zeros(8,1);

    % Para almacenar el tiempo de CPU:
    tiemposcpu   = zeros(8,8);
    tiemposcpuode = zeros(8,1);
```

```

% Para almacenar numero de evaluaciones:
nevals = zeros(8,8);
nevalsode = zeros(8,1);

% calculamos la solucion numerica con cada uno de los
% metodos de Ruth

for ind = 0 : 1
    for i = 1 : 4
        % Metodo de Ruth
        [xx, nevals(:, i+4*ind), tiemposcpu(:, i+4*ind)] = ...
        ruth(@kepler, t0, tF, x0, npasos(i), ind, i+1);

        % ERRORES (en cada uno de los tiempos intermedios)
        for k = 1 : 8
            errorsx(k, i+4*ind) = norm(x0' - xx(k+1,:), 2);
            errorsham(k, i+4*ind) = ...
            norm(ham - hamiltoniano(xx(k+1,:)), 2);
        end
    end
end

% Metodo ode45
options = odeset('AbsTol', 1.0e-9, 'RelTol', 1.0e-9, 'Stats', 'on');

% Vamos a almacenar numero de evaluaciones, tiempo de CPU y
% errores en cada uno de los tiempos intermedios
tic
[~, yy, estado] = ode45(@kepler, [0, times(1)], x0, options);
tcpuode = toc;
neval = estado(3);
tiemposcpuode(1) = tcpuode;
nevalsode(1) = neval;
errorsxode(k, 1) = norm(x0' - yy(end,:), 2);
errorshamode(k, 1) = norm(ham - hamiltoniano(yy(end,:)), 2);

for k = 1 : 7
    tic
    [~, yy, estado] = ode45(@kepler, [times(k), times(k+1)], ...
        yy(end,:), options);
    tcpuode = toc;
    errorsxode(k+1, 1) = norm(x0' - yy(end,:), 2);
end

```

```

    errorshamode(k+1,1) = norm(ham - hamiltoniano(yy(end,:)),2);
    neval = estado(3);
    tiemposcpuode(k+1) = tiemposcpuode(k) + tcpuode;
    nevalsode(k+1) = nevalsode(k) + neval;
end

tiempos = ["2 * pi * 10", "2 * pi * 10 * 3", ...
"2 * pi * 10 * 3^2", "2 * pi * 10 * 3^3", ...
"2 * pi * 10 * 3^4", "2 * pi * 10 * 3^5", ...
"2 * pi * 10 * 3^6", "2 * pi * 10 * 3^7"];
pasos = ["2 * pi / 200", "2 * pi / 400", "2 * pi / 800", ...
"2 * pi / 1600"];

%FIGURAS
% error - tiempo, fijada la longitud de paso utilizada
for i = 1 : 4
    figure(i) % 1,2,3,4
    clf
    t = tiledlayout('horizontal');
    title(t, 'Comparativa del error frente al tiempo', ...
    ['con paso ', num2str(pasos(i))])

    % error en la aproximacion
    nexttile
    loglog(times, errorsx(:,i), 'ro-', 'MarkerSize', 8, ...
    'LineWidth', 1)
    hold on
    loglog(times, errorsx(:,i+4), 'b-', 'Marker', 'square', ...
    'MarkerSize', 8, 'LineWidth', 1) % orden 4
    hold on
    loglog(times, errorsxode, 'k*-', 'MarkerSize', 8, 'LineWidth', 1)
    hold off
    xlabel('TIEMPO')
    ylabel('ERROR')
    title('Error en la aproximacion a la solucion')
    legend('Metodo de Ruth de orden 3', ...
    'Metodo de Ruth de orden 4', 'Ode45', 'Location', 'northwest')

    % error en el hamiltoniano
    nexttile
    loglog(times, errorsham(:,i), 'ro-', 'MarkerSize', 8, ...
    'LineWidth', 1)
    hold on

```

```

loglog(times,errorsham(:,i+4),'b-','Marker','square',...
'MarkerSize',8,'LineWidth',1)
hold on
loglog(times,errorshamode,'k*-','MarkerSize',8,...
'LineWidth',1)
hold off
xlabel('TIEMPO')
ylabel('ERROR')
title('Error en la aproximacion al hamiltoniano')
legend('Metodo de Ruth de orden 3'...
,'Metodo de Ruth de orden 4','Ode45','Location',...
'northwest')

end

for j = 1 : 3
%ERROR - TIEMPO DE CPU
figure(j+4) % 5,6,7
clf
t = tiledlayout('horizontal');
title(t,'Comparativa del error frente al tiempo de cpu',...
['con paso ',num2str(pasos(j+1))])

% error en la aproximacion
nexttile
loglog(tiemposcpu(:,j+1),errorsx(:,j+1),'ro-','MarkerSize',...
8,'LineWidth',1)
hold on
loglog(tiemposcpu(:,j+5),errorsx(:,j+5),'b-','Marker',...
'square','MarkerSize',8,'LineWidth',1)
hold on
loglog(tiemposcpuode,errorsxode,'k*-','MarkerSize',...
8,'LineWidth',1)
hold off
xlabel('TIEMPO DE CPU')
ylabel('ERROR')
title('Error en la aproximacion a la solucion')
legend('Metodo de Ruth de orden 3'...
'Metodo de Ruth de orden 4','Ode45','Location','northwest')

% error en el hamiltoniano
nexttile
loglog(tiemposcpu(:,j+1),errorsham(:,j+1),'ro-',...

```

```

'MarkerSize',8,'LineWidth',1)
hold on
loglog(tiemposcpu(:,j+5),errorsham(:,j+5),'b-',...
,'Marker','square','MarkerSize',8,'LineWidth',1)
hold on
loglog(tiemposcpuode,errorshamode,'k*-', 'MarkerSize',8,...
'LineWidth',1)
hold off
xlabel('TIEMPO DE CPU')
ylabel('ERROR')
title('Error en la aproximacion al hamiltoniano')
legend('Metodo de Ruth de orden 3',...
'Metodo de Ruth de orden 4','Ode45','Location','northwest')

%ERROR – NUMERO DE EVALUACIONES
figure(j+7) % 8,9,10
clf
t = tiledlayout('horizontal');
title(t,'Comparativa del error frente al numero de...
evaluaciones',[ 'con paso ',num2str(pasos(j+1))])

% error en la aproximacion
nexttile
loglog(nevals(:,j+1),errorsx(:,j+1),'ro-', 'MarkerSize',8,...
'LineWidth',1)
hold on
loglog(nevals(:,j+5),errorsx(:,j+5),'b-', 'Marker',...
'square','MarkerSize',8,'LineWidth',1)
hold on
loglog(nevalsode,errorsxode,'k*-', 'MarkerSize',8,...
'LineWidth',1)
hold off
xlabel('NUMERO DE EVALUACIONES')
ylabel('ERROR')
title('Error en la aproximacion a la solucion')
legend('Metodo de Ruth de orden 3',...
'Metodo de Ruth de orden 4','Ode45','Location','northwest')

% error en el hamiltoniano
nexttile
loglog(nevals(:,j+1),errorsham(:,j+1),'ro-', 'MarkerSize',...
8,'LineWidth',1)
hold on

```

```

loglog(nevals(:,j+5),errorsham(:,j+5),'b-','Marker',...
'square','MarkerSize',8,'LineWidth',1)
hold on
loglog(nevalsode,errorshamode,'k*-','MarkerSize',8,...
'LineWidth',1)
hold off
xlabel('NUMERO DE EVALUACIONES')
ylabel('ERROR')
title('Error en la aproximacion al hamiltoniano')
legend('Metodo de Ruth de orden 3',...
'Metodo de Ruth de orden 4','Ode45','Location','northwest')

end

```

end

La función *kepler* evalúa el lado derecho del problema de Kepler escrito como un sistema diferencial de primer orden. Devuelve un vector columna con las componentes de dicha evaluación. Requiere la componente temporal para poder ser utilizada como función a evaluar en *ode45*.

```

function f = kepler(t,y)
    f = [- y(3) / ( sqrt( y(3)^2 + y(4)^2 ) )^3;
        - y(4) / ( sqrt( y(3)^2 + y(4)^2 ) )^3;
        y(1);
        y(2)];
end

```

La siguiente función evalúa el hamiltoniano y devuelve un escalar.

```

function ham = hamiltoniano(y)
    %evalua el hamiltoniano en y
    ham = (y(1)^2+y(2)^2)/2 - 1/sqrt(y(3)^2 + y(4)^2);
end

```

Esta función auxiliar ha sido utilizada para calcular los coeficientes y parámetros de los métodos de Ruth utilizados a partir de unos dados.

```

function [b,B,s] = coefs(b,B,ind)
    switch ind
        case 0
            s = 3;
        case 1
            s = 6;
            b = [b(3),b(2),2*b(1),b(2),b(3),0]./2;
            B = [B(3),B(2),B(1),B(1),B(2),B(3)]./2;
    end
end

```

end

La función *ruth* implementa los métodos de Ruth utilizados en el capítulo 4. Se almacenan únicamente las aproximaciones correspondientes a los tiempos intermedios prefijados con el fin de evitar problemas de memoria.

```
function [xx,neval,tcpu] = ruth(kepler , t0 , tF , x0 , npasos , ind , i)
    %t0 es el tiempo inicial
    %tF es el tiempo final
    %x0 es la condicion inicial , un vector columna
    %ind indica el metodo de Ruth utilizado
    %xx matriz que almacena la sol numerica calculada en cada
    %uno de los tiempos intermedios elegidos

    % Inicializamos
    xx = zeros(9,4); % almacena la sol numerica
    h = (tF - t0) / npasos; % paso
    nevals = 0; % evaluaciones iniciales
    neval = zeros(8,1);
    tcpu = zeros(8,1);
    taux = zeros(9,1);
    tt = 0;

    % coeficientes
    b = [7/24, 3/4, -1/24];
    B = [2/3, -2/3, 1];
    [b,B,s] = coefs(b,B,ind);

    % paso inicial
    pq = x0;
    xx(1,:) = pq;
    j = 1;
    num = 10^3 * 2^(i-1);

    switch ind

        case 0 %Ruth alternativo de orden 3
            for k = 1 : 8
                tic % inicializamos el contador temporal
                for l = j : num
                    for n = 1 : s
                        aux = kepler(tt,pq);
                        pq(1:2) = pq(1:2) + h*b(n)*aux(1:2);
                        % para el siguiente calculo se requiere
```

```

        % otra evaluacion , quedandonos solo con
        % las dos ultimas componentes , que
        % corresponden a la funcion identidad
        % sobre pq(1:2)
        pq(3:4) = pq(3:4) + h*B(n)*pq(1:2);
        nevals = nevals + 2;
    end
end
tcpus = toc; % paramos el contador temporal
% para que los tiempos de CPU que se almacenen
% se tomen desde el inicio , calculamos el tiempo
% que se tarda en obtener la aproximacion entre dos
% tiempos intermedios y sumamos el tiempo anterior.
taux(k+1) = taux(k) + tcpus;
j = l + 1;
% almacenamos la aproximacion a la solucion en
% el tiempo intermedio k-esimo
xx(k+1,:) = pq;
num = num * 3;
neval(k) = nevals;
end
tcpu(1:8) = taux(2:9);

case 1 %Ruth de orden 4
for k = 1 : 8
    tic
    for l = j : num
        for n = 1 : s
            pq(3:4) = pq(3:4) + h*B(n)*pq(1:2);
            aux = kepler(tt,pq);
            pq(1:2) = pq(1:2) + h*b(n)*aux(1:2);
            nevals = nevals + 2;
        end
    end
    tcpus = toc;
    taux(k+1) = taux(k) + tcpus;
    j = l + 1;
    xx(k+1,:) = pq;
    num = num * 3;
    neval(k) = nevals;
end
tcpu(1:8) = taux(2:9);

```



```
        otherwise
            error(['Selecciona 0 o 1 como ' ...
                  'valor para la variable ind'])
    end
end
```