

UNIVERSITAT JAUME I



DEPARTAMENT DE MATEMÀTIQUES

**Métodos de Integración Geométrica.  
Análisis y Algoritmos Numéricos**

Memoria presentada por

Mladen W. Nadinić Cruz

para optar al grado de doctor en Ciencias Matemáticas

Dirigida por Fernando Casas Pérez

CASTELLÓN

ESPAÑA

2015



# Dedicatoria

Este trabajo es el producto de un gran esfuerzo colectivo. Obviamente yo soy el responsable y visible de él, pero obedece al empeño de muchos. Mi profesor Fernando Casas, cuyo consejo, guía y dedicación, se aprecian en cada tema estudiado. Mis colegas y alumnos de la Universidad de Santiago de Chile, que siempre prestaron interés en el avance del proyecto. Y muy especialmente a mi esposa *Pamela* cuya paciencia, cariño y prestancia hicieron posible esta empresa.

*A mis padres*



# Índice general

<b>1. Introducción General</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Objetivos generales del proyecto de tesis doctoral . . . . .	2
1.3. Métodos de integración geométrica . . . . .	5
1.4. Preliminares matemáticos . . . . .	6
<b>2. Integradores en grupos de Lie para problemas lineales dependientes del tiempo</b>	<b>15</b>
2.1. Introducción . . . . .	15
2.2. Desarrollo de Magnus . . . . .	17
2.2.1. Generador de la expansión de Magnus . . . . .	21
2.2.2. La Expansión de Magnus y la teoría de perturbaciones dependiente del tiempo . . . . .	22
2.2.3. Métodos de Integración Numérica basados en la expansión de Magnus . . . . .	23
2.2.4. Integradores de Magnus . . . . .	25
2.3. Procedimiento iterativo de Voslamber . . . . .	31
2.3.1. Elementos previos . . . . .	31
2.3.2. Aproximación de Voslamber . . . . .	34
2.3.3. Un esquema numérico basado en el procedimiento de Voslamber . . . . .	37
2.3.4. Ejemplos numéricos . . . . .	38
2.4. Métodos de splitting basados en matrices triangulares . . . . .	43
2.4.1. Esquema general del algoritmo . . . . .	46
Construcción de un Integrador Geométrico . . . . .	49
Ejemplo numérico . . . . .	52
2.4.2. Algoritmo de orden 6 . . . . .	52
Ejemplo numérico . . . . .	56
2.5. Tamaño de paso adaptativo. . . . .	56
2.6. Aplicación al cálculo de la exponencial de una matriz con traza nula . . . . .	60

2.6.1.	Descripción del proceso . . . . .	61
2.6.2.	Experimentos numéricos . . . . .	64
<b>3.</b>	<b>Tratamiento de problemas no lineales mediante integradores en grupos de Lie</b>	<b>71</b>
3.1.	Métodos de Runge–Kutta–Munthe-Kaas . . . . .	72
3.2.	Desarrollo de Magnus para ecuaciones no lineales . . . . .	75
3.3.	Aplicaciones . . . . .	80
3.3.1.	La ecuación de Emden–Fowler . . . . .	80
	Resultados de un método Magnus modificado de cuarto orden . . . . .	82
	Algoritmo de Magnus con $\Omega^{[1]}$ exacta . . . . .	84
3.3.2.	Ecuación de Lane–Emden . . . . .	85
3.3.3.	Aplicación. Ecuación de Thomas–Fermi . . . . .	86
3.4.	Método de splitting basado en matrices triangulares . . . . .	88
3.4.1.	Procedimiento para el cálculo de $L_k$ . . . . .	91
3.4.2.	Procedimiento para el cálculo de $U_0(t)$ . . . . .	93
3.4.3.	Parte final del método . . . . .	94
3.4.4.	Ejemplo numérico . . . . .	95
<b>4.</b>	<b>Aspectos computacionales de la fórmula de Baker–Campbell–Hausdorff y la fórmula de Zassenhaus</b>	<b>97</b>
4.1.	Introducción . . . . .	97
4.2.	La fórmula de Baker–Campbell–Hausdorff . . . . .	98
4.3.	Una aplicación de la fórmula BCH . . . . .	107
4.3.1.	Ejemplos Numéricos . . . . .	112
4.4.	Fórmula de Zassenhaus . . . . .	113
4.4.1.	Introducción . . . . .	114
4.4.2.	Construcción de los términos de Zassenhaus. Una nueva recurrencia . . . . .	117
4.4.3.	Procedimiento . . . . .	120
4.4.4.	Independencia Lineal . . . . .	120
4.4.5.	Aspectos computacionales . . . . .	122
4.5.	Convergencia de la fórmula de Zassenhaus . . . . .	123
4.5.1.	Implementación . . . . .	125
4.5.2.	Algunos resultados numéricos . . . . .	127
4.5.3.	Comportamiento de la convergencia . . . . .	129
4.6.	Fórmula de Zassenhaus generalizada . . . . .	130
4.6.1.	Algoritmo de Zassenhaus Generalizado . . . . .	135
4.6.2.	Algunos aspectos sobre de la convergencia de la fórmula de Zassenhaus generalizada . . . . .	136

4.6.3. Estudio empírico de la convergencia . . . . .	137
4.7. Desarrollo de Wilcox . . . . .	139
4.7.1. Ejemplos . . . . .	144
<b>5. Conclusiones</b>	<b>149</b>
5.1. Futuros trabajos . . . . .	151
<b>Apéndices</b>	<b>153</b>
<b>A. Cálculo de la exponencial de una matriz</b>	<b>155</b>
A.1. Aproximación de Chebyshev . . . . .	156
A.2. Métodos basados en espacios de Krylov . . . . .	158
A.3. Métodos de splitting . . . . .	158
<b>B. Métodos de Runge–Kutta</b>	<b>161</b>
<b>C. Código expmS()</b>	<b>165</b>





# Capítulo 1

## Introducción General

### 1.1. Motivación

Las últimas décadas han sido muy fructíferas en la construcción de métodos numéricos que mejoran la integración clásica. La idea central de estos nuevos métodos es adaptarse al problema diferencial, por ejemplo al construir trayectorias numéricas que presenten propiedades análogas a las trayectorias exactas. Un ejemplo de ello son los avances en la integración numérica de problemas hamiltonianos, los cuales han contribuido a la consolidación de un campo de investigación denominado *Integración Geométrica* que permite la resolución de ecuaciones diferenciales conservando sus propiedades.

Se debe recordar que un evento que desencadenó el interés por la integración simpléctica de los sistemas hamiltonianos, y más generalmente por la *Integración Geométrica*, fue el descubrimiento simultáneo en los artículos [Las88], [SS88], [Sur88] del hecho de que existen métodos Runge-Kutta simplécticos. Así se encontraba un nexo entre los métodos de integración clásicos y la teoría de la Geometría Simpléctica.

En concreto, está siendo de gran interés el estudio y construcción de métodos numéricos cuyas soluciones muestren el mismo comportamiento asintótico y los mismos invariantes que el problema exacto. Por ello la necesidad de abordar el estudio de estos temas, para desarrollar nuevos métodos de integración geométrica, que puedan ser aplicados en diversas áreas de la ciencia aplicada, con costos de cómputo viables, y en muchos casos mejores que los procedimientos tradicionales, o bien, a veces con una precisión equivalente a la de algoritmos de mayor orden de complejidad.

Para desarrollar una disertación coherente y que combine equilibradamente el desarrollo teórico y práctico de estos temas, la estructura de esta tesis consta de cuatro capítulos. El primero contempla una descripción de los principales objetivos que persigue, así como una presentación introductoria de los temas teóricos que requiere el trabajo para los desarrollos presentados en los demás capítulos.

En el capítulo 2, se detalla el desarrollo de diversos integradores geométricos, entre ellos un nuevo método de *splitting*, presentando diversas experiencias numéricas, que enfatizan sus posibilidades y contratiempos. Además se desarrolla el método de Magnus, y métodos de orden 4, 6 y de paso adaptativo sobre  $SL(n)$ , basados en el método de *splitting* antes mencionado. También se presenta una versión de este algoritmo para el cálculo del caso constante  $e^{At}$  y el método de Voslamber para el problema  $Y' = \frac{1}{\varepsilon}A(t)Y$  cuando el parámetro  $\varepsilon$  es pequeño.

El tercer capítulo se centra en la Expansión de Magnus aplicada a sistemas diferenciales matriciales no lineales. Se analizan y construyen varios métodos de integración numérica, que son probados para problemas altamente oscilatorios como los modelados por las ecuaciones de Emden-Fower, Lane-Emden y Thomas-Fermi. Se finaliza este capítulo con el diseño de una versión especial del método *splitting* para el caso no lineal.

Por último, en el capítulo 4 se presenta la fórmula de Baker–Campbell–Hausdorff (BCH), con desarrollos clásicos de esta serie. Se construyen procedimientos nuevos para la fórmula de Zassenhaus, analizando su convergencia y aplicándola en algunos ejemplos. Se expanden los resultados anteriores para una fórmula de Zassenhaus generalizada, estudiando empíricamente la convergencia. Concluye el capítulo con el desarrollo de un nuevo procedimiento algorítmico para la fórmula de Wilcox [Wil67], que corresponde al caso continuo de la fórmula de Zassenhaus. Se discuten un par de ejemplos donde se aplica este procedimiento.

## 1.2. Objetivos generales del proyecto de tesis doctoral

El estudio y análisis de nuevos integradores geométricos implica una serie de desafíos, tanto teóricos como prácticos. La tesis se centra en ambos ejes,

exponiendo, desarrollando y aplicando la teoría y praxis necesarias en una amplia variedad de casos de estudio. Como toda tesis, ésta se planteó varios objetivos, los cuales se detallan en el mismo orden en que se presentan a lo largo del estudio.

- El objetivo inicial de este estudio es tratar en forma básica, pero precisa, los elementos matemáticos: definiciones, teoremas, proposiciones y resultados teóricos necesarios para el desarrollo posterior de la discusión. En los preliminares matemáticos, se desarrollan los conceptos fundamentales de álgebra y grupo de Lie, así como ejemplos de los principales grupos de Lie matriciales, que representan componentes de gran importancia en varios de los nuevos métodos numéricos desarrollados a lo largo de este trabajo. Para lograr un cabal entendimiento de los elementos abstractos involucrados en la génesis de los integradores geométricos, es necesario manejar la aplicación exponencial. Con ella se puede reconstruir completamente un grupo de Lie a partir de su álgebra de Lie asociada. Ello permite, en términos más prácticos, que su uso logre mantener o preservar propiedades inherentes al contexto físico del problema. En otras palabras, resaltamos el hecho que disponer de procedimientos numéricos eficientes para exponenciar matrices (o el logaritmo de ellas) permite resolver sistemas diferenciales matriciales, en forma eficiente y precisa, lo cual es fundamental en una gran gama de aplicaciones físico/matemáticas.
- El objetivo siguiente es el tratamiento de problemas lineales dependientes del tiempo. Para lograr esto recurrimos a integradores en grupos de Lie. Se requiere desarrollar la expansión de Magnus con modificaciones que la hacen más adecuada. También se desarrolla un procedimiento algorítmico para el método de Voslamber, adecuado para problemas lineales en el régimen adiabático. Se aplicará este nuevo procedimiento a varios problemas en de este tipo, analizando su eficiencia en comparación con otros procedimientos.
- También se estudiará una adaptación del algoritmo de *splitting* basado en matrices triangulares [Cas07], que permite trabajar con paso  $h$  variable de acuerdo a las necesidades de precisión que requiera la aproximación numérica en cuestión.
- Un desafío interesante es el de completar el tratamiento del paper [Cas07], en el sentido de adaptar el proceso para aproximar  $e^A$ , siendo  $A$  una matriz de traza nula. Con ello se contará con un nuevo procedimiento para el cálculo de la exponencial de una matriz.

- Otro de los objetivos importantes del trabajo es el tratamiento del caso no lineal, es decir, poder resolver aproximadamente el sistema de ecuaciones diferenciales no lineales dado por  $Y' = A(t, Y)Y$ . Resulta clave, en particular, la construcción de nuevos métodos de orden 4 y 6.
- La aplicación de los métodos de integración geométrica a problemas de sistemas de ecuaciones diferenciales cuya solución es altamente oscilatoria, y aparecen en diversas áreas de la física, es uno de los desafíos del presente trabajo.
- El objetivo del último capítulo es abordar el estudio y desarrollo de nuevos procedimientos para las fórmulas BCH y Zassenhaus, y su uso en diversas aplicaciones. Para desarrollar este desafío se plantean las siguientes acciones.

- Partimos haciendo una detallada introducción de estas famosas fórmulas, sin pretender ser una especie de estudio del arte en este tópico, pero sí exponiendo los elementos teóricos necesarios e imprescindibles para el trabajo posterior.
- La construcción de un nuevo procedimiento recursivo de los términos de la fórmula de Zassenhaus. Es decir, obtener  $C_n(X, Y) \in \mathcal{L}(X, Y)$  polinomio homogéneo de Lie en  $X$  e  $Y$  de grado  $n$  en el desarrollo

$$e^{X+Y} = e^X e^Y \prod_{n=2}^{\infty} e^{C_n(X,Y)} = e^X e^Y e^{C_2(X,Y)} e^{C_3(X,Y)} \dots e^{C_n(X,Y)} \dots,$$

- Este nuevo método ha de ser capaz de construir una secuencia  $\{C_n\}, n > 1$  de forma eficiente. Además, debe ser capaz cumplir criterios que permitan la convergencia del desarrollo en expansión de la serie.
- Realizar una generalización del nuevo algoritmo al problema

$$e^{\lambda A_1 + \lambda^2 A_2 + \dots + \lambda^n A_n + \dots} = e^{\lambda C_1} e^{\lambda^2 C_2} \dots e^{\lambda^n C_n} \dots$$

- Aprovechando las experiencias anteriores, se analizará una versión continua de Zassenhaus, conocida como desarrollo de Wilcox. Consideramos de nuevo el sistema lineal

$$\frac{dX(t)}{dt} = \lambda A(t)X(t), \quad X(0) = I$$

y desarrollaremos un nuevo procedimiento del tipo

$$X(t) = \prod_{k=1}^{\infty} e^{\lambda^k W_k(t)} = e^{\lambda W_1(t)} e^{\lambda^2 W_2(t)} e^{\lambda^3 W_3(t)} \dots$$

obteniendo recurrencias para generar los términos  $W_n$ .

- Para los nuevos desarrollos expuestos, se analizarán criterios de convergencia, así como se estudiará sus rendimientos computacionales.
- En todos los desarrollos de interés, se detallan las versiones algorítmicas, de pseudo código o en códigos de **Matlab** o *Mathematica*. Así, el lector interesado podrá usar los procedimientos para sus fines.

### 1.3. Métodos de integración geométrica

Algunos de los métodos más simples para hallar aproximaciones numéricas a la solución de las ecuaciones del movimiento de sistemas mecánicos, como por ejemplo el método de Euler o los métodos Runge-Kutta, son adecuados conceptualmente pero, en la práctica, presentan serios inconvenientes, especialmente cuando el sistema físico modelado presenta magnitudes conservadas, tales como energía, momento angular, etc. Incluso métodos más sofisticados y de orden más alto presentan el mismo inconveniente cuando se considera la evolución en períodos largos de tiempo.

Los métodos numéricos que además de minimizar el error muestran correctamente los invariantes de las soluciones (su geometría), reciben el nombre de *integradores geométricos*. *Integración Geométrica* es el término que se suele usar para describir los métodos numéricos utilizados para resolver ecuaciones diferenciales que preservan una o más propiedades físicas/matemáticas del sistema de forma exacta (hasta error de redondeo).

Como ejemplo, bastante estudiado, está el problema de la interacción gravitatoria de dos cuerpos de acuerdo con la ley de Newton, En este caso se preserva la energía y el momento angular. Además, y debido al hecho de que se trata de un sistema hamiltoniano, el sistema posee propiedades geométricas particulares, derivadas de la preservación de la forma simpléctica. Así, en particular, la preservación de la forma simpléctica para un sistema hamiltoniano con un grado de libertad es equivalente a la preservación del área en el espacio de fases, lo cual garantiza la existencia de soluciones periódicas.

Durante las últimas décadas han aparecido nuevos esquemas numéricos que preservan las estructuras consideradas relevantes en un problema dado.

La preservación de estas propiedades tiene importantes consecuencias de cara a la correcta simulación de sistemas físicos provenientes de áreas tan diversas como la mecánica celeste, los aceleradores de partículas, la dinámica molecular, la mecánica cuántica, la dinámica de fluidos, la dinámica del sólido rígido, etc.

La denominación *Integración Geométrica* es el término que se suele usar para describir los métodos numéricos utilizados para resolver ecuaciones diferenciales que preservan una o más propiedades físicas/matemáticas del sistema de forma exacta. Este área tiene su origen (a) en el convencimiento de la importancia de preservar las características cualitativas conocidas del problema continuo subyacente cuando la ecuación se discretiza y (b) en el hecho de que los métodos numéricos tradicionales no preservan esas propiedades cualitativas. En consecuencia, durante los últimos años, han aparecido nuevos esquemas numéricos que preservan las estructuras consideradas relevantes en un problema dado. La preservación de estas propiedades tiene importantes consecuencias en la simulación de ciertos sistemas dinámicos de interés en las aplicaciones.

Como valor añadido, resulta que el hecho de preservar las propiedades geométricas del sistema no sólo produce un comportamiento cualitativo mejorado de la aproximación, sino que también permite efectuar una integración a tiempos más largos que los métodos tradicionales, debido al favorable mecanismo existente en la propagación de errores. La base teórica para este superior comportamiento estriba en la idea del análisis regresivo del error. Lo anterior ha permitido que el área de la integración geométrica haya incrementando su desarrollado intensamente en los últimos tiempos, permitiendo construir nuevos métodos y mejorando potenciales aplicaciones y simulaciones complejas en muy diversos campos.

De entre los integradores geométricos, la clase de métodos diseñados para sistemas de ecuaciones diferenciales que evolucionan en grupos de Lie son sumamente interesantes, debido sobre todo a sus aplicaciones. La próxima sección trata de los principales elementos matemáticos necesarios para los desarrollos expuestos en este trabajo.

## 1.4. Preliminares matemáticos

Para poder seguir y comprender los temas que trata esta tesis, se requiere de varios elementos matemáticos, que en esta sección se definen, así como de cierta notación y terminología pertinente. Como gran parte de la exposición trata de elementos en grupos y álgebras de Lie asociadas, aunque la mayor parte de los resultados discutidos en estas páginas son válidos en un contexto

más general, partiremos considerando los elementos teóricos principales.

Los grupos de Lie son importantes en análisis matemático, física y geometría porque sirven para describir la simetría de estructuras analíticas. Como punto de partida, definimos el concepto de *grupo y álgebra de Lie*, para seguir desarrollando las ideas de *integradores en grupos de Lie matriciales*.

Por un grupo de Lie  $\mathcal{G}$  entendemos un conjunto que combina una estructura algebraica con una topológica. A nivel algebraico, dos elementos de  $\mathcal{G}$  se pueden combinar por una ley interna de composición para producir un tercer elemento también en  $\mathcal{G}$ . La ley requiere ser asociativa, tener un elemento identidad y cada elemento debe tener un inverso.

**Definición.** Un *grupo de Lie* es una variedad diferenciable  $\mathcal{G}$ , con una estructura de grupo, para la que las aplicaciones

$$(a, b) \in \mathcal{G} \times \mathcal{G} \longrightarrow ab \in \mathcal{G}, \quad a \in \mathcal{G} \longrightarrow a^{-1} \in \mathcal{G}$$

son diferenciables. Habitualmente denotaremos con  $e \in \mathcal{G}$  el elemento neutro del grupo y con notación multiplicativa la operación del grupo. Llamaremos *morfismo* de grupos de Lie a todo *homomorfismo* de grupos de Lie que sea diferenciable.

Para cada elemento  $a \in \mathcal{G}$  definimos las aplicaciones

$$R_a : b \in \mathcal{G} \longrightarrow ba \in \mathcal{G}, \quad L_a : b \in \mathcal{G} \longrightarrow ab \in \mathcal{G},$$

consistentes en trasladar por la derecha y por la izquierda respectivamente, el grupo mediante el elemento  $a$ .

Los grupos de Lie se clasifican con respecto a sus propiedades algebraicas (simple, semisimple, resoluble, nilpotente, abeliano), su conexidad (conexo o no conexo) y su compacidad [Mag54].

A cada grupo de Lie podemos asociar un álgebra de Lie que captura totalmente la estructura local del grupo.

Un álgebra de Lie  $\mathfrak{g}$  es un espacio vectorial cuyos elementos pueden ser combinados con una segunda ley, el corchete de Lie, que representamos por  $[A, B] = C$ , con  $A, B, C$  elementos de  $\mathfrak{g}$ , de tal manera que la ley es bilineal, antisimétrica, es decir,

$$[A, B] = -[B, A] \tag{1.1}$$

y satisface la identidad de Jacobi,

$$[A, [B, C]] + [B, [C, A]] + [C, [A, B]] = 0. \tag{1.2}$$

Un ejemplo clásico de álgebra de Lie es el conjunto  $\mathfrak{gl}(n)$  de todas las matrices  $n \times n$  no singulares. En ese caso podemos elegir como corchete de Lie el conmutador,

$$[A, B] = AB - BA, \quad A \in \mathfrak{gl}(n), \quad B \in \mathfrak{gl}(n), \tag{1.3}$$

con  $AB$  representando el producto común de matrices. Para toda la exposición pensamos en matrices reales o complejas.

Si consideramos un álgebra finita de Lie con dimensión  $d$  y denotamos por  $\{A_1, \dots, A_d\}$  a una base, entonces se tendrá

$$[A_i, A_j] = c_{ij}^1 A_1 + \dots + c_{ij}^d A_d, \quad (1.4)$$

donde los coeficientes  $c_{ij}^k$  son las llamadas *constantes de estructura* del álgebra de Lie con respecto a la base dada.

Como ejemplo, podemos citar lo que acontece con el álgebra de Lie  $\mathfrak{sl}(2, \mathbb{R})$  de las matrices  $2 \times 2$  de números reales con traza nula, la cual tiene una base formada por las matrices

$$e_1 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad e_2 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad e_3 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}.$$

Las constantes de estructura no nulas son:  $c_{12}^2 = 2$ ,  $c_{13}^3 = -2$ ,  $c_{23}^1 = 1$ .

Debido a la independencia lineal se puede afirmar que las constantes de estructura cumplen una ley de anticonmutatividad:

$$[A_i, A_j] = -[A_j, A_i] \Rightarrow c_{ij}^k = -c_{ji}^k. \quad (1.5)$$

Asociado con cualquier  $A \in \mathfrak{g}$  se puede definir un operador lineal  $\text{ad}_A : \mathfrak{g} \rightarrow \mathfrak{g}$  que actúa según

$$\text{ad}_A B = [A, B], \quad \text{ad}_A^j B = [A, \text{ad}_A^{j-1} B], \quad \text{ad}_A^0 B = B, \quad j \in \mathbb{N}, B \in \mathfrak{g}. \quad (1.6)$$

También será de interés en el caso matricial la exponencial del operador anterior,

$$\text{Ad}_A = \exp(\text{ad}_A), \quad (1.7)$$

cuya acción sobre  $B \in \mathfrak{g}$  corresponde a

$$\text{Ad}_A(B) = \exp(A) B \exp(-A) = \sum_{k=0}^{\infty} \frac{1}{k!} \text{ad}_A^k B, \quad B \in \mathfrak{g}. \quad (1.8)$$

A lo largo de este trabajo manejaremos con frecuencia tipos especiales de matrices, particularmente ortogonales, unitarias y simplécticas. Presentamos a continuación su caracterización y la notación que se utilizará para el grupo y álgebra de Lie correspondiente.

El grupo ortogonal especial,  $SO(n)$ , es el conjunto de todas las matrices  $n \times n$  sobre los reales con determinante unidad, que satisfacen  $A^T A = A A^T = I$ , donde  $A^T$  es la transpuesta de  $A$  y  $I$  denota la matriz identidad. El



correspondiente álgebra  $\mathfrak{so}(n)$  está formado por las matrices antisimétricas. Una matriz  $A$   $n \times n$  compleja es llamada unitaria si  $A^\dagger A = AA^\dagger = I$ , donde  $A^\dagger$  es la matriz conjugada traspuesta de  $A$ . El grupo especial unitario,  $SU(n)$ , es el conjunto de todas las matrices  $n \times n$  unitarias con determinante uno. Su álgebra asociada  $\mathfrak{su}(n)$  consiste en las matrices antihermíticas de traza cero. Especial relevancia en algunos problemas de la mecánica cuántica que discutiremos tendrá el caso  $n = 2$ . Una base conveniente para  $\mathfrak{su}(2)$  está formada por las matrices de Pauli

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (1.9)$$

las cuales verifican

$$\sigma_j \sigma_k = \delta_{jk} + i\epsilon_{jkl} \sigma_l, \quad (1.10)$$

de donde se deduce la identidad

$$[\sigma_j, \sigma_k] = 2i\epsilon_{jkl} \sigma_l. \quad (1.11)$$

Aquí el símbolo  $\epsilon_{jkl}$  denota el tensor de Levi-Civita, definido mediante

$$\epsilon_{jkl} = \begin{cases} +1 & \text{si } (j, k, l) \text{ es } (1, 2, 3), (2, 3, 1), (3, 1, 2) \\ -1 & \text{si } (j, k, l) \text{ es } (3, 2, 1), (1, 3, 2), (2, 1, 3) \\ 0 & \text{si } j = k \text{ o } k = l \text{ o } l = j \end{cases}$$

Otra identidad útil en la práctica es la siguiente: dados  $\mathbf{a}$  y  $\mathbf{b}$  vectores de  $\mathbb{R}^3$ ,

$$(\mathbf{a} \cdot \boldsymbol{\sigma})(\mathbf{b} \cdot \boldsymbol{\sigma}) = \mathbf{a} \cdot \mathbf{b} I + i(\mathbf{a} \times \mathbf{b}) \cdot \boldsymbol{\sigma}, \quad [\mathbf{a} \cdot \boldsymbol{\sigma}, \mathbf{b} \cdot \boldsymbol{\sigma}] = 2i(\mathbf{a} \times \mathbf{b}) \cdot \boldsymbol{\sigma}. \quad (1.12)$$

En sistemas hamiltonianos, el grupo simpléctico  $Sp(n)$  juega un papel fundamental. Este grupo es el formado por las matrices reales  $2n \times 2n$  que satisfacen

$$A^T J A = J, \quad \text{con} \quad J = \begin{pmatrix} O_n & I_n \\ -I_n & O_n \end{pmatrix} \quad (1.13)$$

La matriz  $I_n$  corresponde a la identidad y  $O_n$  a la matriz de ceros, ambas de dimensión  $n$ . El álgebra de Lie asociada  $\mathfrak{sp}(n)$  está formada por las matrices que verifican  $B^T J + J B = O_{2n}$ . Éste se puede considerar como un caso particular del llamado grupo  $J$ -ortogonal [Pos94], definido como

$$O_J(n) = \{A \in GL(n) : A^T J A = J\}, \quad (1.14)$$

donde  $GL(n)$  es el grupo de todas las matrices  $n \times n$  reales no singulares, y  $J$  es una matriz constante en  $GL(n)$ . Se recupera el grupo ortogonal cuando  $J = I$ ,

el grupo simplectico  $\text{Sp}(n)$  cuando  $J$  es la matriz simpléctica básica dada en (1.13), y el grupo de Lorentz  $\text{SO}(3, 1)$  cuando  $J = \text{diag}(1, -1, -1, -1)$ . El álgebra de Lie asociada es

$$\mathfrak{o}_J(n) = \{B \in \mathfrak{gl}(n, \mathbb{R}) : B^T J + JB = O\}, \quad (1.15)$$

donde  $\mathfrak{gl}(n, \mathbb{R})$  es el álgebra de Lie de todas las matrices  $n \times n$  de números reales. Si  $B \in \mathfrak{o}_J(n)$ , entonces la transformación de Cayley

$$A = (I - \alpha B)^{-1}(I + \alpha B) \quad (1.16)$$

proporciona una matriz  $A$   $J$ -ortogonal.

Otro importante grupo matricial de Lie es  $\text{SL}(n)$ , compuesto por todas las matrices  $n \times n$  de números reales con determinante uno. Su correspondiente álgebra de Lie  $\mathfrak{sl}(n)$  comprende a todas las matrices con traza cero.

A manera de resumen, la siguiente tabla muestra algunos grupos de Lie matriciales involucrados (varios de ellos) en el desarrollo de los integradores geométricos abordados en este trabajo.<sup>1</sup>

Grupo de Lie	Definición	Dimensión
Lineal general, $\mathbb{R}$	$\text{GL}(n, \mathbb{R}) = \{A \in \mathbb{R}^{n \times n} : \det A \neq 0\}$	$n^2$
Lineal especial	$\text{SL}(n, \mathbb{R}) = \{A \in \mathbb{R}^{n \times n} : \det A = 1\}$	$n^2 - 1$
Ortogonal	$\text{O}(n) = \{A \in \mathbb{R}^{n \times n} : A^T A = \mathbf{id}\}$	$n(n + 1)/2$
Especial ortogonal	$\text{SO}(n) = \{A \in \text{O}(n) : \det A = 1\}$	$n(n + 1)/2$
Lineal general, $\mathbb{C}$	$\text{GL}(n, \mathbb{C}) = \{A \in \mathbb{C}^{n \times n} : \det A \neq 0\}$	$4n^2$
Unitario	$\text{U}(n) = \{A \in \mathbb{C}^{n \times n} : A^* A = \mathbf{id}\}$	$n^2$
Especial unitario	$\text{SU}(n) = \{A \in \text{U}(n) : \det A = 1\}$	$n^2 - 1$
Simpléctico	$\text{Sp}(n) = \{A \in \mathbb{H}(n) : A^* A = \mathbf{id}\}$	$2n^2 + 2$

Será común a lo largo del presente trabajo el utilizar el concepto de norma de una matriz. Diremos que la norma de una matriz cuadrada de dimensión  $n$  es un escalar no negativo simbolizado por  $\|A\|$ , que satisface

- a)  $\|A\| \geq 0$  para toda  $A$  y  $\|A\| = 0$  si y sólo si  $A = O_n$ .
- b)  $\|\alpha A\| = |\alpha| \|A\|$ , para todo escalar  $\alpha$ .
- c)  $\|A + B\| \leq \|A\| + \|B\|$ .

Muy a menudo se añade la propiedad sub-multiplicativa

$$\|AB\| \leq \|A\| \|B\|, \quad (1.17)$$

<sup>1</sup>Aquí  $\mathbb{H}$  representa el cuerpo de los cuaterniones.

pero no todas las normas de la matriz satisfacen esta condición [GL96].

Existen diversas normas matriciales. Entre las más comunes figura la llamada  $p$ -norma,  $\|A\|_p$ , y la norma de Frobenius,  $\|A\|_F$ .

Definimos ambas normas, para una matriz  $A$  con elementos  $a_{ij}$ ,  $i, j = 1 \dots n$ , mediante

$$\|A\|_p = \max_{\|\mathbf{x}\|_p=1} \|A\mathbf{x}\|_p \quad (1.18)$$

$$\|A\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^n |a_{ij}|^2} = \sqrt{\text{tr}(A^\dagger A)}, \quad (1.19)$$

donde  $\|\mathbf{x}\|_p = (\sum_{j=1}^n |x_j|^p)^{\frac{1}{p}}$  y  $\text{tr}(A)$  es la traza de la matriz  $A$ . Aunque ambos verifican (1.17), la  $p$ -norma tiene la importante propiedad de que para cada matriz  $A$  y  $\mathbf{x} \in \mathbb{R}^n$  se tiene  $\|A\mathbf{x}\|_p \leq \|A\|_p \|\mathbf{x}\|_p$ . La  $p$ -norma más utilizada corresponde a  $p = 1$ ,  $p = 2$  y  $p = \infty$ . En particular, la 2-norma corresponde a la norma euclídea, también llamada norma espectral y se puede caracterizar como la raíz cuadrada del mayor valor propio de  $A^\dagger A$ . Es frecuente utilizar la inecuación que relaciona la norma de Frobenius y la norma espectral, de la siguiente forma:

$$\|A\|_2 \leq \|A\|_F \leq \sqrt{n} \|A\|_2. \quad (1.20)$$

De hecho, la última desigualdad se puede hacer más precisa [Tyr97]:

$$\|A\|_F \leq \sqrt{\text{rango}(A)} \|A\|_2. \quad (1.21)$$

Considérese una matriz en un álgebra de Lie  $\mathfrak{g}$  y una norma que satisface la propiedad (1.17). Entonces  $\|[A, B]\| \leq 2\|A\|\|B\|$ , y el operador  $\text{ad}$  definido por (1.6) está acotado por

$$\|\text{ad}_A\| \leq 2\|A\|$$

para una matriz  $A$ .

Una norma matricial se dice que es unitariamente invariante si  $\|UAV\| = \|A\|$  siempre que  $U, V$  son matrices unitarias. La norma de Frobenius y la  $p$ -norma son ambas unitariamente invariantes [HJ85].

En algunas de las fórmulas más básicas para la expansión de Magnus aparecerán los llamados números de Bernoulli  $B_n$ , que son definidos a través de la función generatriz [AS65]

$$\frac{te^{zt}}{e^t - 1} = \sum_{n=0}^{\infty} B_n(z) \frac{t^n}{n!}, \quad |t| < 2\pi \quad (1.22)$$

con  $B_n = B_n(0)$ . Equivalentemente,

$$\frac{x}{e^x - 1} = \sum_{n=0}^{\infty} \frac{B_n}{n!} x^n,$$

y

$$\frac{e^x - 1}{x} = \sum_{n=0}^{\infty} \frac{1}{(n+1)!} x^n.$$

Los primeros números de Bernoulli distintos de cero son  $B_0 = 1$ ,  $B_1 = -\frac{1}{2}$ ,  $B_2 = \frac{1}{6}$ ,  $B_4 = -\frac{1}{30}$ . En general se tendrá  $B_{2m+1} = 0$  para  $m \geq 1$ .

Un grupo de Lie  $\mathcal{G}$  tiene asociada un álgebra de Lie  $\mathfrak{g}$ . Para estudiar la relación entre la estructura de estos dos objetos se emplea el mapeo exponencial  $\exp : \mathfrak{g} \rightarrow \mathcal{G}$ , el cual lleva líneas rectas que pasan por el origen en  $\mathfrak{g}$  a subgrupos uniparamétricos<sup>2</sup> de  $\mathcal{G}$ . Formalmente el mapeo exponencial se define de la siguiente manera: Para  $X \in \mathfrak{g}$ , hay un único homomorfismo analítico  $\delta_X : \mathbb{R} \rightarrow \mathcal{G}$ , es decir, un único subgrupo paramétrico, tal que  $\delta'_X(0) = X$ . De esta manera, el mapeo exponencial es dado por  $X \rightarrow \exp(X) \doteq \delta_X(1)$ .

Un hecho interesante es que si tenemos  $\mathfrak{g}$  el álgebra de Lie de un grupo de Lie  $\mathcal{G}$ , y  $\sigma \in \mathcal{G}$ , entonces tenemos un isomorfismo analítico de  $\mathfrak{g}$  sobre él mismo dado por  $I_\sigma : \mathfrak{g} \rightarrow \sigma\mathfrak{g}\sigma^{-1}$ . Luego el mapeo (1.7) es un homomorfismo analítico de  $\mathcal{G}$  en  $GL(\mathfrak{g})$ , que satisface  $\text{Ad}(\exp(X)) = e^{\text{Ad}_X}$ , es decir, el siguiente diagrama conmuta:

$$\begin{array}{ccc} \mathfrak{g} & \xrightarrow{\text{ad}} & \mathfrak{gl}(\mathfrak{g}) \\ \exp \downarrow & & \downarrow e \\ \mathcal{G} & \xrightarrow{\text{Ad}} & GL(\mathfrak{g}) \end{array}$$

Para un tratamiento más exhaustivo de los grupos y álgebras de Lie resulta pertinente la referencia [Gil74].

Para el grupo de Lie  $GL(n, \mathbb{R})$ , cuya álgebra de Lie es  $\mathfrak{gl}(n, \mathbb{R})$  que se identifica con las matrices reales de  $n \times n$ , el mapeo exponencial no es más que la exponencial de matrices  $\exp(A) = e^A$ , para  $A \in \mathfrak{gl}(n, \mathbb{R})$ .

En general, sea  $A \in M_n(\mathbb{K})$ , con  $A = (a_{ij}) | a_{ij} \in \mathbb{K}, i, j = 1, \dots, n$ , donde  $\mathbb{K}$  denota un cuerpo. La exponencial de  $A$ ,  $\exp(A)$ , se define por

---

<sup>2</sup>Un grupo uniparamétrico o subgrupo uniparamétrico es un subconjunto de un grupo de Lie de dimensión uno.

$$\exp(A) \equiv e^A = I + A + \frac{A^2}{2!} + \dots = \sum_{n \geq 0} \frac{A^n}{n!}. \quad (1.23)$$

Notemos que, aparentemente,  $\exp(A)$  debiera ser también una matriz en  $M_n(\mathbb{K})$  ya que en su definición sólo estamos usando las operaciones matriciales básicas en este espacio, por lo que cada término en (1.23) está bien definido. Sin embargo, el problema en este caso es garantizar la convergencia de la serie en (1.23), lo cual nos permitirá estar seguros de que  $\exp(A)$  está bien definida para cualquier matriz  $A$ .

La serie definida en (1.23) converge si cada una de las entradas

$$(I)_{ij} + (A)_{ij} + \left(\frac{A^2}{2!}\right)_{ij} + \dots + \left(\frac{A^k}{k!}\right)_{ij} + \dots$$

converge. Esto nos motiva para el siguiente resultado

**Proposición 1** Sea  $A \in M_n(\mathbb{K})$  y sea  $m = \max\{|a_{ij}|, 1 \leq i, j \leq n\}$ .

Si  $A^p = (a_{ij}^{(p)})$ , entonces  $|a_{ij}^{(p)}| \leq (nm)^p, \forall i, j$ .

En consecuencia la serie (1.23) converge y la matriz  $e^A$  está bien definida.

Se tienen además las siguientes propiedades [Bak02]:

**Proposición 2** Sea  $A \in M_n(\mathbb{K})$ .

- i) Para  $u, v \in \mathbb{K}$ ,  $e^{(u+v)A} = e^{uA}e^{vA}$
- ii)  $e^A \in \text{GL}_{\mathbb{K}}$  y  $(e^A)^{-1} = e^{-A}$
- iii) Si  $A$  y  $B$  conmutan entonces  $e^{(A+B)} = e^A e^B$
- iv) Si  $\|A - I\| < 1$ , entonces  $e^{\ln(A)} = A$



## Capítulo 2

# Integradores en grupos de Lie para problemas lineales dependientes del tiempo

### 2.1. Introducción

Los sistemas de ecuaciones diferenciales lineales aparecen en modelos de problemas de gran importancia en el mundo real. Por ejemplo: problemas mecánicos de acoplamiento de resortes, problemas eléctricos, evolución de poblaciones, modelos depredador/presa, mezclas de distintas sustancias en un estanque, entre muchas otras. La amplia variedad de fenómenos que permiten modelar los sistemas de ecuaciones diferenciales ha originado un extenso estudio de los mismos.

Aunque aparentemente sencillos, los sistemas lineales dependientes del tiempo a menudo requieren para su resolución el uso de métodos numéricos de integración. En este capítulo se analizan varias técnicas que permiten obtener tanto aproximaciones analíticas como numéricas para este tipo de sistemas. Empezaremos presentando las principales características del desarrollo de Magnus para después analizar la modificación propuesta por Voslamber. A continuación presentaremos un esquema particularmente apropiado para sistemas lineales matriciales del tipo  $Y' = A(t)Y$  cuando  $Y$  evoluciona en el grupo  $SL(n)$ . Finalmente, se presenta una versión de este algoritmo para el cálculo del caso constante de la exponencial de una matriz, es decir,  $e^{At}$ . Todos los métodos están acompañados de experiencias numéricas que permiten visualizar las principales características de ellos, así como sus limitaciones.

Para empezar el tratamiento, consideramos el problema de valor inicial para la matriz  $Y(t)$  que satisface la ecuación diferencial matricial lineal

$$Y'(t) = A(t)Y(t), \quad Y(0) = Y_0. \quad (2.1)$$

Aquí  $A(t)$  es una función matricial, con  $t \in \mathbb{R}$ , tal que la existencia y unicidad de la solución está garantizada. En particular tomaremos el caso de  $A : \mathbb{R} \rightarrow M_n(\mathbb{R})$ , donde  $M_n(\mathbb{R})$  representa al conjunto de matrices  $n \times n$  sobre el cuerpo de los reales. En este contexto la existencia y unicidad de soluciones de (2.1) queda establecida por el siguiente teorema [Bel65].

**Teorema 3** *Si  $A(t)$  es una matriz  $n \times n$  continua para  $t \geq 0$ , hay una solución única para la ecuación diferencial vectorial*

$$\frac{dy}{dt} = A(t)y(t), \quad y(0) = y_0, \quad (2.2)$$

con  $y(t) \in \mathbb{R}^n$ . Esta solución existe para  $t \geq 0$  y puede escribirse como

$$y(t) = Y(t)y_0, \quad (2.3)$$

siendo  $Y(t)$  la única matriz que satisface la ecuación diferencial matricial

$$\frac{dY}{dt} = A(t)Y(t), \quad Y(0) = I. \quad (2.4)$$

El teorema 3 nos señala una relación estrecha entre la solución de una ecuación diferencial vectorial y su equivalente matricial.

La ecuación (2.4) es, al menos en principio, fácilmente resuelta por iteración, desarrollando

$$Y(t) = I + \sum_{k=1}^{\infty} P_k(t), \quad (2.5)$$

con  $P_k(0) = 0$ . Luego sustituyendo en la ec. (2.4) se obtiene el algoritmo recursivo

$$\frac{dP_k}{dt} = A(t)P_{k-1}, \quad P_0 \equiv I. \quad (2.6)$$

El esquema anterior recibe diferentes nombres dependiendo del contexto: serie de Neumann, serie de Dyson, etc. La serie en la ec. (2.5) converge para todos los valores de  $t$ . Sin embargo, un serio inconveniente del método es que su convergencia suele ser demasiado lenta y además al truncar la serie generalmente se pierden algunas de las propiedades de la solución exacta. Por ejemplo, supongamos que  $A(t)$  es antihermítica. Entonces,



claramente, la solución de (2.4) es unitaria. No obstante, al truncar la serie (2.5), la correspondiente aproximación perderá este carácter en general. Interesaría, pues, desarrollar procedimientos para obtener aproximaciones que sí respetaran ese carácter, es decir, que preservaran esa propiedad cualitativa de la solución exacta.

## 2.2. Desarrollo de Magnus

Para motivar el desarrollo propuesto por Wilhelm Magnus (1907-1990), consideremos primero el problema (2.4) cuando  $n = 1$ , es decir, en el caso en que tanto  $Y$  como  $A$  son funciones escalares de una variable real. En ese caso la solución se reduce a

$$Y(t) = \exp\left(\int_0^t A(s)ds\right). \quad (2.7)$$

Este es también el caso si  $n > 1$  y para cualquier par de valores de  $t, t_1$  y  $t_2$ , se tiene que  $A(t_1)A(t_2) = A(t_2)A(t_1)$ . Nótese que esto ocurre, en particular, cuando  $A$  es constante. Entonces la solución también se puede representar en la forma (2.7). No obstante, es importante resaltar que (2.7) no es la solución en el caso general de una matriz  $A(t)$  de dimensión  $n > 1$ .

La propuesta de Magnus consiste esencialmente en construir la solución de (2.4) como la exponencial de una matriz  $\Omega(t)$ , y determinar la estructura de esta matriz  $\Omega(t)$ . Específicamente, se construye la solución de (2.4) en la forma

$$Y(t) = \exp \Omega(t), \quad \Omega(0) = 0$$

y se expresa el exponente  $\Omega(t)$  como una serie

$$\Omega(t) = \sum_{k=1}^{\infty} \Omega_k(t)$$

llamada serie de Magnus. Para los primeros términos se tiene las siguientes expresiones:

$$\begin{aligned} \Omega_1(t) &= \int_0^t A(t_1) dt_1, \\ \Omega_2(t) &= \frac{1}{2} \int_0^t dt_1 \int_0^{t_1} dt_2 [A(t_1), A(t_2)] \\ \Omega_3(t) &= \frac{1}{6} \int_0^t dt_1 \int_0^{t_1} dt_2 \int_0^{t_2} dt_3 ([A(t_1), [A(t_2), A(t_3)]] + [A(t_3), [A(t_2), A(t_1)]]) \end{aligned}$$

donde  $A_i \equiv A(t_i)$  y  $[X, Y] \equiv XY - YX$  es el conmutador de  $X$  e  $Y$ .

En general, los  $\Omega_i(t)$  son combinaciones lineales de integrales y conmutadores anidados que involucran la matriz  $A$  en diferentes instantes [IN99].

Para deducir el resultado de Magnus, consideremos primero el problema de calcular la derivada temporal de la exponencial de una matriz dependiente del tiempo. Dada una función escalar  $\omega(t) \in \mathbb{R}$ , la derivada de la exponencial es  $\frac{d e^{\omega(t)}}{dt} = \omega'(t) e^{\omega(t)}$ . Se podría pensar en una fórmula similar para un matriz  $\Omega(t)$ . Sin embargo, este no es el caso, ya que en general  $[\Omega, \Omega'] \neq 0$ . En lugar de ello se tiene el siguiente resultado.

**Lema 4** *La derivada de  $\exp(\Omega(t))$  se puede escribir alternativamente como*

$$(a) \quad \frac{d}{dt} \exp(\Omega(t)) = \text{dexp}_{\Omega(t)}(\Omega'(t)) \exp(\Omega(t)), \quad (2.8)$$

$$(b) \quad \frac{d}{dt} \exp(\Omega(t)) = \exp(\Omega(t)) \text{dexp}_{-\Omega(t)}(\Omega'(t)), \quad (2.9)$$

$$(c) \quad \frac{d}{dt} \exp(\Omega(t)) = \int_0^1 e^{x\Omega(t)} \Omega'(t) e^{(1-x)\Omega(t)} dx, \quad (2.10)$$

donde  $\text{dexp}_{\Omega}(C)$  se define mediante la serie de potencias

$$\text{dexp}_{\Omega}(C) = \sum_{k=0}^{\infty} \frac{1}{(k+1)!} \text{ad}_{\Omega}^k(C) \equiv \frac{\exp(\text{ad}_{\Omega}) - I}{\text{ad}_{\Omega}}(C). \quad (2.11)$$

*Demostración:* Sea  $\Omega(t)$  una función matricial diferenciable y el conjunto de funciones

$$Y(\sigma, t) \equiv \frac{\partial}{\partial t} (\exp(\sigma\Omega(t)) \exp(-\sigma\Omega(t)))$$

para  $\sigma, t \in \mathbb{R}$ . Diferenciando con respecto a  $\sigma$ ,

$$\begin{aligned} \frac{\partial Y}{\partial \sigma} &= \frac{\partial}{\partial t} (\exp(\sigma\Omega)\Omega) \exp(-\sigma\Omega) + \frac{\partial}{\partial t} (\exp(\sigma\Omega)) (-\Omega) \exp(-\sigma\Omega) \\ &= \left( \exp(\sigma\Omega)\Omega' + \frac{\partial}{\partial t} (\exp(\sigma\Omega)) \Omega \right) \exp(-\sigma\Omega) \\ &\quad - \frac{\partial}{\partial t} (\exp(\sigma\Omega)) \Omega \exp(-\sigma\Omega) = \exp(\sigma\Omega)\Omega' \exp(-\sigma\Omega) \\ &= \exp(\text{ad}_{\sigma\Omega})(\Omega') = \sum_{k=0}^{\infty} \frac{\sigma^k}{k!} \text{ad}_{\Omega}^k(\Omega'), \end{aligned}$$

donde la primera igualdad en la última línea sigue fácilmente de (1.7) y (1.8). Por otra parte

$$\frac{d}{dt}(\exp \Omega) \exp(-\Omega) = Y(1, t) = \int_0^1 \frac{\partial}{\partial \sigma} Y(\sigma, t) d\sigma \quad (2.12)$$

ya que  $Y(0, t) = 0$ , y

$$\int_0^1 \frac{\partial}{\partial \sigma} Y(\sigma, t) d\sigma = \int_0^1 \sum_{k=0}^{\infty} \frac{\sigma^k}{k!} \text{ad}_{\Omega}^k(\Omega') d\sigma = \sum_{k=0}^{\infty} \frac{1}{(k+1)!} \text{ad}_{\Omega}^k(\Omega'),$$

por la fórmula (2.8). La convergencia de la serie de potencias (2.11) es una consecuencia del acotamiento del operador  $\text{ad}$ :  $\|\text{ad}_{\Omega}\| \leq 2\|\Omega\|$ .

Multiplicando ambos lados de (2.11) por  $\exp(-\Omega)$ , tenemos

$$e^{-\Omega} \frac{de^{\Omega}}{dt} = e^{-\Omega} d \exp_{\Omega}(\Omega') e^{\Omega} = e^{\text{ad}_{-\Omega}} d \exp_{\Omega}(\Omega') = \frac{e^{\text{ad}_{-\Omega}} - I}{\text{ad}_{-\Omega}} \Omega' = d \exp_{-\Omega}(\Omega')$$

de la que (2.9) sigue fácilmente. Por último, la ecuación (2.10) se obtiene tomando

$$\int_0^1 \frac{\partial}{\partial \sigma} Y(\sigma, t) d\sigma = \int_0^1 \exp(\sigma \Omega) \Omega' \exp(-\sigma \Omega) d\sigma$$

en (2.12).  $\square$

Como consecuencia, el mapeo exponencial tiene una inversa local en las proximidades de un punto  $\Omega$  tal que  $d \exp_{\Omega} = (\exp(\text{ad}_{\Omega}) - I) / \text{ad}_{\Omega}$  es invertible. El siguiente lema establece cuándo se puede aplicar.

**Lema 5** (Baker 1905). *Si los valores propios del operador lineal  $\text{ad}_{\Omega}$  son diferentes de  $2m\pi i$  con  $m \in \{\pm 1, \pm 2, \dots\}$ , entonces  $d \exp_{\Omega}$  es invertible. Además,*

$$d \exp_{\Omega}^{-1}(C) = \frac{\text{ad}_{\Omega}}{e^{\text{ad}_{\Omega}} - I} C = \sum_{k=0}^{\infty} \frac{B_k}{k!} \text{ad}_{\Omega}^k(C) \quad (2.13)$$

y la convergencia de la expansión  $d \exp_{\Omega}^{-1}$  está asegurada si  $\|\Omega\| < \pi$ .

*Demostración:* Los valores propios de  $d \exp_{\Omega}$  son de la forma

$$\mu = \sum_{k \geq 0} \frac{\nu^k}{(k+1)!} = \frac{e^{\nu} - 1}{\nu},$$

donde  $\nu$  es un valor propio de  $\text{ad}_{\Omega}$ . Por supuesto, los valores de  $\mu$  son diferentes de cero, por lo que  $d \exp_{\Omega}$  es invertible. Por definición de los números de

Bernoulli<sup>1</sup>, la composición de (2.13) con (2.11) da la identidad. El hecho de que converge para  $\|\Omega\| < \pi$  se deduce de que  $\|\mathbf{ad}_\Omega\| \leq 2\|\Omega\|$  y de que el radio de convergencia de la expansión de la serie  $x/(e^x - 1)$  es  $2\pi$ .  $\square$

Queda por determinar los valores propios del operador  $\mathbf{ad}_\Omega$ . De hecho, no es difícil demostrar que si  $\Omega$  tiene  $n$  valores propios  $\{\lambda_j, j = 1, 2, \dots, n\}$ , entonces  $\mathbf{ad}_\Omega$  tiene  $n^2$  valores propios  $\{\lambda_j - \lambda_k, j, k = 1, 2, \dots, n\}$ .

Como consecuencia de la discusión anterior, se puede establecer el siguiente teorema.

**Teorema 6** *La solución de la ecuación diferencial  $Y' = A(t)Y$  con condición inicial  $Y(0) = Y_0$  puede ser escrita como  $Y(t) = \exp(\Omega(t))Y_0$ , con  $\Omega(t)$  definido por*

$$\Omega' = d \exp_\Omega^{-1}(A(t)), \quad \Omega(0) = O, \quad (2.14)$$

donde

$$d \exp_\Omega^{-1}(A) = \sum_{k=0}^{\infty} \frac{B_k}{k!} \mathbf{ad}_\Omega^k(A).$$

*Demostración:* La comparación de la derivada de  $Y(t) = \exp(\Omega(t))Y_0$ ,

$$\frac{dY}{dt} = \frac{d}{dt} (\exp(\Omega(t))) Y_0 = d \exp_\Omega(\Omega') \exp(\Omega(t)) Y_0$$

con  $Y' = A(t)Y$ , permite obtener  $A(t) = d \exp_\Omega(\Omega')$ . Aplicando el operador inversa  $d \exp_\Omega^{-1}$  de esta relación se obtiene la ecuación diferencial (2.14) for  $\Omega(t)$ .  $\square$

Teniendo en cuenta los valores numéricos de los primeros números de Bernoulli, la ecuación diferencial (2.14), por tanto, se convierte en

$$\Omega' = A(t) - \frac{1}{2}[\Omega, A(t)] + \frac{1}{12}[\Omega, [\Omega, A(t)]] + \dots,$$

que es no lineal en  $\Omega$ . Definiendo

$$\Omega^{[0]} = O, \quad \Omega^{[1]} = \int_0^t A(t_1) dt_1,$$

y haciendo uso de la iteración de punto fijo de Picard, se obtiene

$$\Omega^{[n]} = \int_0^t \left( A(t_1) dt_1 - \frac{1}{2}[\Omega^{[n-1]}, A] + \frac{1}{12}[\Omega^{[n-1]}, [\Omega^{[n-1]}, A]] + \dots \right) dt_1$$

y  $\lim_{n \rightarrow \infty} \Omega^{[n]}(t) = \Omega(t)$  en un entorno del origen.

---

<sup>1</sup>Los números de Bernoulli (denotados por  $B_n$ ) constituyen una sucesión de números racionales con profundas conexiones en teoría de números. Ellos se definen en (1.22).

En general, la integración de (2.14) en forma iterativa lleva a una serie infinita para  $\Omega$ ,

$$\Omega(t) = \sum_{k=1}^{\infty} \Omega_k(t). \quad (2.15)$$

Fórmulas explícitas para todos los órdenes  $\Omega_k$  se pueden obtener usando teoría de grafos [IN99]. Disponiendo de las expresiones explícitas (o aproximadas) de los términos de la expansión de Magnus (EM), se pueden construir diferentes aproximaciones a la solución de (2.4) a través de la serie truncada de  $\Omega$ .

La EM tiene la muy atractiva propiedad de garantizar la conservación de importantes propiedades cualitativas de la solución exacta para cualquier orden de truncamiento.

Desde la década de 1960, la EM ha sido aplicada con éxito como herramienta analítica en numerosas áreas de la física, química, la física nuclear, atómica y molecular, en la teoría de la resonancia magnética nuclear y la electrodinámica cuántica (veáse [Bak02] para una lista de referencias y [BCOR09] para un tratamiento exhaustivo del desarrollo de Magnus y sus aplicaciones). A finales de los años 90 del siglo pasado, Iserles y Nørsett [IN99] utilizaron herramientas de teoría de grafos para analizar los términos de la expansión, produciendo un procedimiento recursivo para generar y construir algoritmos prácticos para la integración numérica de la ecuación (2.4). Los esquemas resultantes son ejemplos de integradores geométricos, y han demostrado ser muy competitivos con otros esquemas numéricos más convencionales, incluso con respecto a la exactitud y esfuerzo computacional [BCR02].

### 2.2.1. Generador de la expansión de Magnus

Los términos de la expansión de Magnus se pueden generar recursivamente. Si sustituimos  $\Omega(t) = \sum_{n=1}^{\infty} \Omega_n$  en la ecuación (2.8) e igualamos los términos del mismo orden, se obtiene en general

$$\begin{aligned} \Omega'_1 &= A \\ \Omega'_n &= \sum_{j=1}^{n-1} \frac{B_j}{j!} S_n^{(j)}, \quad n \geq 2, \end{aligned} \quad (2.16)$$

donde

$$\begin{aligned} S_n^{(k)} &= \sum_{m=1}^{n-j} \left[ \Omega_m, S_{n-m}^{(j-1)} \right], \quad 2 \leq j \leq n-1 \\ S_n^{(1)} &= [\Omega_{n-1}, A], \quad S_n^{(n-1)} = \text{ad}_{\Omega_1}^{n-1}(A). \end{aligned} \quad (2.17)$$

Después de integrar se obtiene el siguiente resultado

$$\begin{aligned}\Omega_1 &= \int_0^t A(\tau) d\tau \\ \Omega_n &= \sum_{j=1}^{n-1} \frac{B_j}{j!} \int_0^t S_n^{(j)}(\tau) d\tau, \quad n \geq 2.\end{aligned}\tag{2.18}$$

Analizando esta recurrencia, se llega a una expresión de la forma

$$\Omega_n(t) = \sum_{j=1}^{n-1} \frac{B_j}{j!} \sum_{\substack{k_1+\dots+k_j=n-1 \\ k_1 \geq 1, \dots, k_j \geq 1}} \int_0^t \text{ad}_{\Omega_{k_1}}(s) \text{ad}_{\Omega_{k_2}}(s) \dots \text{ad}_{\Omega_{k_j}}(s), \quad n \geq 2.\tag{2.19}$$

Obsérvese que cada término  $\Omega_n(t)$ ,  $n \geq 2$  en la serie de Magnus es una integral múltiple de combinaciones de  $n - 1$  conmutadores anidados que contienen  $n$  operadores  $A(t)$ . Si, en particular,  $A(t)$  pertenece a un álgebra de Lie  $\mathfrak{g}$ , entonces  $\Omega(t)$ , y por ende cualquier serie truncada de la serie de Magnus, también pertenece a  $\mathfrak{g}$  y por consiguiente  $e^\Omega \in \mathcal{G}$ , donde  $\mathcal{G}$  denota, como ya se ha dicho, al grupo de Lie asociado a  $\mathfrak{g}$ .

Otra propiedad importante de la expansión Magnus es que la serie converge. Específicamente, se obtiene el siguiente resultado:

**Teorema 7** *Consideremos la ecuación diferencial  $Y' = A(t)Y$ , donde las matrices pueden ser reales o complejas o más en general definidas en un espacio de Hilbert  $\mathcal{H}$  con  $Y(0) = I$ , y sea  $A(t)$  un operador lineal acotado en  $\mathcal{H}$ . Entonces la serie de Magnus  $\Omega(t) = \sum_{k=1}^{\infty} \Omega_k(t)$ , con  $\Omega_k$  dada por (2.19) converge en el intervalo  $t \in [0, T)$ , estando  $T$  dada implícitamente por la relación*

$$\int_0^T \|A(s)\|_2 ds < \pi$$

*y la suma  $\Omega(t)$  satisface  $\exp \Omega(t) = Y(t)$ . Este resultado también es válido cuando  $\mathcal{H}$  es de dimensión infinita si  $Y$  es un operador normal (en particular, si  $Y$  es unitario).*

### 2.2.2. La Expansión de Magnus y la teoría de perturbaciones dependiente del tiempo

Como mencionamos al comienzo del capítulo, la solución de (2.1) se puede representar por (2.5). De hecho se puede establecer una conexión entre la serie

de Magnus  $\Omega = \sum_k \Omega_k$  y la serie (2.5) [Dys49]. Resolviendo iterativamente (2.6) se llega a

$$P_n(t) = \int_0^t dt_1 \dots \int_0^{t_{n-1}} dt_n A_1 A_2 \dots A_n,$$

donde  $A_i \equiv A(t_i)$ . Entonces, igualando  $Y(t) = \exp \Omega(t)$  con (2.5) tenemos

$$\sum_{j=1}^{\infty} \Omega_j(t) = \log \left( 1 + \sum_{j=1}^{\infty} P_j(t) \right).$$

Insertando el desarrollo de Taylor de la función logaritmo e igualando términos se llega a

$$\Omega_n = P_n - \sum_{j=2}^n \frac{(-1)^j}{j} R_n^{(j)}, \quad n \geq 2, \quad (2.20)$$

donde

$$R_n^{(k)} = \sum P_{i_1} P_{i_2} \dots P_{i_k}, \quad (i_1 + i_2 + \dots + i_k = n).$$

Esta última ecuación se puede expresar recursivamente como

$$\begin{aligned} R_n^{(j)} &= \sum_{m=1}^{n-j+1} R_m^{(1)} R_{n-m}^{(j-1)}, \\ R_n^{(1)} &= P_n, \quad R_n^{(n)} = P_1^n, \end{aligned} \quad (2.21)$$

siendo los primeros términos

$$\begin{aligned} \Omega_1 &= P_1 \\ \Omega_2 &= P_2 - \frac{1}{2} P_1^2 \\ \Omega_3 &= P_3 - \frac{1}{2} (P_1 P_2 + P_2 P_1) + \frac{1}{3} P_1^3. \end{aligned} \quad (2.22)$$

### 2.2.3. Métodos de Integración Numérica basados en la expansión de Magnus

La necesidad de contar con métodos de integración numérica que mantengan las principales características cualitativas del problema resulta clave para muchas aplicaciones. Por ejemplo, es común representar fenómenos por ecuaciones diferenciales ordinarias (EDO) y más concretamente por problemas de valor inicial (PVI) de la forma

$$\mathbf{x}' = \mathbf{f}(\mathbf{t}, \mathbf{x}(\mathbf{t})), \quad \mathbf{x}(\mathbf{t}_0) = \mathbf{x}_0 \in \mathbb{C}^d. \quad (2.23)$$

Métodos de integración numérica estándar, como los métodos de tipo Runge–Kutta y multipaso, proceden iterativamente sobre un intervalo  $[t_0, t_f]$  dividiéndolo en  $n$  subintervalos,  $[t_{i-1}, t_i]$ ,  $i = 1, \dots, n$ , con  $t_n = t_f$  y calculando la secuencia de valores  $\mathbf{x}(t_i)$  mediante

$$\mathbf{x}_{i+1} = \phi(h_i, \mathbf{x}_i, h_{i-1}, \mathbf{x}_{i-1}, \dots, h_0, \mathbf{x}_0) \quad (2.24)$$

partiendo desde  $\mathbf{x}_0$ . Ejemplos conocidos son el método explícito de Euler (orden 1) o el método de Runge-Kutta de orden cuatro.

Cuando este tipo de métodos se aplica a la ecuación lineal  $Y' = A(t)Y$  definida en un grupo de Lie (por ejemplo, si  $Y$  es ortogonal o unitaria), resulta que la aproximación construida no está en dicho grupo. Así, si  $Y$  es ortogonal, las aproximaciones  $Y_i$  obtenidas a partir del esquema (2.24) si se aplica un método Runge–Kutta no serán ortogonales. Ocurre un caso análogo a lo que sucede con la aproximación analítica obtenida al aplicar el formalismo clásico de la teoría de perturbaciones dependiente del tiempo: las propiedades cualitativas de la solución exacta no se conservan. Esto motiva la cuestión de si es posible construir nuevos métodos de integración numérica a partir del desarrollo de Magnus.

Profundizando un poco más en estas cuestiones, podemos decir que la pérdida de la calidad de la solución aproximada entregada por un método numérico, como el descrito genéricamente por (2.24), puede ser inaceptable en términos cualitativos. Por ejemplo: la solución numérica proporcionada por un esquema iterativo de la familia de métodos de Runge-Kutta, y en general por un integrador de la forma descrita se construye como una suma de vectores en  $\mathbb{R}^n$ . Supongamos, por ejemplo, que  $\mathbf{x}$  es un vector conocido que evoluciona en una esfera. Uno no espera que  $\mathbf{x}_i$  construido como una suma de dos vectores conserve esta característica de la solución exacta.

Así pues, caben las siguientes preguntas: ¿Es posible diseñar métodos de integración numérica para la ecuación (2.23) de tal manera que las aproximaciones numéricas correspondientes aún conserven las principales características cualitativas de la solución exacta? Por otra parte, dado que la expansión de Magnus constituye un procedimiento extremadamente útil para la obtención analítica, ¿soluciones aproximadas a la ecuación de evolución lineal,  $Y' = A(t)Y$ , podrían ser utilizadas para construir, en términos factibles, esquemas numéricos eficientes de integración? Resulta que las dos preguntas se pueden contestar de manera afirmativa. Es más, la EM permite diseñar y construir métodos que en la mayoría de las veces reproducen las características cualitativas de la solución de la ecuación diferencial que se está discretizando, en particular sus propiedades geométricas.



La próxima sección presenta dos esquemas numéricos de Magnus que se usan a lo largo de este trabajo.

#### 2.2.4. Integradores de Magnus

Dado que la serie de Magnus converge sólo localmente, para poder entonces integrar sobre un intervalo el procedimiento habitual consiste en dividir el intervalo temporal  $[t_0, t_f]$  en  $N$  pasos tal que la serie de Magnus converge en cada subintervalo  $[t_{n-1}, t_n]$ ,  $n = 1, \dots, N$ , con  $t_N = t_f$ . De esta manera la solución de (2.4) en el tiempo final  $t_f$  está representado por

$$Y(t_N) = \prod_{n=1}^N \exp(\Omega(t_n, t_{n-1})) Y_0, \quad (2.25)$$

y la serie  $\Omega(t_n, t_{n-1})$  puede ser truncada adecuadamente. Los primeros pasos de este enfoque fueron tomados en [CL69], donde esquemas numéricos de segundo y cuarto orden fueron utilizado para los cálculos de colisiones inelásticas y para la dispersión por un potencial, respectivamente.

Estos autores eran conscientes de que el método de integración resultante se convertiría en práctico sólo cuando la ventaja de ser capaz de utilizar tamaños de paso más grandes es mayor que la desventaja en tener que evaluar las integrales que intervienen en la serie de Magnus y luego hacer la exponenciación. Más tarde, siguiendo un enfoque similar, Devries [Dev85] diseñó un procedimiento numérico para determinar una aproximación de cuarto orden para la ecuación de Schrödinger. Sin embargo, fue en [IN99] donde Iserles y Nørsett llevaron a cabo el primer estudio sistemático de la EM con el objetivo de construir integradores numéricos explícitos para problemas lineales. Para ello analizaron detalladamente la dependencia del tiempo de cada término  $\Omega_k$ , y en particular el orden de la aproximación de cada término.

En términos generales, el proceso de hacer de la EM un algoritmo práctico de integración numérica implica tres pasos:

- En primer lugar, la serie  $\Omega$  ha de ser truncada en un orden adecuado.
- En segundo lugar, las integrales múltiples de la serie truncada  $\Omega^{[p]} = \sum_{i=1}^p \Omega$  son reemplazadas por aproximaciones convenientemente elegidas.
- En tercer lugar, la exponencial de la matriz  $\Omega^{[p]}$  tiene que ser calculada.

Consideremos ahora brevemente las dos primeras cuestiones, puesto que el problema general de la evaluación de la exponencial de una matriz es tratado tanto más adelante como en el Apéndice A.

Un hecho destacado de la EM es que es simétrica con respecto al tiempo, es decir, la aplicación  $\varphi^t : Y(t_0) \rightarrow Y(t)$  correspondiente a la ecuación diferencial (2.1) es simétrica:  $\varphi^{-t} \circ \varphi^t = \text{Id}$ , donde la variable de integración  $t$  toma valores entre  $t_0$  (tiempo inicial) y  $t_f$  (tiempo final). Puesto que  $Y(t) = U(t, t_0)Y_0$ , siendo  $U$  una matriz de  $n \times n$ , entonces el operador  $\varphi^t$  puede ser expresado en términos de la matriz fundamental (operador de evolución)  $U(t, t_0)$  como

$$U(t_0, t_f) = U^{-1}(t_f, t_0)$$

o en términos de la EM

$$\Omega(t_f, t_0) = -\Omega(t_0, t_f).$$

Para aprovechar esta característica, escribamos la solución de (2.4) en el tiempo final  $t_f = t_0 + s$  como

$$Y(t_{1/2} + \frac{s}{2}) = \exp\left(\Omega(t_{1/2} + \frac{s}{2}, t_{1/2} - \frac{s}{2})\right) Y(t_{1/2} - \frac{s}{2}), \quad (2.26)$$

donde  $t_{1/2} = (t_0 + t_f)/2$ . Entonces

$$Y(t_{1/2} - \frac{s}{2}) = \exp\left(-\Omega(t_{1/2} + \frac{s}{2}, t_{1/2} - \frac{s}{2})\right) Y(t_{1/2} + \frac{s}{2}). \quad (2.27)$$

Por otro lado, la solución a  $t_0$  puede ser escrita como

$$Y(t_{1/2} - \frac{s}{2}) = \exp\left(\Omega(t_{1/2} - \frac{s}{2}, t_{1/2} + \frac{s}{2})\right) Y(t_{1/2} + \frac{s}{2}), \quad (2.28)$$

de manera que, mediante la comparación de (2.27) y (2.28),

$$\Omega(t_{1/2} - \frac{s}{2}, t_{1/2} + \frac{s}{2}) = -\Omega(t_{1/2} + \frac{s}{2}, t_{1/2} - \frac{s}{2}). \quad (2.29)$$

Como consecuencia de la ecuación (2.29), si  $A(t)$  es una función analítica y se evalúa su serie de Taylor centrada alrededor del punto medio de un subintervalo  $[t_n, t_n + h]$  particular, entonces cada término en  $\Omega_k$  es una función impar de  $h$ , y por lo tanto  $\Omega_{2s+1} = \mathcal{O}(h^{2s+3})$  para  $s \geq 1$ . De manera equivalente,  $\Omega^{[2s-2]} = \Omega + \mathcal{O}(h^{2s+1})$  y  $\Omega^{[2s-1]} = \Omega + \mathcal{O}(h^{2s+1})$ . En otras palabras, para lograr un método de integración de orden  $2s$ , ( $s > 1$ ) solamente se requieren hasta términos  $\Omega_{2s-2}$  en la serie  $\Omega$  [IN99], [BCR02].

Una vez que la expansión de la serie se trunca hasta un orden apropiado, las integrales multidimensionales involucradas tienen que ser calculadas o al

menos convenientemente aproximadas. Aunque a primera vista esto parece ser una empresa muy difícil, resulta que su misma estructura permite aproximar todas las integrales múltiples que aparecen en  $\Omega$  sólo mediante la evaluación de  $A(t)$  en los nodos de una cuadratura [IN99].

Para ilustrar lo tratado, vamos a expandir la matriz  $A(t)$  en serie de potencias en torno al punto medio,  $t_{1/2} \equiv t_n + h/2$ , del subintervalo  $[t_n, t_{n+1}]$ ,

$$A(t) = \sum_{j=0}^{\infty} a_j (t - t_{1/2})^j, \quad \text{donde} \quad a_j = \frac{1}{j!} \left. \frac{d^j A(t)}{dt^j} \right|_{t=t_{1/2}}. \quad (2.30)$$

Insertando la serie (2.30) en la recurrencia de la definición de la EM (2.17), (2.18), se obtiene de forma explícita la expresión de  $\Omega_k$  hasta orden  $h^6$

$$\begin{aligned} \Omega_1 &= ha_0 + h^3 \frac{1}{12} a_2 + h^5 \frac{1}{80} a_4 + \mathcal{O}(h^7) \\ \Omega_2 &= h^3 \frac{-1}{12} [a_0, a_1] + h^5 \left( \frac{-1}{80} [a_0, a_3] + \frac{1}{240} [a_1, a_2] \right) + \mathcal{O}(h^7) \\ \Omega_3 &= h^5 \left( \frac{1}{360} [a_0, a_0, a_2] - \frac{1}{240} [a_1, a_0, a_1] \right) + \mathcal{O}(h^7) \\ \Omega_4 &= h^5 \frac{1}{720} [a_0, a_0, a_0, a_1] + \mathcal{O}(h^7), \end{aligned} \quad (2.31)$$

mientras  $\Omega_5 = \mathcal{O}(h^7)$ ,  $\Omega_6 = \mathcal{O}(h^7)$  y  $\Omega_7 = \mathcal{O}(h^9)$ . Aquí escribimos para mayor claridad  $[a_{i_1}, a_{i_2}, \dots, a_{i_{l-1}}, a_{i_l}] \equiv [a_{i_1}, [a_{i_2}, [\dots, [a_{i_{l-1}}, a_{i_l}] \dots]]]$ . Fijémonos en que, según lo previsto, sólo potencias impares de  $h$  aparecen en  $\Omega_k$  y, en particular se tiene que,  $\Omega_{2i+1} = \mathcal{O}(h^{2i+3})$  para  $i > 1$ .

Alternativamente, introduciendo

$$\alpha_i = h^i \frac{1}{(i-1)!} A^{(i-1)}(t_{1/2})$$

y agrupando en potencias de  $h$ , se tiene hasta orden  $h^6$ :

$$\begin{aligned} \Omega &= \alpha_1 + \frac{1}{12} \alpha_3 - \frac{1}{12} [\alpha_1, \alpha_2] + \frac{1}{240} [\alpha_2, \alpha_3] + \frac{1}{360} [\alpha_1, \alpha_1, \alpha_3] \\ &\quad - \frac{1}{240} [\alpha_2, \alpha_1, \alpha_2] + \frac{1}{720} [\alpha_1, \alpha_1, \alpha_1, \alpha_2] + \mathcal{O}(h^7). \end{aligned} \quad (2.32)$$

Resulta que es posible construir métodos de orden  $p \equiv 2s$  considerando sólo los términos que incluyen  $\alpha_1, \dots, \alpha_s$  en  $\Omega$ . A continuación, estos términos se pueden aproximar por combinaciones lineales adecuadas de la matriz  $A(t)$  evaluada en diferentes puntos. Es así como se llega a los siguientes esquemas numéricos.

**Orden 2.** Sólo se tiene que aproximar  $\alpha_1$  en (2.32). Esto se puede hacer con la regla del punto medio, que conduce al método

$$Y_{n+1} = \exp(hA(t_{1/2}))Y_n. \quad (2.33)$$

Alternativamente, si se utiliza la regla del trapecio para la integral, se tiene

$$Y_{n+1} = \exp(h(A(t_n) + A(t_{n+1})))Y_n.$$

Aunque aparentemente se utilizan dos evaluaciones de  $A$  por paso, esto es sólo así en el primero, porque en los restantes  $A(t_{n+1})$  se reutiliza como la primera evaluación del paso siguiente.

**Orden 4.** Utilizando la regla de cuadratura de Gauss-Legendre,

$$A_1 = A\left(t_n + \left(\frac{1}{2} - \frac{\sqrt{3}}{6}\right)h\right), \quad A_2 = A\left(t_n + \left(\frac{1}{2} + \frac{\sqrt{3}}{6}\right)h\right), \quad (2.34)$$

se tiene

$$\alpha_1 = \frac{h}{2}(A_1 + A_2), \quad \alpha_2 = \frac{h\sqrt{3}}{12}(A_2 - A_1). \quad (2.35)$$

Entonces, sustituyendo en (2.32) y descartando términos de orden superior a 4, obtenemos

$$\begin{aligned} \Omega^{[4]}(h) &= \frac{h}{2}(A_1 + A_2) - h^2 \frac{\sqrt{3}}{12}[A_1, A_2] \\ Y_{n+1} &= \exp(\Omega^{[4]}(h))Y_n. \end{aligned} \quad (2.36)$$

Alternativamente, si evaluamos  $A$  en puntos equidistantes, con  $k = 3$  y  $c_1 = 0, c_2 = 1/2, c_3 = 1$ ;  $b_1 = b_3 = 1/6, b_2 = 2/3$ , es decir, usando la regla de aproximación de Simpson para la integral  $\int_{t_n}^{t_n+h} A(s)ds$ ,

$$A_1 = A(t_n), \quad A_2 = A\left(t_n + \frac{h}{2}\right), \quad A_3 = A(t_n + h)$$

tenemos  $\alpha_1 = \frac{h}{6}(A_1 + 4A_2 + A_3)$ ,  $\alpha_2 = h(A_3 - A_1)$ , luego

$$\Omega^{[4]}(h) = \frac{h}{6}(A_1 + 4A_2 + A_3) - \frac{h^2}{72}[A_1 + 4A_2 + A_3, A_3 - A_1]. \quad (2.37)$$

Aunque aparentemente más evaluaciones de  $A$  son necesarias en (2.37), este no es el caso, ya que  $A_3$  puede ser reutilizado en el siguiente paso de integración.

**Orden 6.** Se puede demostrar que

$$\begin{aligned} C_1 &= [\alpha_1, \alpha_2], \\ C_2 &= -\frac{1}{60}[\alpha_1, 2\alpha_3 + C_1] \\ \Omega^{[6]} &\equiv \alpha_1 + \frac{1}{12}\alpha_3 + \frac{1}{240}[-20\alpha_1 - \alpha_3 + C_1, \alpha_2 + C_2], \end{aligned} \quad (2.38)$$

verifica que  $\Omega^{[6]} = \Omega + \mathcal{O}(h^7)$ .

En términos de los puntos de colocación de Gauss-Legendre, se tendrá

$$A_1 = A\left(t_n + \left(\frac{1}{2} - \frac{\sqrt{15}}{10}\right)h\right), \quad A_2 = A\left(t_n + \frac{1}{2}h\right), \quad A_3 = A\left(t_n + \left(\frac{1}{2} + \frac{\sqrt{15}}{10}\right)h\right)$$

y similarmente obtendremos

$$\alpha_1 = hA_2, \quad \alpha_2 = \frac{\sqrt{15}h}{3}(A_3 - A_1), \quad \alpha_3 = \frac{10h}{3}(A_3 - 2A_2 + A_1), \quad (2.39)$$

que luego se inserta en (2.38) para llegar a la aproximación  $Y_{n+1} = \exp(\Omega^{[6]})Y_n$ .

Si la matriz  $A(t)$  sólo se conoce en puntos equidistantes, podemos utilizar la cuadratura Newton-Cotes, con  $s = 3$  y  $k = 5$ ,  $b_1 = b_5 = 7/90$ ,  $b_2 = b_4 = 32/90$ ,  $b_3 = 12/90$  y  $c_j = (j - 1)/4$ ,  $j = 1, \dots, 5$ . Entonces conseguimos

$$\begin{aligned} \alpha_1 &= \frac{1}{60}(-7(A_1 + A_5) + 28(A_2 + A_4) + 18A_3) \\ \alpha_2 &= \frac{1}{15}(7(A_5 - A_1) + 16(A_4 - A_2)) \\ \alpha_3 &= \frac{1}{3}(7(A_1 + A_5) - 4(A_2 + A_4) - 6A_3). \end{aligned} \quad (2.40)$$

Ambos esquemas implican el número mínimo de conmutadores (tres) y requieren tres o cuatro evaluaciones de la matriz  $A(t)$  por paso de integración (hay que tener en cuenta que  $A_5$  puede ser reutilizado en el siguiente paso en la implementación de Newton-Cotes porque  $c_1 = 0$  y  $c_5 = 1$ ).

Los correspondientes algoritmos se pueden expresar como sigue:

### Magnus de segundo orden

- **Input:** Matriz  $A(t)$ ,  $Y_0 = Y(0)$ ,  $t \in [t_0, t_f = t_N]$

- **Output:** La solución aproximada  $Y_{n+1} \approx Y(t_{n+1})$
  - Para  $n = 0, 1, \dots, N - 1$ 
    1. Calcular las cuadraturas con la regla del punto medio
 
$$A_{t_{1/2}} = A\left(\frac{t_0 + t_f}{2}\right)$$
    2. Aproximación
 
$$Y_{n+1} = \exp(hA(t_{1/2}))Y_n$$
- 

### Magnus de cuarto orden

---

- **Input:** Matriz  $A(t)$ ,  $Y_0 = Y(0)$ ,  $t \in [t_0, t_f = t_N]$
  - **Output:** La solución aproximada  $Y_{n+1} \approx Y(t_{n+1})$
  - Para  $n = 0, 1, \dots, N - 1$ 
    1. Calcular las cuadraturas con la regla de Gauss-Legendre
 
$$A_1 = A\left(t_n + \left(\frac{1}{2} - \frac{\sqrt{3}}{6}\right)h\right), \quad A_2 = A\left(t_n + \left(\frac{1}{2} + \frac{\sqrt{3}}{6}\right)h\right)$$
    2. Calcular el término
 
$$\Omega^{[4]}(h) = \frac{h}{2}(A_1 + A_2) - h^2 \frac{\sqrt{3}}{12}[A_1, A_2]$$
    3. Aproximación
 
$$Y_{n+1} = \exp(\Omega^{[4]}(h))Y_n$$
- 

### Magnus de sexto orden

---

- **Input:** Matriz  $A(t)$ ,  $Y_0 = Y(0)$ ,  $t \in [t_0, t_f = t_N]$
- **Output:** La solución aproximada  $Y_{n+1} \approx Y(t_{n+1})$
- Para  $n = 0, 1, \dots, N - 1$ 
  1. Calcular las cuadraturas con la regla de Gauss-Legendre
 
$$A_1 = A\left(t_n + \left(\frac{1}{2} - \frac{\sqrt{15}}{10}\right)h\right), \quad A_2 = A\left(t_n + \frac{1}{2}h\right)$$

$$A_3 = A\left(t_n + \left(\frac{1}{2} + \frac{\sqrt{15}}{10}\right)h\right)$$

2. Calcular
 
$$\alpha_1 = hA_2, \quad \alpha_2 = \frac{\sqrt{15}h}{3}(A_3 - A_1), \quad \alpha_3 = \frac{10h}{3}(A_3 - 2A_2 + A_1)$$

$$C_1 = [\alpha_1, \alpha_2], \quad C_2 = -\frac{1}{60}[\alpha_1, 2\alpha_3 + C_1]$$
  3. Calcular el término
 
$$\Omega^{[6]}(h) = \alpha_1 + \frac{1}{12}\alpha_3 + \frac{1}{240}[-20\alpha_1 - \alpha_3 + C_1, \alpha_2 + C_2]$$
  4. Aproximación
 
$$Y_{n+1} = \exp(\Omega^{[6]}(h))Y_n$$
- 

## 2.3. Procedimiento iterativo de Voslamber

El método de Voslamber, originalmente descrito en [Vos72], permite el cálculo aproximado del exponente en la representación  $Y(t) = e^{\Omega(t)}$  de la solución del sistema lineal  $Y' = A(t)Y$  de una forma alternativa al desarrollo de Magnus. De hecho, la aproximación conseguida al aplicar este procedimiento se puede interpretar como una resumación de términos en la serie de Magnus, poseyendo además características especiales no compartidas por la correspondiente aproximación obtenida por Magnus. En esta sección se expondrá un nuevo integrador numérico para ecuaciones diferenciales lineales dependientes del tiempo, de cuarto orden, basado en el esquema propuesto por Voslamber, así como su ilustración en algunos ejemplos numéricos.

### 2.3.1. Elementos previos

Consideremos el problema de valor inicial

$$Y' = \delta A(t)Y, \quad Y(0) = I \tag{2.41}$$

con  $\delta > 0$  un cierto parámetro. Como sabemos, para valores suficientemente pequeños de  $t$ , la solución de (2.41) puede ser escrita como

$$Y(t; \delta) = e^{\Omega(t; \delta)}, \tag{2.42}$$

donde

$$\Omega(t; \delta) = \sum_{n=1}^{\infty} \delta^n \Omega_n(t). \tag{2.43}$$

Aunque las aproximaciones obtenidas al truncar la serie (2.43) preservan importantes características de la solución exacta, según hemos visto anteriormente, hay una propiedad notable que no es preservada por dichas

aproximaciones y que no obstante puede ser relevante en ciertas aplicaciones:  $(1/\delta) \sum_{n=1}^m \delta^n \Omega_n(t)$  con  $m > 1$  no está acotada cuando  $\delta \rightarrow \infty$ , aun cuando  $\Omega(t, \delta)/\delta$  es uniformemente convergente con respecto a  $\delta$  bajo unas hipótesis bastante generales sobre la matriz  $A(t)$  [Vos72].

Este comportamiento es particularmente relevante en el caso de sistemas lineales de la forma

$$\frac{dU}{dt} = S(\lambda t)U, \quad \text{con} \quad \lambda \equiv 1/T \ll 1,$$

siendo  $S(\tau)$ ,  $U(\tau)$  matrices complejas  $n \times n$ . Obsérvese que la matriz de coeficientes  $S$  varía con el tiempo a través de  $t/T$ , y que  $T$  determina la escala de tiempo del sistema. De manera equivalente, mediante la introducción de la variable  $\tau = t/T \equiv \lambda t$ , tenemos

$$\frac{dU}{d\tau} = \frac{1}{\lambda} S(\tau)U, \quad U(\tau_0) = I \quad (2.44)$$

con el parámetro  $0 < \lambda \ll 1$ . Entonces, en la ecuación (2.44) hay dos escalas de tiempo diferentes,  $t$  y  $\tau$ , y es el parámetro  $\lambda$  el que controla la separación de escalas de tiempo: cuanto menor  $\lambda$ , más lenta será la variación de  $S(\lambda t)$  en la escala de tiempo rápida fijada a priori por  $t$ . La variable tiempo  $\tau = \lambda t$  en la que  $S$  varía es llamada la escala de tiempo *lenta*. Fijémonos que este caso corresponde a la ecuación (2.41) identificando  $\delta = 1/\lambda$ ,  $t = \tau T$ , etc.

Podemos citar al menos dos ejemplos relevantes con estas características:

1. El tratamiento del *oscilador armónico* dependiente del tiempo en la mecánica clásica. Se describe por la función hamiltoniana

$$H(q, p, t) = \frac{1}{2}(p^2 + \omega^2(\lambda t)q^2). \quad (2.45)$$

Como es bien sabido, en este caso la “variable acción”  $J(\tau) \equiv H(\tau)/\omega(\tau)$  es un *invariante adiabático*: permanece aproximadamente constante durante un intervalo de tiempo del orden de  $1/\lambda$ . De hecho, esta propiedad fue muy importante en la formulación de la vieja teoría cuántica (y fue objeto de estudio por parte de Einstein, Lorentz, etc.).

2. **Mecánica Cuántica.** La ecuación de Schrödinger dependiente del tiempo en el régimen *adiabático*

$$i\hbar \frac{d\psi}{d\tau} = \frac{1}{\lambda} H(\tau)\psi. \quad (2.46)$$

En este caso el *Teorema adiabático cuántico* (Born, Fock) establece que, en el límite  $\lambda \rightarrow 0$ , los valores absolutos de los coeficientes en la representación de los estados propios son invariantes adiabáticos.



Una propiedad importante de la representación (2.42) como solución de la ecuación (2.41), se obtiene cuando aplicamos el teorema de la triangulación unitaria a la solución exacta  $Y(t; \delta)$ :

$$T_\delta = U^\dagger Y U, \quad (2.47)$$

donde  $T_\delta$  es diagonal y  $U$  unitaria. Derivando (2.47) con respecto a  $t$ , tenemos

$$T'_\delta = \delta U^\dagger A U T_\delta + [T_\delta, U^\dagger U'],$$

siendo  $[, ]$  el conmutador. Como el segundo término del lado derecho de la ecuación no incluye a la diagonal principal, se tiene

$$T_\delta = \exp \left( \delta \int_0^t (U^\dagger A U)_\Delta ds \right),$$

donde el subíndice  $\Delta$  denota a la parte triangular superior (incluidos los términos de la diagonal principal) de la correspondiente matriz. Teniendo en cuenta (2.43), se tiene

$$\Omega(t; \delta) = \delta U \left( \int_0^t (U^\dagger A U)_\Delta ds \right) U^\dagger. \quad (2.48)$$

Considerando ahora la norma de Frobenius (que es unitariamente invariante) de ambos lados de esta ecuación, se tiene

$$\begin{aligned} \|\Omega\|_F &= |\delta| \left\| \int_0^t (U^\dagger A U)_\Delta ds \right\|_F \leq |\delta| \int_0^t \|(U^\dagger A U)_\Delta\|_F ds \\ &\leq |\delta| \int_0^t \|U^\dagger A U\|_F ds = |\delta| \int_0^t \|A\|_F ds. \end{aligned} \quad (2.49)$$

Si en lugar de la norma de Frobenius consideramos la norma espectral, de las desigualdades (1.20), (1.21) y (2.49), se concluye que

$$\|\Omega\|_2 \leq \sqrt{\text{rango}(A)} |\delta| \int_0^t \|A\|_2 ds.$$

Es importante subrayar aquí que para la solución exacta se tiene que  $\Omega(t; \delta)/\delta$  es uniformemente convergente con respecto del parámetro  $\delta$ . El único requerimiento de  $A$  es que sea de dimensión finita y sus elementos sean funciones integrables.

### 2.3.2. Aproximación de Voslamber

Voslamber propone un algoritmo de aproximaciones sucesivas de la expresión (2.42) de tal manera que cualquier truncamiento preserve la notable propiedad citada en el último párrafo de la sección anterior, es decir, que  $\Omega^{[m]}(t; \delta)/\delta$  converge con respecto a  $\delta$  cuando  $m \rightarrow \infty$ .

El punto de partida consiste en considerar el operador

$$\Gamma \equiv e^{\Omega/2} \Omega' e^{-\Omega/2}. \quad (2.50)$$

Insertando  $\Omega' = d \exp_{\Omega}^{-1}(A)$  en (2.50), se tiene

$$\begin{aligned} \Gamma &= e^{\text{ad}_{\Omega/2}} \Omega' = d \exp_{\Omega}^{-1}(\delta A) = e^{\text{ad}_{\Omega/2}} \frac{\text{ad}_{\Omega}}{e^{\text{ad}_{\Omega}} - 1}(\delta A) \\ &= \frac{\text{ad}_{\Omega}}{\sinh \Omega/2}(\delta A) = \sum_{n=0}^{\infty} \frac{B_n(1/2)}{n!} \text{ad}_{\Omega}^n(\delta A). \end{aligned}$$

Como  $B_k(1/2) = -(1 - 2^{1-k})B_k$ , llegamos finalmente a

$$\Gamma = \sum_{n=0}^{\infty} \frac{2^{1-n} - 1}{n!} B_n \text{ad}_{\Omega}^n(\delta A). \quad (2.51)$$

A fin de expresar  $\Gamma$  como una serie de potencias de  $\delta$ , se tiene que introducir la serie de Magnus en la ecuación (2.51). Obtenemos entonces

$$\Gamma(t; \delta) = \sum_{n=0}^{\infty} \delta^n \Gamma_n(t), \quad (2.52)$$

donde los términos  $\Gamma_n$  se pueden expresar como función de  $\Omega_k$  con  $k \leq n-2$ , a través de un proceso recursivo. Específicamente se tiene

$$\begin{aligned} \text{ad}_{\Omega}(\delta A) &= \sum_{j=1}^{n-1} \delta^j \text{ad}_{\Omega_j}(\delta A) + \mathcal{O}(\delta^n) \\ &= \sum_{j=1}^{n-1} \delta^{j+1} \text{ad}_{\Omega_j} A + \mathcal{O}(\delta^n), \end{aligned}$$

de manera que

$$\sum_{k=0}^{\infty} c_k \text{ad}_{\Omega}^k(\delta A) = \delta A + \delta \sum_{j=1}^{n-1} c_j \text{ad}_{\Omega}^j A + \mathcal{O}(\delta^{n+1}).$$

Aquí

$$c_j \equiv \frac{2^{1-j} - 1}{j!} B_j,$$

con  $c_{2j+1} = 0, c_2 = -1/24, c_4 = 7/5760, \text{ etc.}$  Ahora

$$\begin{aligned} \delta \sum_{j=1}^{n-1} c_j \text{ad}_{\Omega}^j A &= \delta \sum_{j=1}^{n-1} c_j \sum_{l=j}^{n-1} \delta^l \sum_{p \wedge q} \text{ad}_{\Omega_{k_1}} \text{ad}_{\Omega_{k_2}} \dots \text{ad}_{\Omega_{k_j}} A + \mathcal{O}(\delta^{n+1}) \\ &= \sum_{l=2}^n \delta^l \sum_{j=1}^{l-1} c_j \sum_{p \wedge q} \text{ad}_{\Omega_{k_1}} \text{ad}_{\Omega_{k_2}} \dots \text{ad}_{\Omega_{k_j}} A + \mathcal{O}(\delta^{n+1}), \end{aligned}$$

donde  $p$  y  $q$  corresponden a las representaciones de las condiciones lógicas  $k_1 + \dots + k_j = l$  y  $k_1 \geq 1 \wedge \dots \wedge k_j \geq 1$ , respectivamente, que se deben cumplir simultáneamente.

Entonces resulta claro que  $\Gamma_1 = A$  y

$$\Gamma_l = \sum_{j=2}^{l-1} c_j \sum_{p \wedge q} \text{ad}_{\Omega_{k_1}} \text{ad}_{\Omega_{k_2}} \dots \text{ad}_{\Omega_{k_j}} A,$$

pero  $c_1 = 0$ , de manera que  $\Gamma_2 = 0$ . Entonces queda finalmente

$$\begin{aligned} \Gamma_1 &= A, & \Gamma_2 &= 0, \\ \Gamma_n &= \sum_{j=2}^{n-1} c_j \sum_{p \wedge q} \text{ad}_{\Omega_{k_1}} \text{ad}_{\Omega_{k_2}} \dots \text{ad}_{\Omega_{k_j}} A, & n &\geq 3. \end{aligned} \quad (2.53)$$

En particular

$$\begin{aligned} \Gamma_3 &= -\frac{1}{24} [\Omega_1, [\Omega_1, A]] \\ \Gamma_4 &= -\frac{1}{24} ([\Omega_1, [\Omega_2, A]] + [\Omega_2, [\Omega_1, A]]). \end{aligned} \quad (2.54)$$

Ahora por la definición de  $\Gamma$ , la ecuación (2.50), se escribe

$$\Omega' = e^{-\Omega/2} \Gamma e^{\Omega/2},$$

que, después de integrarse sobre  $t$ , puede ser usado para la construcción de la sucesión de aproximaciones de  $\Omega$  una vez que los términos  $\Gamma_n$  son determinados en términos de  $\Omega_k, k \leq n - 2$ . Así, la  $n$ -ésima aproximación,  $\Omega^{(n)}$ , queda definida por

$$\Omega^{(n)}(t) = \int_0^t e^{-\frac{1}{2}\Omega^{(n-1)}(s)} \Gamma^{(n)} e^{\frac{1}{2}\Omega^{(n-1)}(s)} ds, \quad n = 1, 2, \dots \quad (2.55)$$

donde la dependencia de  $\delta$  ha sido omitida por simplicidad y  $\Gamma^{(n)} = \sum_{k=1}^n \delta^k \Gamma_k$ ,  $\Omega^{(0)} = 0$ . Así, las dos primeras aproximaciones explícitas son

$$\begin{aligned}\Omega^{(1)}(t; \delta) &= \delta \Omega_1(t) = \delta \int_0^t A(s) ds \\ \Omega^{(2)}(t; \delta) &= \delta \int_0^t e^{-\frac{1}{2}\Omega^{(1)}(s, \delta)} A(s) e^{\frac{1}{2}\Omega^{(1)}(s, \delta)} ds.\end{aligned}\tag{2.56}$$

Con este procedimiento la solución aproximada está dada por  $Y(t) \simeq e^{\Omega^{(n)}}$ . Se debe observar que  $\Omega^{(n)}$  contiene contribuciones de una infinidad de órdenes en  $\delta$ , mientras que el  $n$ -ésimo término en la serie de Magnus (2.50) es proporcional a  $\delta^n$ . Además,  $\Omega^{(n)}$  contiene  $\sum_{k=1}^n \delta^k \Omega_k$  y también potencias mayores de  $\delta^m$ ,  $m > n$ .

Analizamos a continuación el caso específico  $n = 2$ . Es claro que

$$\delta e^{-\frac{1}{2}\Omega^{(1)}} A e^{\frac{1}{2}\Omega^{(1)}} = \delta \sum_{k=0}^{\infty} \frac{(-1)^k}{2^k k!} \text{ad}_{\Omega^{(1)}}^k A = \delta \sum_{k=0}^{\infty} \frac{(-1)^k}{2^k k!} \delta^k \text{ad}_{\Omega_1}^k A,$$

de manera que

$$\begin{aligned}\Omega^{(2)}(t) &= \delta \sum_{k=0}^{\infty} \frac{(-1)^k}{2^k k!} \delta^k \int_0^t \text{ad}_{\Omega_1(s)}^k A(s) ds \\ &= \delta \int_0^t A(s) ds - \frac{1}{2} \delta^2 \int_0^t [\Omega_1(s), A(s)] ds + \delta \sum_{k=2}^{\infty} \dots \\ &= \delta \Omega_1(t) + \delta^2 \Omega_2(t) + \sum_{k=2}^{\infty} \frac{(-1)^k}{2^k k!} \delta^{k+1} \int_0^t \text{ad}_{\Omega_1(s)}^k A(s) ds.\end{aligned}$$

Entonces, finalmente tenemos

$$\Omega^{(2)}(t; \delta) = \delta \Omega_1(t) + \delta^2 \Omega_2(t) + R^{(2)}(\delta, t),$$

donde

$$R^{(2)}(\delta, t) \equiv \sum_{k=3}^{\infty} \sum_{l=k+1}^{\infty} r_{kl} \delta^k t^l = \sum_{k=3}^{\infty} \frac{(-1)^{k-1}}{2^{k-1} (k-1)!} \delta^k \int_0^t \text{ad}_{\Omega_1(s)}^{k-1} A(s) ds.$$

Se puede apreciar que  $\Omega^{(2)}$  contiene a  $\delta \Omega_1 + \delta^2 \Omega_2$  y a un número infinito de términos en  $\delta$  y  $t$ . Como consecuencia, consideramos  $\Omega^{(2)}$  como una aproximación de orden  $t^4$ , pero ahora incluidos una infinidad de términos en  $\delta$  de tal manera que  $\|\Omega^{(2)}\|/|\delta|$  tiene límite. A partir de la estructura de

la expresión (2.55) es también posible hallar el comportamiento asintótico de  $\Omega^{(n)}/\delta$  ( $n \geq 3$ ) para  $\delta \rightarrow \infty$  y demostrar que sigue siendo acotada, como es la solución exacta. Esta propiedad del algoritmo iterativo de Voslamber puede conducir a una mejor aproximación de  $Y(t)$  cuando el parámetro  $\delta$  no es muy pequeño, ya que en ese caso se espera que  $\Omega^{(n)}/\delta$  permanezca cerca de  $\Omega/\delta$ .

### 2.3.3. Un esquema numérico basado en el procedimiento de Voslamber

El procedimiento iterativo de Voslamber, aunque es más complicado que el de Magnus, presenta varias ventajas adicionales, por lo que parece conveniente tratar de diseñar un esquema numérico incorporando las principales propiedades del método. Específicamente, se construye un integrador numérico de cuarto orden a partir de la expresión (2.55). Este algoritmo se basa en expresar  $\Omega^{(2)}$  para un paso de tamaño  $h$  como

$$\Omega^{(2)}(h) = e^{-(\alpha_1 A_1 + \alpha_2 A_2)h} h (\beta_1 A_2 + \beta_2 A_1) e^{(\alpha_1 A_1 + \alpha_2 A_2)h}, \quad (2.57)$$

donde  $A_1$  y  $A_2$  son evaluaciones en los nodos de la regla de cuadratura de Gauss–Legendre de cuarto orden,

$$A_1 \equiv A(t_n + c_1 h), \quad A_2 \equiv A(t_n + c_2 h),$$

con

$$c_1 = \frac{1}{2} - \frac{\sqrt{3}}{6}, \quad c_2 = \frac{1}{2} + \frac{\sqrt{3}}{6}$$

y  $\alpha_i, \beta_i$  son coeficientes a determinar de manera que  $\Omega^{(2)}(h)$  coincida con la expresión exacta hasta orden  $h^4$ . Esto se puede hacer mediante la expansión en series de potencia de  $h$ , tanto de la expresión exacta

$$\Omega^{(2)}(t_n + h, t_n) = \int_{t_n}^{t_n+h} e^{-\frac{1}{2}\Omega^{(1)}(s)} A(s) e^{\frac{1}{2}\Omega^{(1)}(s)} ds$$

como de la expresión propuesta en (2.56) para  $A(t) = a_0 + a_1 t + a_2 t^2 + \dots$ . Procediendo así llegamos a

$$\begin{aligned} \Omega^{(2)}(t_n + h, t_n) = & a_0 h + \frac{1}{2} h^2 a_1 + h^3 \left( \frac{1}{3} a_2 - \frac{1}{12} [a_0, a_1] \right) \\ & + h^4 \left( \frac{1}{4} a_3 - \frac{1}{12} [a_0, a_2] + \frac{1}{64} [a_0, [a_0, a_1]] \right) + \mathcal{O}(h^5). \end{aligned}$$

Por otro lado, expandiendo (2.57) obtenemos

$$\begin{aligned}\Omega^{(2)}(h) &= h(\beta_1 + \beta_2)a_0 + h^2\beta_1\alpha_1 \\ &+ h^3a_2\frac{1}{6}(-(-2 + \sqrt{3})\beta_1 + (2 + \sqrt{3})\beta_2) + h^3[a_0, a_1]\frac{\alpha_2\beta_1 - \alpha_1\beta_2}{\sqrt{3}} \\ &+ h^4a_3\frac{1}{36}((9 - 5\sqrt{3})\beta_1 + (9 + 5\sqrt{3})\beta_2) + h^4[a_0, a_1]\frac{\alpha_2\beta_1 - \alpha_1\beta_2}{\sqrt{3}} \\ &+ h^4[a_0, [a_0, a_1]]\frac{(\alpha_1 + \alpha_2)(-\alpha_2\beta_1 + \alpha_1\beta_2)}{2\sqrt{3}}.\end{aligned}$$

Igualando esta expresión con la precedente se obtienen los valores de los coeficientes como

$$\alpha_1 = \frac{9 + 4\sqrt{3}}{48}, \quad \alpha_2 = \frac{9 - 4\sqrt{3}}{48}, \quad \beta_1 = \beta_2 = \frac{1}{2}.$$

En consecuencia, el método propuesto para avanzar la solución en el paso de  $t_n$  a  $t_{n+1} = t_n + h$  está dado por

$$\Omega^{(2)}(h) = \frac{h}{2}e^{-(\alpha_1 A_1 + \alpha_2 A_2)h}(A_1 + A_2)e^{(\alpha_1 A_1 + \alpha_2 A_2)h} \quad (2.58)$$

$$Y_{n+1} = e^{\Omega^{(2)}(h)}Y_n. \quad (2.59)$$

Este método de cuarto orden requiere dos evaluaciones de la matriz  $A$  por paso y dos exponenciales matriciales, pero así se evita el cálculo de conmutadores. En cualquier caso, conviene notar que una de las exponenciales es la inversa de la otra.

### 2.3.4. Ejemplos numéricos

Ilustramos a continuación el procedimiento de Voslamber en comparación con los métodos basados en el desarrollo de Magnus en dos ejemplos sencillos.

**Primer ejemplo.** El primer caso que analizamos corresponde a la ecuación

$$\frac{dY(\tau)}{d\tau} = \frac{1}{\epsilon}A(\tau)Y(\tau), \quad (2.60)$$

con

$$A(\tau) = \begin{pmatrix} \tau & \gamma \\ \gamma & -\tau \end{pmatrix}. \quad (2.61)$$

Esta matriz tiene valores propios  $\mu(\tau) = \pm\sqrt{\tau^2 + \gamma^2}$ . Por lo tanto, el parámetro  $\gamma > 0$  determina la distancia mínima entre los valores propios.

Elegimos  $\gamma = 2$  y calculamos la solución del sistema en el intervalo  $[-1, 1]$  con distintos tamaño de paso  $h$ .

Este ejemplo admite una solución en términos de funciones hipergeométricas, pero muy poco *amigable*, que se pueden obtener con un software de manipulación simbólica, como Mathematica.

Inicialmente estudiamos el tiempo de CPU versus el error relativo de los métodos de Voslamber, Magnus de orden 4 y 6, aplicados al problema. Para el cálculo de la exponencial de las matrices, en el método de Voslamber descrito en (2.58) y (2.59), se utilizó la función `expm()` de **Matlab**, además de la condición inicial  $Y_0 \equiv Y(-1) = I$ . Se puede notar, en la figura 2.1, que el esquema iterativo de Voslamber es competitivo con el método de Magnus de cuarto orden, mejorando ligeramente en precisión y tiempo de CPU, lo cual es consistente con la teoría, dado que el esquema numérico implementado del método de Voslamber es de orden 4. Por otra parte el método de Magnus de orden 6 presenta una pendiente más empinada (por su mayor orden) que el resto, además de invertir más tiempo de CPU.

Ahora consideramos el mismo problema, con la misma condición inicial, pero esta vez estudiamos el comportamiento de las aproximaciones cuando  $\epsilon \rightarrow 0$ , es decir,  $\delta = \frac{1}{\epsilon} \rightarrow \infty$ . Para conseguir esto, graficamos el comportamiento del error relativo de  $\Omega^{(n)}/\delta$  cuando  $n = 2$  en (2.58) y usando como tamaño de paso constante  $h = 10^{-4}$ . En la figura 2.2 se muestra los resultados. Cuando el parámetro  $\epsilon$  es pequeño, se aprecia que el error relativo de  $\Omega^{(2)}/\delta$ , decrece rápidamente, es decir,  $\|\Omega^{(n)}/\delta - \Omega/\delta\| \rightarrow 0$ , con  $n = 2$ . Situación que ya habíamos descrito, y que corresponde a una propiedad del método de Voslamber. Con este ejemplo se ilustra el hecho que el procedimiento de Voslamber construido va mejorando sus aproximaciones, superando a Magnus de cuarto orden para este caso, cuando los valores de  $\epsilon$  son pequeños.

**Segundo Ejemplo: modelo de Rosen–Zener.** Consideremos ahora un problema típico en mecánica cuántica: un sistema de dos niveles. El Hamiltoniano genérico de un sistema de dos niveles tiene la forma

$$H(t) = \begin{pmatrix} E_1(t) & C(t) \\ C^*(t) & E_2(t) \end{pmatrix}, \quad (2.62)$$

donde  $E_1(t), E_2(t)$  son funciones reales y  $C(t)$  es, en general, una función compleja de  $t$ . Se define la parte resoluble del Hamiltoniano como la matriz diagonal

$$H_0(t) = \begin{pmatrix} E_1(t) & 0 \\ 0 & E_2(t) \end{pmatrix} \quad (2.63)$$

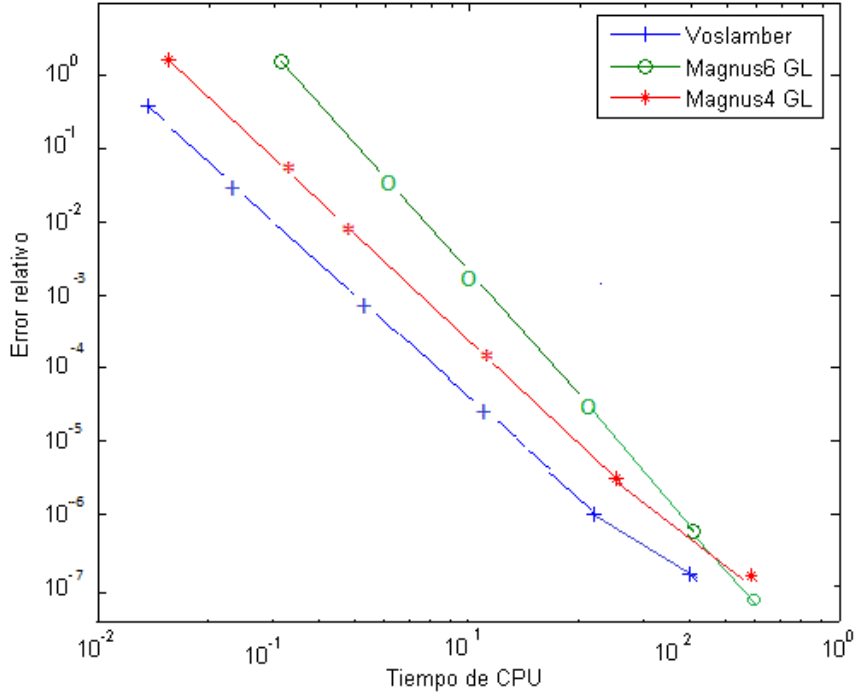


Figura 2.1: Tiempos de CPU VS error relativo de tres métodos: Voslamber, Magnus de orden 4 y 6. Los procedimientos numéricos se aplican a la ecuación (2.60) con  $A(\tau)$  dada por (2.61),  $\gamma = 2$ ,  $\epsilon = 1/25$ . Para los tres métodos se itera sobre el intervalo  $[-1, 1]$ , con  $Y_0 = I$ . Tomándose distintos valores de  $h$ , según la fórmula  $h = \frac{1}{2^k}$ , con  $k = 0, 1, \dots, 5$ . Se observa que el compartamiento del método de Voslamber se asemeja al de Magnus de cuarto orden. El método de Magnus de orden 6 presenta una gráfica de mayor pendiente, como era de esperar por su mayor orden. Se utilizó como referencia del valor exacto, en la comparación de todos los métodos, el dado por Magnus de sexto orden con  $h = 10^{-6}$ .

y toda la interacción dependiente del tiempo está descrita por la función  $C(t)$ , la cual se considera como una perturbación. En la imagen de interacción el



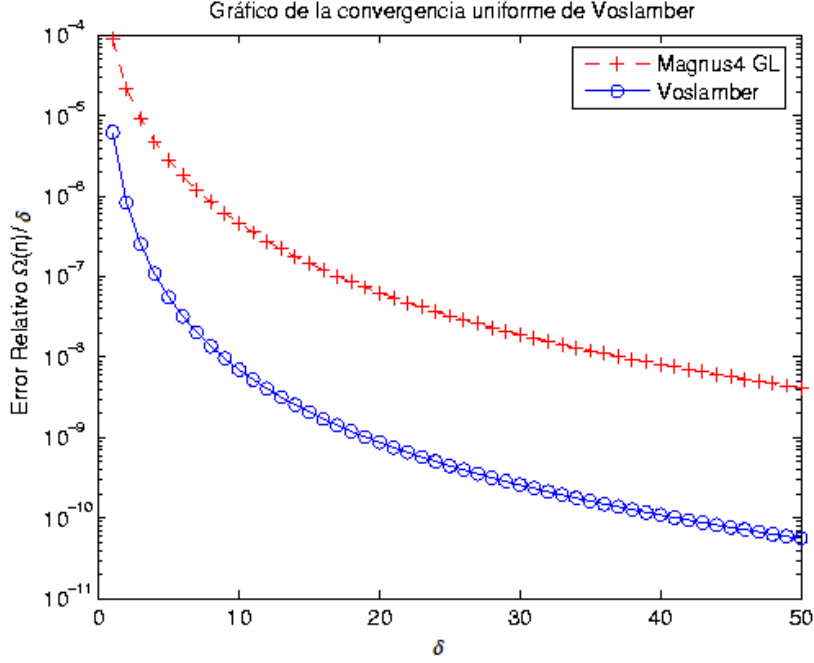


Figura 2.2: Comportamiento del error relativo de  $\Omega^{(n)}/\delta$ , es decir, se grafica en el eje vertical la expresión  $\|\Omega^{(n)}/\delta - \Omega/\delta\|/\|\Omega/\delta\|$  para el procedimiento de Voslamber y el de Magnus de orden 4, usando como aproximación de  $\Omega/\delta$  el método de Magnus de 6° orden con  $h = 10^{-6}$ , y barriendo los valores de  $\delta$  en el intervalo  $[0, 50]$ .

nuevo Hamiltoniano viene dado por

$$H_I(t) = \begin{pmatrix} 0 & C(t) \exp\left(i \int_0^t dt' \omega(t')\right) \\ C^*(t) \exp\left(-i \int_0^t dt' \omega(t')\right) & 0 \end{pmatrix} \quad (2.64)$$

con  $\omega = (E_1 - E_2)/\hbar$ . Supongamos que  $H_0$  no depende del tiempo. Entonces  $U(t) = \exp(-iH_0 t/\hbar) U_I(t)$ . Sin pérdida de generalidad, se puede considerar que  $H_0$  tiene traza nula, de manera que  $E_1 = -E_2 \equiv E$ . Así  $\pm E$  denota las energías propias correspondientes a los vectores propios  $|+\rangle \equiv (1, 0)^T$ ,  $|-\rangle \equiv (0, 1)^T$  de  $H_0$ , el sistema no perturbado. En términos de las matrices

de Pauli el Hamiltoniano se puede expresar como

$$H(t) = \frac{1}{2}\hbar\omega\sigma_3 + f(t)\sigma_1 + g(t)\sigma_2, \quad (2.65)$$

con  $f = \text{Re}(C)$  y  $g = -\text{Im}(C)$ .

Como  $H_0$  es diagonal, la probabilidad de transición entre los estados propios  $|+\rangle$ ,  $|-\rangle$  de  $H_0$  está dada por

$$P(t) = |\langle +|U_I(t)|-\rangle|^2. \quad (2.66)$$

Como caso particular consideramos el modelo de Rosen-Zener. En dicho modelo la interacción  $C(t)$  en (2.62) está dada por la función real

$$V(t) = V_0 \text{sech}(t/T) = \frac{2V_0}{e^{t/T} + e^{-t/T}},$$

donde  $T$  determina la escala temporal. Utilizamos la notación  $\gamma = \pi V_0 T/\hbar$  y  $\xi = \omega T = 2ET/\hbar$ .

El Hamiltoniano correspondiente, escrito en términos de las matrices de Pauli, es

$$H(t) = E\sigma_3 + V(t)\sigma_1 \equiv \mathbf{a}(t) \cdot \boldsymbol{\sigma}, \quad V(t) = V_0/\cosh(t/T), \quad (2.67)$$

con  $\mathbf{a} \equiv (V(t), 0, E)$ . En la imagen de interacción se tiene

$$H_I(s) = V(s)(\sigma_1 \cos(\xi s) - \sigma_2 \sin(\xi s)) \quad (2.68)$$

en términos de la variable temporal adimensional  $s = t/T$ . El resultado exacto para la probabilidad de transición (considerando un intervalo temporal que se extiende desde  $-\infty$  hasta  $+\infty$ ) es

$$P_{ex} = |(U)_{12}(-\infty, +\infty)|^2 = \frac{\sin^2 \gamma}{\cosh^2(\pi\xi/2)}. \quad (2.69)$$

Para aplicar el método numérico de Voslamber descrito en (2.58) y (2.59) usamos  $Y_{n+1} = e^{\Omega^{(2)}(h)}Y_n$ . Además, para este experimento numérico, tomamos los datos presentados en la siguiente tabla

Variable	Significado	Valor
$T$	Escala de tiempo	1
$E$	Energía	1
$V_0$	Potencial inicial	$\frac{2}{\pi}$
$\hbar$	Constante reducida de Planck	1
$s$	Intervalo de Tiempo	$[-25, 25]$

Las figuras 2.3 y 2.4 muestran las aproximaciones numéricas para la probabilidad de transición (2.69) del modelo de Rosen-Zener con distintos valores de  $\xi$ . Tanto el método de Voslamber como el de Magnus de cuarto orden se ejecutaron en el intervalo de tiempo  $[-25, 25]$ , como una referencia a un intervalo de tiempo que se extienda de  $-\infty$  a  $+\infty$ , para una colección de valores  $\gamma$  igualmente espaciados en el intervalo  $[0, 2\pi]$ .

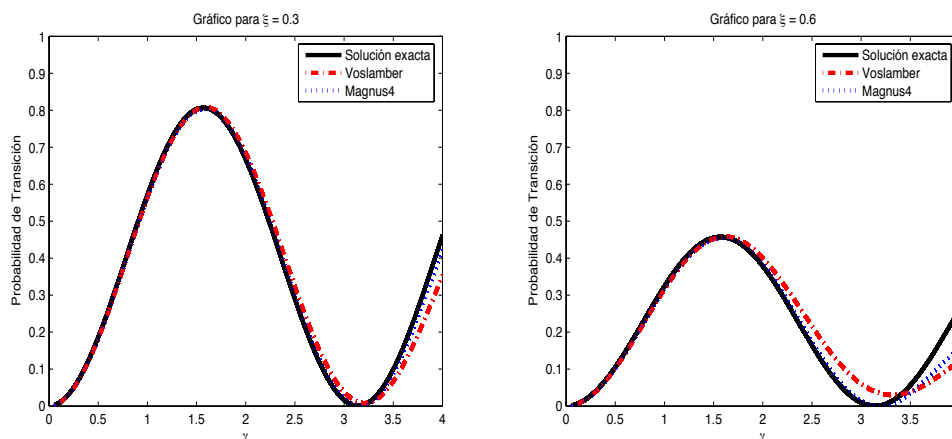


Figura 2.3: Gráfica de la probabilidad de transición, con  $\xi = 0.3$  y  $\xi = 0.6$ , para el modelo de Rosen-Zener.

Completamos el tratamiento de este ejemplo, figura 2.5, estudiando el comportamiento de las gráficas obtenidas para la probabilidad de transición, pero esta vez correspondientes a tomar un valor fijo de  $\gamma$  y variar  $\xi$ . En particular se tomaron:  $\gamma = 1$  y  $\gamma = 1.5$  con  $\xi \in [0, 2]$ . Se observa que al método de Voslamber le va muy bien en este caso, pues casi coincide con la curva analítica desplegada en la gráfica.

## 2.4. Métodos de splitting basados en matrices triangulares.

En las secciones anteriores se han tratado diversas técnicas para la resolución aproximada de la ecuación diferencial lineal matricial

$$Y' = A(t)Y, \quad Y(0) = I \quad (2.70)$$

que son particularmente apropiados cuando, como ocurre a menudo en las aplicaciones, la ecuación está definida en un grupo de Lie matricial  $\mathcal{G}$ .

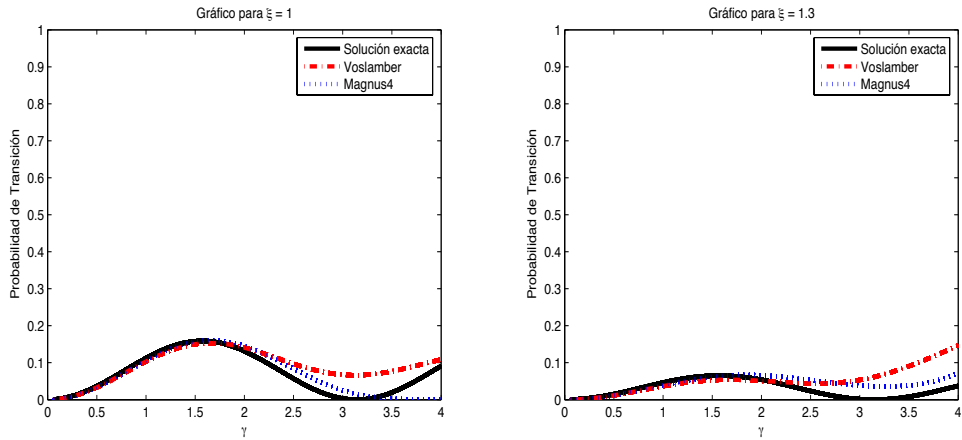


Figura 2.4: Gráfica de la probabilidad de transición, continuación. Se puede apreciar que la aproximación dada por el esquema numérico basado en la formulación de Voslamber tiende a estar próxima a la solución exacta, aunque en el tramo mostrado es superada por la aproximación dada por el método de Magnus de cuarto orden, para ambos valores de  $\xi$ ,  $\xi = 1$  y  $\xi = 1.3$ .

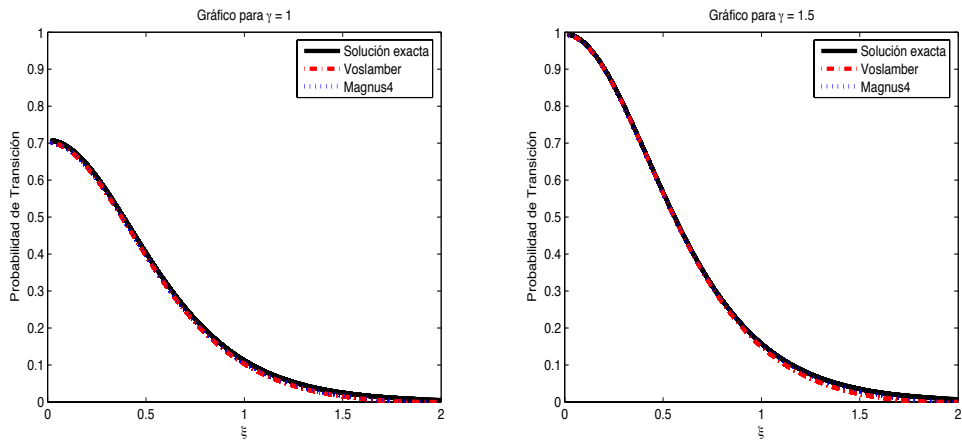


Figura 2.5: Gráfica de la probabilidad de transición, con  $\gamma = 1$  y  $\gamma = 1.5$ , en ambos casos haciendo variar  $\xi \in [0, 2]$  para el modelo de Rosen Zener.

Ejemplos relevantes aparecen en sistemas hamiltonianos ( $\mathcal{G} = \text{Sp}(n)$ ) y en mecánica cuántica ( $\mathcal{G} = \text{SU}(n)$ ). En ese sentido, los métodos (tanto analíticos como numéricos) basados en el desarrollo de Magnus y en el procedimiento iterativo de Voslamber tienen la peculiaridad de proporcionar

aproximaciones a la solución exacta que están definidas en el mismo grupo de Lie  $\mathcal{G}$ , independientemente de dónde se trunquen los desarrollos.

Ya se ha comentado (ver Apéndice A) que el problema de calcular exponenciales matriciales tiene una larga tradición en análisis numérico, y que existen diferentes procedimientos para calcularla, a menudo no exentos de problemas y limitaciones: aproximantes racionales, métodos basados en subespacios de Krylov, polinomios de Chebyshev, etc. En el caso de integradores en grupos de Lie es crucial que la aproximación usada transforme el álgebra de Lie  $\mathfrak{g}$  donde  $A(t)$  está definida en el grupo  $\mathcal{G}$ . Obviamente, esto siempre se puede lograr si la exponencial se calcula hasta error de redondeo, pero el procedimiento es costoso computacionalmente cuando la dimensión  $n$  es grande y el resultado puede estar influido por efectos no deseados debido a la acumulación de errores.

Otra posibilidad consiste en aproximar  $\exp(z)$  por una función  $R(z)$  diferenciable en una vecindad de  $z = 0$  de tal manera que  $e^z = R(z) + \mathcal{O}(h^{p+1})$ , donde  $p \geq 1$  es el orden del método de grupo de Lie y  $R(\mathfrak{g}) \subseteq \mathcal{G}$ . Este es el caso, por ejemplo, de aproximaciones basadas en desarrollos de Padé diagonales o la transformación de Cayley para grupos de Lie  $J$ -ortogonales, tales como el grupo ortogonal, el simpléctico y el grupo de Lorentz [VC04].

Hay casos, sin embargo, en que la aproximación de  $\exp(z)$  por dicha función  $R(z)$  no pertenece a  $\mathcal{G}$  y por lo tanto el método general no conserva la estructura de grupo de Lie. Esto se produce, en particular, cuando  $\mathcal{G} = \text{SL}(n)$ , el grupo lineal especial [Bak02]. Equivalentemente, cuando la matriz en (2.70) es tal que  $\text{tr}(A) = 0$ . En otras palabras, en el caso de que el problema (2.70) está definido en el grupo  $\text{SL}(n)$ , si queremos aplicar un método numérico basado en el desarrollo de Magnus, hemos de calcular necesariamente la correspondiente exponencial matricial (hasta error de redondeo) para que la aproximación resultante esté definida también en  $\text{SL}(n)$ . Usando cualesquiera de las aproximaciones a la exponencial enumeradas en el Apéndice A *no* tendremos garantizado que la solución numérica resultante está en  $\text{SL}(n)$ .

¿Por qué es conveniente que la aproximación esté en el grupo  $\text{SL}(n)$ ? Un conocido resultado de la teoría de ecuaciones diferenciales lineales establece que

$$\frac{d}{dt} \det Y = \text{tr}(A(t)) \det Y.$$

Por consiguiente, la solución de (2.70) preserva el volumen a lo largo del tiempo, mientras que esto sólo es cierto en el caso del desarrollo de Magnus si la exponencial se calcula hasta error de redondeo.

La preservación de volumen juega un papel importante en muchos sistemas dinámicos que surgen en las aplicaciones físicas, tales como perturbaciones de los sistemas hamiltonianos, en la discretización de las

ecuación de ondas, etc.

En esta sección presentamos un procedimiento para construir nuevos integradores numéricos para la ecuación (2.70) que no requieren el cálculo de la exponencial matricial y que sin embargo preservan la estructura del grupo de Lie  $SL(n)$  cuando  $\text{tr}(A) = 0$ . La solución aproximada se construye como un producto infinito de matrices triangulares superiores e inferiores obtenidos explícitamente como soluciones de ciertas ecuaciones diferenciales lineales en términos de cuadraturas. El procedimiento es iterativo, pero con pocas iteraciones es posible alcanzar órdenes extraordinariamente altos. El procedimiento se presentó en [Cas07], pero aquí haremos un análisis más exhaustivo y lo extenderemos hasta orden 6. Basado en estos resultados, presentaremos un procedimiento para aproximar la exponencial de una matriz con traza nula de manera que, por construcción, dicha aproximación posea determinante unidad. En otras palabras, construiremos un algoritmo computacionalmente efectivo capaz de aproximar la exponencial de un elemento de  $\mathfrak{sl}(n)$  de manera que pertenezca a  $SL(n)$  para todo  $n$ .

### 2.4.1. Esquema general del algoritmo

Partimos del sistema lineal  $Y' = A(t)Y$ , con  $Y_0 \equiv Y, A_0 \equiv A$ , y separamos la matriz como

$$A_0(t) = A_{0+}(t) + A_{0-}(t),$$

donde  $A_{0+} \in \nabla_n$  es una matriz triangular superior estricta, es decir, todos los elementos en la diagonal principal y abajo de ella son ceros. En forma análoga,  $A_{0-} \in \tilde{\Delta}_n$  es una triangular inferior. Representaremos

$$Y_0(t) = L_0(t)Z(t),$$

donde

$$L_0' = A_{0-}L_0, \quad L_0(0) = I.$$

Consecuentemente,  $L_0(t)$  es también triangular inferior y

$$Z' = C_0(t)Z, \quad Z(0) = I \quad \text{donde} \quad C_0(t) = L_0^{-1}(t)A_{0+}(t)L_0(t).$$

A continuación representamos

$$Z(t) = U_0(t)Y_1, \quad \text{por lo tanto} \quad Y_0(t) = L_0(t)U_0(t)Y_1(t),$$

con

$$U_0' = C_{0+}(t)U_0, \quad U_0(0) = I. \tag{2.71}$$

Aquí  $C_{0+} \in \tilde{\nabla}_n$  es la parte triangular superior de  $C_0$ , de manera que  $U_0(t)$  también es triangular superior y se puede obtener resolviendo (2.71). Ahora bien, es fácil demostrar que  $Y_1$  satisface

$$Y_1' = A_1(t)Y_1, \quad Y_1(0) = I, \quad (2.72)$$

con

$$A_1(t) = U_0^{-1}(t)C_{0-}(t)U_0(t). \quad (2.73)$$

De esta forma hemos completado el primer ciclo del procedimiento. A continuación se procede del mismo modo con  $Y_1$  y  $A_1$ . Así, separamos

$$A_1 = A_{1+} + A_{1-}, \quad A_{1+} \in \nabla_n, A_{1-} \in \tilde{\Delta}_n$$

y escribimos  $Y_1 = L_1U_1Y_2$  con  $L_1' = A_{1-}L_1$ ,  $L_1(0) = I$  y así sucesivamente. Al final se logra una factorización como la siguiente

$$Y(t) \equiv Y_0 = L_0(t)U_0(t)L_1(t)U_1(t) \cdots L_k(t)U_k(t)Y_{k+1}, \quad k = 0, 1, 2, \dots \quad (2.74)$$

con

$$\begin{aligned} A_k &= A_{k+} + A_{k-}, \quad A_{k+} \in \nabla_n, \quad A_{k-} \in \tilde{\Delta}_n, \\ L_k' &= A_{k-}L_k, \quad L_k(0) = I, \\ C_k &\equiv L_k^{-1}A_{k+}L_k = C_{k+} + C_{k-}, \quad C_{k+} \in \tilde{\nabla}_n, C_{k-} \in \Delta_n, \\ U_k' &= C_{k+}U_k, \quad U_k(0) = I, \\ A_{k+1} &\equiv U_k^{-1}C_{k-}U_k, \quad Y_{k+1}' = A_{k+1}Y_{k+1}, \quad k = 0, 1, 2, \dots \end{aligned} \quad (2.75)$$

El esquema iterativo se trunca haciendo  $Y_{k+1} = I$ , para algún  $k$ . En lo que sigue analizamos la aproximación obtenida. Para ello introducimos un parámetro  $\varepsilon > 0$  en la matriz  $A$ . El esquema original se obtiene sin más que hacer  $\varepsilon = 1$ . Consideramos primero la expansión de la serie de Taylor alrededor de  $t = 0$ ,

$$A(t) = \varepsilon \sum_{i=0}^{\infty} a_i t^i.$$

Cuando esta serie es insertada en (2.75), y después de un cálculo sencillo, se tiene

$$L_0(t) = I + \varepsilon t \alpha_1^{(0)} + \frac{1}{2} \varepsilon t^2 \alpha_2^{(0)} + \dots$$

donde  $\alpha_j^{(0)}$  es la parte  $a_{j-1}$ ,  $j \geq 1$  de matriz triangular. Entonces, por el cálculo del producto  $L_0^{-1}A_{0+}L_0$  y agrupando las potencias de  $\varepsilon$  y  $t$  obtenemos

$$\begin{aligned} C_{0+} &= \varepsilon \gamma_1^{(0)} + t(\varepsilon \gamma_2^{(0)}) + \varepsilon^2 \gamma_3^{(0)} + \dots, \\ C_{0-}(t) &= t \varepsilon^2 \delta_1^{(0)} + t^2(\varepsilon^2 \delta_2^{(0)} + \varepsilon^3 \delta_3^{(0)}) + \dots \end{aligned}$$

para los coeficientes determinados  $\gamma_j^{(0)}, \delta_j^{(0)}$  en función de  $a_i$ , y entonces

$$U_0(t) = I + \varepsilon t \beta_1^{(0)} + t^2(\varepsilon \beta_2^{(0)} \varepsilon^2 \beta_3^{(0)}) + \dots,$$

con  $\beta_1^{(0)} = a_0 \alpha_1^{(0)}$ . A continuación se calcula  $A_1 = A_{1+} + A_{1-}$  con

$$\begin{aligned} A_{1-}(t) &= t \varepsilon^2 \alpha_1^{(1)} + t^2(\varepsilon^2 \alpha_2^{(1)} + \varepsilon^3 \alpha_3^{(1)}) + \dots, \\ A_{1+}(t) &= t^2 \varepsilon^3 \beta_1^{(1)} + t^3(\varepsilon^3 \beta_2^{(1)} + \varepsilon^4 \beta_3^{(1)}) + \dots, \end{aligned}$$

donde, en particular,  $\alpha_j^{(1)} = \delta_j^{(0)}, j = 1, 2, 3$ . Procediendo por inducción se obtiene en general

$$\begin{aligned} A_{j-} &= t^{n_j} \varepsilon^{n_j} (\varepsilon \alpha_1^{(j)} + t(\varepsilon \alpha_2^{(j)} + \varepsilon^2 \alpha_3^{(j)}) + \mathcal{O}(t^2)), \\ A_{j+} &= t^{m_j} \varepsilon^{m_j} (\varepsilon \beta_1^{(j)} + t(\varepsilon \beta_2^{(j)} + \varepsilon^2 \beta_3^{(j)}) + \mathcal{O}(t^2)). \end{aligned} \quad (2.76)$$

Al insertar (2.76) en el algoritmo (2.75) y repitiendo todo el procedimiento se obtiene, para el paso  $j$ -ésimo:

$$\begin{aligned} L_j(t) &= I + \frac{1}{n_j + 1} (t\varepsilon)^{n_j+1} \alpha_1^{(j)} + \frac{1}{n_j + 2} t^{n_j+2} \varepsilon^{n_j} (\varepsilon \alpha_2^{(j)} + \varepsilon^2 \alpha_3^{(j)}) + \dots, \\ U_j(t) &= I + \frac{1}{m_j + 1} (t\varepsilon)^{m_j+1} \beta_1^{(j)} + \frac{1}{m_j + 2} t^{m_j+2} \varepsilon^{m_j} (\varepsilon \beta_2^{(j)} + \varepsilon^2 \beta_3^{(j)}) + \dots \end{aligned} \quad (2.77)$$

y expresiones del tipo (2.76) para  $A_{(j+1)-}(t)$  y  $A_{(j+1)+}(t)$ , pero ahora con

$$\begin{aligned} n_{j+1} &= n_j + m_j + 1, \\ m_{j+1} &= n_j + 2m_j + 2, \quad j = 1, 2, \dots \end{aligned} \quad (2.78)$$

Los primeros valores de  $n_j$  y  $m_j$  aparecen en la siguiente tabla.

j	1	2	3	4	5
$n_j$	1	4	12	33	88
$m_j$	2	7	20	54	143

Examinando la tabla y la expresión (2.78), es claro que al truncar la factorización (2.74) con  $k = 0$  ( $Y_1 = I$ ) se tiene una aproximación de primer orden en  $t$ , mientras que con  $k = 1$  se tiene

$$Y(t) = L_0(t)U_0(t)L_1(t)U_1(t) + \mathcal{O}(t^5\varepsilon^5). \quad (2.79)$$



Análogamente  $Y(t) \approx L_0 U_0 L_1 U_1 L_2$  produce una aproximación de orden 7,  
 $Y(t) \approx L_0 U_0 L_1 U_1 L_2 U_2$  produce una aproximación de orden 12,  
 $Y(t) \approx L_0 U_0 L_1 U_1 L_2 U_2 L_3$  produce una aproximación de orden 20,  
 $Y(t) \approx L_0 U_0 L_1 U_1 L_2 U_2 L_3 U_3$  produce una aproximación de orden 33.

Estos resultados también muestran que el algoritmo (2.74)-(2.75) es particularmente adecuado para los sistemas de la forma  $Y' = (B_0 + \varepsilon B_1)Y$  cuando la ecuación  $U' = B_0 U$  puede ser resuelta exactamente. En este caso la solución puede ser factorizada como  $Y = UV$  con  $V' = \varepsilon U^{-1} B_1 UV$  y se aplican las consideraciones anteriores a esta última ecuación.

Por otro lado, el comportamiento del procedimiento con respecto a la preservación de la estructura de grupo Lie queda descrito por el siguiente teorema:

**Teorema 8** *Si la matriz de coeficientes  $A(t)$  en  $Y' = A(t)Y$  pertenece a  $\mathfrak{sl}(n)$ , es decir, al álgebra de Lie de las matrices de  $n \times n$  con traza 0, entonces el algoritmo descrito en (2.74)-(2.75) produce una solución aproximada  $Y^{[k]}(t)$  en el correspondiente grupo de Lie  $SL(n)$ .*

*Demostración.* Procederemos por inducción. Para esto es suficiente analizar el paso  $k$ -ésimo del algoritmo ( $k \geq 0$ ), [Cas07].

Vamos a dividir  $A_k$ , en dos matrices,  $A_{k+} \in \nabla_n$  matriz triangular superior estricta y una triangular inferior débil  $A_{k-} \in \Delta_n$ . Entonces  $A_k$  pertenece a una subálgebra soluble de  $\mathfrak{sl}(n)$ , ya que  $A_k$  tiene traza nula. Por lo tanto la solución  $L_k(t)$  del problema de valor inicial  $L'_k = A_{k-} L_k$  puede obtenerse en forma explícita, en un subgrupo soluble de  $SL(n)$ . A continuación, observamos que la traza de  $A_{k+}$  es cero y como la traza es invariante bajo una transformación de semejanza, tenemos  $\text{tr}(C_k) = \text{tr}(L_k^{-1} A_{k+} L_k) = \text{tr}(A_{k+}) = 0$ , luego  $C_k \in \mathfrak{sl}(n)$ .

La nueva matriz es separada como  $C_k = C_{k+} + C_{k-}$ , con  $C_{k+} \in \tilde{\nabla}_n$  y  $C_{k-} \in \Delta_n$ . Aplicando un argumento similar al aplicado anteriormente, resulta claro que la solución del problema de valor inicial  $U'_k = C_{k+} U_k$ , con  $U_k(0) = I$ , también está en  $SL(n)$  y  $A_{K+1} = U_k^{-1} C_{k-} U_k \in \mathfrak{sl}(n)$ .  $\square$

A continuación se construye un nuevo integrador geométrico basado en este desarrollo, y posteriormente lo ilustramos en un ejemplo numérico sencillo.

## Construcción de un Integrador Geométrico

Este nuevo algoritmo, en su versión de cuarto orden, requiere dos ciclos del proceso anterior. A continuación se detallan los pasos del proceso. Denotamos

por  $a_{ij}(t)$  los elementos de  $A_0 \equiv A(t)$ ,  $L_{ij}(t)$  serán los elementos de  $L_0(t)$ ,  $j \leq i$  y finalmente

$$A_{ii}(t) \equiv \int_0^t a_{ii}(s) ds.$$

**Paso 1.-** En primer paso se elige un tamaño de paso  $h$  y se consideran los elementos diagonales  $A_{ii}(h)$ ,  $i = 1, 2, \dots, n$ , que han de aproximarse hasta orden cuatro. Esto se puede hacer con la regla de Simpson,

$$\begin{aligned} A_{ii}(h) &= \int_0^h a_{ii}(t) dt = \frac{h}{6} \{a_{ii}(0) + 4a_{ii}(h/2) + a_{ii}(h)\} + \mathcal{O}(h^5) \\ &\equiv \tilde{A}_{ii}(h) + \mathcal{O}(h^5). \end{aligned}$$

Además

$$\begin{aligned} A_{ii}(h/2) &= \frac{h}{24} \{a_{ii}(0) + 8a_{ii}(h/2) - a_{ii}(h)\} + \mathcal{O}(h^4) \\ &\equiv \tilde{A}_{ii}(h/2) + \mathcal{O}(h^4). \end{aligned}$$

**Paso 2.-** El cómputo de  $L_0$  procede como sigue: primero se determinan los elementos de la diagonal principal de la matriz, es decir,

$$\begin{aligned} L_{ii}(h) &= e^{\tilde{A}_{ii}(h)} + \mathcal{O}(h^5), \quad i = 1, 2, \dots, n \\ L_{ii}(h/2) &= e^{\tilde{A}_{ii}(h/2)} + \mathcal{O}(h^4), \quad i = 1, 2, \dots, n-1. \end{aligned}$$

Luego las correspondientes aproximaciones de  $L_{ij}(h)$ , y  $L_{ij}(h/2)$ ,  $j < i$ , se hacen

$$\begin{aligned} L_{ij}(h) &= e^{A_{ii}(h)} \int_0^h F_{ij}(t) dt, \quad \text{donde} \\ F_{ij}(t) &= e^{-A_{ii}(t)} \sum_{k=j}^{i-1} a_{ik}(t) L_{kj}(t). \end{aligned}$$

Así

$$\begin{aligned} L_{ij}(h) &= e^{\tilde{A}_{ii}(h)} \frac{h}{6} (a_{ij}(0) + 4F_{ij}(h/2) + F_{ij}(h)) + \mathcal{O}(h^5), \\ L_{ij}(h/2) &= e^{\tilde{A}_{ii}(h/2)} \frac{h}{24} (5a_{ij}(0) + 8F_{ij}(h/2) - F_{ij}(h)) + \mathcal{O}(h^4), \end{aligned}$$

donde  $F_{ij}(h/2)$  y  $F_{ij}(h)$  pueden obtenerse hasta orden  $h^3$ .

**Paso 3.-** Ahora la matriz  $C_0$  puede aproximarse como

$$\begin{aligned} C_0(0) &= A_{0+}(0) \\ C_0(h/2) &= L_0^{-1}(h/2) A_{0+}(h/2) L_0(h/2) \end{aligned}$$

con error  $\mathcal{O}(h^4)$  y

$$C_0(h) = L_0^{-1}(h)A_{0+}(h)L_0(h)$$

con error  $\mathcal{O}(h^5)$ .

**Paso 4.-** En forma análoga al paso 1, pero ahora con la matriz  $C$ , se tiene

$$\begin{aligned}\tilde{C}_{ii}(h) &= \frac{h}{6}(C_{ii}(0) + 4C_{ii}(h/2) + C_{ii}(h)) \\ \tilde{C}_{ii}(h/2) &= \frac{h}{24}(5C_{ii}(0) + 8C_{ii}(h/2) - C_{ii}(h)).\end{aligned}$$

**Paso 5.-** Calculamos la matriz  $U$ . Primero su diagonal:

$$\begin{aligned}U_{ii}(h) &= e^{\tilde{C}_{ii}(h)} + \mathcal{O}(h^5), \\ U_{ii}(h/2) &= e^{\tilde{C}_{ii}(h/2)} + \mathcal{O}(h^4),\end{aligned}$$

y para los demás elementos

$$\begin{aligned}U_{ij}(h) &= e^{C_{ii}(h)} \int_0^h G_{ij}(t) dt, \quad \text{donde} \\ G_{ij}(t) &= e^{-C_{ii}(h)} \sum_{k=i+1}^j C_{ik}(t) U_{kj}(t),\end{aligned}$$

y entonces

$$\begin{aligned}U_{ij}(h) &= e^{\tilde{C}_{ii}(h)} \frac{h}{6}(C_{ij}(0) + G_{ij}(h/2) + G_{ij}(h)) + \mathcal{O}(h^5), \\ U_{ij}(h/2) &= e^{\tilde{C}_{ii}(h/2)} \frac{h}{24}(5C_{ij}(0) + 8G_{ij}(h/2) - G_{ij}(h)) + \mathcal{O}(h^4).\end{aligned}$$

Para el segundo ciclo, se calcula la matriz  $A_1$  en los nodos de cuadratura

$$\begin{aligned}A_1(0) &= C_{0-}(0) \\ A_1(h/2) &= U_0^{-1}(h/2)C_{0-}(h/2)U_0(h/2), \quad \text{con error } \mathcal{O}(h^4) \\ A_1(h) &= U_0^{-1}(h)C_{0-}(h)U_0(h), \quad \text{con error } \mathcal{O}(h^5)\end{aligned}$$

y los pasos del 1 al 5 se reemplaza con  $A_1$ .

**Paso 6.-** Una vez alcanzado el orden del esquema de aproximación deseado se aplica la factorización (2.74).

### Ejemplo numérico

Este esquema de integración de orden 4 se aplica a continuación al problema lineal (2.1), con  $Y_0 = I$  y dos matrices de coeficientes distintas.

La primera está dada por

$$A_{ij} = \sin[t(i^2 - j^2)], \quad 1 \leq i \leq j \leq n \quad (2.80)$$

con  $A_{ij} = A_{ji}$  y  $n = 10$ ,  $t \in [0, 10]$ . Se observa que  $\text{tr}(A) = 0$ . En este caso la solución ortogonal  $Y(t)$  oscila con el tiempo, principalmente debido a la dependencia del tiempo de  $A(t)$ . Esto queda de manifiesto al observar la gráfica de componente  $Y_{11}(t)$  de la solución, figura (2.6)(izquierda).

En la gráfica derecha de la misma figura 2.6 se representa la solución aproximada para el mismo problema diferencial, pero esta vez con la matriz

$$A_{ij} = \log \left( 1 + t \frac{j-i}{j+i} \right), \quad 1 \leq i \leq j \leq n. \quad (2.81)$$

Por otra parte, en la figura 2.7 se muestra los tiempos de CPU que requiere el algoritmo de splitting anterior y el que utiliza la expansión de Magnus, ambos de orden cuatro, para el problema diferencial matricial con  $A$  dado por (2.80).

A manera de síntesis, el nuevo algoritmo basado en *splitting* tiene un tiempo de CPU menor que el requerido por el integrador numérico de cuarto orden equivalente basado en Magnus, y además una mejora en la precisión, al dar un error relativo menor al de Magnus para este ejemplo.

#### 2.4.2. Algoritmo de orden 6

Se puede lograr una aproximación de orden 6 en  $t$  con el procedimiento anterior si se aplican dos ciclos y medio, es decir, con  $L_0 U_0 L_1 U_1 L_2$ . Esto requerirá además utilizar reglas de cuadratura de más precisión que las consideradas previamente. En particular, poremos utilizar la regla de Bode para el cálculo aproximado de integrales:

$$L_{ij}(h) \approx \sum_{k=j}^{i-1} a_{ik} e^{a_{ii}h} \left( \frac{2h}{45} \right) [7\delta_{kj} + 32F_{kj}(h/4) + 12F_{kj}(h/2) + 32F_{kj}(3h/4) + 7F_{kj}(h)]$$

$$L_{ij}(h/2) \approx \sum_{k=j}^{i-1} a_{ik} e^{a_{ii}(h/2)} \left( \frac{2h}{45} \right) [7\delta_{kj} + 32F_{kj}(h/8) + 12F_{kj}(h/4) + 32F_{kj}(3h/8) + 7F_{kj}(h/2)]$$

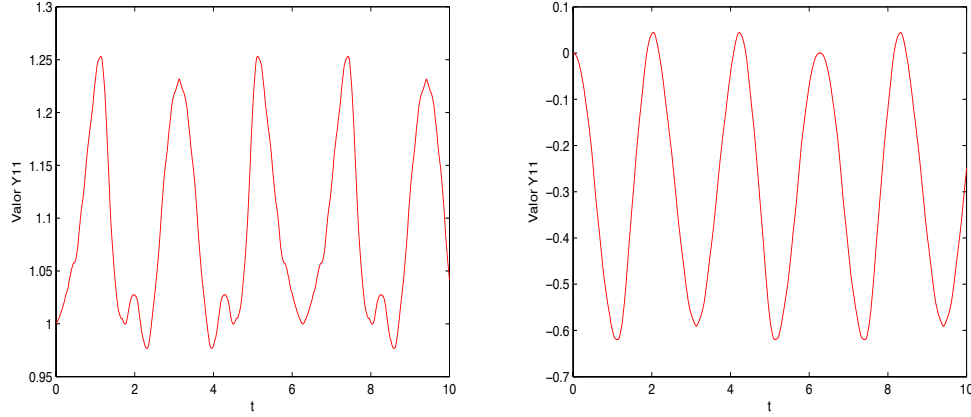


Figura 2.6: Gráfico (izquierda) de la componente  $Y(1, 1)$  de la solución aproximada del problema (2.70), con  $A_{ij} = \sin[t(i^2 - j^2)]$ ,  $1 \leq i \leq j \leq 10$ ;  $t \in [0, 10]$ , y la condición inicial  $Y_0 = I$ , dada por el método basado en *splitting*. La gráfica (derecha) del componente  $Y(1, 2)$ , para el problema diferencial matricial (2.70), con  $A_{ij} = \log\left(1 + t\frac{j-i}{j+i}\right)$ ,  $1 \leq i \leq j \leq 10$ ;  $t \in [0, 10]$ . En ambos casos se puede apreciar la forma altamente oscilatoria de esta componente de la matriz solución.

También es preciso aproximar  $L_{ij}(\frac{h}{4})$ . Partimos de la expresión

$$\begin{aligned} L_{ij}\left(\frac{h}{4}\right) &= e^{A_{ii}(h/4)} \int_0^{h/4} F_{ij}(t) dt \\ &\equiv e^{\tilde{A}_{ii}(h/4)} h [\alpha_1 F_{ij}(0) + \alpha_2 F_{ij}(h/4) + \alpha_3 F_{ij}(h/2) + \alpha_4 F_{ij}(3h/4) + \alpha_5 F_{ij}(h)] + \\ &\quad \mathcal{O}(h^7), \end{aligned}$$

donde  $\tilde{A}_{ii}(h/4)$  usaría los mismos puntos de cuadratura en la integración aproximada de  $A_{ii}(h/4)$ . Usando la serie de Taylor para  $F_{ij}(t)$  (en torno a 0), tenemos

$$F_{ij}(t) = F_{ij}(0) + F'_{ij}(0)t + F''_{ij}(0)\frac{t^2}{2} + F'''_{ij}(0)\frac{t^3}{6} + F^{(4)}_{ij}(0)\frac{t^4}{24} + \mathcal{O}(h^5).$$

Así

$$\begin{aligned} \int_0^{h/4} F_{ij}(t) dt &= F_{ij}(0)\frac{h}{4} + F'_{ij}(0)\frac{1}{2}\left(\frac{h}{4}\right)^2 + F''_{ij}(0)\frac{1}{3!}\left(\frac{h}{4}\right)^3 + F'''_{ij}(0)\frac{1}{4!}\left(\frac{h}{4}\right)^4 + \\ &\quad F^{(4)}_{ij}(0)\frac{1}{5!}\left(\frac{h}{4}\right)^5 + \mathcal{O}(h^6). \end{aligned}$$

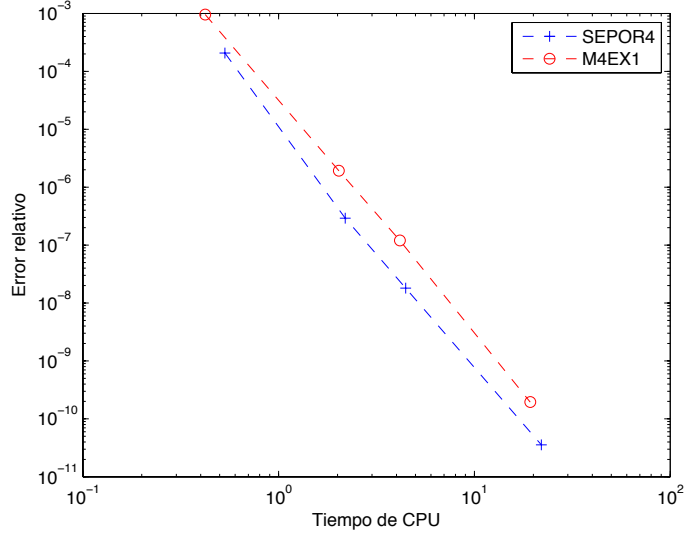


Figura 2.7: Gráfico de tiempos de CPU y error relativos del algoritmo basado en división (SEPOR4) y el integrador de Magnus (M4EX1) para el problema (2.80) Se observa que ambos métodos presentan pendientes semejantes, al ser del mismo orden. Sin embargo, el método basado en *splitting* registra una mejor precisión (error relativo menor), invirtiendo menos tiempo.

Luego

$$\begin{aligned}
\int_0^{h/4} F_{ij}(t) dt &= h [d_1 F_{ij}(0) + d_2 F_{ij}(h/4) + d_3 F_{ij}(h/2) + d_4 F_{ij}(3h/4) + d_5 F_{ij}(h)] \\
&+ \mathcal{O}(h^7) \\
&= h d_1 F_{ij}(0) + \\
&h d_2 \left[ F_{ij}(0) + F'_{ij}(0) \frac{h}{4} + F''_{ij}(0) \frac{h^2}{32} + F'''_{ij}(0) \frac{h^3}{384} + F_{ij}^{(4)}(0) \frac{h^4}{6144} \right] + \\
&h d_3 \left[ F_{ij}(0) + F'_{ij}(0) \frac{h}{2} + F''_{ij}(0) \frac{h^2}{8} + F'''_{ij}(0) \frac{h^3}{48} + F_{ij}^{(4)}(0) \frac{h^4}{384} \right] + \\
&h d_4 \left[ F_{ij}(0) + F'_{ij}(0) \frac{3h}{4} + F''_{ij}(0) \frac{9h^2}{32} + F'''_{ij}(0) \frac{9h^3}{128} + F_{ij}^{(4)}(0) \frac{81h^4}{6144} \right] + \\
&h d_5 \left[ F_{ij}(0) + F'_{ij}(0) h + F''_{ij}(0) \frac{h^2}{2} + F'''_{ij}(0) \frac{h^3}{6} + F_{ij}^{(4)}(0) \frac{h^4}{24} \right].
\end{aligned}$$

Agrupando adecuadamente se obtiene

$$\begin{aligned}
& hF_{ij}(0)(d_1 + d_2 + d_3 + d_4 + d_5) + \\
& h^2F'_{ij}(0) \left( \frac{1}{4}d_2 + \frac{1}{2}d_3 + \frac{3}{4}d_4 + d_5 \right) + \\
& h^3F''_{ij}(0) \left( \frac{1}{32}d_2 + \frac{1}{8}d_3 + \frac{9}{32}d_4 + \frac{1}{2}d_5 \right) + \\
& h^4F'''_{ij}(0) \left( \frac{1}{384}d_2 + \frac{1}{48}d_3 + \frac{9}{128}d_4 + \frac{1}{6}d_5 \right) + \\
& h^5F^{(4)}_{ij}(0) \left( \frac{1}{6144}d_2 + \frac{1}{384}d_3 + \frac{81}{6144}d_4 + \frac{1}{24}d_5 \right).
\end{aligned}$$

Como, por otra parte,

$$\int_0^{h/4} F_{ij}(t)dt = \frac{1}{4}F_{ij}(0)h + \frac{1}{32}F'_{ij}(0)h^2 + \frac{1}{384}F''_{ij}(0)h^3 + \frac{1}{6144}F'''_{ij}(0)h^4 + \frac{1}{122880}F^{(4)}_{ij}(0)h^5,$$

se obtiene el sistema de ecuaciones lineales

$$\begin{aligned}
d_1 + d_2 + d_3 + d_4 + d_5 &= \frac{1}{4} \\
\frac{1}{4}d_2 + \frac{1}{2}d_3 + \frac{3}{4}d_4 + d_5 &= \frac{1}{32} \\
\frac{1}{32}d_2 + \frac{1}{8}d_3 + \frac{9}{32}d_4 + \frac{1}{2}d_5 &= \frac{1}{384} \\
\frac{1}{384}d_2 + \frac{1}{48}d_3 + \frac{9}{128}d_4 + \frac{1}{6}d_5 &= \frac{1}{6144} \\
\frac{1}{6144}d_2 + \frac{1}{384}d_3 + \frac{81}{6144}d_4 + \frac{1}{24}d_5 &= \frac{1}{122880},
\end{aligned}$$

cuya solución viene dada por

$$\begin{aligned}
d_1 &= \frac{6977}{32760} & d_2 &= \frac{-2263}{131040} & d_3 &= \frac{391}{4368} \\
d_4 &= \frac{-5729}{131040} & d_5 &= \frac{557}{65520}.
\end{aligned}$$

En forma análoga se obtienen los coeficientes en las cuadraturas correspondientes a  $\int_0^{h/2} F_{ij}(t)dt$ ,  $\int_0^{3h/4} F_{ij}(t)dt$  y  $\int_0^h F_{ij}(t)dt$ . Estos valores se muestran en la siguiente tabla.

$t$	$w_1$	$w_2$	$w_3$	$w_4$	$w_5$
$\frac{h}{2}$	$\frac{-6069}{155360}$	$\frac{1839}{3884}$	$\frac{327141}{1242880}$	$\frac{-7101}{1242880}$	$\frac{282}{4855}$
$\frac{3h}{4}$	$\frac{2551}{14565}$	$\frac{1592}{8739}$	$\frac{35447}{233040}$	$\frac{313}{233040}$	$\frac{-941}{87390}$
$h$	$\frac{-2218}{14565}$	$\frac{5248}{8739}$	$\frac{4639}{14565}$	$\frac{-199}{14565}$	$\frac{10789}{43695}$

### Ejemplo numérico

La implementación del método de splitting desarrollado anteriormente se aplica al problema lineal definido por la matriz de coeficientes

$$A_{ij} = \sin[t(i^2 - j^2)], \quad 1 \leq i \leq j \leq n.$$

La figura 2.8 muestra los tiempos de CPU vs el error relativo del método de orden 6 y el de orden 4, para el sistema matricial diferencial  $Y' = A(t)Y, t \in [0, 10]$ . Las gráficas se contruyen usando los valores de  $h = 10^{-1}, 10^{-2}, 10^{-3}$  y  $10^{-4}$ . Se aprecia en el gráfico que el método de orden 6 disminuye el error relativo más que el método de orden 4. Sin embargo, esto requiere invertir más tiempo de CPU, lo que se aprecia como una traslación de la recta del método de orden 6, a la derecha de la figura. En otras palabras, no parece que el nuevo método de orden 6 sea más eficiente que usar el método de orden 4 con un tamaño de paso más pequeño.

## 2.5. Tamaño de paso adaptativo.

Para lograr resultados numéricos más precisos, pero sin el dilema de invertir más recursos de CPU, se puede recurrir a una estrategia de paso adaptativo, o de selección automática del paso de integración. De esa forma, es posible cambiar el paso de integración de manera que el error cometido  $\gamma$  se encuentre siempre por debajo de cierta tolerancia  $\delta$ . Esto siempre es posible en el caso de métodos de un paso que permitan la estimación del error. Un algoritmo de estas características ampliamente usado es el siguiente:

### Algoritmo *h-adaptativo*

---

1. En cada iteración, dado un paso  $h$ , el algoritmo entrega dos aproximaciones de  $Y(t)$ ,  $\hat{Y}$  e  $Y$ , de orden  $\hat{p} = 2$  y  $p = 4$  respectivamente, con

$$\hat{Y} = L_0 U_0 L_1$$



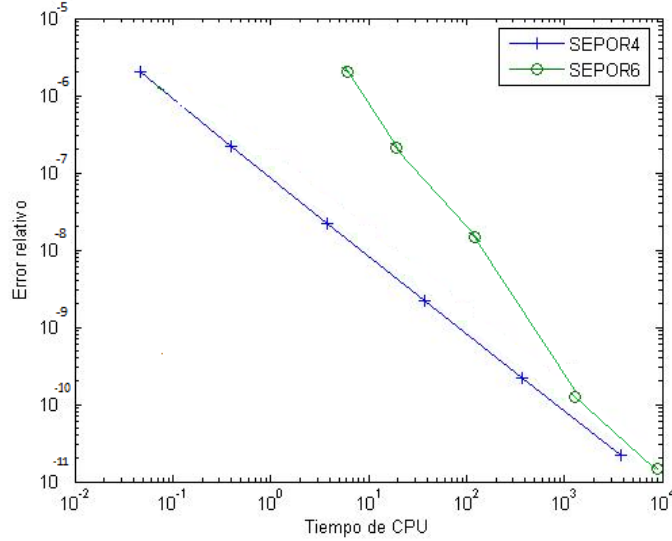


Figura 2.8: Gráfico de tiempos de CPU y error relativo del algoritmo basado en división de orden 6 (SEPOR6), comparado con el de orden 4 (SEPOR4). El método de orden 6 resulta de mayor precisión, sin embargo, invierte más tiempo de CPU. Lo cual queda de manifiesto al observar la pendiente más pronunciada del método *splitting* de sexto orden y su desplazamiento a la derecha de la gráfica, respectivamente. Se utiliza una expansión de Magnus de orden sexto como referencia, con  $h = 10^{-6}$ .

$$Y = L_0 U_0 L_1 U_1$$

2. Deseamos que  $\|\widehat{Y} - Y\| \leq Tol$ ,  $1 \leq i, j \leq n$ ,

$$\text{donde } Tol_{ij} = ER_{ij} + \max\{|\widehat{Y}_{ij}| - |Y_{ij}|\} \cdot EA_{ij}$$

3. Calculamos  $err = \sqrt{\frac{1}{n} \sum_{i=1}^n \left( \frac{\widehat{Y}_{ij} - Y_{ij}}{Tol_{ij}} \right)^2}$

4. El nuevo paso se obtiene de la fórmula:

$$h_{new} = h \cdot \min\{f_{max}, w\}$$

$$\text{donde } w = \max\left\{f_{min}, f \cdot \left(\frac{1}{err}\right)^{\frac{1}{q+1}}\right\}$$

con  $q = \min\{\widehat{p}, p\} = 2$ ,  $f_{max} \in [1.5; 5]$ ,  $f_{min} = h \cdot 10^{-3}$  y el factor de seguridad  $f$ , que se suele tomar del conjunto  $\{0.8; 0.9; (\frac{1}{4})^{\frac{1}{q+1}}\}$

5. El nuevo paso,  $h_{new}$ , calculado en el paso anterior, se utilizará efectivamente en la próxima iteración del método si  $err \leq 1$ ; si no,

se utilizará el mismo valor de paso  $h$ .

---

Estudiamos el comportamiento del método de *splitting con  $h$  adaptativo*, para nuestra matriz  $A$  usada con anterioridad,

$$A_{ij} = \sin[t(i^2 - j^2)], \quad 1 \leq i \leq j \leq n$$

con  $A_{ij} = A_{ji}$  y  $n = 10$ , pero ahora la integración se lleva a cabo en el intervalo  $t \in [0, 1]$ .

En la figura 2.9, se registra el comportamiento del valor del paso  $h$  en el método de *splitting con paso adaptativo*. Se puede apreciar la evolución totalmente coincidente del tamaño del paso y el error relativo producido en el intervalo de tiempo  $[0, 1]$ . El perfil de ambas gráficas es muy similar y refleja la incidencia directa de la elección del paso y la precisión obtenida en ese instante. Además se puede apreciar cómo el tamaño de paso va oscilando, contrayéndose y elongándose, a medida que la precisión obtenida en cada paso cumpla o no los criterios establecidos para el algoritmo. El comportamiento del paso adaptativo sigue relación con el carácter oscilante de  $Y(t)$  en el tiempo.

Para dar una referencia del comportamiento del método adaptativo frente al de paso fijo, la tabla n°1 registra el tiempo de CPU que requieren ambas versiones para cotas de error relativo dadas, al ser comparadas con la solución de referencia dada por el esquema de Magnus de orden 6 con  $h = 10^{-6}$ . El tiempo utilizado por el algoritmo *h-adaptativo* del método *splitting* aumenta casi en forma aritmética (alrededor de 1 seg/CPU). En contraste el método de paso fijo [Cas07] lo hace casi geoméricamente (aumentando 10 veces).

Cota ER	Tiempo CPU	
	Paso fijo	Paso variable
0.015	0.8268	0.4212
$10^{-3}$	4.0092	1.9188
$10^{-4}$	38.4364	3.0567
$10^{-5}$	378.4428	4.0248

**Tabla n° 1**

En estas experiencias se utilizaron los valores de los parámetros dados en la tabla n°2.

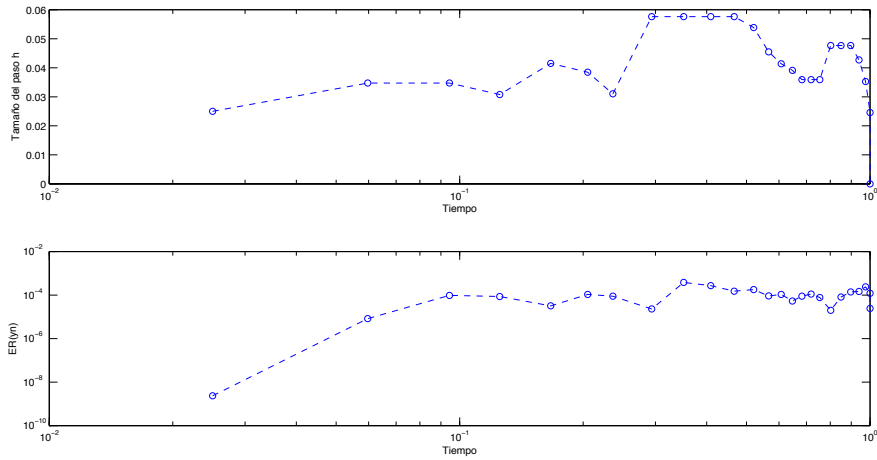


Figura 2.9: Evolución del tamaño del paso  $h$ , (gráfico superior) y el error relativo (gráfico inferior) en el intervalo de tiempo  $t \in [0, 1]$ , al resolver un sistema de ecuaciones diferenciales matricial (2.1) con  $A_{ij} = \sin[t(i^2 - j^2)]$ . El tamaño de paso inicial es de  $h = 10^{-2}$ . En el gráfico superior se aprecian los ciclos que sufre el tamaño del paso, situación que obedece a la naturaleza cíclica de la solución de este sistema diferencial. Como era de esperar el comportamiento del error relativo sigue una relación a corde con el tamaño del paso, obviamente teniendo en cuenta la holgura dispuesta como parámetro de entrada del procedimiento, es decir, el error se ve incrementado con la elección de pasos grandes y se ve reducido cuando se elige pasos más reducidos. Se tomaron los valores de la Tabla n°2 para los parámetros del algoritmos adaptativo.

Parámetro	Valor usado
$f_{max}$	2.5
$f_{min}$	0.001
$f$	0.8
$ER$	0.00001
$EA$	0.001

**Tabla n° 2**

A manera de síntesis, sobre el método de splitting con  $h$ -adaptativo se puede señalar que en las experiencias numéricas realizadas se encontró:

- El error relativo es comparable, en general, con el obtenido con el de paso fijo.

- Se disminuye ostensiblemente la cantidad de iteraciones necesarias para conseguir una precisión similar a la del método común.
- El punto anterior implica una disminución en los tiempo de CPU.
- Tanto la precisión lograda como el tiempo de CPU que requiere el método adaptativo se verá limitada por los valores de los parámetros del algoritmo, de esta manera superado los valores de la tolerancia,  $\delta$ , el método estanca sus resultados y el tiempo requerido se vuelve constante, pues no invierte en ese contexto más cálculos.

Hay que señalar por último que, aunque se aplique un tamaño de paso diferente a lo largo de la integración, el resultado seguirá estando en el grupo  $SL(n)$  por construcción.

## 2.6. Aplicación al cálculo de la exponencial de una matriz con traza nula

En esta sección aplicaremos el algoritmo desarrollado anteriormente para construir un algoritmo eficiente que permita calcular, dada una matriz  $A \in \mathfrak{sl}(n)$ , su exponencial (o mejor dicho una aproximación a la misma) que por construcción estará en  $SL(n)$ .

Para ilustrar el procedimiento descrito se calculará la exponencial de matrices cuyo resultado analítico se conoce en la mayoría de los casos. También los resultados obtenidos con este algoritmo se comparará con los resultados que entrega el software **Matlab** con la función **expm**(A), así como con los resultados entregados por otras técnicas, tales como la aproximación de Padé ([ML03]) (*splitting-Padé-squaring*) dada por la expresión:

$$e^A \simeq \left( \frac{I + B^2 \frac{3}{28} + B^4 \frac{1}{1680} + B \frac{1}{2} + B^3 \frac{1}{84}}{I + B^2 \frac{3}{28} + B^4 \frac{1}{1680} - B \frac{1}{2} - B^3 \frac{1}{84}} \right)^4, \quad (2.82)$$

con  $B = \frac{A}{4}$ , y la expansión truncada de la serie de Taylor.

### 2.6.1. Descripción del proceso

Al ser la matriz  $A$  independiente de la variable  $t$ , el procedimiento que construye el integrador numérico descrito anteriormente se transforma en los siguientes pasos algorítmicos:

1. Aproximamos  $e^A$  por la factorización:

$$Y(t) \equiv Y_0(t) = L_0 U_0 L_1 U_1 \dots L_k U_k Y_{k+1}(t), \quad k = 0, 1, 2, \dots \quad (2.83)$$

con

$$A_k = A_{k+} + A_{k-}.$$

Aquí  $A_{k+}$  y  $A_{k-}$  denotan las matrices triangulares superior e inferior de  $A$ , respectivamente.

$$L' = A_{k-} L_k, \quad L_k(0) = I$$

$$C_k \equiv L_k^{-1} A_{k+} L_k = C_{k+} + C_{k-}$$

$$U' = C_{k+} U_k, \quad U_k(0) = I$$

$$A_{k+1} \equiv U_k^{-1} C_{k-} U_k, \quad A'_{k+1} = A_{k+1} Y_{k+1}, \quad k = 0, 1, 2, \dots$$

Usualmente la factorización (2.83) es truncada en el paso  $k$ -ésimo, cuando  $Y_{k+1} = I$ .

2. Por simplicidad tomaremos el caso particular de  $k = 0$ , y denotaremos por  $a_{ij}$  los elementos de  $A_0$ ,  $i, j = 1, \dots, n$ . En forma análoga para  $L_{ij}(t)$ ,  $i \leq j$ . Así, definimos

$$A_{ij}(t) = \int_0^t a_{ij}(s) ds.$$

Como  $A = A(t)$ , tenemos

$$A_{ij}(h) = \int_0^h a_{ij}(s) ds = a_{ij} h,$$

$$A_{ij}(h/2) = \int_0^{h/2} a_{ij}(s) ds = a_{ij} \left( \frac{h}{2} \right).$$

También denotamos

$$C_0(t) = (c_{ij}), \quad U_0(t) = U_{ij}, \quad C_{ij}(h) = \int_0^h c_{ij}(s) ds.$$

3. La matriz triangular  $L_0(t)$  se obtiene por la integración de la ecuación  $L'_0 = A_{0-}L_0$ . Para el cálculo de  $L_k$  primero se calcula su diagonal, para los argumentos  $h$  y  $h/2$

$$L_{ii}(h) = e^{a_{ii}h}, \quad L_{ii}(h/2) = e^{a_{ii}h/2}.$$

Las expresiones para  $L_{ij}(h)$  y  $L_{ij}(h/2)$ ,  $j < i$ , son:

$$L_{ij}(h) = e^{a_{ii}h} \int_0^h F_{ij}(t) dt \quad \text{con} \quad F_{ij}(t) \equiv e^{-a_{ii}t} \sum_{k=j}^{i-1} a_{ik} L_{kj}(t).$$

Se aproximan los  $L_{ij}(h)$  y  $L_{ij}(h/2)$ ,  $j < i$ , con la regla de Simpson, dando:

$$L_{ij}(h) = e^{a_{ii}h} \frac{h}{6} (a_{ij} + 4F_{ij}(h/2) + F_{ij}(h)) + \mathcal{O}(h^5)$$

$$L_{ij}(h/2) = e^{a_{ii}(h/2)} \frac{h}{24} (5a_{ij} + 8F_{ij}(h/2) - F_{ij}(h)) + \mathcal{O}(h^4)$$

4. La matriz  $C_0(0)$  se puede aproximar mediante

$$C_0(0) = A_{0+}(0),$$

$$C_0(h/2) = L_0^{-1}(h/2) A_{0+}(h/2) L_0(h/2),$$

$$C_0(h) = L_0^{-1}(h) A_{0+}(h) L_0(h).$$

En forma similar al paso 3, construimos las matrices  $C_{ij}(h)$  y  $C_{ij}(h/2)$ , con orden 4 y 3 respectivamente:

$$\tilde{C}_{ii}(h) = \frac{h}{6} (c_{ii}(0) + 4c_{ii}(h/2) + c_{ii}(h)),$$

$$\tilde{C}_{ii}(h/2) = \frac{h}{24} (5c_{ii}(0) + 8c_{ii}(h/2) - c_{ii}(h)).$$

5. Los cálculos de la diagonal de la matriz  $U$  son

$$U_{ii}(h) \approx e^{\tilde{C}_{ii}(h)}, \quad U_{ii}(h/2) \approx e^{\tilde{C}_{ii}(h/2)}.$$

Los demás términos de la matriz se aproximan por

$$U_{ij}(h) \approx e^{\tilde{C}_{ii}(h)} \int_0^h G_{ij}(t) dt \quad \text{con} \quad G_{ij}(t) \equiv e^{-\tilde{C}_{ii}(t)} \sum_{k=i+1}^j c_{ik} U_{kj}(t),$$

luego

$$U_{ij}(h) \approx e^{\tilde{C}_{ii}(h)} \frac{h}{6} (c_{ij}(0) + 4G_{ij}(h/2) + G_{ij}(h)),$$

$$U_{ij}(h/2) \approx e^{\tilde{C}_{ii/2}(h)} \frac{h}{24} (5c_{ij}(0) + 8G_{ij}(h/2) - G_{ij}(h)).$$

6. Para la segunda iteración se calcula la matriz  $A_1$  por:

$$A_1(0) = C_{0-}(0),$$

$$A_1(h/2) = U_0^{-1}(h/2)C_{0-}(h/2)U_0(h/2),$$

$$A_1(h) = U_0^{-1}(h)C_{0-}(h)U_0(h).$$

Puesto que el proceso es iterativo se repiten los pasos de (1) a (5), usando la matriz  $A_1$ .

Finalmente, el producto  $Y_{n+1} = L_0 U_0 Y_n$  es la aproximación de  $Y(t_{n+1} = t_n + h)$ , donde  $h$  es el tamaño de paso del método.

Para concluir esta sección, y a manera de síntesis, se presenta una versión algorítmica del método discutido (una implementación en **Matlab** se encuentra en el Apéndice C). En la práctica el ciclo se hace dos veces ( $k = 1$ ) para un cierto valor de  $h$ .

**Algoritmo** expmS(A) \_\_\_\_\_

- **Input:** La matriz  $A \in \mathbb{R}^{n \times n}$ , tal que  $\text{tr}(A) = 0$   
**Output:** La matriz  $Y \approx e^A$
- Inicializar:  $A_0 = A = (a_{ij})_{n \times n}$ ,  $Y = I$
- Para  $n = 0, 1, 2, \dots, k$ . Hacer

1. Calcular las Matriz  $(Ah2)^n, (Ah)^n$ , etapa  $n$ -ésima  
 $(Ah2)_{ij}^n = \left(\frac{h}{2}\right) a_{ij}$   
 $(Ah)_{ij}^n = ha_{ij}$
  2. Calcular las Matriz  $Lh2_n, Lh_n$   
 $(Lh2)_{ii}^n = \exp((Ah2)_{ij}^n)$   
 $(Lh)_{ii}^n = \exp((Ah)_{ij}^n)$   
 $(Lh2)_{ij}^n = \left(\frac{h}{24}\right) \exp((Ah2)_{ij}^n)(5a_{ij} + 8F_{ij}(h/2) \left(\frac{h}{2}\right) - F_{ij}(h))$   
 $(Lh)_{ij}^n = \left(\frac{h}{6}\right) \exp((Ah)_{ij}^n)(a_{ij} + 4F_{ij} \left(\frac{h}{2}\right) + F_{ij}(h))$   
con  $F_{ij}(t) = \sum_{k=j}^{i-1} a_{ik} (L_{kj})^n(t)$
  3. Calcular las matrices  $C_n$  y  $\tilde{C}$   
 $(C0)^n = (A0)_{\Delta}^n$ ,  
donde  $\Delta$  corresponde a la triangular superior de la matriz.  
 $(Ch2)^n = ((Lh2)^n)^{-1}((Ah2)^n)_{\Delta}((Lh2)^n)$   
 $(Ch)^n = ((Lh)^n)^{-1}((Ah)^n)_{\Delta}((Lh)^n)$   
 $(\tilde{C}h2)_{ij}^n = \left(\frac{h}{24}\right) (5(C0)_{ij}^n + 8(Ch2)_{ij}^n - (Ch)_{ij}^n)$   
 $(\tilde{C}h)_{ij}^n = \left(\frac{h}{6}\right) ((C0)_{ij}^n + 4(Ch2)_{ij}^n + (Ch)_{ij}^n)$
  4. Calcular la matriz  $U$   
 $(Uh2)_{ii}^n = \exp((\tilde{C}h2)_{ij}^n)$   
 $(Uh)_{ii}^n = \exp((\tilde{C}h)_{ij}^n)$   
 $(Uh2)_{ij}^n = (Uh2)_{ii}^n \left(\frac{h}{24}\right) (5(C0)_{ij}^n + 8G_{ij} \left(\frac{h}{2}\right) - G_{ij}(h))$   
 $(Uh)_{ij}^n = (Uh)_{ii}^n \left(\frac{h}{6}\right) ((C0)_{ij}^n + 4G_{ij} \left(\frac{h}{2}\right) + G_{ij}(h))$   
con  $G_{ij}(t) = \sum_{k=j}^{i-1} (C_{ij})^n(t)$
  5. Preparación de  $A$  para la próxima iteración:  
 $(A0)^{n+1} = ((C0)^n)_{\nabla}$ ,  
donde  $\nabla$  representa la triangular inferior de la matriz.  
 $(Ah2)^{n+1} = ((Uh2)^n)^{-1}((Ch2)^n)_{\nabla}(Uh2)^n$   
 $(Ah)^{n+1} = ((Uh)^n)^{-1}((Ch)^n)_{\nabla}(Uh)^n$ .
- Factorización de la solución aproximada:  
 $Y = (Lh)^0(Uh)^0(Lh)^1(Uh)^1 \dots (Lh)^k(Uh)^k Y$
- 

## 2.6.2. Experimentos numéricos

A continuación se muestran diferentes ejemplos numéricos que ponen de manifiesto las principales características de este algoritmo.



**Ejemplo 1.** Nuestro primer ejemplo corresponde al cálculo de la exponencial de la matriz  $2 \times 2$

$$A = \begin{bmatrix} -49 & 24 \\ 64 & 31 \end{bmatrix}.$$

Es fácil calcular la matriz aproximada  $e^A$  por el método de Padé dado por la fórmula (2.82), o por el método de la serie truncada de la expansión de Taylor [ML03], [GL96], dado por

$$e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!}.$$

Para este ejemplo la figura 2.10 muestra el comportamiento del error relativo del método de Taylor, Padé y el método de splitting, con distinto número de términos en el caso de la serie de Taylor. Se aprecia claramente que el algoritmo de Taylor reduce el error relativo lentamente, de hecho requiere al menos 60 términos para alcanzar una precisión un poco mejor que la de (2.82). Sin embargo, el método basado en splitting presenta un error relativo menor que los demás procedimientos. Utilizamos el resultado proporcionado por el comando `expm()` de **Matlab** como solución de referencia.

Esta matriz es un caso particular de

$$A = \begin{bmatrix} \lambda & \alpha \\ 0 & \mu \end{bmatrix},$$

cuya exponencial está dada por

$$e^{tA} = \begin{bmatrix} e^{\lambda t} & \alpha \frac{e^{\lambda t} - e^{\mu t}}{\lambda - \mu} \\ 0 & e^{\mu t} \end{bmatrix}.$$

**Ejemplo 2.** Aplicamos el método a la matriz  $A$  cuyos elementos son

$$A_{ij} = \sin[i^2 - j^2], \quad 1 \leq i \leq j \leq n$$

con  $A_{ij} = A_{ji}$  y  $n = 20$  y  $100$  respectivamente. Se observa que  $\text{tr}(A) = 0$ .

Los resultados de aplicar el método usando esta matriz se muestran en las figura 2.11. Se utiliza en ellos una escala logarítmica para el error relativo.

Los resultados se compararon con los dados por la instrucción `expm()` de **Matlab**, y para el error se utiliza la norma matricial de Frobenius. Más concretamente

$$ER = \frac{\|e_{ap}^A - e_{ref}^A\|}{\|e_{ref}^A\|}, \quad (2.84)$$

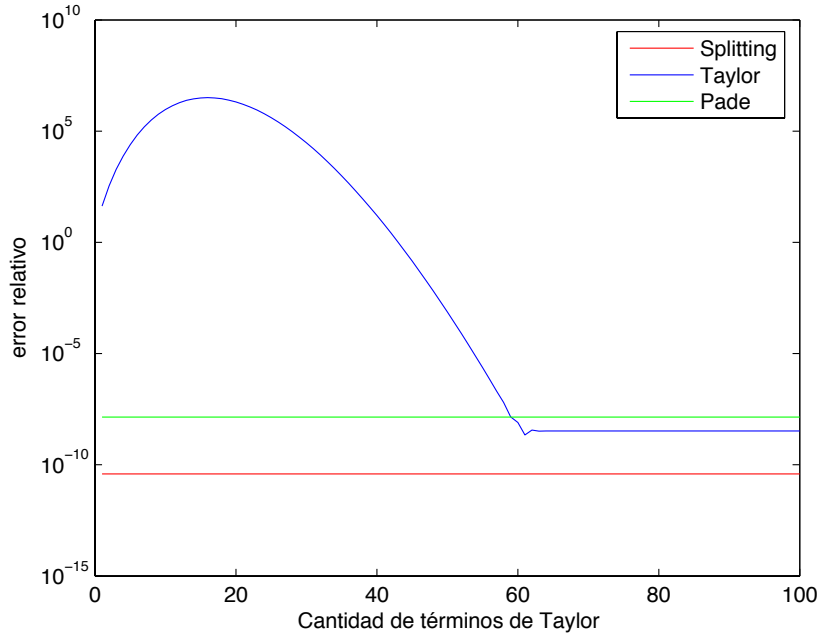


Figura 2.10: Comportamiento de la expansión de Taylor para aproximar  $e^A$  con la matriz del primer ejemplo. Se observa que el error relativo a partir de expansiones con más de 60 términos estanca su valor de error relativo. Sin embargo, el método basado en *splitting* tiene un error relativo menor a los demás métodos, aun cuando para este caso  $\text{tr}(A) \neq 0$ .

donde  $e_{ap}^A$  y  $e_{ref}^A$ , son los valores dados por el método basado en *splitting* y la rutina de **Matlab**, respectivamente.

Se desprende de estos resultados que el método *splitting* mantiene la misma tendencia de error relativo, no afectándose por la dimension de la matriz  $A$ , sin embargo, disminuye ligeramente su precisión comparada con la función `expm()`. Nótese el orden 4 de la aproximación.

El teorema 8 establece el comportamiento de  $Y^{[k]}(t)$  cuando  $A(t) \in \mathfrak{sl}(n)$ : las aproximaciones de  $e^A$  están en el grupo  $SL(n)$ . Esto queda de manifiesto en la gráfica de la figura 2.12, donde se representa el valor  $\det(\text{expmS}(A))$  de la matriz usada en la figura 2.11 con  $n = 20$ . Se puede apreciar que  $\det(\text{expmS}(A)) - 1 \rightarrow 0$ , cuando  $h \rightarrow 0$  (figura izquierda), lo cual es una muestra empírica de la validez del teorema.

**Ejemplo 3.** Esta experiencia replica la estudiada en [ML03], con una matriz de tamaño  $n = 25$ . Esta matriz  $A$  se forma de la siguiente manera

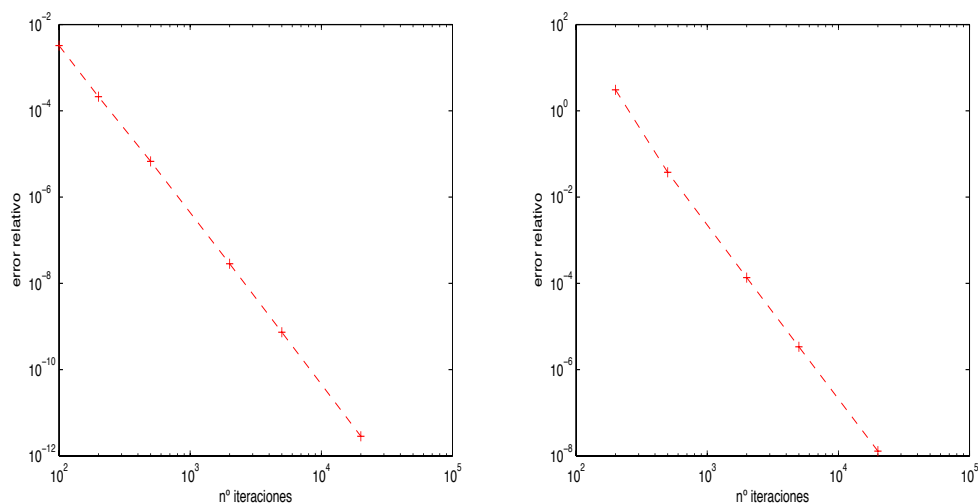


Figura 2.11: Gráfica del error relativo para los casos de una matriz simétrica  $n \times n$ , con elementos  $A_{ij} = \sin[i^2 - j^2] = A_{ji}$  para  $n = 20$  (gráfico izquierdo) y  $n = 100$  (gráfico derecho). Se aprecia en ambas situaciones que la pendiente es igual y sólo varía el nivel de precisión, el cual disminuye cuando la dimensión de la matriz aumenta. El eje de abscisas corresponde esencialmente al valor  $1/h$ .

según la sintaxis de **Matlab**:

```
n=25;
C=diag(ones(n-1,1),1); C(n,1)=1;
D=diag(exp(2*pi*i*(0:n-1)/n));
I=eye(n); A=C+D-1.1*I;
```

Este ejemplo ilustra el fenómeno de ‘joroba’ de la gráfica de  $\|e^{tA}\|$  con respecto a  $t$ ,  $t \in [0, 50]$ . Véase la figura 2.13. Los resultados son muy similares a los obtenidos en [ML03], al aplicar la función `expm`.

**Ejemplo 4** Aplicaremos el algoritmo basado en splitting sobre una matriz de Hilbert, definida por

$$(H_{ij})_{n \times n} = \int_0^1 x^{i+j-2} dx = \frac{1}{i+j-1}, \quad \text{con } n = 5.$$

Las matrices de Hilbert son ejemplos clásicos de matrices mal condicionadas, haciéndolas muy difíciles de utilizar en el cálculo numérico [Sal87]. Por

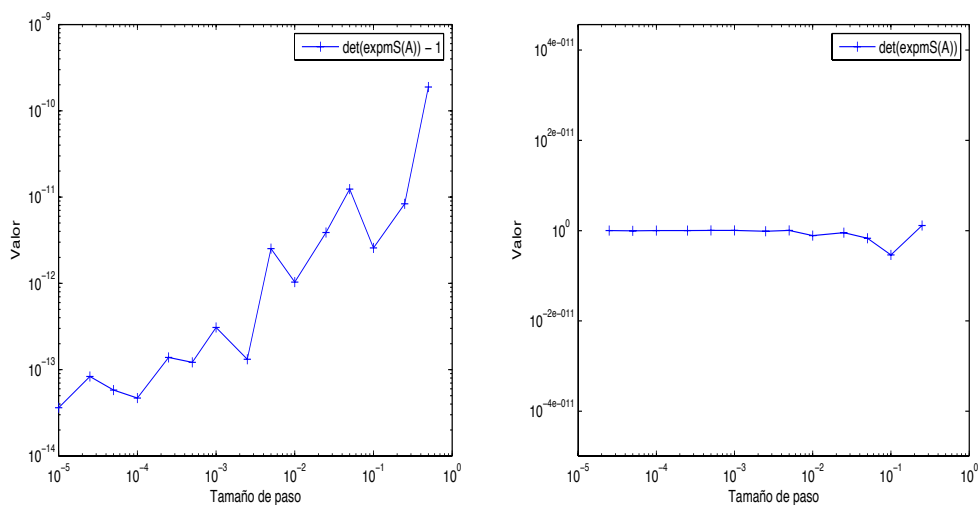


Figura 2.12: Observamos que  $\det(\text{expmS}(A)) - 1$  prácticamente toma un valor cero, para el rango de valores de  $h$ ,  $h \in [10^{-5}, 10^{-1}]$ , figura izquierda. Cabe notar que para todos los valores de  $h$  el error se mantiene acotado por debajo de  $10^{-9}$ . O bien, en otras palabras  $\det(\text{expmS}(A)) \rightarrow 1$ , cuando  $h \rightarrow 0$ , figura derecha.

ejemplo, el número de condición<sup>2</sup> de la matriz anterior es alrededor de  $\kappa(H) = 4,8 \cdot 10^5$ , cuando se utiliza norma 2. Al aplicar el algoritmo de *splitting* a esta matriz de Hilbert y comparar sus resultados con los dados por  $\text{expm}(H)$ , se obtuvo la gráfica del *error relativo* de la figura 2.14. Se aprecia que el error relativo del método de *splitting* de cuarto orden va disminuyendo considerablemente a medida que aumenta el número de iteraciones, es decir, a medida que se toman tamaños de paso  $h$  más pequeños. Llega un momento en que el error relativo se mantiene, esto se explica por el error de redondeo que produce **Matlab**, y esto no depende ostensiblemente de la dimensión de la matriz de Hilbert.

Si bien las experiencias numéricas discutidas anteriormente no tienen un carácter exhaustivo de prueba para el algoritmo basado en *splitting*, en su versión para la exponencial de una matriz  $A$  constante, sí ponen de manifiesto que este nuevo algoritmo es tanto o más eficiente, en precisión, que otros métodos de exponenciación matricial existentes o estándares en software numérico para los casos analizados.

Además, el algoritmo desarrollado podría ser usado en conjunción con

<sup>2</sup>Las matrices mal condicionadas son aquellas en que pequeñas perturbaciones en los coeficientes de un sistema de ecuaciones lineales ( $Ax = b$ ) provocan grandes perturbaciones en la solución del sistema [War77]. Se le define por  $\kappa(A) = \|A\| \cdot \|A^{-1}\|$

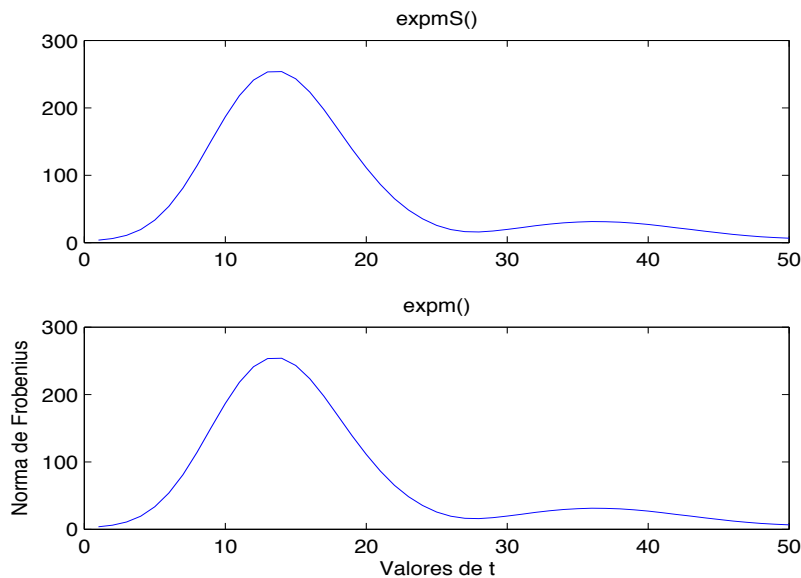


Figura 2.13: Comportamiento, virtualmente idéntico, de  $\|e^{tA}\|$  para los algoritmos de `expmS()` (arriba) basado en `splitting` y `expm()`, con  $n = 25$ . El método de *splitting* implementado en el código `expmS()` se comporta igual al código `expm()` de **Matlab**.

otros esquemas numéricos de integración de forma consistente, ya que en principio no es necesario ir a error de redondeo para que el resultado final esté en el grupo  $SL(n)$ . Por ejemplo, podríamos usar un método de Magnus de orden 4 y calcular la correspondiente exponencial matricial con ayuda de este algoritmo (también de orden 4), produciendo resultados en  $SL(n)$  de manera más rápida que usando otros procedimientos, y asegurándonos a la vez que el esquema global es de orden 4.

Este algoritmo se podría de hecho generalizar para el cálculo de la exponencial de una matriz cuadrada  $A$   $n \times n$  con traza no necesariamente nula de la siguiente forma. Supongamos que  $\text{tr}(A) = \tau \neq 0$ . Entonces formamos la nueva matriz  $B = A - \frac{\tau}{n}I$ , la cual tiene ahora traza nula. Por consiguiente

$$e^A = e^{\frac{\tau}{n}I} e^{A - \frac{\tau}{n}I} = e^{\frac{\tau}{n}I} e^B$$

y ahora el algoritmo desarrollado en este capítulo se puede aplicar al cálculo de  $e^B$ .

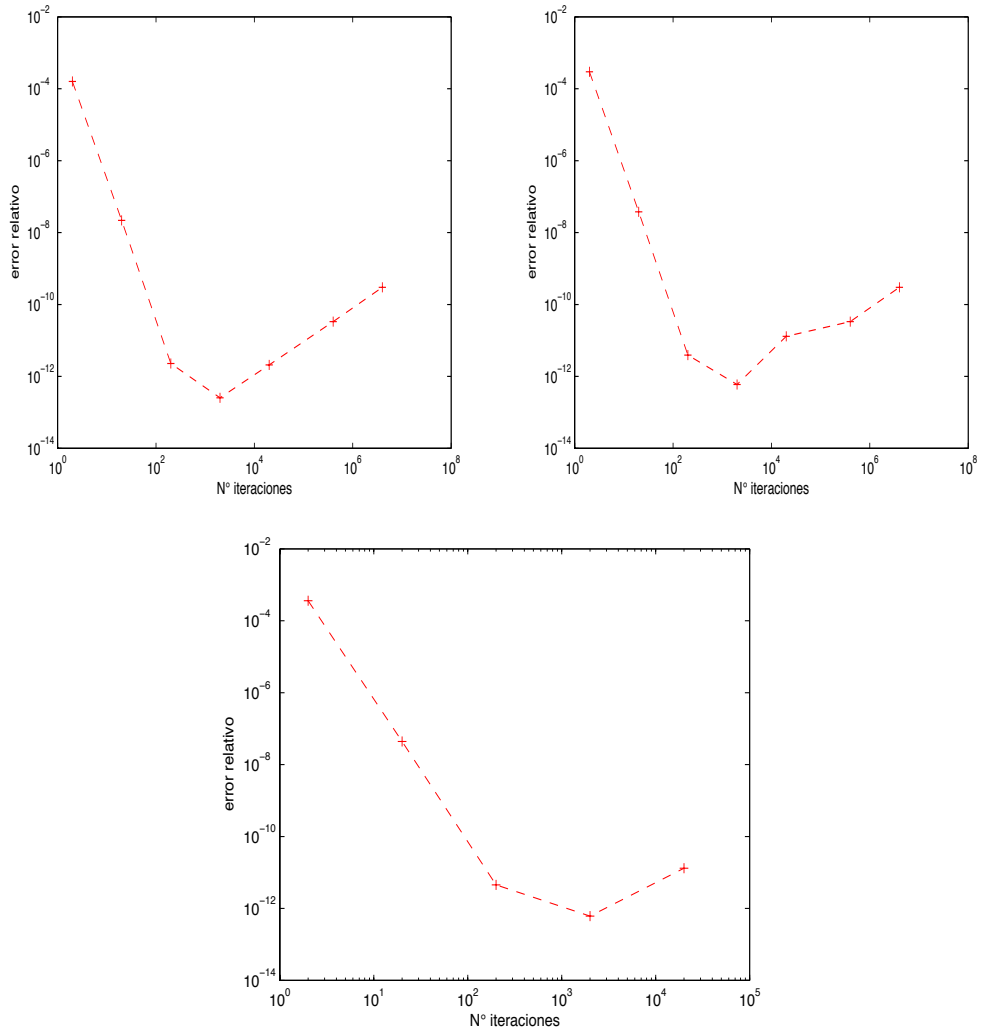


Figura 2.14: Gráfica error relativo de  $e^{H_n}$ , con  $H_n$  una matriz de Hilbert,  $n=10, 25$  y  $50$ , respectivamente, usando un método numérico basado en *splitting* de cuarto orden. Este es un ejemplo del comportamiento numérico del algoritmo, cuando se aplica a una matriz mal condicionada.

## Capítulo 3

# Tratamiento de problemas no lineales mediante integradores en grupos de Lie

En este capítulo nos centramos en el diseño y análisis de diversos procedimientos numéricos para la resolución de la ecuación diferencial no lineal matricial

$$Y'(t) = A(t, Y)Y, \quad Y(0) = Y_0 \in \mathcal{G}, \quad (3.1)$$

donde  $\mathcal{G}$  es un grupo de Lie matricial,  $A : \mathbb{R}_+ \times \mathcal{G} \rightarrow \mathfrak{g}$  y  $\mathfrak{g}$  denota el álgebra de Lie correspondiente al grupo  $\mathcal{G}$ .

La ecuación (3.1) aparece en muchos ámbitos físicos, como por ejemplo en la dinámica del sólido rígido y en el cálculo de los exponentes de Lyapunov ( $\mathcal{G} \equiv \text{SO}(n)$ ), en sistemas dinámicos hamiltonianos ( $\mathcal{G} \equiv \text{Sp}(n)$ ) y la mecánica cuántica ( $\mathcal{G} \equiv \text{SU}(n)$ ). Además, se puede demostrar que cada ecuación diferencial que evoluciona en un grupo de Lie matricial se puede escribir en la forma (3.1). Por otra parte, el análisis de las ecuaciones diferenciales genéricas definidas en espacios homogéneos se puede reducir a la ecuación en un grupo de Lie [HLW06]. Es por ello de gran interés desarrollar técnicas aproximadas de resolución de la ecuación (3.1) que a la vez respeten la estructura algebraica subyacente.

Una técnica ampliamente usada es de hecho una generalización del procedimiento que sirvió para formular el desarrollo de Magnus: se introduce una ecuación diferencial alternativa en el correspondiente álgebra de Lie  $\mathfrak{g}$ , se resuelve dicha ecuación y por último se transforma la solución de nuevo a  $\mathcal{G}$  por medio de la función exponencial. Una posibilidad a este respecto consiste en aproximar la solución de la ecuación diferencial en el álgebra de Lie  $\mathfrak{g}$  por

medio de un método clásico de Runge-Kutta. Esta es la idea que subyace a los llamados métodos de tipo Runge-Kutta-Munthe-Kaas (RKMK)[MK98].

Otra posibilidad es formular directamente una generalización del desarrollo de Magnus llevado a cabo para problemas lineales en el capítulo anterior. Analizamos a continuación estos métodos. Por otra parte, también aquí son aplicables los comentarios realizados entonces cuando el grupo de Lie  $\mathcal{G}$  es el grupo lineal especial  $SL(n)$ : a menos que se calcule de forma exacta la exponencial matricial, ni los métodos RKMK ni los basados en el desarrollo de Magnus no lineal van a preservar la estructura algebraica y por tanto el volumen en el espacio de fases, al contrario que la solución exacta. Por este motivo, estudiamos la generalización del tratamiento realizado por medio de matrices triangulares al caso no lineal definido por (3.1).

### 3.1. Métodos de Runge-Kutta-Munthe-Kaas

Como en el caso lineal, la solución de (3.1) se puede representar por

$$Y(t) = \exp(\Omega(t, Y_0))Y_0, \quad (3.2)$$

donde  $\Omega \in \mathfrak{g}$  satisface la ecuación diferencial

$$\Omega' = d \exp_{\Omega}^{-1} (A(t, e^{\Omega} Y_0)), \quad \Omega(0) = O, \quad (3.3)$$

y el operador  $d \exp_{\Omega}^{-1}$  se ha definido en (2.13). Esta ecuación constituye, de hecho, el punto de partida para diseñar métodos de integración que proporcionan aproximaciones numéricas en  $\mathcal{G}$  para todo  $t$ . Esto se puede hacer mediante la aplicación de un método de Runge-Kutta a (3.3), en lugar de a la ecuación original (3.1).

La serie de la definición del operador  $d \exp^{-1}$ ,

$$d \exp_u^{-1}(v) = \sum_{k=0}^{\infty} \frac{B_k}{k!} \text{ad}_u^k v = v - \frac{1}{2}[u, v] + \frac{1}{12}[u, [u, v]] + \dots$$

puede ser truncada ya que se utiliza sólo en los casos en que el argumento  $u = \mathcal{O}(h)$ . De esta manera una aproximación  $v \approx \Omega(t_n + h)$  es construida, y la solución numérica es obtenida por  $Y_{n+1} = \exp(v)Y_n$ . Más específicamente, con un método Runge-Kutta de orden  $p$  con  $s$  etapas, se puede escribir el correspondiente esquema Munthe-Kaas (RKMK) de la siguiente forma, donde  $h$  es el tamaño de paso y  $a_{ij}, b_i$  son los coeficientes del método Runge-Kutta.



```

for  $i = 1 : s$  do
     $u_i = \sum_{j=1}^s a_{ij} \text{dexpinv}(u_j, k_j, p - 1)$ 
     $k_i = hA(\exp(t_n + c_i h, e^{u_i} Y_n))$ 
end do
 $v = \sum_i b_i \text{dexpinv}(u_i, k_i, p)$ 
 $Y_{n+1} = \exp(v) Y_n$ 

```

Aquí  $\text{dexpinv}(u, v, p)$  denota un orden de aproximación  $p$  a  $d \exp u^{-1}(v)$ , es decir,  $d \exp_u^{-1}(v) - \text{dexpinv}(tu, v, p) = \mathcal{O}(t^{p+1})$  para todos  $u, v \in \mathfrak{g}$ . En otras palabras, incluso si  $d \exp^{-1}$  se aproxima de este modo, el algoritmo resultante es del mismo orden que el esquema Runge-Kutta subyacente, y la solución está en el grupo de Lie donde la ecuación diferencial (3.1) está definida.

A continuación se muestran algunos ejemplos de métodos particulares de esta clase:

- Método de Euler explícito:

$$Y_{n+1} = \exp(hA(t_n, Y_n))Y_n.$$

- Regla implícita del punto medio

$$v = \frac{h}{2}A(t_n + \frac{h}{2}, e^v Y_n), \quad Y_{n+1} = \exp(v)Y_n.$$

- Regla del trapecio (implícita)

$$\begin{aligned}
 u_1 &= hA(t_n, Y_n) \\
 u_2 &= hA(t_n + h, e^{\frac{1}{2}(u_1+u_2)} Y_n) \\
 v &= \frac{1}{2}(u_1 + u_2) \\
 Y_{n+1} &= \exp(v)Y_n.
 \end{aligned}$$

- Método de Heun

$$\begin{aligned}
 u_1 &= hA(t_n, Y_n) \\
 u_2 &= hA(t_n + \frac{1}{2}h, e^{\frac{1}{2}u_1} Y_n) \\
 v &= \frac{1}{2}(u_1 + u_2) \\
 Y_{n+1} &= \exp(v)Y_n.
 \end{aligned}$$

Para los métodos de orden superior, se debe calcular un número significativo de conmutadores a la hora de evaluar  $\text{dexpinv}(u_i, k_i, p)$  directamente a partir de la serie (2.13). En su lugar, resulta ventajoso introducir unas nuevas variables  $Q_i$  por medio de

$$Q_i = \sum_{j=1}^i V_{ij} k_j = \mathcal{O}(h^{q_i}), \quad i = 1, \dots, s, \quad (3.4)$$

donde las constantes  $V_{ij}$  se eligen de tal manera que los números enteros resultantes  $q_i$  son tan grandes como sea posible. Entonces es evidente que

$$[Q_{i_1}, [Q_{i_2}, \dots, [Q_{i_{m-1}}, Q_{i_m}] \dots]] = \mathcal{O}(h^{q_{i_1} + \dots + q_{i_m}}),$$

lo que hace que sea fácil desechar términos de orden más alto que el método en sí. La aplicación de esta técnica al método RK clásico de cuarto orden (ver Apéndice B) da como resultado el siguiente algoritmo optimizado:

$$\begin{aligned} u_1 &= 0 \\ k_1 &= hA(t_n, Y_n) & Q_1 &= k_1 = \mathcal{O}(h) \\ u_2 &= \frac{1}{2}Q_1 \\ k_2 &= hA(t_n + \frac{1}{2}h, e^{u_2}Y_n) & Q_2 &= k_2 - k_1 = \mathcal{O}(h^2) \\ u_3 &= \frac{1}{2}Q_1 + \frac{1}{2}Q_2 - \frac{1}{8}[Q_1, Q_2] \\ k_3 &= hA(t_n + \frac{1}{2}h, e^{u_3}Y_n) & Q_3 &= k_3 - k_2 = \mathcal{O}(h^3) \\ u_4 &= Q_1 + Q_2 + Q_3 \\ k_4 &= hA(t_n + h, e^{u_4}Y_n) & Q_4 &= k_4 - 2k_2 + k_1 = \mathcal{O}(h^3) \\ v &= Q_1 + Q_2 + \frac{1}{3}Q_3 + \frac{1}{6}Q_4 - \frac{1}{6}[Q_1, Q_2] - \frac{1}{12}[Q_1, Q_4] \\ Y_{n+1} &= \exp(v)Y_n. \end{aligned}$$

El método requiere 4 evaluaciones de la exponencial de  $A$  y sólo 2 conmutadores. Esta técnica se puede combinar con otras estrategias para obtener métodos RKMK con una reducción significativa en el número de conmutadores.

## 3.2. Desarrollo de Magnus para ecuaciones no lineales

Como antes, el punto de partida es la representación de la solución de (3.1) en la forma

$$Y(t) = e^{\Omega(t)}Y_0, \quad (3.5)$$

donde  $\Omega$  satisface la ecuación

$$\Omega' = d \exp_{\Omega}^{-1}(A(t, e^{\Omega}Y_0)), \quad \Omega(0) = O, \quad (3.6)$$

con

$$d \exp_{\Omega}^{-1}(c) = \sum_{k=0}^{\infty} \frac{B_k}{k!} \text{ad}_{\Omega}^k C.$$

En el caso lineal, es decir, cuando  $A$  depende sólo del tiempo  $t$ , la serie de Magnus para  $\Omega$  se puede obtener por la iteración de Picard,

$$\Omega^{[0]}(t) = O$$

$$\Omega^{[m+1]}(t) = \int_0^t d \exp_{\Omega^{[m]}(s)}^{-1} A(s) ds = \int_0^t \sum_{k=0}^{\infty} \frac{B_k}{k!} \text{ad}_{\Omega^{[m]}(s)}^k A(s) ds, \quad m \geq 0.$$

De forma análoga, el mismo procedimiento se puede aplicar para el caso no lineal, obteniéndose

$$\begin{aligned} \Omega^{[m+1]}(t) &= \int_0^t d \exp_{\Omega^{[m]}(s)}^{-1} A(s, e^{\Omega^{[m]}(s)}Y_0) ds \\ &= \int_0^t \sum_{k=0}^{\infty} \frac{B_k}{k!} \text{ad}_{\Omega^{[m]}(s)}^k A(s, e^{\Omega^{[m]}(s)}Y_0) ds, \quad m \geq 0. \end{aligned} \quad (3.7)$$

El siguiente paso para obtener aproximaciones de forma explícita es truncar adecuadamente el operador  $d \exp^{-1}$  en la expansión (3.7). En términos generales, cuando se considera toda la serie de  $d \exp^{-1}$ , el desarrollo en serie de potencias de  $\Omega^{[k]}(t)$ ,  $k \geq 1$ , sólo reproduce la expansión de la solución  $\Omega(t)$  hasta cierto orden, digamos  $\mathcal{O}(t^m)$ . En consecuencia, las series (infinitas) de potencias de  $\Omega^{[k]}(t)$  y  $\Omega^{[k+1]}(t)$  difieren en términos de orden  $\mathcal{O}(t^{m+1})$ . La idea es entonces descartar en  $\Omega^{[k]}(t)$  los términos de orden mayor que  $\mathcal{O}(t^m)$ . Por supuesto, esto requiere un análisis cuidadoso de cada término en la expansión.

Por ejemplo,  $\Omega^{[0]} = O$  lo que implica que  $(\Omega^{[1]})' = A(t, Y_0)$  y de esta manera

$$\Omega^{[1]}(t) = \int_0^t A(s, Y_0) ds = \Omega(t) + \mathcal{O}(t^2),$$

con

$$A(s, e^{\Omega^{[1]}(s)}Y_0) = A(0, Y_0) + \mathcal{O}(s),$$

lo que implica que

$$-\frac{1}{2} \int_0^t [\Omega^{[1]}(s), A(s, e^{\Omega^{[1]}(s)}Y_0)] ds = \mathcal{O}(t^3).$$

Así,

$$\Omega^{[2]}(t) = \int_0^t A(s, e^{\Omega^{[1]}(s)}Y_0) ds.$$

En general

$$\begin{aligned} \Omega^{[1]}(t) &= \int_0^t A(s, Y_0) ds \\ \Omega^{[m]}(t) &= \sum_{k=0}^{m-2} \frac{B_k}{k!} \int_0^t \text{ad}_{\Omega^{[m-1]}(s)} A(s, e^{\Omega^{[m-1]}(s)}Y_0) ds. \quad m \geq 2 \end{aligned} \quad (3.8)$$

El orden de la aproximación se concluye a partir del siguiente resultado:

**Teorema 9** *Sea  $\Omega(t)$  la solución exacta del problema de valor inicial (3.1) y  $\Omega^{[m]}(t)$  la aproximación dada por el esquema (3.8). Entonces se cumple que*

$$\Omega(t) - \Omega^{[m]}(t) = \mathcal{O}(t^{m+1}).$$

Es decir, usando  $\Omega^{[m]}(t)$  en la ecuación (3.5), se logra tener una aproximación explícita para la solución de (3.1) que es correcta hasta el orden  $\mathcal{O}(t^{m+1})$ .

*Demostración.* Para simplificar las cosas, consideraremos el caso autónomo, es decir,  $Y' = A(Y)Y$ . La extensión para el caso general es sencilla. En este caso la solución exacta de (3.6) se puede escribir como la serie infinita

$$\Omega(t) = \sum_{l=1}^{\infty} t^l w_l$$

con  $w_1 = A(Y_0)$ ,  $w_2 = \frac{1}{2}G_1$  y, para  $l \geq 3$ ,

$$\begin{aligned}
lw_l &= G_{l-1} + \sum_{j=1}^{l-1} \frac{B_j}{j!} \sum_{\substack{k_1+\dots+k_j=l-1 \\ k_1 \geq 1, \dots, k_j \geq 1}} \text{ad}_{w_{k_1}} \dots \text{ad}_{w_{k_j}} A(Y_0) \\
&+ \sum_{j=1}^{l-2} \left( \sum_{s=0}^j \frac{B_s}{s!} \sum_{\substack{k_1+\dots+k_j=l-1 \\ k_1 \geq 1, \dots, k_j \geq 1}} \text{ad}_{w_{k_1}} \dots \text{ad}_{w_{k_s}} \right) G_{l-1-j} \\
&+ \sum_{j=1}^{l-2} \frac{B_j}{j!} \sum_{\substack{k_1+\dots+k_j=l-1 \\ k_1 \geq 1, \dots, k_j \geq 1}} \text{ad}_{w_{k_1}} \dots \text{ad}_{w_{k_j}} G_1,
\end{aligned}$$

siendo  $G_k$  una función dependiente de  $Y_0, w_1, \dots, w_k$ , es decir,

$$G_k \equiv G_k(Y_0; w_1, w_2, \dots, w_k), \quad k \geq 1.$$

Por otra parte, si descartamos todos los términos de orden superior a  $\mathcal{O}(t^m)$  en  $\Omega^{[m]}(t)$  dado por (3.8), entonces

$$\Omega(t) = \sum_{l=1}^{\infty} t^l \widehat{w}_l,$$

donde  $\widehat{w}_1 = A(Y_0)$  y  $\widehat{w}_l$ ,  $2 \leq l \leq m$ , es dado por misma expresión desarrollada para  $lw_l$  con las sustituciones

$$w_k \longrightarrow \widehat{w}_k, \quad G_k \longrightarrow \widehat{G}_k,$$

para la nueva  $\widehat{G}_k \equiv \widehat{G}_k(Y_0; \widehat{w}_1, \widehat{w}_2, \dots, \widehat{w}_{m-1})$ ,  $k = 1, \dots, m$ . Para  $\widehat{w}_1 = w_1$ , tenemos  $\widehat{G}_1 = G_1$  y por inducción

$$\widehat{w}_l = w_l, \widehat{G}_l = G_l, \text{ para } l = 1, \dots, m-1,$$

pero

$$\widehat{G}_k \equiv \widehat{G}_k(Y_0; \widehat{w}_1, \widehat{w}_2, \dots, \widehat{w}_{m-1}),$$

mientras que  $G_m \equiv G_m(Y_0; w_1, w_2, \dots, w_m)$ , de modo que  $\widehat{G}_m \neq G_m$ . En consecuencia

$$\Omega'(t) - (\Omega^{[m]}(t))' = t^m (G_m - \widehat{G}_m) + \mathcal{O}(t^{m+1})$$

y así se tiene que  $\Omega(t) - \Omega^{[m]}(t) = \mathcal{O}(t^{m+1})$ .  $\square$

Para construir un integrador numérico práctico basado en la expansión de Magnus no lineal (3.8), las integrales que aparecen en la formulación tienen que ser convenientemente evaluadas. Para la mayoría de los problemas, sin embargo, sólo la aproximación de primer orden  $\Omega^{[1]}(t)$  puede ser calculada exactamente, y así las integrales restantes necesarias de órdenes superiores debe ser reemplazadas por cuadraturas apropiadas, dependiendo del problema particular. También la existencia de varios conmutadores y exponenciales matriciales en las etapas intermedias requiere un tratamiento detallado para reducir la complejidad computacional y hacer de los esquemas de integración procedimientos prácticos.

Para ilustrar los diferentes temas involucrados en la construcción de métodos numéricos para (3.8), consideramos aquí los esquemas de órdenes 2, 3 y 4. En todos los casos se elige una cuadratura con puntos equidistantes a lo largo del intervalo  $[t_n, t_n + h]$ .

**Orden 2.** Este caso corresponde a  $m = 2$  en (3.8), luego

$$\Omega^{[1]}(t) = \int_0^t A(s, Y_0) ds, \quad (3.9)$$

$$\Omega^{[2]}(t) = \int_0^t A(s, e^{\Omega^{[1]}(s)} Y_0) ds. \quad (3.10)$$

Si  $A$  es tal que la integral (3.9) se puede calcular con exactitud, todo lo que se requiere para obtener un integrador de segundo orden es reemplazar la integral (3.10) por una regla de cuadratura de orden 2. Por ejemplo, si discretizamos  $\Omega^{[2]}$  con la regla del trapecio, tendremos

$$\Omega^{[2]}(h) = \frac{h}{2} \left( A(0, Y_0) + A(h, e^{\Omega^{[1]}(h)} Y_0) \right) + \mathcal{O}(h^3). \quad (3.11)$$

De hecho, se puede construir un esquema de segundo orden que no evalúa exactamente la integral (3.9), sino sólo una aproximación de primer orden de ella. Si, por ejemplo, se utiliza el método de Euler,  $\Omega^{[1]}(h) = hA(0, y_0) + \mathcal{O}(h^2)$ , se tendrá un nuevo esquema explícito de segundo orden, dado por

$$\begin{aligned} v_1 &\equiv \frac{h}{2} \left( A(0, Y_0) + A(h, e^{hA(0, Y_0)} Y_0) \right) = \Omega^{[2]}(h) + \mathcal{O}(h^3) \\ Y_1 &= e^{v_1} Y_0, \end{aligned} \quad (3.12)$$

que es precisamente el método de dos etapas de Runge-Kutta-Munthe-Kaas (RKMK), con tablero de Butcher (ver Apéndice B)

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

Vale la pena mencionar que el método (3.12) es diferente de (3.11) cuando la matriz  $A$  depende explícitamente del tiempo, aunque ambos son aproximaciones de segundo orden a  $\Omega(t)$ .

Si  $\Omega^{[2]}$  se discretiza con la regla del punto medio se tendrá

$$\begin{aligned} v_2 &\equiv hA\left(\frac{h}{2}, e^{\frac{h}{2}A(0, Y_0)}Y_0\right) = \Omega^{[2]}(h) + \mathcal{O}(h^3) \\ Y_1 &= e^{v_2}Y_0. \end{aligned} \quad (3.13)$$

Vemos que, en el orden 2, los esquemas explícitos RKMK son producidos por la EM al utilizar una aproximación de  $\Omega^{[1]}$  con una cuadratura de primer orden. Surgen nuevos esquemas, sin embargo, cuando  $\Omega^{[1]}$  es evaluado exactamente.

**Orden 3.** En forma análoga a las ecuaciones (3.9) y (3.10), tenemos que trabajar además con

$$\Omega^{[3]}(t) = \int_0^t \left( A_2(s) - \frac{1}{2}[\Omega^{[2]}(s), A_2(s)] \right) ds, \quad (3.14)$$

donde  $A_2(s) \equiv A(s, e^{\Omega^{[2]}(s)}Y_0)$ . Si utilizamos la regla de Simpson para aproximar (3.14), tenemos

$$\begin{aligned} \Omega^{[3]}(h) &= \frac{h}{6} (A(0, Y_0) + 4A_2(h/2) + A_2(h)) \\ &\quad - \frac{h}{3} [\Omega^{[2]}(h/2), A_2(h/2)] - \frac{h}{12} [\Omega^{[2]}(h), A_2(h)] + \mathcal{O}(h^4). \end{aligned}$$

Ahora  $\Omega^{[1]}$  se puede aproximar con Euler y  $\Omega^{[2]}(h)$  con la regla del punto medio, ecuación (3.13), mientras que

$$\Omega^{[2]}(\frac{h}{2}) = \frac{h}{4} \left( A(0, Y_0) + \frac{h}{4} A\left(\frac{h}{2}, e^{\frac{h}{2}A(0, Y_0)}Y_0\right) \right) + \mathcal{O}(h^3)$$

para obtener un esquema de orden 3. El algoritmo se puede formular como

$$\begin{aligned}
u_1 &= 0 \\
k_1 &= hA(0, Y_0) \\
u_2 &= \frac{1}{2}k_1 \\
k_2 &= hA(h/2, e^{u_2}Y_0) \\
u_3 &= \frac{1}{4}(k_1 + k_2) \\
k_3 &= hA(h/2, e^{u_3}Y_0) \\
u_4 &= k_2 \\
k_4 &= hA(h, e^{u_4}Y_0) \\
v_3 &= \frac{1}{6}(k_1 + 4k_3 + k_4) - \frac{1}{3}[u_3, k_3] - \frac{1}{12}[u_4, k_4] \\
Y_1 &= e^{v_3}Y_0
\end{aligned} \tag{3.15}$$

Este método se asemeja mucho al esquema RKMK basado en el tablero de Butcher

$$\begin{array}{c|ccc}
0 & & & \\
\frac{1}{2} & & \frac{1}{2} & \\
1 & -1 & 2 & \\
\hline
& \frac{1}{6} & \frac{2}{3} & \frac{1}{6}
\end{array} \tag{3.16}$$

La elección de un método como los descritos o los de tipo RKMK se puede basar en el coste computacional que involucran [CI06], el cual puede variar según el orden deseado.

**Orden 4.** Con  $m = 4$  podemos usar la regla de Simpson para aproximar  $\Omega^{[4]}(h)$ . Los cálculos que se llevan a cabo para  $\Omega^{[3]}(h)$ , se pueden reutilizar aquí, pero todavía es necesario calcular  $\Omega^{[3]}(h/2)$  hasta orden  $\mathcal{O}(h^3)$  y realizar dos nuevas evaluaciones de  $A$ . El algoritmo resultante se presenta en el apéndice de [CI06] y requiere seis evaluaciones de  $A$ , dos conmutadores y seis exponenciales de matrices por cada paso de tiempo.

### 3.3. Aplicaciones

#### 3.3.1. La ecuación de Emden–Fowler

Vamos a considerar la ecuación diferencial de segundo orden

$$y'' + a(t, y, y') = 0, \quad y(0) = y_0, \quad y'(0) = y'_0, \tag{3.17}$$



la cual se puede representar como un sistema de EDO

$$\mathbf{y}' = A(t, \mathbf{y})\mathbf{y}, \quad \mathbf{y}(0) = \mathbf{y}_0, \quad (3.18)$$

donde  $A = \begin{pmatrix} 0 & 1 \\ -a(t, \mathbf{y}) & 0 \end{pmatrix}$ ,  $\mathbf{y} = (y, y')^T$ .

Estudiaremos inicialmente la ecuación de *Emden–Fowler*, que se obtiene de (3.17) cuando  $a(t, y, y') = ty^2$ . Se debe señalar que esta ecuación es un caso particular de la ecuación  $y'' = kt^n y^m$ , que tiene solución analítica sólo para algunos valores de  $n$  y  $m$  [Zwi97].

En concreto, estudiaremos inicialmente la ecuación diferencial de segundo orden

$$y'' + ty^3 = 0, \quad y(0) = y_0 = 1, \quad y'(0) = y'_0 = 0 \quad (3.19)$$

con  $t \in [0, 60]$ . A este modelo teórico le aplicamos diversos métodos numéricos y contrastamos los resultados.

Comenzamos el estudio de la ecuación (3.19), mostrando su naturaleza altamente oscilante. Ello se aprecia claramente al resolver el problema diferencial con **Matlab**.

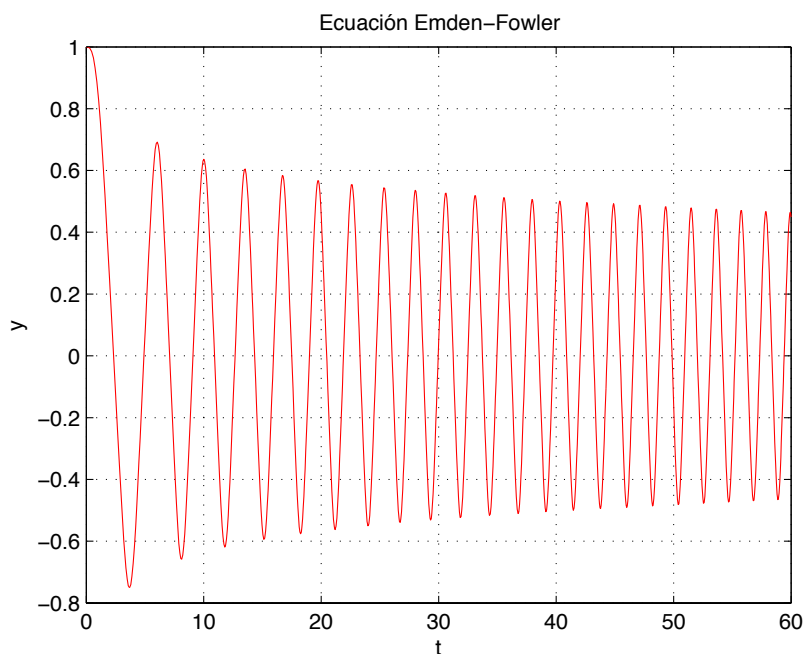


Figura 3.1: Solución aproximada, con alta precisión, obtenida por **Matlab**. Nótese la característica oscilante de la solución.

Por otra parte, la solución aproximada  $y(t)$  de la ecuación (3.19) que entrega Magnus de cuarto orden al comparar el error relativo cometido versus el tamaño de paso  $h$ , figura 3.2, nos indica el orden del método.

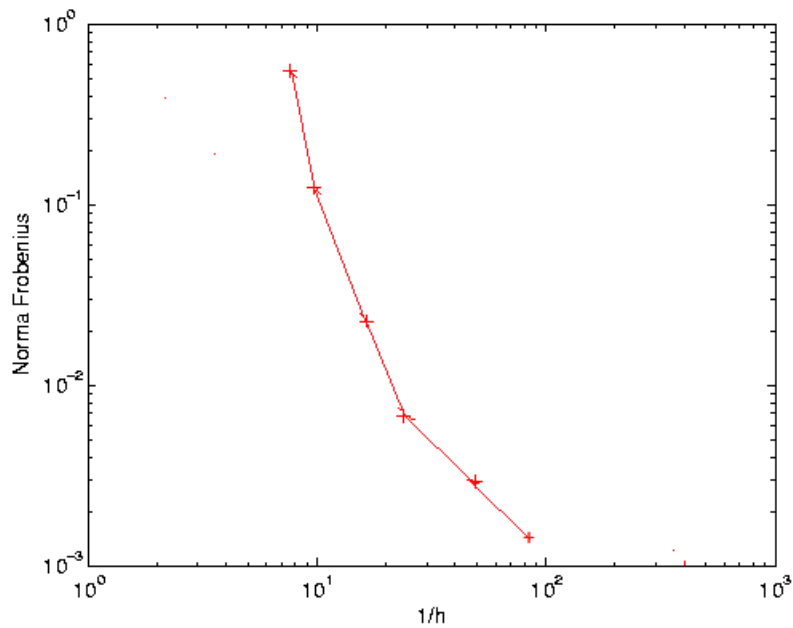


Figura 3.2: Comportamiento del error relativo cometido por el método de Magnus de cuarto orden en contraste al tamaño de paso para el problema dado por el modelo (3.19).

Para contrastar el método de RK de cuarto orden y el método de Magnus de igual orden, se presenta en la figura 3.3 las gráficas del error relativo de ambos procedimientos, comparados con RK de orden 6 (con  $h = 10^{-6}$ ), como referencia. Se integra con  $t \in [0, 10]$ , y tamaños de paso indicados. El carácter altamente oscilatorio de la solución de (3.19) también se aprecia en la distribución del error relativo a lo largo del intervalo de la integración.

### Resultados de un método Magnus modificado de cuarto orden

En el artículo [CI06] se describe con detalle un método de Magnus modificado. El algoritmo, presentado a continuación, corresponde a esta variación del método de Magnus, que contempla una disminución de ciertas operaciones aritméticas.

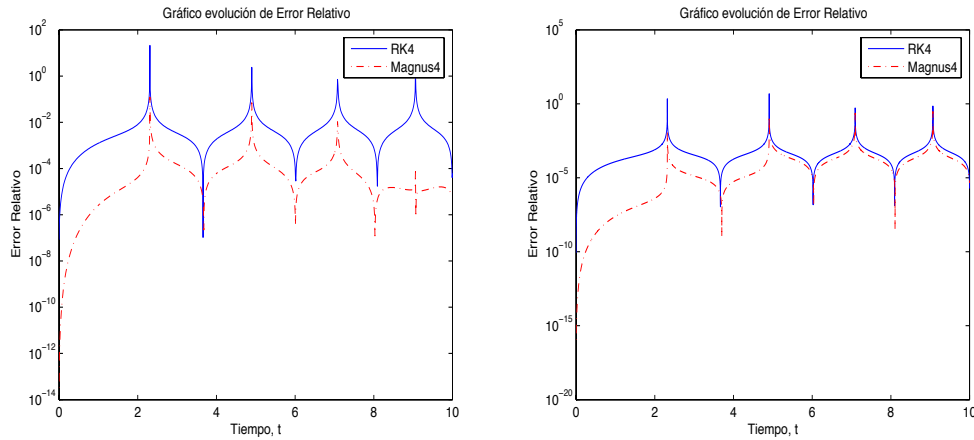


Figura 3.3: Los gráficos corresponden a los errores relativos de los métodos de Runge-Kutta y Magnus, ambos de cuarto orden. El gráfico de la izquierda utiliza un paso  $h = 0.1$  y el de la derecha  $h = 0.001$ . En las dos experiencias se observa que el método de Magnus tiene un error relativo menor, aunque se estrecha la brecha de la diferencia cuando refinamos el paso de integración.

A continuación se detalla el algoritmo para la integración numérica de cuarto orden de la ecuación  $Y' = A(t, Y)Y$  basado en la expansión no lineal de Magnus (3.1).

### Algoritmo *Magnus modificado*

---

En cada iteración se obtendrá la aproximación  $Y_{n+1}$ , usando  $t_{n+1} = t_n + h$

1.  $u_1 = 0$   
 $k_1 = hA(t_n, Y_n), \quad Q_1 = k_1$
2.  $u_2 = \frac{1}{2}Q_1$   
 $k_2 = hA\left(t_n + \frac{h}{2}, e^{u_2}Y_0\right), \quad Q_2 = k_2 - k_1$
3.  $u_3 = \frac{1}{2}Q_1 + \frac{1}{4}Q_2$   
 $k_3 = hA\left(t_n + \frac{h}{2}, e^{u_3}Y_0\right), \quad Q_3 = k_3 - k_2$

$$\begin{aligned}
4. \quad & u_4 = Q_1 - Q_2 \\
& k_4 = hA(t_n + h, e^{u_4}Y_0), \quad Q_4 = k_4 - 2k_2 - k_1 \\
5. \quad & u_5 = \frac{1}{2}Q_1 + \frac{1}{4}Q_2 + \frac{1}{3}Q_3 - \frac{1}{24}Q_4 - \frac{1}{48}[Q_1, Q_2] \\
& k_5 = hA\left(t_n + \frac{h}{2}, e^{u_5}Y_0\right), \quad Q_5 = k_5 - k_2 \\
6. \quad & u_6 = Q_1 + Q_2 + \frac{2}{3}Q_3 + \frac{1}{6}Q_4 - \frac{1}{6}[Q_1, Q_2] \\
& k_6 = hA(t_n + h, e^{u_6}Y_0), \quad Q_6 = k_6 - 2k_2 + k_1 \\
7. \quad & v = Q_1 + Q_2 + \frac{2}{3}Q_5 + \frac{1}{6}Q_6 - \frac{1}{6}\left[Q_1, Q_2 - Q_3 + Q_5 + \frac{1}{2}Q_6\right] \\
& Y_{n+1} = e^v Y_n.
\end{aligned}$$


---

Aplicamos esta variación del método de Magnus a la ecuación de Emden–Fowler presentada en (3.19).

Comparamos los tiempos de CPU que requiere el algoritmo modificado con el de RK del mismo orden, figura 3.4, con respecto al mismo tamaño de paso. Se aprecia que el algoritmo de Magnus modificado requiere de menos tiempo de CPU, en particular para tamaños de paso  $h$  decrecientes.

### Algoritmo de Magnus con $\Omega^{[1]}$ exacta

En esta sección modificamos el algoritmo de la sección anterior, usando la evaluación de forma exacta para la primera integral, es decir,  $\Omega^{[1]}$ , en vez de considerar una cuadratura. Esto resulta sencillo para el caso que nos ocupa.

Recordemos que

$$\Omega^{[1]}(t) = \int_0^t A(s, Y_0) ds.$$

Examinando la fórmula (3.8) y usando  $m = 2$ , obtenemos  $\Omega^{[1]}$  para (3.18), en cada uno de los tramos de la partición del intervalo para  $t$ , es decir,

$$\Omega^{[1]}(t_{n+1}) = \int_{t_n}^{t_n+h} A(s, Y_n) ds = \int_{t_n}^{t_n+h} \begin{pmatrix} 0 & 1 \\ -sy_n^2 & 0 \end{pmatrix} ds = \begin{pmatrix} 0 & h \\ -\beta & 0 \end{pmatrix},$$

donde  $\beta = -\frac{1}{2}y_n^2 h(2t_n + h)$ . Esto genera una nueva versión del algoritmo del apéndice de [CI06].

El gráfico de la figura 3.5 muestra los errores relativos de  $y_n \approx y(t_n)$  para ambos algoritmos, el que calcula  $\Omega^{[1]}$  en forma aproximada y en forma exacta.

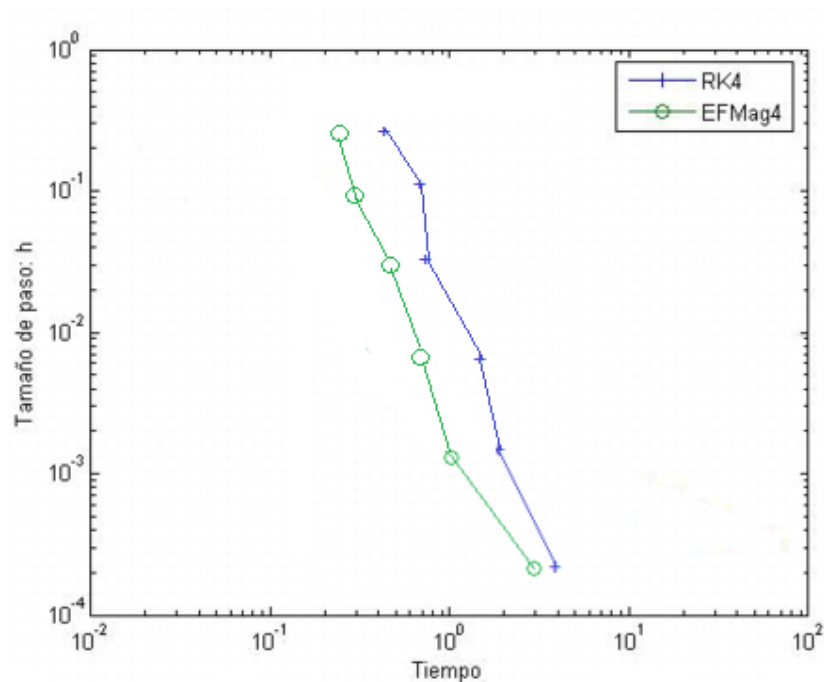


Figura 3.4: Gráfico del tiempo de CPU para RK4 y EFMag4. El método de Magnus modificado presenta menor requerimiento de tiempo CPU debido a que el algoritmo presentado en el apéndice de [CI06] requiere seis evaluaciones de  $A$ , dos conmutadores y seis exponenciales de matrices por cada paso de tiempo. Esto es un poco menos de cálculo de punto flotante que su similar RK4.

Como era de esperar, la versión que utiliza el cálculo exacto de  $\Omega^{[1]}$  tiene un comportamiento con menor error relativo que su versión aproximada.

### 3.3.2. Ecuación de Lane–Emden

Al igual que hicimos para la ecuación de Emden–Fowler, resolvemos el problema (3.18), pero ahora con la ecuación de Lane–Emden, también de frecuente uso en modelos físicos. En este caso la función  $a(t, Y) = \left(\frac{y}{t}\right)^{n-1}$ . Nosotros tomaremos  $n = 3$ . La solución aproximada, con una alta precisión, obtenida por el software **Matlab** se muestra en la figura 3.6. La gráfica 3.7 representa el diagrama de eficiencia: error versus tiempo de CPU del

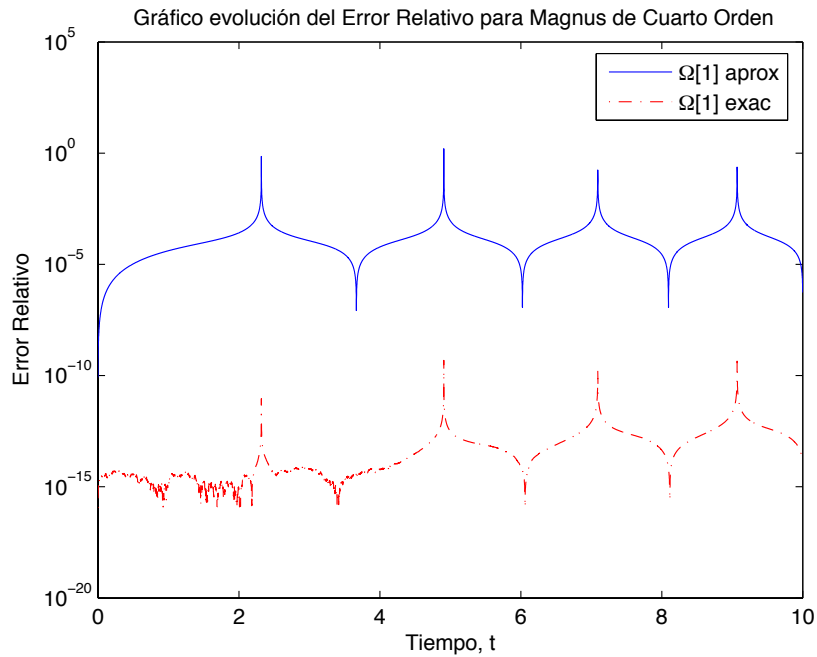


Figura 3.5: Gráfico del error relativo del método de Magnus con  $\Omega^{[1]}$  aproximada y exacta, con tamaño de paso  $h = 10^{-2}$ , y evolucionando el tiempo  $t \in [0, 10]$ . Nuevamente como referencia de la solución exacta se usó RK de orden 6, con  $h = 10^{-6}$ . Al igual que en la figura 3.3, se aprecia las formas de cúpulas periódicas que va tomando el error relativo a medida que la variable  $t$  va recorriendo el intervalo de integración.

método de Magnus modificado, con las versiones de  $\Omega^{[1]}$  aproximada y exacta. Se aprecia que ambas versiones alcanzan una precisión similar, con un costo en tiempo de cómputo ligeramente mayor en el caso de  $\Omega^{[1]}$  calculada exactamente.

### 3.3.3. Aplicación. Ecuación de Thomas–Fermi

La última ecuación que se estudiará es la de Thomas–Fermi<sup>1</sup>. En este caso se tiene que  $a(t, Y) = -\sqrt{\frac{y}{t}}$ . La solución de esta ecuación, dada por **Matlab**

<sup>1</sup>Esta ecuación aparece en el modelo de Thomas–Fermi, que es una teoría de la mecánica cuántica para explicar la estructura electrónica de sistemas de muchos cuerpos, desarrollado de forma semiclásica poco tiempo después de la introducción de la ecuación de Schrödinger

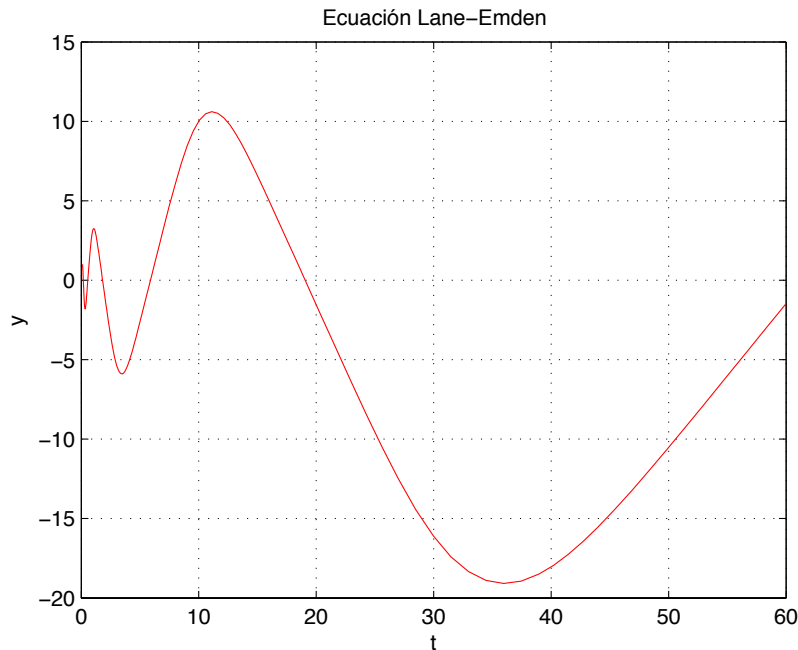


Figura 3.6: Solución para la evolución en el tiempo  $t, t \in (0, 60]$  obtenida por **Matlab** para la ecuación de Lane-Emden con  $n = 3$ .

con alta precisión, también presenta un aspecto oscilatorio y uniforme, como se aprecia en la gráfica de la figura 3.8.

Para contrastar los métodos de RK de cuarto orden y el método de Magnus de igual orden, se presentan en la figura 3.9 las gráficas del error relativo de ambos procedimientos, comparados con RK de orden 6 como referencia. Se integra con  $t \in (0, 70]$ . El carácter oscilatorio de la gráfica de la solución aproximada también se aprecia en la distribución del error relativo a lo largo del intervalo de integración. En la gráfica 3.10 se presenta el diagrama de eficiencia del método de Magnus modificado con ambas versiones de cálculo de  $\Omega^{[1]}$  aproximada y exacta. En esta experiencia se llega a utilizar por ambas versiones del método de Magnus un valor de paso de  $h = 10^{-5}$ , por ello se tiene un tiempo de CPU bastante elevado. Sin embargo, al igual que en las experiencias anteriores (Emden-Fowler y Lane-Emden) se aprecia que ambas versiones alcanzan una precisión similar, con un costo en tiempo de cómputo ligeramente mayor en el caso de  $\Omega^{[1]}$  calculada exactamente.

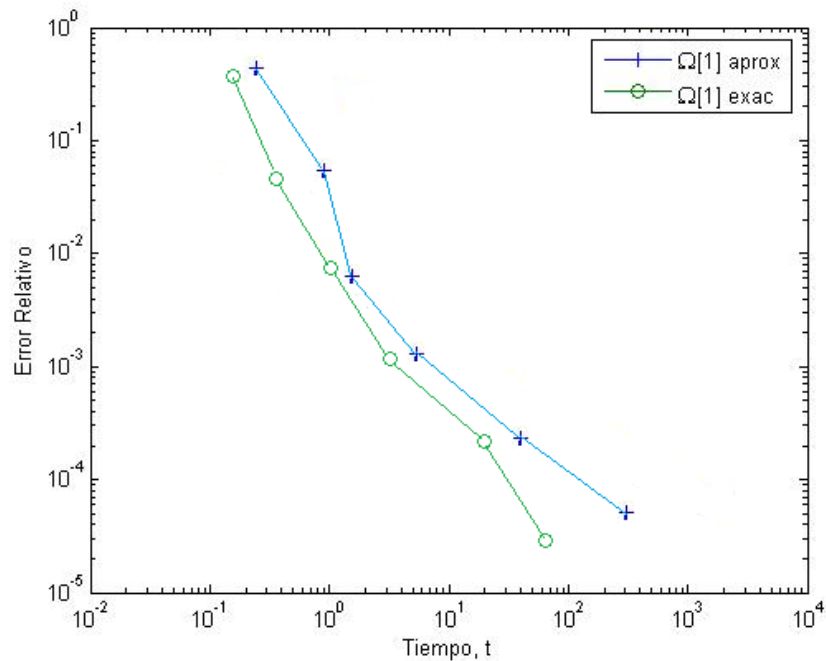


Figura 3.7: Diagrama de la eficiencia del método de Magnus modificado, con las versiones de  $\Omega^{[1]}$  aproximada y exacta, para el modelo gobernado por la ecuación de Lane–Emden con  $n = 3$ . Se utiliza RK de orden 6, con  $h = 10^{-6}$ , como referencia de la solución exacta.

### 3.4. Método de splitting basado en matrices triangulares

En esta sección se amplía el estudio del método de tipo *splitting* desarrollado en el capítulo 2, para problemas no lineales. Se describe detalladamente el procedimiento para calcular los diversos componentes que requiere este nuevo método, descrito en ([Cas07]). En concreto, discutiremos el problema *no lineal* representado por la ecuación

$$Y'(t) = A(t, Y(t))Y(t), \quad \text{con } Y(t_0) = Y_0 \quad \text{en } \text{GL}(n), \quad (3.20)$$

donde  $A(t, Y)$  es una matriz que depende de  $t$  e  $Y(t)$ . Por simplicidad se tomará el caso  $t_0 = 0$  e  $Y_0 = I$ . Por igual motivo se analizará sólo el primer ciclo del procedimiento.



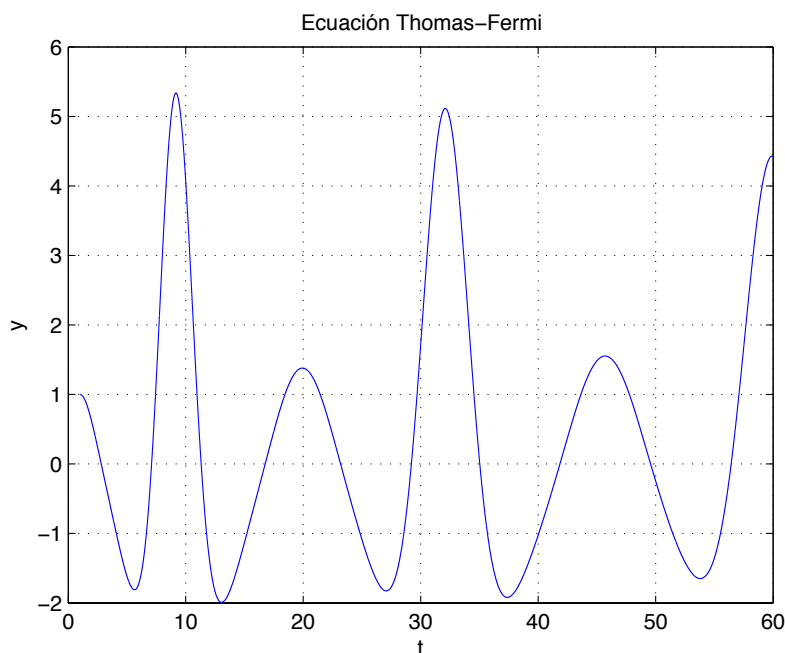


Figura 3.8: Solución dada por **Matlab** para la ecuación de Thomas–Fermi. Al igual que en las otras ecuaciones estudiadas, la ecuación de Thomas-Fermi presenta una solución oscilante.

Nuestro punto de partida será la construcción de la solución del sistema

$$L'_0 = A_-(t, L_0)L_0, \quad L_0(0) = I, \quad (3.21)$$

que puede ser integrado exactamente. Aquí  $A_-$  es triangular inferior.

Denotaremos las filas de  $L_0$  por  $\mathbf{l}_1^T, \mathbf{l}_2^T, \dots, \mathbf{l}_n^T$ . En forma análoga, las filas de  $A_-$  se denotan por  $\mathbf{a}_i^T$  y las de  $I$  por  $\mathbf{i}_i^T$ .

Definiremos la matriz extendida

$$C_m = [ \mathbf{l}_1^T \quad \mathbf{l}_2^T \quad \dots \quad \mathbf{l}_{m-1}^T \quad \mathbf{i}_m^T \quad \dots \quad \mathbf{i}_n^T ]^T, \quad m = 1, 2, \dots, n.$$

Observamos que  $C_1 = Y(0) = I$ . Así el primer renglón de la matriz  $A_-$  será

$$\mathbf{a}_1^T(t, L_0) = [a_{11}(t, C_1), 0, \dots, 0]$$

y entonces

$$\mathbf{l}_1^{T'} = a_{11}(t, C_1)\mathbf{l}_1^T,$$

que es soluble exactamente por cuadraturas.

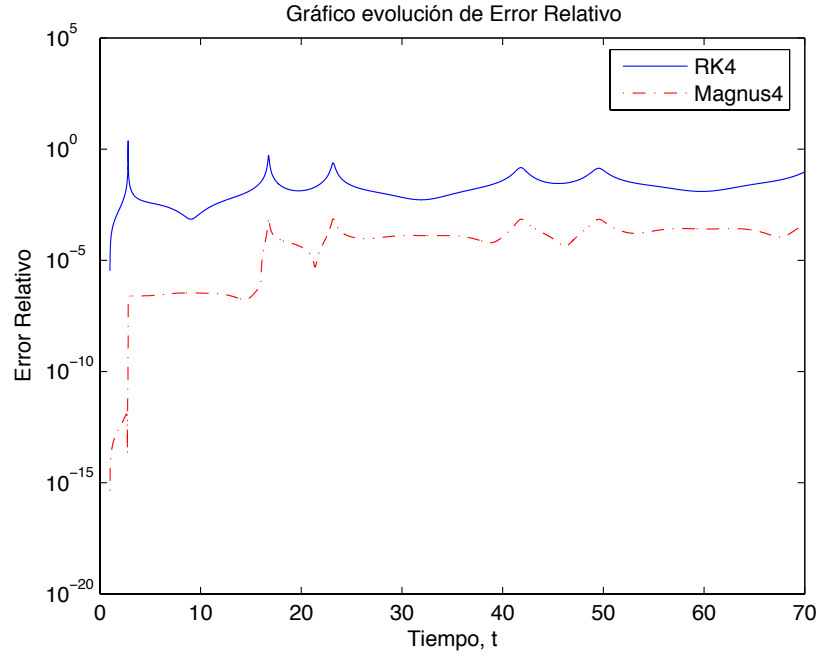


Figura 3.9: Evolución de error relativo en el tiempo  $t \in [0, 70]$ , con las versiones de  $\Omega^{[1]}$  aproximada y exacta, para la ecuación de Thomas–Fermi. Se utiliza RK de orden 6, con  $h = 10^{-6}$ , como referencia de la solución exacta.

La siguiente ecuación surge del segundo renglón de  $A_-$ ,

$$\mathbf{a}_2^T(t, L_0) = [a_{21}(t, C_2), a_{22}(t, C_2), 0, \dots, 0].$$

Así tendremos la ecuación

$$l_2^{T'} = a_{21}(t, C_2)l_1^T + a_{22}(t, C_2)l_2^T,$$

también soluble exactamente.

En general tendremos que el  $m$ -ésimo renglón de la matriz  $A_-$ ,

$$\mathbf{a}_m^T(t, L_0) = [a_{m1}(t, C_m), a_{m2}(t, C_m), \dots, a_{mm}(t, C_m), 0, \dots, 0], \quad m = 1, 2, \dots, n.$$

Así se forma la ecuación diferencial

$$l_m^{T'} = \sum_{j=1}^m a_{mj}(t, C_j)l_j^T, \quad m = 1, 2, \dots, n.$$

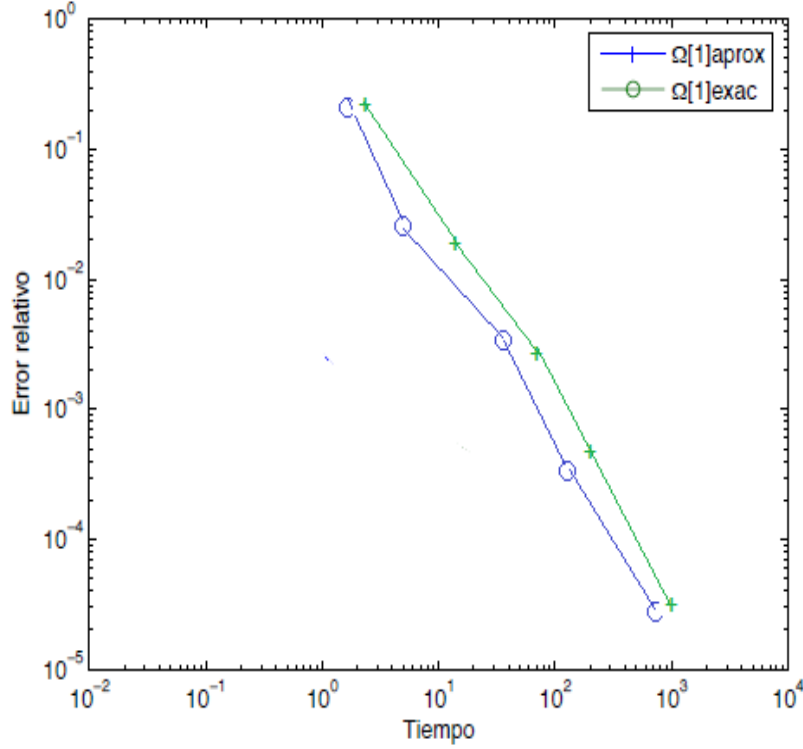


Figura 3.10: Diagrama de la eficiencia del método de Magnus modificado, con las versiones de  $\Omega^{[1]}$  aproximada y exacta, para la ecuación de Thomas–Fermi. Se utiliza RK de orden 6, con  $h = 10^{-6}$ , como referencia de la solución exacta.

### 3.4.1. Procedimiento para el cálculo de $L_k$

Desarrollaremos el procedimiento para el cálculo de  $L_k$ , en particular para  $k = 0$  y  $n = 2$ .

Para obtener la matriz  $L_0 \Rightarrow l_1^T, l_2^T, \dots, l_n^T$ , construimos cada fila de la matriz, resolviendo secuencialmente las siguientes ecuaciones:

$$1^{\text{a}} Ec. \quad l_1^{T'} = a_{11}(t, C_1)l_1^T$$

$$2^{\text{a}} Ec. \quad l_2^{T'} = a_{21}(t, C_2)l_1^T + a_{22}(t, C_2)l_2^T$$

$$i^{\text{ésima}} Ec. \quad l_m^{T'} = \sum_{j=1}^m a_{mj}(t, C_j)l_j^T.$$

Podemos resolver las ecuaciones a través de una cuadratura como la *regla Simpson*. Por ejemplo, para la primera ecuación se tiene:

$$l_1^T(t) = e^{\int_0^t a_{11}(h, C_1) dh} = e^{A_{11}(t)},$$

con

$$\int_0^t a_{11}(h, C_1) dh \simeq \frac{h}{6} [a_{11}(0, C_1) + 4a_{11}(t/2, C_1) + a_{11}(t, C_1)].$$

Se utiliza este resultado para obtener  $l_2^{T'}$ :

$$\begin{aligned} l_2^{T'} &= a_{21}(t, C_2)l_1^T + a_{22}(t, C_2)l_2^T \implies l'_{21} = a_{21}(t, C_2)l_{11} + a_{22}(t, C_2)l_{21} \\ &\implies l'_{21} = a_{21}(t, C_2)e^{A_{11}(t)} + a_{22}(t, C_2)l_{21}. \end{aligned}$$

De esta manera

$$l_{21}(t) = e^{\int_0^t a_{22}(s, C_2) ds} \left[ l_{21}(0) + \int_0^t e^{-\int_0^s a_{22}(s, C_2) ds} a_{21}(s, C_2) e^{A_{11}(s)} ds \right].$$

Dado que  $l_{21}(0) = 0$ , entonces la expresión anterior se reduce a

$$l_{21}(t) = e^{A_{22}(t)} \int_0^t e^{-A_{22}(s)} a_{21}(s, C_2) l_{11}(s) ds,$$

donde  $A_{22}$  es el término que se obtiene, en forma análoga a  $l_{11}$ , para  $l_{22}$

$$l_{22}(t) = e^{\int_0^t a_{22}(s, C_2) ds} = e^{A_{22}(t)}.$$

En términos más precisos utilizamos las siguientes fórmulas:

$$A_{11}(h) = \int_0^h a_{11}(t, C_1) dt \approx \frac{h}{6} \left[ a_{11}(0, C_1) + 4a_{11}\left(\frac{h}{2}, C_1\right) + a_{11}(h, C_1) \right] \equiv \tilde{A}_{11}(h).$$

Hemos usado la *regla de Simpson* para aproximar  $A_{11}$ . Análogamente.

$$A_{11}\left(\frac{h}{2}\right) \approx \frac{h}{24} \left[ 5a_{11}(0, C_1) + 8a_{11}\left(\frac{h}{2}, C_1\right) - a_{11}(h, C_1) \right] \equiv \tilde{A}_{11}\left(\frac{h}{2}\right),$$

luego

$$l_{11}(h) = e^{\tilde{A}_{11}(h)}$$

$$l_{11}\left(\frac{h}{2}\right) = e^{\tilde{A}_{11}\left(\frac{h}{2}\right)}$$

$$C_2(0) = I, \quad C_2(h) = \begin{bmatrix} l_{11}(h) & 0 \\ 0 & 1 \end{bmatrix}$$

$$C_2\left(\frac{h}{2}\right) = \begin{bmatrix} l_{11}\left(\frac{h}{2}\right) & 0 \\ 0 & 1 \end{bmatrix}.$$

De esta manera

$$A_{22}(h) = \int_0^h a_{11}(t, C_2) dt \approx \frac{h}{6} \left[ a_{22}(0, C_0) + 4a_{22}\left(\frac{h}{2}, C_2\right) + a_{22}(h, C_2) \right] \equiv \tilde{A}_{22}(h),$$

$$A_{22}\left(\frac{h}{2}\right) \approx \frac{h}{24} \left[ 5a_{22}(0, C_2) + 8a_{22}\left(\frac{h}{2}, C_2\right) - a_{22}(h, C_2) \right] \equiv \tilde{A}_{22}\left(\frac{h}{2}\right)$$

Por último,

$$l_{22}(h) \approx e^{\tilde{A}_{22}(h)}$$

$$l_{22}\left(\frac{h}{2}\right) \approx e^{\tilde{A}_{22}\left(\frac{h}{2}\right)}$$

$$l_{21}(h) = e^{\tilde{A}_{22}(h)} \frac{h}{6} \left( a_{21}(0, C_2(0)) + 4F_{21}\left(\frac{h}{2}\right) + F_{21}(h) \right)$$

$$l_{21}\left(\frac{h}{2}\right) = e^{\tilde{A}_{22}\left(\frac{h}{2}\right)} \frac{h}{24} \left( 5a_{21}(0, C_2(0)) + 8F_{21}\left(\frac{h}{2}\right) - F_{21}(h) \right),$$

donde

$$F_{21}(t) = e^{-A_{22}(t)} a_{21}(t, C_2(t)) l_{11}(t).$$

Así, tenemos construida la matriz

$$L_0(t) = \begin{bmatrix} l_{11}(t) & 0 \\ l_{21}(t) & l_{22}(t) \end{bmatrix}, t > 0, \quad L_0(0) = I.$$

### 3.4.2. Procedimiento para el cálculo de $U_0(t)$

El procedimiento para resolver el problema no lineal (3.20) requiere, después del cálculo de  $L_k$ , la factorización

$$Y(t) = L_0(t)Z(t) \Rightarrow Z' = C_0(t, Z)Z, \quad Z(0) = I,$$

donde

$$C_0(t, Z) = L_0^{-1}[A(t, L_0 Z) - A_-(t, L_0)]L_0 = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

$$Z(t) = U_0(t)Y_1(t)$$

$$U_0' = C_{0+}(t, U_0)U_0, \quad U_0(0) = I,$$

con

$$C_{0+}(t, U_0) = \begin{bmatrix} c_{11}(t, U_0) & c_{12}(t, U_0) \\ 0 & c_{22}(t, U_0) \end{bmatrix}$$

$$U_0 = \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix}.$$

Definimos

$$D_1 = I, \quad D_2 = \begin{bmatrix} 1 & 0 \\ 0 & u_{22}(t) \end{bmatrix},$$

luego

$$u_{11}(t) = e^{C_{11}(t)}, \quad C_{11}(t) \equiv \int_0^t c_{11}(s, D_2)ds$$

$$u_{12}(t) = e^{C_{11}(t)} \int_0^t e^{-C_{11}(s)} c_{12}(s, D_2) u_{22}(s) ds.$$

Al igual que en el cálculo de  $L_0$ , se utiliza la *regla de Simpson* para aproximar los elementos de  $U_0(t)$ .

De esta manera construimos

$$U_0 = \begin{bmatrix} u_{11}(t) & u_{12}(t) \\ 0 & u_{22}(t) \end{bmatrix}.$$

### 3.4.3. Parte final del método

Una vez calculada la matriz  $U_0(t)$ , obtenemos  $Z(t)$  dado por

$$Z(t) = U_0(t)Y_1(t),$$

donde

$$Y_1' = U_0^{-1}(C_0 - C_{0+})U_0Y_1 \equiv A_1(t, Y_1)Y_1, \quad Y_1(0) = I.$$

Así se completa el primer ciclo. Este procedimiento iterativo genera la siguiente factorización

$$Y(t) \equiv Y_0(t) = L_0 U_0 L_1 U_1 \dots L_k U_k Y_{k+1}(t), \quad k = 0, 1, 2, \dots$$

Cabe notar que este procedimiento reproduce el algoritmo de la sección 2.4 en el caso de una ecuación lineal.

### 3.4.4. Ejemplo numérico

Tomamos la ecuación diferencial (3.17), que se puede representar como una ecuación diferencial matricial (3.18). En particular, estudiamos el caso de la ecuación de Emden-Fowler [IN99], es decir,  $a(t, y) = ty^2$ . En la gráfica de la figura 3.11 se representa los diagramas de eficiencia del método basado en splitting desarrollado en esta sección y el de Magnus.

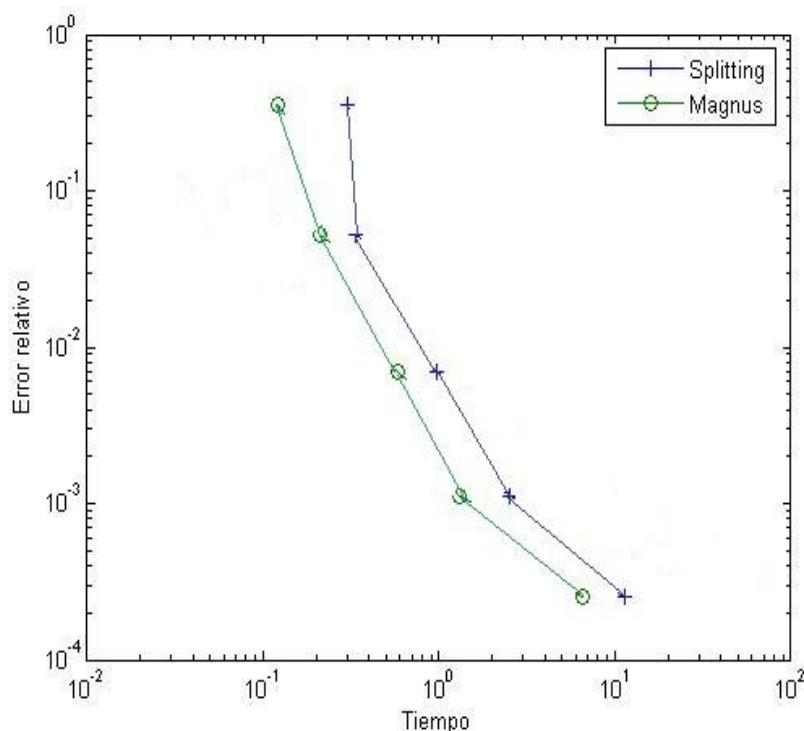


Figura 3.11: Diagrama de eficiencia, error relativo versus tiempo de CPU, para los métodos de splitting no lineal y el de Magnus desarrollado al inicio del capítulo. Se aprecia que el método de splitting requiere un poco más de tiempo pero proporciona una precisión similar a Magnus. Ambos procedimientos se aplican a la ecuación de Emden-Fowler, usando como solución exacta el resultado dado por un método de Runge-Kutta de sexto orden, con  $h = 10^{-6}$ .





## Capítulo 4

# Aspectos computacionales de la fórmula de Baker–Campbell–Hausdorff y la fórmula de Zassenhaus

### 4.1. Introducción

Como hemos visto en los capítulos anteriores, los desarrollos exponenciales son de gran importancia a la hora de construir aproximaciones (tanto analíticas como numéricas) a sistemas de ecuaciones diferenciales que poseen propiedades cualitativas. Un ejemplo de tales desarrollos exponenciales es el desarrollo de Magnus. Consideremos en particular el problema de valor inicial

$$U' = A(t)U, \quad U(0) = I, \quad (4.1)$$

con

$$A(t) = \begin{cases} Y & 0 \leq t \leq 1 \\ X & 1 < t \leq 2, \end{cases}$$

siendo  $X$  e  $Y$  dos matrices  $n \times n$  constantes que, en general, no conmutan:  $[X, Y] = XY - YX \neq 0$ . La solución exacta de este problema en  $t = 2$  es claramente

$$U(t = 2) = e^X e^Y.$$

Por otra parte, si aplicamos el desarrollo de Magnus, podremos expresar

$$U(t = 2) = e^{\Omega(2)},$$

donde  $\Omega(2)$  depende de  $X$ ,  $Y$  y sus conmutadores anidados. Éste es de hecho un caso particular de la llamada fórmula de Baker–Campbell–Hausdorff: el producto de la exponencial de dos matrices que no conmutan es igual a la exponencial de otra matriz obtenida a partir de las dos anteriores mediante una combinación lineal (con coeficientes racionales) de sus conmutadores anidados.

Supongamos ahora que tenemos  $e^{X+Y}$ . Sabemos, por lo que hemos comentado anteriormente, que en general

$$e^{X+Y} \neq e^X e^Y.$$

Ahora bien, ¿es posible expresar  $e^{X+Y}$  como un producto de la forma

$$e^{X+Y} = e^X e^Y e^{C_2} e^{C_3} \dots \quad ?$$

Tal desarrollo, dual del anterior, recibe el nombre de fórmula de Zassenhaus, y ha encontrado aplicación recientemente en varios campos.

En este capítulo se estudiará con detalle tanto la fórmula de Baker–Campbell–Hausdorff como la de Zassenhaus, atendiendo principalmente a sus aspectos computacionales (esto es, a la forma más eficiente de generarlas) y a las condiciones que se han de verificar para que se tenga asegurada su convergencia.

Así como en cierto sentido el desarrollo de Magnus puede considerarse como una versión “continua” de la fórmula de Baker–Campbell–Hausdorff, analizaremos también el análogo continuo de la fórmula de Zassenhaus, conocido como el desarrollo de Wilcox.

La fórmula de Baker–Campbell–Hausdorff (BCH, para abreviar) fue establecida (cualitativamente) por Campbell ([Cam97] y [Cam98]) a finales del siglo XIX analizando el problema de la construcción de un grupo de Lie directamente a partir de un álgebra de Lie [Gil74]. Esencialmente, este resultado establece que  $e^X e^Y = e^Z$ , donde  $Z$  se puede escribir como una serie que contiene únicamente conmutadores de  $X$  e  $Y$ . Posteriormente Baker [Bak05] y Hausdorff [Hau06] de forma independiente también formularon un resultado análogo dentro de un marco abstracto más amplio, aunque ninguno de ellos proporcionó una demostración rigurosa del resultado. En palabras de Bourbaki, “*chacun considère que les démonstrations de ses prédécesseurs ne sont pas convaincantes*”. Finalmente, Dynkin logró encontrar una fórmula explícita para  $Z$ .

## 4.2. La fórmula de Baker–Campbell–Hausdorff

La fórmula de Baker–Campbell–Hausdorff (BCH, para abreviar) fue establecida (cualitativamente) por Campbell ([Cam97] y [Cam98]) a finales

del siglo XIX analizando el problema de la construcción de un grupo de Lie directamente a partir de un álgebra de Lie [Gil74]. Esencialmente, este resultado establece que  $e^X e^Y = e^Z$ , donde  $Z$  se puede escribir como una serie que contiene únicamente conmutadores de  $X$  e  $Y$ . Posteriormente Baker [Bak05] y Hausdorff [Hau06] de forma independiente también formularon un resultado análogo dentro de un marco abstracto más amplio, aunque ninguno de ellos proporcionó una demostración rigurosa del resultado. En palabras de Bourbaki, “*chacun considère que les démonstrations de ses prédécesseurs ne sont pas convaincantes*”. Finalmente, Dynkin logró encontrar una fórmula explícita para  $Z$ .

El exponente  $Z$  en  $e^X e^Y = e^Z$  se puede determinar usando el desarrollo en serie de la función exponencial

$$e^X e^Y = \sum_{p,q=0}^{\infty} \frac{1}{p!q!} X^p Y^q, \quad (4.2)$$

y sustituyendo esta serie en la serie formal de la función logaritmo,

$$\log(Z) = \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k} (Z - I)^k. \quad (4.3)$$

Así se obtiene

$$\log(e^X e^Y) = \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k} \sum \frac{X^{p_1} Y^{q_1} \dots X^{p_k} Y^{q_k}}{p_1! q_1! \dots p_k! q_k!},$$

donde los términos no negativos de la suma interna  $p_1, q_1, \dots, p_k, q_k$  cumplen  $p_i + q_i > 0$  ( $i = 1, \dots, k$ ). Agrupando los términos para los que  $p_1 + q_1 + p_2 + q_2 + \dots + p_k + q_k = m$ , podemos escribir

$$Z = \log(e^X e^Y) = \sum_{m=1}^{\infty} P_m(X, Y), \quad (4.4)$$

donde  $P_m(X, Y)$  es un polinomio homogéneo de grado  $m$  en las variables no conmutativas  $X$  e  $Y$ . Como hemos dicho, Campbell [Cam98], Baker [Bak05] y Hausdorff [Hau06] abordaron la cuestión de si  $Z$  se puede representar en general como una serie de conmutadores anidados de  $X$  e  $Y$ , pero fue Dynkin quien finalmente derivó una fórmula explícita para  $Z$  como

$$Z = \sum_{k=1}^{\infty} \sum_{p_i, q_i} \frac{(-1)^{k+1}}{k} \frac{[X^{p_1} Y^{q_1} \dots X^{p_k} Y^{q_k}]}{\left(\sum_{i=1}^k (p_i + q_i)\right) p_1! q_1! \dots p_k! q_k!}. \quad (4.5)$$

Aquí la suma interna se hace tomando todos los enteros no negativos  $p_1, q_1, \dots, p_k, q_k$  tal que  $p_i + q_i > 0$ , para  $i = 1, \dots, k$  y  $[X^{p_1}Y^{q_1} \dots X^{p_k}Y^{q_k}]$  denota el conmutador anidado hacia la derecha basado en la *palabra*  $X^{p_1}Y^{q_1} \dots X^{p_k}Y^{q_k}$  [Dyn47]. Así, por ejemplo,

$$[XY^2X^2Y] \equiv [X, [Y, [Y, [X, [X, Y]]]]].$$

La expresión (4.5) es conocida como la serie BCH en la forma de *Dynkin*. Agrupando términos se llega finalmente a

$$Z = \log(e^X e^Y) = X + Y + \sum_{m=2}^{\infty} z_m(X, Y), \quad (4.6)$$

donde los  $z_m(X, Y)$  son polinomios homogéneos de Lie en  $X$  e  $Y$  de grado  $m$ , es decir, una combinación lineal de los conmutadores de la forma  $[V_1, [V_2, \dots, [V_{m-1}, V_m] \dots]]$  con  $V_i \in \{X, Y\}$  para  $i = 1, \dots, m$ , siendo los coeficientes números racionales. Los primeros términos en forma explícita son

$$\begin{aligned} z_2 &= \frac{1}{2}[X, Y] \\ z_3 &= \frac{1}{12}[X, [X, Y]] - \frac{1}{12}[Y, [X, Y]] \\ z_4 &= \frac{1}{24}[X, [Y, [Y, X]]]. \end{aligned}$$

La expresión  $e^X e^Y = e^Z$  es llamada fórmula de *Baker-Campbell-Hausdorff*, aunque se utilizan comúnmente en la literatura otros nombres (por ejemplo, Campbell–Baker–Hausdorff, Baker–Hausdorff, Campbell–Hausdorff). La fórmula (4.5) es ciertamente difícil de usar debido a la complejidad de las sumas involucradas. Nótese, en particular, que diferentes opciones de  $p_i, q_i, k$  en (4.6) pueden dar lugar a un mismo conmutador. Así, por ejemplo,  $[X^3Y^1] = [X^1Y^0X^2Y^1] = [X, [X, [X, Y]]]$ . Una dificultad adicional surge del hecho de que no todos los conmutadores son independientes, debido a la identidad de Jacobi

$$[X_1, [X_2, X_3]] + [X_2, [X_3, X_1]] + [X_3, [X_1, X_2]] = 0.$$

La fórmula BCH juega un papel fundamental en diversos campos de las matemáticas. Por ejemplo: en la teoría de ecuaciones diferenciales [Mag54], grupos de Lie [GOV97], análisis numérico [HLW06], la física teórica (teoría de perturbaciones [DF76], mecánica cuántica [WM62], mecánica estadística [Kum65] [Wil67], computación cuántica [SS99]) y teoría de control

(en análisis y diseño de leyes de mando no lineales, filtros no lineales, estabilización de los cuerpos rígidos, etc. [TTM05]).

En particular, en el análisis numérico de ecuaciones diferenciales la fórmula BCH a veces se utiliza como sigue. Supongamos que tenemos una ecuación diferencial

$$\frac{du}{dt} = X(u), \quad u(0) = q \quad (4.7)$$

definida en un cierto grupo de Lie (por ejemplo, un sistema Hamiltoniano), y que el campo vectorial  $X$  puede ser separado como  $X(u) = A(u) + B(u)$  de manera que las ecuaciones correspondientes a  $A(u)$  y  $B(u)$ ,

$$\frac{du}{dt} = A(u), \quad \frac{du}{dt} = B(u), \quad u(0) = q$$

pueden resolverse de forma exacta. Entonces, una práctica habitual consiste en considerar un método de splitting de la forma

$$\Psi_h \equiv e^{ha_1A} e^{hb_1B} \dots e^{ha_kA} e^{hb_kB}$$

para la solución exacta  $e^{h(A+B)}$  de (4.7) en un paso de tiempo  $h$ . La idea es entonces obtener las condiciones que deben cumplir los coeficientes  $a_i, b_i$  de modo que

$$\Psi_h(q) = u(h) + \mathcal{O}(h^{p+1}),$$

y esto se puede hacer mediante la aplicación sucesiva de la fórmula BCH para la expresión de  $\Psi$  hasta el grado requerido por el orden de la aproximación  $p$ .

Además de la forma de Dynkin hay otros procedimientos estándar para construir de forma explícita los términos de la serie BCH. Una de ellos es la forma asociativa explícita

$$Z = X + Y + \sum_{m=2}^{\infty} \sum_{w, |w|=m} g_m w, \quad (4.8)$$

donde  $w$  denota una cadena de símbolos  $X$  e  $Y$  (una palabra), es decir,  $w = w_1 w_2 \dots w_m$ , con  $w_i \in \{X, Y\}$ ,  $g_w$  es un coeficiente racional y todas las cadenas  $w$  tienen longitud, o cantidad de símbolos,  $|w| = m$ . Estos coeficientes se pueden calcular a través de un proceso basado en polinomios calculados de forma recursiva [Gol56].

Aunque la ecuación (4.8) carece de conmutadores, siempre es posible escribir esta expresión como

$$Z = X + Y + \sum_{m=2}^{\infty} \frac{1}{m} \sum_{w, |w|=m} g_m[w], \quad (4.9)$$

es decir, los términos individuales son los mismos que en la serie asociativa (4.8) excepto que la palabra  $w = w_1 w_2 \cdots w_m$  se sustituye por el conmutator anidado por la derecha  $[w] = [w_1, [w_2, \dots [w_{m-1}, w_m] \dots]]$  y el coeficiente de  $g_w$  se divide por la longitud  $m$  de la palabra correspondiente [Tho82]. Esto es un caso particular del teorema de Dynkin–Specht–Wever [Jac79].

Si  $Z = Z(\tau)$  es la solución de  $e^Z = e^{\tau X} e^{\tau Y}$ , diferenciando esta relación con respecto a  $\tau$  y considerando la serie  $Z = \sum_{k \geq 1} \tau^k z_k$ , es posible derivar la fórmula de recurrencia [Var84]

$$\begin{aligned} z_1 &= X + Y \\ (n+1)z_{n+1} &= \frac{1}{2}[X - Y, z_n] + \sum_{p=1}^{[n/2]} \frac{B_{2p}}{(2p)!} \sum [z_{k_1}, [\dots [z_{k_{2p}}, X + Y] \dots]], \quad n \geq 1, \end{aligned} \tag{4.10}$$

donde la segunda sumatoria se extiende sobre todos los enteros positivos tal que  $k_1 + \dots + k_{2p} = n$  y  $B_{2p}$  son los números de Bernoulli [AS65]. Alternativamente, si consideramos  $e^{Z(t)} = e^{tX} e^Y$ , entonces  $Z(t)$  verifica la ecuación diferencial

$$\frac{dZ}{dt} = \sum_{k=1}^{\infty} \frac{B_k}{k!} \text{ad}_Z^k X, \quad Z(0) = Y.$$

De aquí es posible obtener la siguiente recurrencia para los términos de la serie  $Z(t) = \sum_{n \geq 1} z_n(t)$  [BC04]:

$$\begin{aligned} z_1(t) &= Xt + Y \\ z_n(t) &= \sum_{j=1}^{n-1} \frac{B_j}{j!} \sum_{k_1 + \dots + k_j = n-1} \int_0^t \text{ad}_{z_{k_1}(s)} \text{ad}_{z_{k_2}(s)} \cdots \text{ad}_{z_{k_j}(s)} X ds, \quad n \geq 2, \end{aligned} \tag{4.11}$$

donde para la sumatoria interna se debe cumplir que  $k_1 \geq 1, \dots, k_j \geq 1$ . Esta última fórmula es particularmente conveniente desde un punto de vista computacional, además si en (4.11) se toma  $t = 1$ , se vuelve a la serie BCH.

Las fórmulas explícitas (4.5), (4.9), así como las recurrencias (4.10), (4.11) se pueden utilizar, en principio, para la construcción de la serie BCH hasta grado arbitrario en términos de conmutadores. Varios cálculos sistemáticos de la serie han sido llevado a cabo a lo largo de los años, empezando con el trabajo realizado por Richtmyer y Greenspan en 1965

[RG65], en el que informan de resultados hasta grado ocho. Más tarde, Newman y Thompson obtuvieron los coeficientes  $g_w$  en (4.9) hasta cadenas de longitud 20 [NT87], mientras que Bose [Bos89] propuso un algoritmo para calcular directamente el coeficiente de un conmutador determinado en la forma de Dynkin (4.5), y Oteo [Ote91] y Kolsrud [Kol93] presentaron una expresión simplificada de (4.5) en términos de conmutadores anidados por la derecha, hasta grado ocho y nueve, respectivamente. Más recientemente, Reinsch [Rei00] ha propuesto una formulación basada en operaciones con matrices para el cálculo de los polinomios  $P_m(X, Y)$  en (4.4) que puede ser implementado fácilmente en cualquier paquete de álgebra simbólica. En ese caso, se tiene que recurrir de nuevo al teorema de Dynkin–Specht–Wever para escribir las expresiones resultantes en términos de conmutadores .

Como se mencionó antes, todos estos procedimientos exhiben una limitación clave: los conmutadores iterados no son linealmente independientes debido a la identidad de Jacobi (y otras identidades que implican conmutadores anidados de grado superior obtenidas como consecuencia de la misma [Ote91]). En otras palabras, no se proporcionan expresiones directamente en términos de una base del álgebra de Lie libre generada por  $X$  e  $Y$ ,  $\mathcal{L}(X, Y)$ , y por lo tanto es difícil de estudiar con ellas las características específicas de la serie, tales como la distribución de los coeficientes y otras propiedades combinatorias [NT87]. Por supuesto, siempre es posible expresar las fórmulas resultantes en términos de una base de  $\mathcal{L}(X, Y)$  (por ejemplo, la base Hall o la de Lyndon), pero este proceso de reescritura es muy lento y requiere grandes recursos de memoria de un computador. En la práctica, más allá del grado  $n = 10$ , se trata de una tarea difícil [Kos93], [TTM05], ya que el número de términos que intervienen en la serie crece exponencialmente.

En la referencia [CM09], sus autores desarrollan un nuevo algoritmo para la expresión de la serie BCH como

$$Z = \log(\exp(X) \exp(Y)) = \sum_{i \geq 1} z_i E_i, \quad (4.12)$$

donde  $z_i \in \mathbb{Q}$  ( $i \geq 1$ ) y  $\{E_i : i = 1, 2, 3, \dots\}$  es una base de  $\mathcal{L}(X, Y)$  cuyos elementos son de la forma

$$E_1 = X, \quad E_2 = Y, \quad \text{y} \quad E_i = [E_{i'}, E_{i''}] \quad i \geq 3, \quad (4.13)$$

para números enteros adecuados  $i', i'' < i$  (para  $i = 3, 4, \dots$ ). Cada  $E_i$  en (4.13) es un polinomio de Lie homogéneo de grado  $|i|$ , donde

$$|1| = |2| = 1, \quad \text{y} \quad |i| = |i'| + |i''| \quad \text{for} \quad i \geq 3. \quad (4.14)$$

En ese trabajo se considera una clase general de bases para un álgebra de Lie libre  $\mathcal{L}(X, Y)$ , que la literatura actual refiere como bases de Hall generalizadas y también como bases Hall-Viennot [C.R93], [Vie78]. En concreto, se propone un nuevo procedimiento para escribir la serie BCH (4.12) en términos de una base Hall-Viennot arbitraria. Tal algoritmo se basa en los resultados obtenidos en [Mur06].

El algoritmo ha sido implementado en *Mathematica*<sup>TM</sup> (aunque también se puede programar en Fortran o C para una mayor eficiencia). El procedimiento resultante da la serie BCH hasta un grado prescrito directamente en términos de una base Hall-Viennot de  $\mathcal{L}(X, Y)$  y es capaz de conseguir la serie hasta el grado  $n = 20$  en cuestión de minutos. La expresión resultante tiene 109697 coeficientes no conocidos de 111013 elementos  $E_i$  de grado  $|i| \leq 20$  en la base Hall.

Por ejemplo, el último elemento de grado 20 en la base de Hall es

$$E_{111013} = \begin{aligned} &[[[[[Y, X], Y], [Y, X]], [[Y, X], X], [Y, X]], \\ &[[[[Y, X], Y], [Y, X]], [[[[Y, X], Y], Y], Y]], \end{aligned}$$

y el coeficiente correspondiente  $z_i$  en (4.12) es

$$z_{111013} = -\frac{19234697}{140792940288}.$$

La convergencia de la serie BCH también ha sido un tema importante en el tratamiento de la fórmula BCH a lo largo de los años, y muchos resultados están disponibles en la literatura. Supongamos que introducimos una norma  $\|\cdot\|$  sub-multiplicativa, tal que

$$\|[X, Y]\| \leq \mu \|X\| \|Y\| \tag{4.15}$$

para algún  $\mu > 0$ . Entonces la serie (4.5) es absolutamente convergente siempre que  $\|X\| + \|Y\| < \log 2/\mu$  [Bou89]. De hecho, se han obtenido varias cotas mejoradas para las diferentes presentaciones. Así, en particular, la presentación de Lie (4.9) converge absolutamente si  $\|x\| \leq 1/\mu$  y  $\|Y\| \leq 1/\mu$  en un álgebra de Lie  $\mathfrak{g}$  dotada de una norma que satisface (4.15) [NT87], [Tho89]. Se ha demostrado también, a partir del procedimiento recursivo (4.10), que la serie  $Z(X, Y) = \sum_{n \geq 1} Z_n(X, Y)$  es absolutamente convergente para todos los  $X, Y \in \mathcal{S} \equiv \{V \in \mathfrak{g} : \|V\| < \frac{\delta}{2\mu}\}$ , donde la constante  $\delta$  está



dada por [NT87]

$$\delta = \int_0^{2\pi} \frac{1}{2 + \frac{1}{2}y - \frac{1}{2}\cot(\frac{1}{2}y)} dy = 2.17373740\dots$$

Por otra parte, en [BC04] se demostró que la serie  $Z = \sum_{m \geq 1} Z_m$ , es decir, expresada como una serie de polinomios de Lie homogéneos en  $X$  e  $Y$  converge absolutamente en el dominio  $D_1 \cup D_2$  de  $\mathfrak{g} \times \mathfrak{g}$ , donde

$$D_1 = \left\{ (X, Y) : \mu \|X\| < \int_{\mu \|Y\|}^{2\pi} \frac{1}{g(x)} dx \right\} \quad (4.16)$$

$$D_2 = \left\{ (X, Y) : \mu \|Y\| < \int_{\mu \|X\|}^{2\pi} \frac{1}{g(x)} dx \right\} \quad (4.17)$$

y  $g(x) = 2 + \frac{x}{2}(1 - \cot \frac{x}{2})$ . Por último, si  $X$  e  $Y$  son elementos acotados en un espacio de Hilbert  $\mathcal{H}$  con  $\dim \mathcal{H} \geq 2$ , entonces la fórmula BCH en la forma (4.8), es decir, expresada como una serie de polinomios homogéneos de Lie en  $X$  e  $Y$ , converge cuando  $\|X\| + \|Y\| < \pi$  [CM09], mientras que la serie diverge en general si  $\|X\| + \|Y\| \geq \pi$  cuando  $\mu = 2$  [Mic74]. Este resultado se puede generalizar a cualquier conjunto finito de operadores  $X_1, X_2, \dots, X_k$ : la serie BCH correspondiente es convergente si  $\|X_1\| + \dots + \|X_k\| < \pi$  en la norma 2.

Ilustramos los resultados anteriores con un ejemplo sencillo. Dadas

$$X = \begin{pmatrix} \alpha & 0 \\ 0 & -\alpha \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & \beta \\ 0 & 0 \end{pmatrix}, \quad (4.18)$$

con  $\alpha, \beta \in \mathbb{C}$ , un simple cálculo nos da

$$\log(e^X e^Y) = X + \frac{2\alpha}{1 - e^{-2\alpha}} Y,$$

que es una función analítica para  $|\alpha| < \pi$  con las primeras singularidades en  $\alpha = \pm i\pi$ . Por lo tanto, la fórmula BCH no puede converger si  $|\alpha| \geq \pi$ , independientemente de  $\beta \neq 0$ . Al tomar la norma espectral, es evidente que  $\|X\| = |\alpha|$ ,  $\|Y\| = |\beta|$ , por lo que el dominio de la convergencia es  $|\alpha| + |\beta| < \pi$ . Nótese que en el límite  $|\beta| \rightarrow 0$  este dominio es óptimo.

En términos generales, sin embargo, el límite anterior es conservador, es decir, la serie BCH converge para valores mayores de  $\|X\|$  e  $\|Y\|$ . Así, en este ejemplo, para cualquier  $\alpha$  y  $\beta$  con  $|\alpha| < \pi$  y  $|\alpha| + |\beta| \geq \pi$ , la serie BCH también converge. Resulta que una caracterización más precisa del dominio

de convergencia de la serie es posible cuando  $X$  e  $Y$  son matrices (real o complejas) en términos de los valores propios de  $e^Z$  [CM09].

A veces es necesario calcular la serie de Lie  $W$  definida por

$$\exp\left(\frac{1}{2}X\right)\exp(Y)\exp\left(\frac{1}{2}X\right) = \exp(W). \quad (4.19)$$

Esto ocurre, por ejemplo, si uno está interesado en la obtención de las condiciones de orden que deben satisfacer los métodos de composición simétricos temporales para la integración numérica de ecuaciones diferenciales [Yos90]. La aplicación de la fórmula habitual BCH da entonces la expresión de  $W$  en la base Hall de  $\mathcal{L}(X, Y)$ .

Se obtiene un procedimiento más eficiente, sin embargo, mediante la introducción de un parámetro  $\tau$  en (4.19) de tal forma que

$$W(\tau) = \log(e^{\tau X/2} e^Y e^{\tau X/2}) \quad (4.20)$$

y derivando la ecuación diferencial satisfecha por  $W(\tau)$ . A través de la derivada de la función exponencial, se obtiene

$$\frac{dW}{d\tau} = X + \sum_{n=2}^{\infty} \frac{B_n}{n!} \text{ad}_W^n X, \quad W(0) = Y, \quad (4.21)$$

de donde es posible construir de forma explícita  $W$  como la serie  $W(\tau) = \sum_{k=0}^{\infty} W_k(\tau)$ , con

$$\begin{aligned} W_1(\tau) &= X\tau + Y \\ W_2(\tau) &= 0 \\ W_l(\tau) &= \sum_{j=2}^{l-1} \frac{B_j}{j!} \int_0^\tau (\text{ad}_W^j X)_l ds, \quad l \geq 3 \end{aligned} \quad (4.22)$$

donde en general  $W_{2m} = 0$  para  $m \geq 1$ .

Siguiendo un método similar al utilizado con la ecuación (4.10) en la serie BCH habitual, la recursión (4.22) nos permite expresar  $W$  en (4.19) como

$$W = \sum_{i \geq 1} w_i E_i. \quad (4.23)$$

Con respecto a la convergencia de la serie, del resultado obtenido anteriormente para la serie BCH se deduce que el presente desarrollo es convergente al menos cuando  $\|X\| + \|Y\| < \pi$ .

### 4.3. Una aplicación de la fórmula BCH

En la referencia [Cer74] Cernusca analiza el siguiente problema. Dados  $X$  e  $Y$  dos operadores no conmutativos, se trata de expresar la exponencial  $\exp(X + \lambda Y)$ , donde  $\lambda \in \mathbb{R}$ , como el producto de  $\exp(A)$  veces la exponencial de una serie de potencias de  $\lambda$ , es decir,

$$e^{X+\lambda Y} = e^X e^{U(\lambda)}, \quad (4.24)$$

donde  $U(\lambda) = \lambda u_1 + \lambda^2 u_2 + \dots$ . Cernusca propone dos procedimientos diferentes para calcular esta serie. El primero consiste simplemente en escribir

$$e^{U(\lambda)} = e^{-X} e^{X+\lambda Y}$$

y aplicar la fórmula BCH, reordenando los términos sucesivos de acuerdo con la potencia de  $\lambda$ . El segundo trata de obtener primero la ecuación diferencial satisfecha por  $U$  y luego resolverla por iteración. Con ambos procedimientos, Cernusca es capaz de obtener los primeros términos de la serie  $U(\lambda)$  como

$$\begin{aligned} \lambda u_1 &= \left( Y - \frac{1}{2}[X, Y] + \frac{1}{6}[X, [X, Y]] - \frac{1}{24}[X, [X, [X, Y]]] + \dots \right) \lambda \\ \lambda^2 u_2 &= \left( \frac{1}{12}[Y, [X, Y]] - \frac{1}{48}[X, [Y, [X, Y]]] - \frac{1}{48}[Y, [X, [X, Y]]] + \dots \right) \lambda^2, \end{aligned}$$

pero no proporciona expresiones generales. Nuestro propósito aquí es calcular la serie  $U(\lambda)$  hasta un grado arbitrario en  $\lambda$  con las técnicas que se han desarrollado para el estudio de la fórmula BCH.

A continuación analizamos el problema un poco más general definido por

$$e^{t(X+\lambda Y)} = e^{tX} e^{U(t,\lambda)}, \quad (4.25)$$

donde  $t \in \mathbb{R}$ . El problema original se recupera tomando  $t = 1$ . En realidad, nada nuevo se puede decir sobre el primer procedimiento para obtener  $U(t, \lambda)$ : se aplica el algoritmo optimizado desarrollado previamente para la fórmula BCH con las sustituciones obvias

$$X \mapsto -tX, \quad Y \mapsto t(X + \lambda Y)$$

y se reordena la serie correspondiente en potencias de  $t$  y/o  $\lambda$ .

En lo que sigue analizamos con detalle la segunda forma, con el objetivo de obtener un procedimiento más eficiente que nos permita construir la serie que define  $U(t, \lambda)$ . Para continuar, primero diferenciamos (4.25) con respecto a  $t$ . Entonces

$$\begin{aligned}
(X + \lambda Y)e^{t(X+\lambda Y)} &= \frac{d}{dt} (e^{tX} e^U) = X e^{tX} U + e^{tX} \frac{d}{dt} e^U \\
&= X e^{t(X+\lambda Y)} + e^{t(X+\lambda Y)} d \exp_{-U}(U'),
\end{aligned}$$

de modo que

$$\lambda e^{-t(X+\lambda Y)} Y e^{t(X+\lambda Y)} = \lambda e^{\text{ad}_{-t(X+\lambda Y)}} Y = d \exp_{-U}(U')$$

y finalmente

$$U' = \lambda d \exp_{-U}^{-1} (e^{-\text{ad}_{t(X+\lambda Y)}} Y), \quad U(0) = I. \quad (4.26)$$

Recordemos que

$$d \exp_{\Omega}^{-1}(C) = \frac{\text{ad}_{\Omega}}{e^{\text{ad}_{\Omega}} - I} C = \sum_{k=0}^{\infty} \frac{B_k}{k!} \text{ad}_{\Omega}^k C$$

y

$$F \equiv e^{\text{ad}_{-t(X+\lambda Y)}} Y = \sum_{k=0}^{\infty} \frac{(-1)^k}{k!} t^k \text{ad}_{(X+\lambda Y)}^k Y \equiv \sum_{k=0}^{\infty} f_k t^k.$$

Construiremos  $U$  como una serie de potencias en  $t$ , es decir,

$$U(t, \lambda) = \sum_{n=1}^{\infty} t^n U_n(\lambda) = t U_1(\lambda) + t^2 U_2(\lambda) + \dots$$

y nuestro objetivo ahora es conseguir una recurrencia para los términos  $U_j(\lambda)$ . Es claro que

$$\frac{\partial U}{\partial t}(t, \lambda) = \sum_{k=0}^n (k+1) t^k U_{k+1} + \mathcal{O}(t^{n+1})$$

y

$$\text{ad}_U = t \text{ad}_{U_1} + t^2 \text{ad}_{U_2} + \dots + t^n \text{ad}_{U_n} + \mathcal{O}(t^{n+1}).$$

En general

$$\text{ad}_{U(t, \lambda)}^j = \sum_{l=j}^n t^l \sum_{\substack{k_1 + \dots + k_j = l \\ k_1 \geq 1, \dots, k_j \geq 1}} \text{ad}_{U_{k_1}} \text{ad}_{U_{k_2}} \dots \text{ad}_{U_{k_j}} + \mathcal{O}(t^{n+1}).$$

Por otra parte,

$$\begin{aligned}
d \exp_{-U}^{-1}(F) &= F + \sum_{j=1}^n (-1)^j \frac{B_j}{j!} \text{ad}_U^j(F) + \mathcal{O}(t^{n+1}) \\
&= \sum_{k=0}^n f_k t^k + \sum_{j=1}^n (-1)^j \frac{B_j}{j!} \text{ad}_U^j \left( \sum_{k=0}^{n-1} f_k t^k \right) + \mathcal{O}(t^{n+1})
\end{aligned}$$

Analicemos este último término:

$$\begin{aligned}
\sum_{j=1}^n (-1)^j \frac{B_j}{j!} \text{ad}_U^j \left( \sum_{k=0}^{n-1} f_k t^k \right) &= \sum_{j=1}^n (-1)^j \frac{B_j}{j!} \text{ad}_U^j f_0 + \sum_{j=1}^n (-1)^j \frac{B_j}{j!} \text{ad}_U^j \left( \sum_{k=1}^{n-1} f_k t^k \right) \\
&= \sum_{l=1}^n t^l \sum_{j=1}^l (-1)^j \frac{B_j}{j!} \sum_{\substack{k_1+\dots+k_j=l \\ k_1 \geq 1, \dots, k_j \geq 1}} \text{ad}_{U_{k_1}} \text{ad}_{U_{k_2}} \cdots \text{ad}_{U_{k_j}} f_0 \\
&\quad + \mathcal{O}(t^{n+1}) + \sum_{l=1}^n t^l \sum_{s=1}^l \sum_{j=1}^s (-1)^j \frac{B_j}{j!} \text{ad}_U^j \left( \sum_{k=1}^{n-1} f_k t^k \right) \\
&= \sum_{l=1}^n t^l \sum_{j=1}^l (-1)^j \frac{B_j}{j!} \sum_{\substack{k_1+\dots+k_j=s \\ k_1 \geq 1, \dots, k_j \geq 1}} \text{ad}_{U_{k_1}} \text{ad}_{U_{k_2}} \cdots \text{ad}_{U_{k_j}} f_{l-s} \\
&\quad + \mathcal{O}(t^{n+1}).
\end{aligned}$$

Por lo tanto, igualando potencias en  $t$ , obtenemos

$$\begin{aligned}
U_1 &= f_0 \equiv \text{ad}_{X+\lambda Y}^0 Y = Y \\
(n+1)U_{n+1} &= f_n + \sum_{s=1}^n \sum_{j=1}^s (-1)^j \frac{B_j}{j!} \sum_{\substack{k_1+\dots+k_j=s \\ k_1 \geq 1, \dots, k_j \geq 1}} \text{ad}_{U_{k_1}} \text{ad}_{U_{k_2}} \cdots \text{ad}_{U_{k_j}} f_{n-s}
\end{aligned}$$

o equivalentemente,

$$\begin{aligned}
U_1 &= f_0 \tag{4.27} \\
nU_n &= f_{n-1} + \sum_{l=1}^n \sum_{j=1}^l (-1)^j \frac{B_j}{j!} \sum_{\substack{k_1+\dots+k_j=l \\ k_1 \geq 1, \dots, k_j \geq 1}} \text{ad}_{U_{k_1}} \text{ad}_{U_{k_2}} \cdots \text{ad}_{U_{k_j}} f_{n-1-l}.
\end{aligned}$$

Computacionalmente es más ventajoso introducir los siguientes generadores:

$$\begin{aligned}
S_{n,k}^{(1)} &= [U_{n-1}, f_k], & S_{n,k}^{(n-1)} &= \text{ad}_{U_1}^{n-1} f_k \\
S_{n,k}^{(j)} &= \sum_{m=1}^{n-j} [U_m, S_{n-m,k}^{j-1}], & k &= 0, 1, \dots, n-2.
\end{aligned} \tag{4.28}$$

Entonces,

$$U_n = \frac{1}{n} \left( f_{n-1} + \sum_{l=1}^{n-1} \sum_{j=1}^l (-1)^j \frac{B_j}{j!} S_{l+1, n-1-l}^{(j)} \right), \quad n \geq 2. \tag{4.29}$$

Hemos puesto en marcha esta recurrencia en *Mathematica*<sup>TM</sup> y calculado hasta el término  $U_{14}(\lambda)$ . Reproducimos los resultados obtenidos mediante la aplicación de la fórmula BCH (primer procedimiento). En general se reduce el tiempo de cálculo y los recursos de memoria que requiere el uso de las recurrencias (4.29) son menos estrictos. Para la comparación, se incluyen en la siguiente tabla estos datos, obtenidos con un computador personal con 2 GB de RAM.

$n$	Tiempo de CPU (segundos)		Memoria (MBytes)	
	<i>BCH</i>	<i>Rec</i>	<i>BCH</i>	<i>Rec</i>
12	11.54	10.92	59.49	27.01
13	44.28	30.85	129.49	37.33
14	190.91	93.37	478.03	65.08

Cuadro 4.1: Tiempo de CPU y memoria necesarios para el cómputo de los términos  $U_1, U_2, \dots, U_n$  hasta el valor determinado de  $n$ , utilizando la fórmula de Baker–Campbell–Hausdorff (BCH) y la recurrencia (4.29) (*Rec*). Se aprecia que esta última recurrencia es más eficiente al crecer el  $n$ , es decir, requiere de menos recursos de tiempo de CPU y memoria, frente al mismo input.

A manera de ilustración, los primeros términos generados por (4.29) son:

$$\begin{aligned}
U_1(\lambda) &= \lambda Y \\
U_2(\lambda) &= -\frac{1}{2}\lambda[X, Y] \\
U_3(\lambda) &= \frac{1}{6}\lambda[X, [X, Y]] + \frac{1}{12}\lambda^2[Y, [X, Y]] \\
U_4(\lambda) &= -\frac{1}{24}\lambda[X, [X, [X, Y]]] - \frac{1}{24}\lambda^2[Y, [X, [X, Y]]] \\
U_5(\lambda) &= \frac{1}{120}\lambda[X, [X, [X, [X, Y]]]] + \frac{1}{80}\lambda^2[Y, [X, [X, [X, Y]]]] + \\
&\quad \frac{1}{720}\lambda^3[Y, [Y, [X, [X, Y]]]] - \frac{1}{720}\lambda^4[Y, [Y, [Y, [X, Y]]]] + \\
&\quad \frac{1}{120}\lambda^2[[X, Y], [X, [X, Y]]] - \frac{1}{240}\lambda^3[[X, Y], [Y, [X, Y]]]
\end{aligned}$$

Reordenando la serie y tomando  $t = 1$  podemos escribir  $U(\lambda) = \lambda u_1 + \lambda^2 u_2 + \dots$ , es decir, resolvemos el problema planteado por Cernusca. De esta manera

tenemos

$$\begin{aligned}
u_1 &= Y - \frac{1}{2}[X, Y] + \frac{1}{6}[X, [X, Y]] - \frac{1}{24}[X, [X, [X, Y]]] + \dots \\
&\quad + \frac{1}{120}[X, [X, [X, [X, Y]]]] + \dots \\
u_2 &= \frac{1}{12}[Y, [X, Y]] - \frac{1}{24}[Y, [X, [X, Y]]] + \frac{1}{80}[Y, [X, [X, [X, Y]]]] + \\
&\quad \frac{1}{120}[[X, Y], [X, [X, Y]]] + \dots \\
u_3 &= \frac{1}{720}[Y, [Y, [X, [X, Y]]]] - \frac{1}{240}[[X, Y], [Y, [X, Y]]] + \dots \quad (4.30)
\end{aligned}$$

donde cada término  $u_i$  contiene infinitos términos. La pregunta es entonces la siguiente: ¿existe alguna fórmula cerrada para  $u_i$  proporcional a cada potencia de  $\lambda$ ?

Para resolver este problema, tomamos de nuevo como punto de partida nuestra ecuación (4.25), que ahora se escribe como

$$e^{t(X+\lambda Y)} = e^{tX} V, \quad (4.31)$$

y derivamos la ecuación diferencial satisfecha por  $V$ :

$$\frac{dV}{dt} = \lambda e^{-tX} Y e^{tX} V \equiv \lambda \tilde{Y}(t) V, \quad V(0) = I. \quad (4.32)$$

A continuación aplicamos la expansión Magnus a (4.32), por lo que  $V$  se puede escribir como la exponencial de una serie de potencias en  $\lambda$ :

$$V(t, \lambda) = e^{U(t, \lambda)}, \quad U(t, \lambda) = \sum_{k=1}^{\infty} \lambda^k u_k(t).$$

Los primeros términos de esta serie están dados por

$$\begin{aligned}
u_1(t) &= \int_0^t \tilde{Y}(t_1) dt_1, \\
u_2(t) &= \frac{1}{2} \int_0^t dt_1 \int_0^{t_1} dt_2 [\tilde{Y}(t_1), \tilde{Y}(t_2)] \\
u_3(t) &= \frac{1}{6} \int_0^t dt_1 \int_0^{t_1} dt_2 \int_0^{t_2} dt_3 ([\tilde{Y}(t_1), [\tilde{Y}(t_2), \tilde{Y}(t_3)]] + [\tilde{Y}(t_3), [\tilde{Y}(t_2), \tilde{Y}(t_1)]])
\end{aligned} \quad (4.33)$$

Estos son de hecho las expresiones cerradas que estábamos buscando. Para verificarlo, examinemos la primera:

$$u_1(t) = \int_0^t \tilde{Y}(t_1) dt_1 = \int_0^t e^{-t_1 X} Y e^{t_1 X} dt_1 = \int_0^t e^{-s \text{ad}_X} Y ds = \frac{1 - e^{t \text{ad}_X}}{\text{ad}_X} Y \quad (4.34)$$

y esta expresión es la serie de potencias formales cuyos primeros términos se incluyen en la primera fórmula de (4.30).

### 4.3.1. Ejemplos Numéricos

Para ejemplificar el comportamiento numérico de la recurrencia para  $U_k$  presentada en (4.29), tomamos las matrices

$$X = \begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \quad (4.35)$$

y comparamos  $\exp(X + \lambda Y)$  con  $\exp(X)\exp(U^{[k]})$  donde  $U^{[k]} \equiv \sum_{i=1}^k \lambda^i U_i$ . Asumiendo como una mejor aproximación el resultado  $\eta = \expm(X + \lambda Y)$ , dado por **Matlab**, la expresión para el *error relativo* será:

$$Er_k = \frac{\|\eta - e^X e^{U^{[k]}}\|}{\|\eta\|}.$$

El resultado de computar  $Er_k$  para  $k = 1, \dots, 20$ , se aprecia en la figura 4.1 para  $\lambda = 1$ . A medida que se añaden términos a la serie  $U(\lambda)$  se logra un menor error relativo de la misma, es decir, va aumentando su precisión.

Ahora realizamos otra experiencia numérica, esta vez con matrices  $3 \times 3$  del tipo

$$X = \begin{bmatrix} \alpha & a/2 & b \\ -2a & \beta & 0 \\ b & -c & -\delta \end{bmatrix}, \quad Y = \begin{bmatrix} -\alpha & a/2 & 0 \\ a & -\beta & c \\ -2b & 0 & \delta \end{bmatrix}. \quad (4.36)$$

Tomamos  $A = X + \lambda Y$ , de manera que  $A \in \mathfrak{so}(3)$  y

$$e^A = \exp \left( \begin{bmatrix} 0 & a & b \\ -a & 0 & c \\ -b & -c & 0 \end{bmatrix} \right) = I + \frac{\sin \sigma}{\sigma} A + \frac{1 - \cos \sigma}{\sigma^2} A^2,$$

donde  $\sigma = \sqrt{a^2 + b^2 + c^2}$ . Elegimos  $\lambda = 1$ , y las demás constantes  $\alpha, \beta, \delta, a, b, c$  toman valores aleatorios entre 0 y 1.

Al igual que en el caso anterior, graficamos, figura 4.2, los errores relativos, es decir,  $Er_k$  para  $k = 1, \dots, 20$ . Nuevamente tenemos la misma pendiente de decrecimiento del error relativo de la serie, con respecto a la cantidad de términos de la misma.

Como conclusión de estas pruebas numéricas, hallamos que para el problema propuesto por Cernusca las dos experiencias estudiadas tienen un resultado muy preciso al utilizar la recurrencia obtenida en (4.29), lo que muestra su efectividad en este tipo de casos.



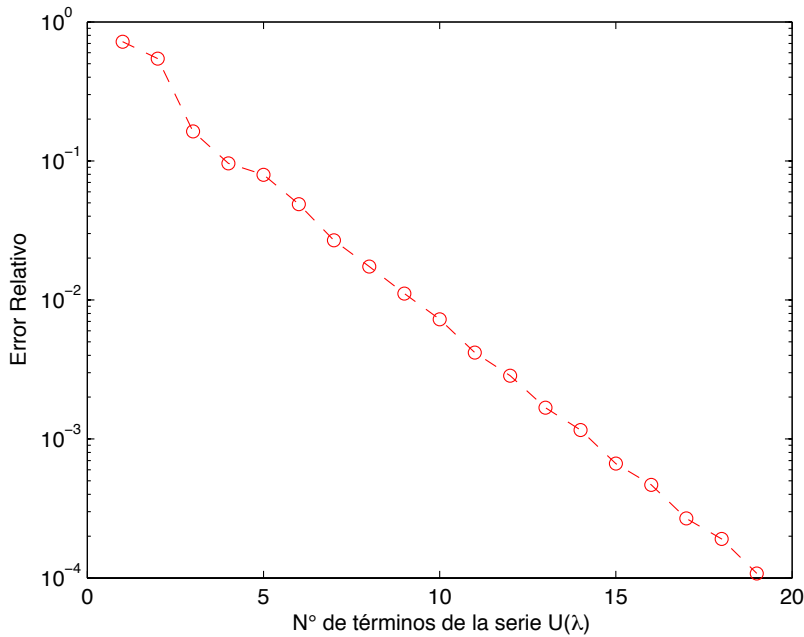


Figura 4.1: Gráfica error relativo para el problema propuesto por Cernusca. Se utilizó la función `expm()`, que dispone **Matlab**, como referencia del resultado exacto. Con parámetro  $\lambda = 1$  y las matrices  $X$  e  $Y$  de (4.35).

## 4.4. Fórmula de Zassenhaus

En el artículo [CMN12] se desarrolla un nuevo procedimiento para generar los términos de la serie de Zassenhaus. También se analiza en términos generales las cotas que deben cumplir las matrices para garantizar su convergencia. En el resto del capítulo se desarrolla los elementos necesarios para explicar este procedimiento innovador de la fórmula de Zassenhaus, así como un estudio empírico sobre lo práctico del algoritmo con matrices cercanas a las cotas que deben cumplir las matrices. Para ello se desarrollaron códigos en **Matlab** cuyos resultados se comentan. Además se presenta el desarrollo de la fórmula de Zassenhaus generalizada, con su estudio de la región de convergencia asociada y finalmente se presenta en detalle el desarrollo de Wilcox, que resulta ser el análogo continuo de Zassenhaus.

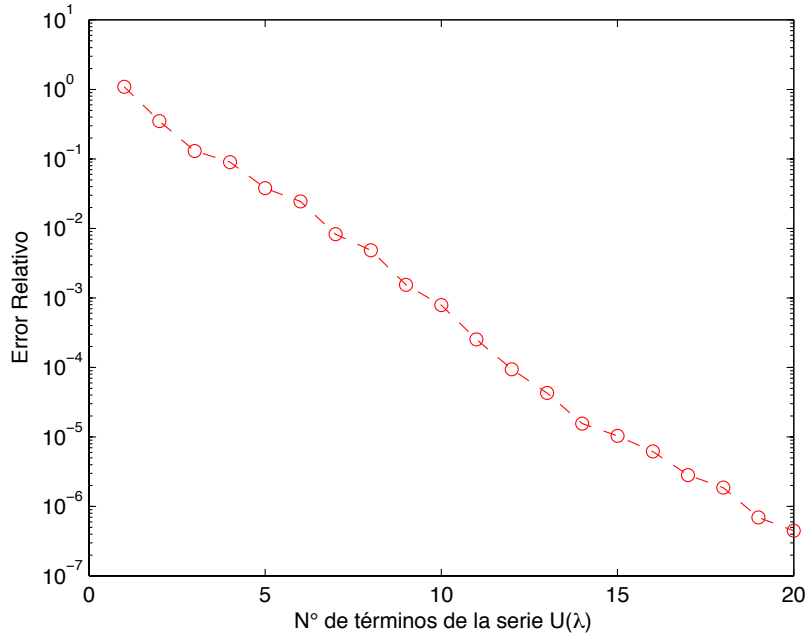


Figura 4.2: Gráfica error relativo para el problema propuesto por Cernusca. Con parámetro  $\lambda = 1$  y las matrices  $X$  e  $Y$  de (4.36). Se puede apreciar que el error relativo disminuye más que en el ejemplo previo, debido a que en este caso la solución de referencia es la exacta, y no una aproximación.

#### 4.4.1. Introducción

En el artículo sobre la expansión que lleva su nombre, Magnus [Mag54] cita una referencia inédita de Zassenhaus, comentando que existe una fórmula que puede ser llamada dual de la fórmula de Baker–Campbell–Hausdorff. Este resultado puede expresarse como sigue.

**Teorema 10** (*Fórmula de Zassenhaus*). Sea  $\mathcal{L}(X, Y)$  el álgebra de Lie libre generada por  $X$  e  $Y$ . Entonces  $e^{X+Y}$  se puede descomponer de manera única como

$$e^{X+Y} = e^X e^Y \prod_{n=2}^{\infty} e^{C_n(X,Y)} = e^X e^Y e^{C_2(X,Y)} e^{C_3(X,Y)} \dots e^{C_n(X,Y)} \dots, \quad (4.37)$$

donde  $C_n(X, Y) \in \mathcal{L}(X, Y)$  es un polinomio homogéneo de Lie en  $X$  e  $Y$  de grado  $n$ .

La existencia de tal fórmula es una consecuencia inmediata del teorema BCH. De hecho, está claro que  $e^{-X} e^{X+Y} = e^{Y+D}$ , donde  $D$  involucra

polinomios de Lie de grado  $> 1$ . Ahora  $e^{-Y}e^{Y+D} = e^{C_2+\tilde{D}}$ , donde  $\tilde{D}$  involucra polinomios de Lie de grado  $> 2$  y el proceso se repite de nuevo. El resultado general se obtiene entonces por inducción.

Mediante la comparación con la fórmula BCH es posible obtener los primeros términos de la fórmula (4.37) como

$$C_2(X, Y) = -\frac{1}{2}[X, Y], \quad C_3(X, Y) = \frac{1}{3}[Y, [X, Y]] + \frac{1}{6}[X, [X, Y]].$$

Aunque menos conocida que la fórmula BCH, la fórmula Zassenhaus constituye, sin embargo, una herramienta estándar en varios campos, por ejemplo, en mecánica estadística, teorías de muchos cuerpos, óptica cuántica,  $q$ -análisis en grupos cuánticos, etc. También en la teoría de aceleradores de partículas, Dragt y sus colaboradores han utilizado la fórmula de Zassenhaus [DF76].

En la literatura se pueden encontrar diversos cálculos sistemáticos de los términos  $C_n$  para  $n > 3$  en la fórmula de Zassenhaus. En particular, en [Wil67] se presenta un procedimiento recursivo que ha sido posteriormente utilizado para obtener expresiones explícitas hasta  $C_6$  en términos de conmutadores anidados. Por otro lado, Volkin [Vol68] propuso una técnica general para expresar una función general de operadores no conmutativos como un desarrollo de conmutadores (de orden cada vez más alto) de los operadores involucrados. De esta forma, fue capaz de obtener fórmulas recursivas hasta  $C_9$ . Siguiendo una idea ya sugerida por Wilcox en [Wil67], Suzuki [Suz77] obtuvo los términos sucesivos  $C_n(X, Y)$  en

$$e^{\lambda(X+Y)} = e^{\lambda X} e^{\lambda Y} e^{\lambda^2 C_2} e^{\lambda^3 C_3} \dots \quad (4.38)$$

mediante la diferenciación de ambos lados con respecto a  $\lambda$  y evaluando en  $\lambda = 0$  después de cada diferenciación. De este modo

$$\begin{aligned} C_2 &= \frac{1}{2} \left( \frac{d^2}{d\lambda^2} (e^{-\lambda Y} e^{-\lambda X} e^{\lambda(X+Y)}) \right)_{\lambda=0} = \frac{1}{2}[Y, X] \\ C_3 &= \frac{1}{3!} \left( \frac{d^3}{d\lambda^3} (e^{-\lambda^2 C_2} e^{-\lambda Y} e^{-\lambda X} e^{\lambda(X+Y)}) \right)_{\lambda=0} = \frac{1}{3}[C_2, X + 2Y] \end{aligned}$$

y en general, para  $n \geq 3$ ,

$$C_n = \frac{1}{n!} \left( \frac{d^n}{d\lambda^n} (e^{-\lambda^{n-1} C_{n-1}} \dots e^{-\lambda^2 C_2} e^{-\lambda Y} e^{-\lambda X} e^{\lambda(X+Y)}) \right)_{\lambda=0}. \quad (4.39)$$

Por último, Baues [Bau81] dio fórmulas explícitas para los términos de Zassenhaus utilizando elementos de la teoría de homotopías.

Todas estas propuestas dan resultados para  $C_n$  como una combinación lineal de conmutadores anidados. Por contra, Scholz y Weyrauch [SW06] propusieron un procedimiento recursivo basado en matrices triangulares superiores que se pueden implementar fácilmente en un paquete de álgebra simbólica. En este caso, sin embargo, las expresiones para  $C_n$  no están escritas explícitamente en términos de polinomios homogéneos de Lie. Más recientemente [WS09], los mismos autores han aplicado una técnica relacionada con una utilizada anteriormente por Witschel [Wit75] para obtener  $C_n$  hasta  $n = 15$  en menos de 2 minutos de tiempo de CPU. Aquí también los términos Zassenhaus se escriben como

$$C_n = \sum_{w, |w|=n} g_w w, \quad (4.40)$$

donde  $g_w$  es un coeficiente racional y la suma se toma sobre todas las palabras o cadenas  $w$  con longitud  $|w| = n$  con los símbolos  $X$  e  $Y$ , es decir,  $w = a_1 a_2 \cdots a_n$ , cada  $a_i$  es  $X$  o  $Y$ . Como sucedía con la fórmula BCH, se puede aplicar el teorema de Dynkin-Specht-Wever [Jac79], y expresar  $C_n$  como

$$C_n = \frac{1}{n} \sum_{w, |w|=n} g_w [w], \quad (4.41)$$

es decir, los términos individuales son los mismos que en la serie asociativa (4.40) excepto que la palabra  $w = a_1 a_2 \cdots a_n$  se sustituye por el conmutador anidado por la derecha  $[w] = [a_1, [a_2, \dots [a_{n-1}, a_n] \dots]]$  y el coeficiente  $g_w$  está dividida por la longitud  $n$ . De esta manera  $C_n$  se construye como una combinación lineal de conmutadores anidados homogéneos de grado  $n$ , es decir, como una combinación lineal de los elementos del subespacio homogéneo  $\mathcal{L}(X, Y)_n$  de grado  $n$  del álgebra de Lie libre  $\mathcal{L}(X, Y)$ .

En la referencia [CMN12] se presenta una nueva recurrencia que permite expresar los términos de Zassenhaus  $C_n$  directamente como una combinación lineal de elementos independientes del subespacio homogéneo  $\mathcal{L}(X, Y)_n$ . En otras palabras, el procedimiento, que puede ser implementado fácilmente en un paquete de álgebra simbólica, da  $C_n$  hasta un grado prescrito directamente en términos de conmutadores independientes que implican  $n$  operadores  $X$  e  $Y$ . De esta manera, no es necesario ningún proceso de reescritura en una base  $\mathcal{L}(X, Y)$ , ahorrando considerable tiempo de cálculo y recursos de memoria. Por otra parte, los  $C_n$  se expresan directamente con un número mínimo de conmutadores requeridos en cada grado  $n$ .

El segundo aspecto analizado en [CMN12] se refiere a la convergencia de la fórmula de Zassenhaus en un álgebra de Banach. A este respecto, sólo había dos resultados previos en la literatura. El primero de ellos lo obtuvo

Suzuki [Suz77] a partir de la recurrencia (4.39). Específicamente, fue capaz de demostrar que si  $|\lambda|(\|X\| + \|Y\|) \leq \log 2 - 1/2$ , entonces el producto infinito (4.38) converge. Posteriormente, Bayen [Bay79] generaliza el análisis, demostrando que el producto (4.38) converge si  $|\lambda|(\|X\| + \|Y\|) \leq r$ , donde  $r \approx 0.596705$  es una raíz de una cierta ecuación trascendental. En [CMN12], siguiendo la metodología de Bayen, se obtuvieron límites más precisos para los términos de la fórmula Zassenhaus que muestran que el producto (4.37) converge en un dominio más extenso.

Una observación simple pero importante es la siguiente: en algunas aplicaciones se considera la fórmula Zassenhaus ‘orientada a la izquierda’

$$e^{X+Y} = \dots e^{\hat{C}_4(X,Y)} e^{\hat{C}_3(X,Y)} e^{\hat{C}_2(X,Y)} e^Y e^X \quad (4.42)$$

en vez de (4.37). Es fácil ver entonces que los exponentes  $\hat{C}_i$  y  $C_i$  se relacionan como

$$\hat{C}_i(X, Y) = (-1)^{i+1} C_i(X, Y), \quad i \geq 2$$

y así podemos limitarnos a analizar la fórmula ‘orientada a la derecha’ (4.37).

#### 4.4.2. Construcción de los términos de Zassenhaus. Una nueva recurrencia

Para derivar un procedimiento recursivo es conveniente utilizar un parámetro  $\lambda$  como en (4.38),

$$e^{\lambda(X+Y)} = e^{\lambda X} e^{\lambda Y} e^{\lambda^2 C_2} e^{\lambda^3 C_3} e^{\lambda^4 C_4} \dots \quad (4.43)$$

Cuando se hace  $\lambda = 1$ , se tiene la fórmula clásica. Ahora se considera la composición

$$R_1(\lambda) = e^{-\lambda X} e^{-\lambda Y} e^{\lambda(X+Y)}, \quad n \geq 2. \quad (4.44)$$

Generalizando, definimos para  $n \geq 2$

$$R_n(\lambda) = e^{-\lambda^n C_n} \dots e^{-\lambda^2 C_2} e^{-\lambda X} e^{-\lambda Y} e^{\lambda(X+Y)} = e^{-\lambda^n C_n} R_{n-1}(\lambda), \quad (4.45)$$

que también se puede expresar como

$$R_n(\lambda) = e^{\lambda^{n+1} C_{n+1}} e^{\lambda^{n+2} C_{n+2}} \dots \quad (4.46)$$

A continuación introducimos

$$F_n(\lambda) \equiv \left( \frac{d}{d\lambda} R_n(\lambda) \right) R_n(\lambda)^{-1}, \quad n \geq 1, \quad (4.47)$$

donde  $F_n(\lambda)$ , es una serie de elementos del álgebra de Lie libre  $\mathcal{L}(X, Y)$ .

Una adecuada manipulación algebraica de (4.45) nos permite obtener, para  $n \geq 2$

$$\begin{aligned}
F_n(\lambda) &= -n C_n \lambda^{n-1} + e^{-\lambda^n C_n} \left( \frac{d}{d\lambda} R_{n-1}(\lambda) \right) R_{n-1}(\lambda)^{-1} e^{\lambda^n C_n} \\
&= -n C_n \lambda^{n-1} + e^{-\lambda^n C_n} F_{n-1}(\lambda) e^{\lambda^n C_n} \\
&= -n C_n \lambda^{n-1} + e^{-\lambda^n \text{ad}_{C_n}} F_{n-1}(\lambda) \\
&= e^{-\lambda^n \text{ad}_{C_n}} (F_{n-1}(\lambda) - n C_n \lambda^{n-1}). \tag{4.48}
\end{aligned}$$

Por otra parte, diferenciando la expresión (4.46) con respecto a  $\lambda$  y usando (4.47), se tiene

$$F_n(\lambda) = (n+1) C_{n+1} \lambda^n + \sum_{j=n+2}^{\infty} j \lambda^{j-1} e^{\lambda^{n+1} \text{ad}_{C_{n+1}}} \dots e^{\lambda^{j-1} \text{ad}_{C_{j-1}}} C_j. \tag{4.49}$$

O bien,

$$F_n(\lambda) = (n+1) C_{n+1} \lambda^n + G_{n+1}(\lambda), \quad n \geq 1,$$

donde  $G_{n+1}^{(k)}(0) = 0, \forall k \geq 0$ , y para  $n \geq 1$ . En consecuencia, tenemos para  $n \geq 1$ ,

$$F_{n+1}(\lambda) = e^{-\lambda^{n+1} \text{ad}_{C_{n+1}}} G_{n+1}(\lambda), \tag{4.50}$$

$$C_{n+1} = \frac{1}{(n+1)!} F_n^{(n)}(0), \tag{4.51}$$

$$G_{n+1}(\lambda) = F_n(\lambda) - \frac{\lambda^n}{n!} F_n^{(n)}(0). \tag{4.52}$$

Las expresiones (4.50)-(4.52) pueden ser calculadas recursivamente para hallar los términos  $C_n$  partiendo de  $F_1(\lambda)$ . La secuencia es

$$F_n(\lambda) \longrightarrow C_{n+1} \longrightarrow G_{n+1}(\lambda) \longrightarrow F_{n+1}(\lambda) \longrightarrow \dots, \quad n \geq 1.$$

Vamos a analizar con más detalle este procedimiento, con el objetivo de proporcionar un algoritmo optimizado desde el punto de vista computacional. Para  $n = 1$ , y teniendo en cuenta (4.44), formamos

$$\begin{aligned}
F_1(\lambda) &= -Y - e^{-\lambda Y} X e^{\lambda Y} + e^{-\lambda Y} e^{-\lambda X} (X + Y) e^{\lambda X} e^{\lambda Y} \\
&= -Y - e^{-\lambda \text{ad}_Y} X + e^{-\lambda \text{ad}_Y} e^{-\lambda \text{ad}_X} (X + Y) \\
&= e^{-\lambda \text{ad}_Y} (e^{-\lambda \text{ad}_X} - I) Y.
\end{aligned}$$

Esto da

$$F_1(\lambda) = \sum_{i=0}^{\infty} \sum_{j=1}^{\infty} \frac{(-\lambda)^{i+j}}{i! j!} \text{ad}_Y^i \text{ad}_X^j Y \tag{4.53}$$

o equivalentemente

$$F_1(\lambda) = \sum_{k=1}^{\infty} f_{1,k} \lambda^k, \quad \text{con} \quad f_{1,k} = \sum_{j=1}^k \frac{(-1)^k}{j!(k-j)!} \text{ad}_Y^{k-j} \text{ad}_X^j Y. \quad (4.54)$$

En general, a partir de (4.48) un cálculo sencillo muestra que para  $n \geq 2$ ,

$$F_n(\lambda) = \sum_{k=n}^{\infty} f_{n,k} \lambda^k, \quad \text{con} \quad f_{n,k} = \sum_{j=0}^{[k/n]-1} \frac{(-1)^j}{j!} \text{ad}_{C_n}^j f_{n-1,k-nj}, \quad k \geq n. \quad (4.55)$$

Aquí  $[k/n]$  denota la parte entera de  $k/n$ . Por otra parte, un examen más detenido de (4.49) revela que

$$\begin{aligned} F_n(\lambda) &= (n+1)C_{n+1}\lambda^n + (n+2)e^{\lambda^{n+1}\text{ad}_{C_{n+1}}}C_{n+2}\lambda^{n+1} + \dots \\ &= (n+1)C_{n+1}\lambda^n + (n+2)C_{n+2}\lambda^{n+1} + \dots \\ &\quad + (2n+2)C_{2n+2}\lambda^{2n+1} + \lambda^{2n+2}[C_{n+1}, C_{n+2}] + \dots \\ &= \sum_{k=n+1}^{2n+2} k C_k \lambda^{k-1} + \lambda^{2n+2} H_n(\lambda), \end{aligned} \quad (4.56)$$

donde  $H_n(\lambda)$  involucra conmutadores de  $C_j, j \geq n+1$ . Se debe tener en cuenta que los términos  $C_{n+1}, \dots, C_{2n+2}$  de la fórmula Zassenhaus pueden entonces obtenerse directamente de  $F_n(\lambda)$ . En particular,

$$C_{n+1} = \frac{1}{n+1} f_{1,n} = \frac{1}{n+1} \sum_{i=0}^{n-1} \frac{(-1)^n}{i!(n-j)!} \text{ad}_Y^i \text{ad}_X^{n-j} Y, \quad \text{para } n = 1, 2, 3. \quad (4.57)$$

Explícitamente,

$$\begin{aligned} C_2 &= -\frac{1}{2} [X, Y], \\ C_3 &= \frac{1}{3} [Y, [X, Y]] + \frac{1}{6} [X, [X, Y]], \\ C_4 &= -\frac{1}{8} ([Y, [Y, [X, Y]]] + [Y, [X, [X, Y]]]) - \frac{1}{24} [X, [X, [X, Y]]]. \end{aligned}$$

Al comparar (4.55) con (4.56) se obtiene la expresión general

$$C_{n+1} = \frac{1}{n+1} f_{[n/2],n} \quad n \geq 5, \quad (4.58)$$

donde la expresión  $f_{n,k}$  se puede obtener recursivamente por (4.55).

A partir de las ideas expuestas, se desarrolla un procedimiento que permite el cálculo eficiente de los polinomios homogéneos  $C_n, n \geq 2$ , de la serie de Zassenhaus (4.37).

### 4.4.3. Procedimiento

El método requiere de los siguientes pasos algorítmicos:

#### Procedimiento de Zassenhaus

---

- **Input:** Operadores  $X$  e  $Y$ .
- **Output:** Los operadores  $C_1, C_2, \dots$ .
- $f_{1,k} = \sum_{j=1}^k \frac{(-1)^k}{j!(k-j)!} \text{ad}_Y^{k-j} \text{ad}_X^j Y$ .
- $f_{n,k} = \sum_{j=0}^{\lfloor k/n \rfloor - 1} \frac{(-1)^j}{j!} \text{ad}_{C_n}^j f_{n-1, k-nj}$ .

Luego:

- $C_n = \frac{1}{n} f_{1, n-1}, \quad n = 2, 3, 4$
  - $C_n = \frac{1}{n} f_{\lfloor (n-1)/2 \rfloor, n-1} \quad n \geq 5$ .
- 

Esto constituye una nueva forma recursiva para obtener directamente el término  $C_n$  como un polinomio Lie homogéneo en  $X$  e  $Y$  de grado  $n$  arbitrariamente alto, que puede ser implementado fácilmente con un paquete de álgebra simbólica.

### 4.4.4. Independencia Lineal

El algoritmo anterior, o equivalentemente el procedimiento dado por las identidades (4.50)-(4.52) proporcionan expresiones para  $C_n$  que, por construcción, involucran sólo conmutadores independientes. En otras palabras, no se puede simplificar más mediante el uso de la identidad de Jacobi y la propiedad de antisimetría del conmutador.

Para probar esta afirmación, es conveniente obtener una expresión más explícita de  $F_n(\lambda)$  y  $C_{n+1}$  a partir de (4.50)-(4.52). Para ello, se tiene en cuenta que para  $n \geq 1$  los conjuntos  $\mathcal{J}_n$  y  $\mathcal{I}_n$  de  $(n+1)$ -tuplas de números enteros no negativos recursivamente definidas de la siguiente manera:

$$\begin{aligned} \mathcal{J}_1 &= \{(i_0, i_1) \in \mathbb{N}^2 : i_0 \geq 1\}, \\ \mathcal{I}_n &= \{(i_0, i_1, \dots, i_n) \in \mathcal{J}_n : i_0 + i_1 + 2i_2 + \dots + ni_n = n\}, \\ \mathcal{J}_{n+1} &= (\mathcal{J}_n \setminus \mathcal{I}_n) \times \mathbb{N}. \end{aligned}$$



El conjunto  $\mathcal{I}_n$  se puede definir directamente como el conjunto de las  $(n+1)$ -tuplas de números enteros no negativos que satisfacen  $i_0 + i_1 + 2i_2 + \cdots + ni_n = n$  y

$$j+1 \leq i_0 + i_1 + 2i_2 + \cdots + ji_j \quad \text{para } j = 0, \dots, n-1. \quad (4.59)$$

Entonces, en particular  $\mathcal{I}_1 = \{(1, 0)\}$ ,  $\mathcal{I}_2 = \{(1, 1, 0), (2, 0, 0)\}$ , etc. Observamos, que por construcción, cada  $(i_0, i_1, \dots, i_n) \in \mathcal{I}_n$  es tal que  $i_m = 0$  si  $m > n/2$ .

Con (4.50)-(4.52), se puede demostrar por inducción sobre  $n$  que, para  $n \geq 1$ ,

$$\begin{aligned} F_n(\lambda) &= \sum_{(i_0, i_1, \dots, i_n) \in \mathcal{I}_n} \frac{(-1)^{i_0 + \cdots + i_n} \lambda^{i_0 + i_1 + 2i_2 + \cdots + ni_n}}{i_0! i_1! \cdots i_n!} \text{ad}_{C_n}^{i_n} \cdots \text{ad}_{C_2}^{i_2} \text{ad}_Y^{i_1} \text{ad}_X^{i_0} Y, \\ C_{n+1} &= \frac{1}{n+1} \sum_{(i_0, i_1, \dots, i_n) \in \mathcal{I}_n} \frac{(-1)^{i_0 + \cdots + i_n}}{i_0! i_1! \cdots i_n!} \text{ad}_{C_n}^{i_n} \cdots \text{ad}_{C_2}^{i_2} \text{ad}_Y^{i_1} \text{ad}_X^{i_0} Y. \end{aligned} \quad (4.60)$$

De hecho, esto es claramente cierto para  $n = 1$  (ecuaciones (4.53) y (4.57), respectivamente), mientras que la aplicación sucesiva de (4.50)-(4.52) conduce al resultado general.

Ahora, aplicando repetidamente el principio de eliminación de Lazard [C.R93], junto con  $\mathcal{I}_1 = \{(1, 0)\}$ ,  $\{C_2\} = \{-\frac{1}{2}[X, Y]\} = \{-\frac{1}{2}\text{ad}_Y^{i_1} \text{ad}_X^{i_0} Y : (i_0, i_1) \in \mathcal{I}_1\}$ , muestra que, como un espacio vectorial,

$$\begin{aligned} \mathcal{L}(X, Y) &= \text{spn}\{X\} \oplus \mathcal{L}(\{\text{ad}_X^j Y : j \geq 0\}) \\ &= \text{spn}\{X\} \oplus \text{spn}\{Y\} \oplus \mathcal{L}(\{\text{ad}_Y^i \text{ad}_X^j Y : i \geq 0, j \geq 1\}) \\ &= \text{spn}\{X, Y\} \oplus \mathcal{L}(\{\text{ad}_Y^{i_1} \text{ad}_X^{i_0} Y : (i_0, i_1) \in \mathcal{I}_1\}) \\ &= \text{spn}\{X, Y\} \oplus \mathcal{L}(\{C_2\} \cup \{\text{ad}_Y^{i_1} \text{ad}_X^{i_0} Y : (i_0, i_1) \in \mathcal{I}_1 \setminus \mathcal{I}_1\}) \\ &= \text{spn}\{X, Y, C_2\} \oplus \mathcal{L}(\{\text{ad}_{C_2}^{i_2} \text{ad}_Y^{i_1} \text{ad}_X^{i_0} Y : (i_0, i_1, i_2) \in \mathcal{I}_2\}). \end{aligned}$$

Más en general, la aplicación de la eliminación de Lazard junto con (4.60) da

$$\begin{aligned} \mathcal{L}(X, Y) &\subset \text{spn}\{X, Y, C_2, \dots, C_n\} \\ &\quad \oplus \mathcal{L}(\{\text{ad}_{C_n}^{i_n} \cdots \text{ad}_{C_2}^{i_2} \text{ad}_Y^{i_1} \text{ad}_X^{i_0} Y : (i_0, \dots, i_n) \in \mathcal{I}_n\}) \\ &\subset \text{spn}\{X, Y, C_2, \dots, C_n\} \\ &\quad \oplus \mathcal{L}(\{C_{n+1}\} \cup \{\text{ad}_{C_n}^{i_n} \cdots \text{ad}_{C_2}^{i_2} \text{ad}_Y^{i_1} \text{ad}_X^{i_0} Y : (i_0, \dots, i_n) \in \mathcal{I}_n \setminus \mathcal{I}_n\}) \\ &\subset \text{spn}\{X, Y, C_2, \dots, C_{n+1}\} \\ &\quad \oplus \mathcal{L}(\{\text{ad}_{C_{n+1}}^{i_{n+1}} \cdots \text{ad}_{C_2}^{i_2} \text{ad}_Y^{i_1} \text{ad}_X^{i_0} Y : (i_0, \dots, i_{n+1}) \in \mathcal{I}_{n+1}\}). \end{aligned}$$

En consecuencia, los términos  $\{\text{ad}_{C_m}^{i_m} \cdots \text{ad}_{C_2}^{i_2} \text{ad}_Y^{i_1} \text{ad}_X^{i_0} Y : (i_0, i_1, \dots, i_m) \in \mathcal{I}_n\}$  son linealmente independientes en el álgebra de Lie libre  $\mathcal{L}(X, Y)$  y lo mismo es cierto para la representación (4.60) de los términos de Zassenhaus.

#### 4.4.5. Aspectos computacionales

Se ha implementado el procedimiento recursivo descrito en la sección 4.4.2 en *Mathematica*<sup>TM</sup> de la siguiente manera

```

Clear[Cmt, ad, ff, cc];
$RecursionLimit= 1024;
Cmt[a_, a_]:= 0;
Cmt[a___, 0, b___]:= 0;
Cmt[a___, c_ + d_, b___] := Cmt[a, c, b] + Cmt[a, d, b];
Cmt[a___, n_ c_Cmt, b___]:= n Cmt[a, c, b];
Cmt[a___, n_ X, b___]:= n Cmt[a, X, b];
Cmt[a___, n_ Y, b___]:= n Cmt[a, Y, b];
Cmt /: Format[Cmt[a_, b_]:= SequenceForm["[" , a, "," , b, "]""];

ad[a_, 0, b_]:= b;
ad[a_, j_Integer, b_]:= Cmt[a, ad[a, j-1, b]];
ff[1, k_]:= ff[1, k] =
    Sum[((-1)^k/(j! (k-j)!)) ad[Y, k-j, ad[X, j, Y]], {j, 1, k}];
cc[2] = (1/2) ff[1, 1];
ff[p_, k_]:= ff[p, k] =
    Sum[((-1)^j/j!) ad[cc[p], j, ff[p-1, k - p j]], {j, 0,
        IntegerPart[k/p] - 1}];
cc[p_Integer]:= cc[p] =
    Expand[(1/p) ff[IntegerPart[(p-1)/2], p-1]];

```

El objeto  $Cmt[x_1, x_2, \dots, x_{n-1}, x_n]$  representa el conmutador anidado  $[x_1, [x_2, \dots [x_{n-1}, x_n] \dots]]$ . Sólo se dotado de la propiedad de linealidad (no hay necesidad de agregar a dicho objeto la propiedad de antisimetría y la identidad de Jacobi). El símbolo  $ad$  representa el operador adjunto y sus potencias  $ad_a^j b$ , mientras que  $ff[1, k]$ ,  $ff[p, k]$  y  $cc[p]$  corresponden a expresiones (4.54), (4.55) y (4.58), respectivamente. Procediendo de esta manera, hemos obtenido las expresiones de  $C_n$  hasta  $n = 20$  con un razonable consumo de tiempo y memoria de cálculo. Así, por ejemplo, la construcción de los términos hasta grado  $n = 20$  con una computadora personal (2,4 GHz Intel Core 2 Duo con 2 GBytes de RAM) tarda menos de 20 segundos de tiempo de CPU y 35 MB de memoria. La expresión para  $C_{20}$  tiene 48.528 términos, todos ellos independientes. Las expresiones resultantes para  $C_8$  son idénticas a las expresadas en la base clásica de Hall.

En la tabla 4.2 recogemos el tiempo de CPU (en segundos) y la memoria (en MBytes) necesaria para construir los términos  $C_2, C_3, \dots, C_n$  hasta un determinado valor de  $n$ , con la recurrencia descrita en la sección 4.4.2 (*Nuevo*)

y la implementación proporcionada en [WS09] usando una variante del llamado *método de comparación*, introducido previamente en [Wit75]. Nótese que con este método, que es el más eficiente de todos los procedimientos analizados en [WS09], los exponentes Zassenhaus  $C_j$  se expresan como combinaciones lineales de palabras de longitud  $j$  y no directamente en términos de conmutadores independientes. Como comparación,  $C_{16}$  tiene 54.146 términos cuando se expresan como combinaciones de palabras, pero sólo 3711 un acuerdo con la nueva formulación. Esto se traduce directamente en los requisitos de memoria de ambos algoritmos, como es evidente a partir de los resultados recogidos en la tabla.

$n$	Tiempo CPU (segundos)		Memoria (MBytes)	
	<i>W-S</i>	<i>Nuevo</i>	<i>W-S</i>	<i>Nuevo</i>
14	29.18	0.14	122.90	0.88
16	203.85	0.59	764.32	4.09
18		3.01		11.12
20		19.18		35.27

Cuadro 4.2: Tiempo de CPU y de memoria necesarios para el cómputo de los términos  $C_2, C_3, \dots, C_n$  en la fórmula de Zassenhaus hasta el valor dado de  $n$  usando el algoritmo presentado en [WS09] y la recurrencia (4.58) (*Nuevo*).

## 4.5. Convergencia de la fórmula de Zassenhaus

Supongamos ahora que  $X$  e  $Y$  se definen en un álgebra de Banach  $\mathcal{A}$ , es decir, un álgebra que es también un espacio lineal normado completo cuya norma es submultiplicativa,

$$\|XY\| \leq \|X\| \|Y\| \quad (4.61)$$

para cualquier par de elementos de  $\mathcal{A}$ . Entonces para el conmutador se tiene  $\|[X, Y]\| \leq 2 \|X\| \|Y\|$  y tiene sentido analizar la convergencia de la fórmula de Zassenhaus (4.37).

Sólo hay dos resultados previos acerca de la la convergencia, siendo ambos de la forma  $\|X\| + \|Y\| < r$  con  $r > 0$ . Específicamente, Suzuki [Suz77] obtiene  $r_s = \log 2 - \frac{1}{2} \approx 0.1931$ , mientras que Bayen [Bay79] demostró que el

dominio de la convergencia puede ser aumentado a un valor de  $r_b$  dado por la solución positiva única de la ecuación

$$z^2 \left( 1 + 2 \int_0^z \frac{e^{2w} - 1}{w} dw \right) = 4(2 \log 2 - 1).$$

Un cálculo numérico muestra que  $r_b = 0.59670569 \dots$ . Entonces para  $\|X\| + \|Y\| < r_b$  tenemos

$$\lim_{n \rightarrow \infty} e^X e^Y e^{C_2} \dots e^{C_n} = e^{X+Y}. \quad (4.62)$$

A continuación, se utiliza la recursividad (4.50)-(4.52) para demostrar que (4.62) se cumple, para  $(x, y) \equiv (\|X\|, \|Y\|) \in \mathbb{R}^2$  en un dominio que es mayor que  $\{(x, y) \in \mathbb{R}^2 : 0 \leq x + y < r_b\}$ .

Claramente, (4.62) se cumple si

$$\lim_{n \rightarrow \infty} \|R_n(1)\| = 1, \quad (4.63)$$

donde  $R_n(\lambda)$  es dado por (4.46), y es la solución de un problema de valor inicial

$$\frac{d}{d\lambda} R_n(\lambda) = F_n(\lambda) R_n(\lambda), \quad R_n(0) = I. \quad (4.64)$$

Es bien sabido que, si  $\int_0^1 \|F_n(\lambda)\| d\lambda < \infty$ , entonces existe una única solución  $R_n(\lambda)$  de (4.64) para  $0 \leq \lambda \leq 1$ , y que  $\|R_n(1)\| \leq \exp(\int_0^1 \|F_n(\lambda)\| d\lambda)$ . En consecuencia, (4.63) converge siempre que  $(x, y) = (\|X\|, \|Y\|) \in \mathbb{R}^2$  es tal que

$$\lim_{n \rightarrow \infty} \int_0^1 \|F_n(\lambda)\| d\lambda = 0.$$

De (4.60) tenemos que  $\|C_{n+1}\| \leq \delta_{n+1}$ , donde  $\delta_2 = xy$  y para  $n \geq 2$ ,

$$\delta_{n+1} = \frac{1}{n+1} \sum_{(i_0, i_1, \dots, i_n) \in \mathcal{I}_n} \frac{2^{i_0 + \dots + i_n}}{i_0! i_1! \dots i_n!} \delta_n^{i_n} \dots \delta_2^{i_2} y^{i_1} x^{i_0} y.$$

Similarmente,  $\|F_n(\lambda)\| \leq f_n(\lambda)$ , donde

$$f_1(\lambda) = \sum_{i_1=0}^{\infty} \sum_{i_0=1}^{\infty} \frac{(2\lambda)^{i_0+i_1}}{i_0! i_1!} y^{i_1} x^{i_0} y = e^{2\lambda y} (e^{2\lambda x} - 1) y, \quad (4.65)$$

y para  $n \geq 2$ ,

$$f_n(\lambda) = \sum_{(i_0, i_1, \dots, i_n) \in \mathcal{I}_n} \frac{2^{i_0 + \dots + i_n} \lambda^{i_0 + i_1 + 2i_2 + \dots + ni_n}}{i_0! i_1! \dots i_n!} \delta_n^{i_n} \dots \delta_2^{i_2} y^{i_1} x^{i_0} y.$$

Esto implica

$$\int_0^1 f_n(\lambda) d\lambda \leq \sum_{k=n}^{\infty} \delta_k,$$

de modo que (4.63) se garantiza si la serie  $\sum_{k=2}^{\infty} \delta_k$  converge. Analicemos cada término de esta serie imitando el procedimiento recursivo dado por el algoritmo de la sección 4.4.2. De (4.54) (o alternativamente de (4.65)) y (4.55), obtenemos

$$\begin{aligned} \|f_{1,k}\| &\leq d_{1,k} \equiv 2^k y \sum_{j=1}^k \frac{1}{j!(k-j)!} x^j y^{k-j} = \frac{2^k}{k!} y ((x+y)^k - y^k) \\ \|f_{n,k}\| &\leq d_{n,k} = \sum_{j=0}^{\lfloor k/n \rfloor - 1} \frac{2^j}{j!} \delta_n^j d_{n-1, k-nj}, \end{aligned} \quad (4.66)$$

de donde

$$\|C_n\| \leq \delta_n = \frac{1}{n} d_{\lfloor (n-1)/2 \rfloor, n-1}, \quad n \geq 3. \quad (4.67)$$

Una condición suficiente para la convergencia se obtiene mediante la imposición de

$$\lim_{n \rightarrow \infty} \frac{\delta_{n+1}}{\delta_n} < 1. \quad (4.68)$$

En este punto vale la pena señalar que, aunque no se refleja en la notación, tanto  $d_{n,k}$  y  $\delta_n$  dependen  $(x, y) = (\|X\|, \|Y\|)$ , por lo que la condición (4.68) implica, de hecho, una restricción en el dominio de la convergencia  $(x, y) \in \mathbb{R}^2$  de la fórmula de Zassenhaus. En la figura 4.3 se muestra gráficamente el dominio  $\mathcal{D}$  de tales puntos  $(x, y)$ . Esto ha sido obtenido mediante el cálculo para cada punto de los coeficientes  $d_{n,k}$  y  $\delta_n$  hasta  $n = 1000$  (de hecho, teniendo en cuenta un valor menor de  $n$  no hay cambio significativo). También se han incluido para la comparación los resultados anteriores:  $x+y < 0.1931$  y  $x+y < 0.5967$ , de Suzuki y Bayen, respectivamente. Observamos que el nuevo dominio de convergencia es considerablemente más extenso. En particular, contiene la región  $x+y < 1.054$ , pero no se limita a eso. Por ejemplo, el dominio de convergencia contiene los conjuntos  $\{(x, 2.9216) \mid x < 0.00292\}$  y  $\{(2.893, y) \mid y < 0.0145\}$ , y también los puntos  $(x, 0)$ ,  $(y, 0)$  con valor de  $x$  o  $y$  arbitrariamente grande.

### 4.5.1. Implementación

Para hacer posible el estudio numérico de esta serie, se implementó en código Matlab el procedimiento descrito con anterioridad.

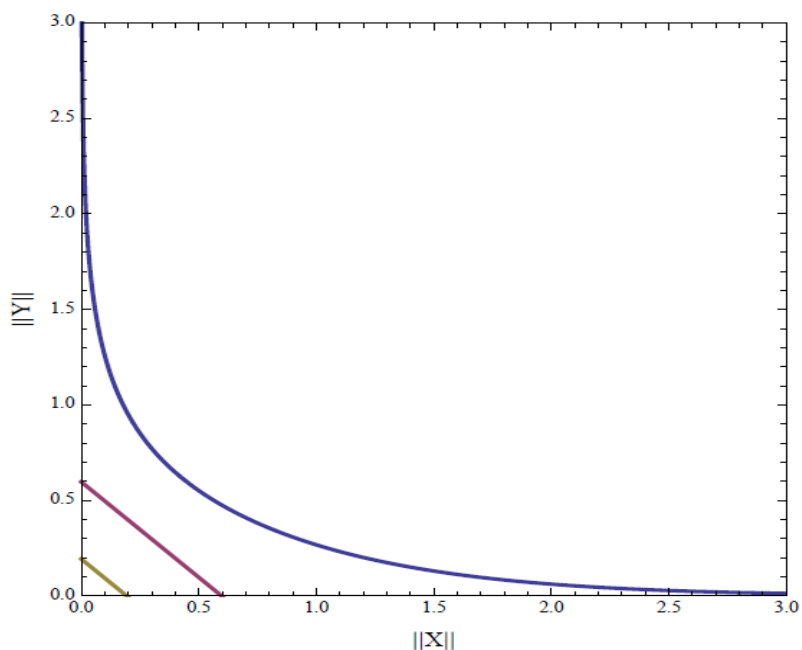


Figura 4.3: Límite superior del dominio de la convergencia de la fórmula Zassenhaus obtenido con el procedimiento (4.66) - (4.67). Resultados anteriores  $0 < \|X\| + \|Y\| < 0.1931$  y  $0 < \|X\| + \|Y\| < 0.5967$  también se representan para la comparación. El nuevo dominio contiene la región  $\|X\| + \|Y\| < 1.054$ .

```
function CC=Cn(A,B,n,m)
*operador Cn :
* Cn matrices cuadradas de los términos de Zassenhaus
* n natural
* Se calcula Cn para n ≥ 1
* A y B matrices m × m
*****
    C = cell(15,1);
    for i=1:15 Ci,1=zeros(m); end
    for i=2:n if i_j=4
    Ci,1=(1/i)*flk(A,B,C,1,i-1,m); end
    if i ≥ 5
    Ci,1=(1/i)*fkn(A,B,C,floor((i-1)/2),i-1,m);
    end
    end
```

```

    CC = C ; end

function fn=fnk(A,B,C,n,k,m)
* operador fn,k :
* A,B, C matrices cuadradas
* n,k,m natural
*****
S=zeros(m);
for j=0:floor(k/n)-1;
if n-1==1
S = S + (((-1)^j)/(factorial(j))) * adj(Cn, 1, j, f1k(A, B, C, n - 1, k - n *
j, m));
else
S = S + (((-1)^j)/(factorial(j))) * adj(Cn, 1, j, fnk(A, B, C, n - 1, k - n *
j, m));
end
fn=S; end
*****

function f1=f1k(A,B,C,n,k,m)
* operador f1,k :
* A y B matrices cuadradas
* k natural, m tamaño matriz
*****
if n==1
S=zeros(m);
for j=1:k;
S = S + (((-1)^k)/(factorial(k-j)*factorial(j))) * adj(B, k-j, adj(A, j, B));
end
f1=S;
end

end

```

#### 4.5.2. Algunos resultados numéricos

Los términos  $C_n$ ,  $n \geq 2$  de la fórmula de Zassenhaus generados por el nuevo procedimiento forman un conjunto linealmente independiente [CMN12]. Para comparar la efectividad numérica del método, generamos matrices  $X$  e  $Y$   $10 \times 10$  aleatorias, que cumplen con la cota  $\|X\| + \|Y\| \leq 1.054$ .

La figura 4.4 muestra el comportamiento del error absoluto. Observamos que el error absoluto va disminuyendo a medida que el número de términos de la serie de Zassenhaus va aumentando, lo cual es de esperar dado que la serie infinita converge para matrices en la región señalada en [CMN12].

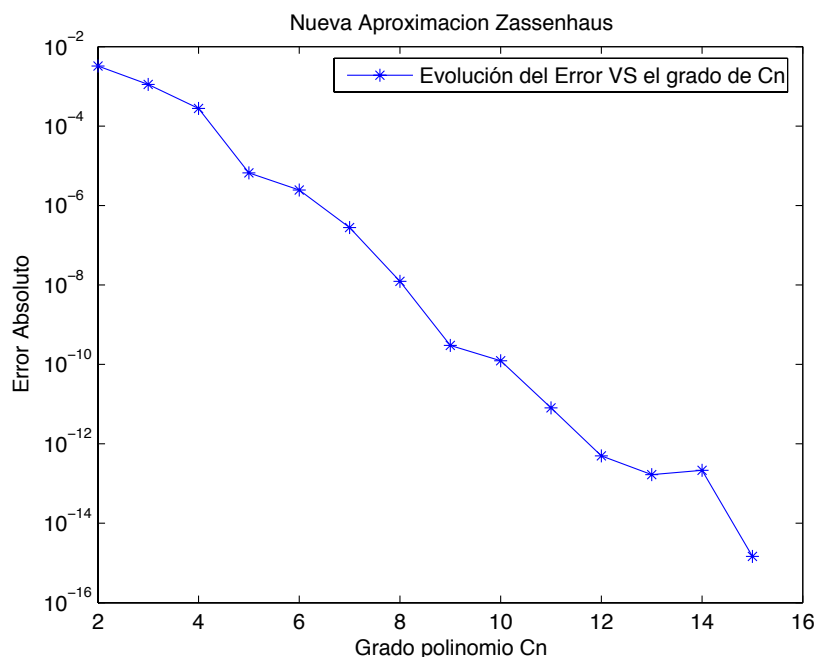


Figura 4.4: Gráfica del error absoluto. Se utilizó la función `expm()`, que dispone **Matlab**, para el cálculo de referencia. Con parámetro  $\lambda = 1$ . Se aprecia que el error absoluto decrece rápidamente al aumentar el número de términos de la serie, es decir, al crecer el grado de  $C_n$ , no importando las matrices utilizadas en la zona de convergencia dada por la cota.

En la figura 4.5 se representa el comportamiento de los errores absolutos de distintas series de Zassenhaus, tomándose tres valores del parámetro  $\lambda$ , a saber, ( $\lambda = 0.5; 1.0; 1.5$ ). Se utilizaron matrices  $2 \times 2$  generadas al azar con **Matlab**. Se puede apreciar que el parámetro  $\lambda = 0.5$  ofrece una caída más brusca del error absoluto de la serie de Zassenhaus al compararla con la expresión  $e^{\lambda(X+Y)}$  aproximada por **Matlab**.

Estas experiencias indican que para lograr acelerar la convergencia el parámetro  $\lambda$  juega un papel importante, más allá de la forma de las matrices utilizadas, encontrándose éstas en la región de convergencia.



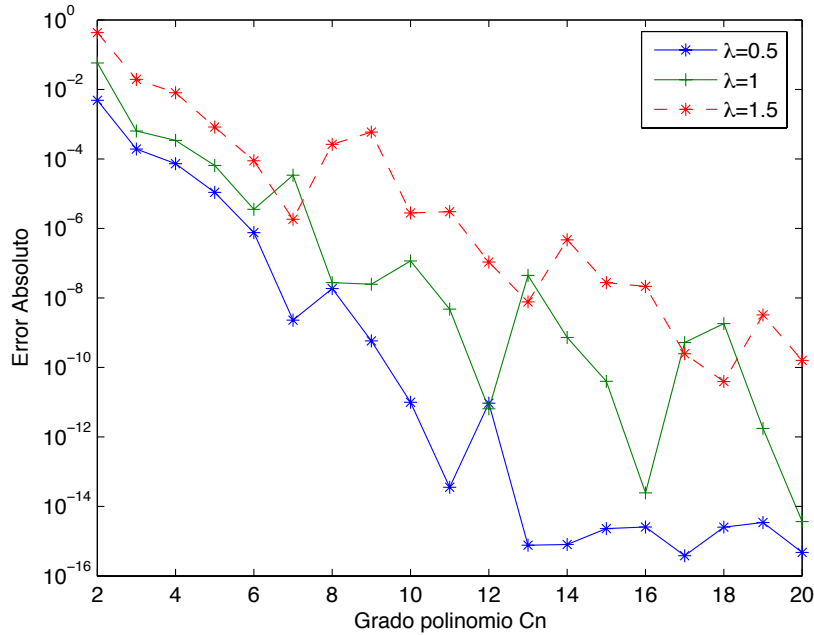


Figura 4.5: Gráfica error absoluto. Se utilizó la función `expm()`, que dispone **Matlab**. Con distintos valores del parámetro  $\lambda$ . En términos precisos se utilizó para esta prueba  $\lambda \in \{0.5, 1, 1.5\}$ .

### 4.5.3. Comportamiento de la convergencia

Como se citó antes, una cota (entre otras [Suz77],[Bay79]) para la convergencia de las matrices en el nuevo procedimiento de la fórmula de Zassenhaus está dada en términos de la norma de las matrices  $X$  e  $Y$ . Más en concreto, si las matrices  $(X, Y) \in V_{0;\kappa}$ , con

$$V_{0;\kappa} = \{A, B \in M_{n \times n} / 0 \leq \|A\| + \|B\| \leq \kappa\}, \quad (4.69)$$

entonces el procedimiento converge cuando  $\kappa = 1,054$ .

Una cuestión que puede plantearse es si, a partir de matrices cuya norma sea tal que estén fuera de la región de convergencia garantizada, pero relativamente cerca, ¿sigue convergiendo el procedimiento? Se trata de determinar numéricamente si la región de convergencia garantizada por el teorema es el mejor resultado que se puede conseguir o no. Para contestar esta pregunta estudimos el comportamiento de parejas de matrices aleatorias en las cercanías de la región de convergencia.

Describimos lo anterior en los siguientes pasos:

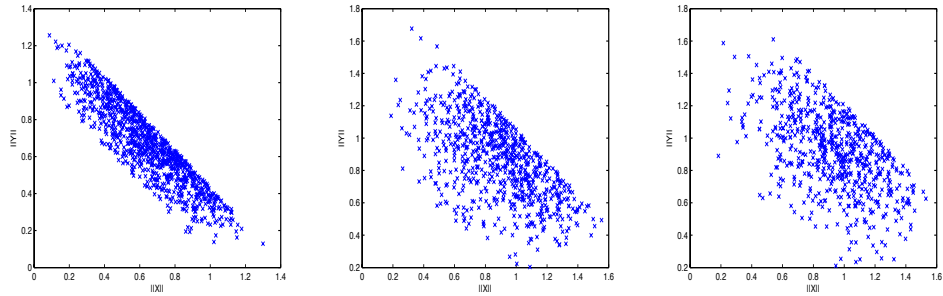


Figura 4.6: Gráfico de distintas regiones de convergencias,  $\mathcal{V}_{1,054;\tau}$ , para  $\tau = 0.3821; 1; 1.1573$  respectivamente. Estos valores de  $\tau$  se generaron aleatoriamente, por una distribución uniforme sobre el intervalo  $(0,2)$ . Se aprecia claramente cómo a medida que el parámetro  $\tau$  crece la nube de puntos se va disipando.

- Generar aleatoriamente  $(X, Y)$ , tal que

$(X, Y) \in \mathcal{V}_{1,054;\tau}$ , donde

$$\mathcal{V}_{1,054;\tau} = \{A, B \in M_{2 \times 2} / 1,054 \leq \|A\| + \|B\| \leq 1,054 + \tau, \tau \in B(0, 2)\}$$

- Calcular:

$$\mathcal{E} = \|e^{\lambda X} e^{\lambda Y} e^{\lambda^2 C_2} e^{\lambda^3 C_3} \dots e^{\lambda^n C_n} - e^{\lambda(X+Y)}\|$$

- Graficar en el plano  $\mathbb{R}^2$  el punto  $(\|X\|, \|Y\|)$  si  $\mathcal{E} \leq 0.000001$ .

La figura 4.6 reseña la experiencia con  $\tau = 0.3821; 1; 1.1573$ . Se aprecia que a medida que  $\tau$  crece, las parejas de matrices con una convergencia adecuada ( $\mathcal{E} \leq 0.000001$ ) se encuentran distribuidas de forma más dispersa, es decir, cada vez hay menos matrices que cumplen la cota y la precisión de convergencia requeridas. Sin embargo, se observa que la cota teórica de 1.054 puede mejorarse en la práctica.

## 4.6. Fórmula de Zassenhaus generalizada

La expresión generalizada de Zassenhaus que abordamos es:

$$e^{\lambda A_1 + \lambda^2 A_2 + \dots + \lambda^n A_n + \dots} = e^{\lambda C_1} e^{\lambda^2 C_2} \dots e^{\lambda^n C_n} \dots \quad (4.70)$$

donde  $A_i, C_i; i = 1, 2, \dots$  son matrices. Las  $C_i$  serán los términos de la serie de Zassenhaus generalizada que deseamos hallar.

Partimos definiendo

$$f(\lambda) = \lambda A_1 + \lambda^2 A_2 + \dots = \sum_{i=1}^{\infty} \lambda^i A_i, \quad (4.71)$$

de manera que

$$f'(\lambda) = \sum_{i=1}^{\infty} i \lambda^{i-1} A_i. \quad (4.72)$$

Como ya hemos visto, la derivada de la exponencial de  $f(\lambda)$  se expresa como

$$\frac{d}{d\lambda} e^{f(\lambda)} = d \exp_{f(\lambda)}(f'(\lambda)) e^{f(\lambda)}, \quad (4.73)$$

donde  $d \exp$  se define mediante la serie de potencias, siempre que converja,

$$d \exp_{f(\lambda)}(f'(\lambda)) = \sum_{k=0}^{\infty} \frac{1}{(k+1)!} \text{ad}_{f(\lambda)}^k f'(\lambda) \quad (4.74)$$

Por otra parte, al derivar  $e^{f(\lambda)}$  se tiene

$$\begin{aligned} \frac{d}{d\lambda} e^{f(\lambda)} &= \frac{d}{d\lambda} [e^{\lambda C_1} e^{\lambda^2 C_2} \dots e^{\lambda^n C_n} \dots] \\ &= C_1 e^{\lambda C_1} e^{\lambda^2 C_2} \dots + 2\lambda e^{\lambda C_1} C_2 e^{\lambda^2 C_2} \dots + 3\lambda^2 e^{\lambda C_1} e^{\lambda^2 C_2} \lambda^3 C_3 \dots \end{aligned} \quad (4.75)$$

Aplicando la relación (4.73), y el hecho de que

$$e^{-f} = \dots e^{-\lambda^n C_n} \dots e^{-\lambda^2 C_2} e^{-\lambda C_1} \quad (4.76)$$

se obtiene

$$\begin{aligned} d \exp_f f' &= \left( \frac{d}{d\lambda} e^f \right) e^{-f} \\ &= C_1 + \sum_{j=2}^{\infty} j \lambda^{j-1} e^{\text{ad}_{\lambda C_1}} \dots e^{\text{ad}_{\lambda^{j-1} C_{j-1}}} C_j. \end{aligned} \quad (4.77)$$

Por otra parte, tenemos

$$d \exp_f f' = \sum_{k=1}^{\infty} \frac{1}{(k+1)!} \text{ad}_f^k f' = \frac{1}{2!} \text{ad}_f f' + \frac{1}{3!} \text{ad}_f^2 f' + \dots \quad (4.78)$$

Sea

$$f_1 \equiv \text{ad}_f f' = [f, f'] \quad (4.79)$$

Observamos que se puede construir (4.79) como una serie infinita en  $\lambda$ . Para ello

$$\begin{aligned} f_1 = [f, f'] &= [\lambda A_1 + \lambda^2 A_2 + \cdots + \lambda^n A_n + \cdots, A_1 + 2\lambda A_2 \cdots + n\lambda^{n-1} A_n \cdots] \\ &= \lambda^2(2[A_1, A_2] + [A_2, A_1]) \implies \lambda^2[A_1, A_2] \\ &+ \lambda^3(3[A_1, A_3] + [A_3, A_1]) \implies 2\lambda^3[A_1, A_3] \\ &+ \lambda^4(2[A_3, A_2] + 3[A_2, A_3] + 4[A_1, A_4] + [A_4, A_1]) \implies \lambda^4(3[A_1, A_4] + [A_2, A_3]) \\ &+ \lambda^5(4[A_1, A_5] + 2[A_2, A_4]) + \cdots \\ &+ \lambda^j \sum_{k=1}^{[j/2]} (j-k+1) \text{ad}_{A_k} A_{j+k-1} \cdots \end{aligned}$$

Por consiguiente

$$f_1 = \sum_{j=2}^{\infty} \lambda^j f_{1,j}, \quad (4.80)$$

con

$$f_{1,j} = \sum_{k=1}^{[j/2]} (j-k+1) \text{ad}_{A_k} A_{j+k-1} = \sum_{k=1}^{[j/2]} (j-k+1) [A_k, A_{j+k-1}]. \quad (4.81)$$

A continuación definimos

$$f_2 \equiv \text{ad}_f^2 f' = [f, \text{ad}_f f']. \quad (4.82)$$

Desarrollando esta última expresión tenemos

$$\begin{aligned} [f, \text{ad}_f f'] &= [\lambda A_1 + \lambda^2 A_2 + \cdots, f_1] \\ &= [\lambda A_1 + \lambda^2 A_2 + \lambda^3 A_3 \cdots, \lambda^2 f_{1,2} + \lambda^3 f_{1,3} + \lambda^4 f_{1,4} + \cdots] \\ &= \lambda^3 (A_1 f_{1,2} - f_{2,1}) \Rightarrow \lambda^3 [A_1, f_{1,2}] \\ &+ \lambda^4 (A_1 f_{1,3} - f_{3,1} A_1 + A_2 f_{1,2} - f_{2,1} A_2) \Rightarrow \lambda^4 ([A_1, f_{1,3}] + [A_2, f_{1,2}]) \\ &+ \lambda^5 (A_1 f_{1,4} - f_{4,1} A_1 + A_2 f_{1,3} - f_{3,1} A_2 + A_3 f_{1,2} - f_{1,2} A_3) \\ &\Rightarrow \lambda^5 ([A_1, f_{1,4}] + [A_2, f_{1,3}] + [A_3, f_{1,2}]) + \cdots \end{aligned}$$

Luego

$$f_2 = \sum_{j=3}^{\infty} \lambda^j f_{2,j}, \quad (4.83)$$

donde

$$f_{2,j} = \sum_{k=1}^{j-2} \text{ad}_{A_k} f_{1,j-k}.$$

Antes de hacer una generalización de los  $f_i$ , definimos

$$f_3 \equiv \text{ad}_f^3 f' = [f, \text{ad}_f^2 f'] = [f, f_2]. \quad (4.84)$$

En forma análoga a los casos anteriores, tenemos que

$$\begin{aligned} f_3 &= [\lambda A_1 + \lambda^2 A_2 + \dots, f_2] \\ &= \lambda^4 [A_1, f_{2,3}] + \lambda^5 ([A_1, f_{2,4}] + [A_2, f_{2,3}]) + \dots, \end{aligned}$$

que se puede escribir como

$$f_3 = \sum_{j=4}^{\infty} \lambda^j \sum_{k=1}^{j-3} \text{ad}_{A_k} f_{2,j-k} = \sum_{j=3}^{\infty} \lambda^j f_{3,j}. \quad (4.85)$$

Con lo anterior en mente, estamos en condiciones de generalizar a  $f_k$ ,

$$f_k = \text{ad}_f^k f' = \sum_{j=k+1}^{\infty} \lambda^j f_{k,j}, \quad f_{k,j} = \sum_{l=1}^{j-k} \text{ad}_{A_l} f_{k-1,j-l}, \quad k \geq 2. \quad (4.86)$$

Ahora podemos continuar el desarrollo de la expresión (4.74), pero omitiendo el primer término de la serie, así que

$$\begin{aligned} d \exp_{f(\lambda)}(f'(\lambda)) &= \sum_{k=1}^{\infty} \frac{1}{(k+1)!} \text{ad}_{f(\lambda)}^k f'(\lambda) = \sum_{k=1}^{\infty} \frac{1}{(k+1)!} f_k \\ &= \frac{1}{2!} \sum_{j=2}^{\infty} \lambda^j f_{1,j} + \frac{1}{3!} \sum_{j=3}^{\infty} \lambda^j f_{2,j} + \dots \\ &\quad + \frac{1}{n!} \sum_{j=n}^{\infty} \lambda^j f_{n-1,j} + \dots \end{aligned}$$

Escribimos esta última expresión como una serie en  $\lambda$ ,

$$\begin{aligned} \sum_{k=1}^{\infty} \frac{1}{(k+1)!} \text{ad}_{f(\lambda)}^k f'(\lambda) &= \lambda^2 \frac{1}{2!} f_{1,2} + \lambda^3 \left( \frac{1}{2!} f_{1,3} + \frac{1}{3!} f_{2,3} \right) \\ &\quad + \lambda^4 \left( \frac{1}{2!} f_{1,4} + \frac{1}{3!} f_{2,4} + \frac{1}{4!} f_{3,4} \right) + \dots, \end{aligned}$$

de manera que

$$\sum_{k=1}^{\infty} \frac{1}{(k+1)!} \text{ad}_{f(\lambda)}^k f'(\lambda) = \sum_{j=2}^{\infty} \lambda^j \sum_{l=1}^{j-1} \frac{1}{(l+1)!} f_{l,j}.$$

Es decir,

$$\begin{aligned} d \exp_f f' &= \sum_{k=0}^{\infty} \frac{1}{(k+1)!} \text{ad}_f^k f' = f' + \sum_{k=1}^{\infty} \frac{1}{(k+1)!} \text{ad}_f^k f' \\ &= A_1 + \sum_{j=1}^{\infty} (j+1) \lambda^j A_{j+1} + \sum_{k=1}^{\infty} \frac{1}{(k+1)!} \text{ad}_f^k f' \\ &= A_1 + \sum_{j=1}^{\infty} (j+1) \lambda^j A_{j+1} + \sum_{j=2}^{\infty} \lambda^j \sum_{l=1}^{j-1} \frac{1}{(l+1)!} f_{l,j} \\ &= A_1 + 2\lambda A_2 + \sum_{j=2}^{\infty} \lambda^j \left( (j+1) A_{j+1} + \sum_{l=1}^{j-1} \frac{1}{(l+1)!} f_{l,j} \right). \end{aligned} \tag{4.87}$$

Al aplicar la ecuación (4.77) y asumiendo  $C_1 = A_1$ , se tiene

$$d \exp_f f' - C_1 = \sum_{j=2}^{\infty} j \lambda^{j-1} e^{\text{ad}_{\lambda C_1}} \dots e^{\text{ad}_{\lambda^{j-1} C_{j-1}}} C_j. \tag{4.88}$$

Por otra parte,

$$\begin{aligned} d \exp_f f' - A_1 &= 2\lambda A_2 + \sum_{j=2}^{\infty} \lambda^j \left( (j+1) A_{j+1} + \sum_{l=1}^{j-1} \frac{1}{(l+1)!} f_{l,j} \right) \\ &\equiv \sum_{n=1}^{\infty} \lambda^n h_n, \end{aligned} \tag{4.89}$$

donde

$$\begin{cases} h_1 = 2A_2 \\ h_n = (n+1)A_{n+1} + \sum_{l=1}^{n-1} \frac{1}{(l+1)!} f_{l,n}, \quad n \geq 2. \end{cases} \tag{4.90}$$

A continuación, procedemos definiendo

$$G_1 \equiv e^{-\text{ad}_{\lambda A_1}} (d \exp_f f' - A_1). \tag{4.91}$$

Aplicando un poco de manipulación algebraica a esta última expresión,

$$\begin{aligned}
G_1 &= \sum_{k=0}^{\infty} \frac{(-\lambda)^k}{k!} \text{ad}_{-\lambda A_1}^k (d \exp_f f' - A_1) \\
&= \sum_{k=0}^{\infty} \frac{(-\lambda)^k}{k!} \text{ad}_{-\lambda A_1}^k \left( \sum_{j=1}^{\infty} \lambda^j h_j \right) \\
&= \sum_{k=0}^{\infty} \frac{(-\lambda)^k}{k!} \sum_{j=1}^{\infty} \text{ad}_{A_1}^k \lambda^j h_j = \sum_{k=0}^{\infty} \frac{(-\lambda)^k}{k!} \sum_{j=1}^{\infty} \lambda^j \text{ad}_{A_1}^k h_j.
\end{aligned}$$

Desarrollando la serie,

$$\begin{aligned}
G_1 &= \frac{(-\lambda)^0}{0!} (\lambda h_1 + \lambda^2 h_2 + \cdots + \lambda^n h_n) \\
&\quad - \frac{\lambda}{1!} (\lambda^2 \text{ad}_{A_1} h_1 + \lambda^3 \text{ad}_{A_1} h_2 + \cdots + \lambda^n \text{ad}_{A_1} h_{n-1} + \cdots) \\
&\quad + \frac{1}{2!} (\lambda^3 \text{ad}_{A_1}^2 h_1 + \lambda^4 \text{ad}_{A_1}^2 h_2 + \cdots + \lambda^n \text{ad}_{A_1}^2 h_{n-1} + \cdots) \\
&\quad - \frac{1}{3!} (\lambda^4 \text{ad}_{A_1}^3 h_1 + \lambda^5 \text{ad}_{A_1}^3 h_2 + \cdots + \lambda^n \text{ad}_{A_1}^3 h_{n-3} + \cdots) + \cdots \\
&= \lambda h_1 + \lambda^2 (h_2 - \frac{1}{1!} \text{ad}_{A_1} h_1) + \lambda^3 (h_3 - \frac{1}{1!} \text{ad}_{A_1} h_2 + \frac{1}{2!} \text{ad}_{A_1}^2 h_1) \\
&\quad + \lambda^4 (h_4 - \frac{1}{1!} \text{ad}_{A_1} h_3 + \frac{1}{2!} \text{ad}_{A_1}^2 h_2 - \frac{1}{3!} \text{ad}_{A_1}^3 h_1) + \cdots .
\end{aligned}$$

Así, se tiene

$$G_1 = \sum_{l=1}^{\infty} q_{1,l} \lambda^l \quad \text{donde} \quad q_{1,l} \equiv \sum_{j=1}^l \frac{(-1)^{l-j}}{(l-j)!} \text{ad}_{A_1}^{l-j} h_j. \quad (4.92)$$

De esta manera se obtiene una recurrencia análoga a la obtenida para la fórmula de Zassenhaus original (4.58), ahora en términos de las funciones  $G_n(\lambda)$ :

$$G_n(\lambda) = \sum_{k=1}^{\infty} q_{n,k} \lambda^k, \quad q_{n,k} = \sum_{j=0}^{[k/n]-1} \frac{(-1)^j}{j!} \text{ad}_{C_n}^j q_{n-1, k-nj}. \quad (4.93)$$

#### 4.6.1. Algoritmo de Zassenhaus Generalizado

En términos prácticos, la aplicación de este nuevo procedimiento para Zassenhaus se sintetiza en los siguientes pasos algorítmicos:

## Zassenhaus Generalizado

---

- **Input** : las matrices  $A_1, A_2, \dots$   
**Output** : las matrices  $C_1, C_2, \dots$
- Calcular:

$$C_1 = A_1$$

$$C_2 = A_2$$

$$f_{1,j} = \sum_{k=1}^{[j/2]} (j + 2k - 1) \text{ad}_{A_k} A_{j-k+1}; \quad j = 2, 3, \dots$$

$$f_{k,j} = \sum_{l=1}^{j-k} \text{ad}_{A_l} f_{k-1,j-l}; \quad k = 2, 3, \dots$$

$$h_j = (j + 1)A_{j+1} + \sum_{l=1}^{j-1} \frac{1}{(l+1)!} f_{l,j}; \quad j = 2, 3, \dots$$

- Calcular:

$$q_{1,k} = \sum_{j=1}^k \frac{(-1)^{k-j}}{(k-j)!} \text{ad}_{A_1}^{k-j} h_j; \quad k = 1, 2, 3, \dots$$

$$q_{p,k} = \sum_{j=0}^{[k/p]-1} \frac{(-1)^j}{j!} \text{ad}_{C_p}^j q_{p-1,k-pj}; \quad k \geq p$$

$$C_p = \frac{1}{p} q_{[\frac{p-1}{2}, p-1]; \quad p \geq 3.$$


---

### 4.6.2. Algunos aspectos sobre de la convergencia de la fórmula de Zassenhaus generalizada

En el trabajo de Suzuki ([Suz77]) se presenta una extensión de la fórmula de Zassenhaus, en los siguientes términos.

Para un conjunto de operadores  $A_1, A_2, \dots, A_p$  de un álgebra de Banach,

$$e^{\lambda A_1} e^{\lambda A_2} \dots e^{\lambda A_p} e^{\lambda^2 C_2} \dots e^{\lambda^n C_n} \longrightarrow e^{\lambda \sum_{j=1}^{j=p} A_j}, \text{ cuando } n \rightarrow \infty \quad (4.94)$$



para

$$|\lambda| \sum_{j=1}^p \|A_j\| \leq \log 2 - \frac{1}{2} \quad (4.95)$$

Se puede ver que la expresión (4.6.2) es menos general que la desarrollada a partir de la expresión (4.6). Sin embargo, podemos utilizarla para generar una cota de convergencia, aunque no óptima. En particular podemos afirmar que

$$e^{\lambda A_1 + \lambda^2 A_2 + \dots + \lambda^n A_n + \dots} = e^{\lambda C_1} e^{\lambda^2 C_2} \dots e^{\lambda^n C_n} \dots \quad (4.96)$$

$$\sum_{j=1}^{\infty} \|A_j\| \leq \frac{1}{|\lambda|} \left( \log 2 - \frac{1}{2} \right). \quad (4.97)$$

### 4.6.3. Estudio empírico de la convergencia

En el paper [CMN12] se estudió la convergencia de la fórmula clásica de Zassenhaus. La cota hallada fue de  $R = 1.054$ , pero en la práctica se puede tener matrices  $X$  e  $Y$  tales que  $\|X\| + \|Y\| < R + \tau$ , con  $\tau$  pequeño y que aún permitan convergencia, como se indicó en la sección 4.5.3 de este capítulo.

Es de pensar que para el caso generalizado ocurra algo similar.

**Primer enfoque:** La situación gráfica 4.7 muestra el comportamiento del error relativo de tres grupos de matrices  $10 \times 10$ , cada grupo de diez experiencias y cada experiencia con cinco matrices generadas aleatoriamente, que cumplen con distinta cota  $c_k$ , es decir,

$$\sum_{i=1}^5 \|A_i\| \leq c_k, \quad c_k \in \Theta \equiv \{c_1, c_2, c_3\}.$$

Para realizar este experimento numérico se utilizó el conjunto de cotas  $\Theta = \{1, 0.5, 0.2\}$ . El grupo de experiencias que presenta, ostensiblemente, un menor error relativo, es el grupo con cota de  $c_3 = 0.2$ . En términos promedio se obtuvo

$$\frac{\left\| e^{\sum_{i=1}^5 \lambda^i A_i} - \prod_{i=1}^5 e^{\lambda^i C_i} \right\|}{\left\| e^{\sum_{i=1}^5 \lambda^i A_i} \right\|} \leq 0.00055$$

Parece razonable suponer entonces que una cota adecuada para Zassenhaus generalizado, podría ser

$$\sum_{j=1}^{\infty} \|A_j\| \leq R_{ZG} \equiv \frac{R}{|\lambda|} \text{ donde } R = 1.054. \quad (4.98)$$

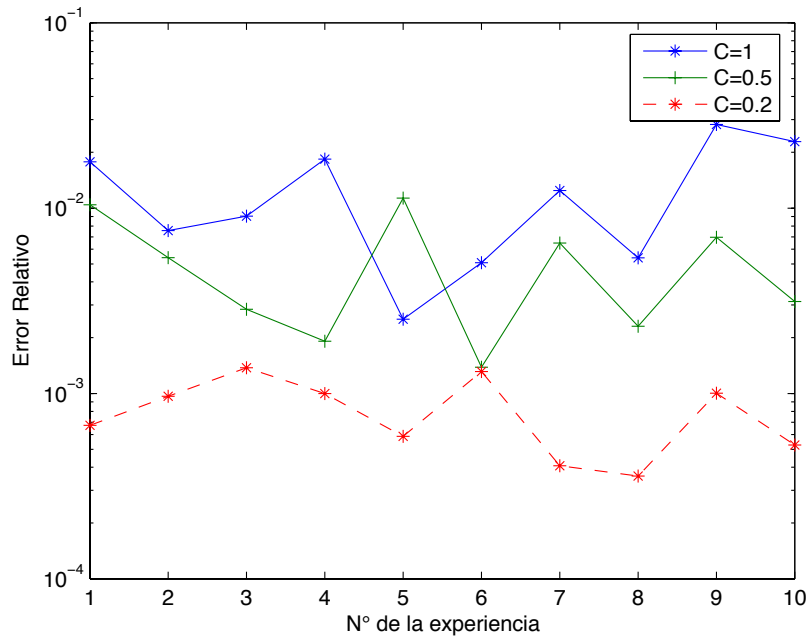


Figura 4.7: Comportamiento del error relativo de distintos grupos de experiencias. Cada grupo implica diez experiencias y cada experiencia involucra a cinco matrices  $10 \times 10$ , generadas aleatoriamente por **Matlab**. Las cotas usadas, para los grupos son  $c_1 = 1, c_2 = 0.5$  y  $c_3 = 0.2$ , respectivamente. Para todos los casos se utilizó  $\lambda = 1$ .

Se obtiene una mejora considerable de las aproximaciones dadas por la expansión de Zassenhaus generalizado, es decir, una reducción importante de los errores relativos, cuando se utiliza  $\lambda = 0.5$ . Véase la figura 4.8. De hecho en el grupo que utilizó la menor cota,  $c_3 = 0.2$ , se obtuvo un error relativo promedio inferior a 0.00001. En los términos de la cota presentada en la sección 4.5.3, un valor  $|\lambda| \rightarrow 1$  implicará una cota más grande para la suma de las normas matriciales de las matrices  $A_i$ .

**Segundo enfoque:** Podemos analizar la convergencia de la expresión (4.6), haciendo una analogía al caso del análisis de Zassenhaus (sección 4.5).

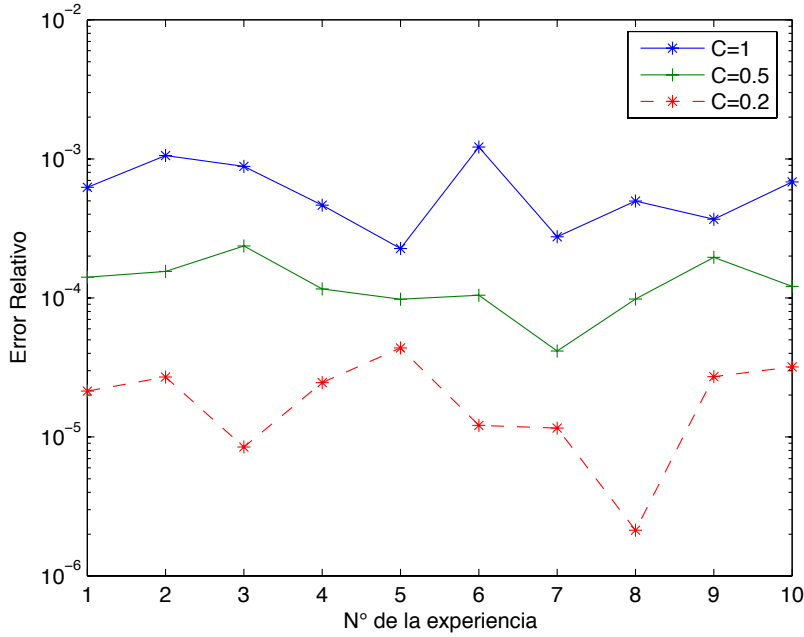


Figura 4.8: Comportamiento del error relativo análogo al estudio de la figura 4.7, pero con  $\lambda = 0.5$ .

Para ello definimos

$$\Upsilon_n(\lambda) = \sum_{i=1}^n \lambda^i A_i, \quad n \geq 1. \quad (4.99)$$

Luego

$$e^{A_1+A_2+\dots+A_n+\dots} = e^{\Upsilon_n(1)} e^{\Upsilon_{m>n}(1)}. \quad (4.100)$$

De esta manera tendremos que una cota para la convergencia será  $\|\Upsilon_n(1)\| + \|\Upsilon_{m>n}(1)\| < 1.054$ .

## 4.7. Desarrollo de Wilcox

Consideramos de nuevo el sistema lineal

$$\frac{dX_\lambda(t)}{dt} = \lambda A(t)X_\lambda(t), \quad X_\lambda(0) = I. \quad (4.101)$$

El objetivo ahora es construir la solución como un producto infinito de exponenciales de la forma

$$X_\lambda(t) = \prod_{k=1}^{\infty} e^{\lambda^k W_k(t)} = e^{\lambda W_1(t)} e^{\lambda^2 W_2(t)} e^{\lambda^3 W_3(t)} \dots, \quad (4.102)$$

es decir, se trata de construir el análogo continuo de la fórmula de Zassenhaus. Tal desarrollo fue considerado por Wilcox, atribuyéndolo erróneamente a Fer.

Sustituyendo (4.102) en la ecuación (4.101), derivando y multiplicando por la derecha por el operador inverso

$$[X_\lambda(t)]^{-1} = \dots e^{-\lambda^3 W_3(t)} e^{-\lambda^2 W_2(t)} e^{-\lambda W_1(t)}$$

tenemos

$$\begin{aligned} & \frac{d(e^{\lambda W_1(t)})}{dt} e^{-\lambda W_1(t)} + e^{\lambda W_1(t)} \frac{d(e^{\lambda^2 W_2(t)})}{dt} e^{-\lambda^2 W_2(t)} e^{-\lambda W_1(t)} + \\ & = e^{\lambda W_1(t)} e^{\lambda^2 W_2(t)} \frac{d(e^{\lambda^3 W_3(t)})}{dt} e^{-\lambda^3 W_3(t)} e^{-\lambda^2 W_2(t)} e^{-\lambda W_1(t)} \dots = \lambda A(t). \end{aligned} \quad (4.103)$$

Esta última expresión se puede reescribir como

$$\begin{aligned} & \sum_{n=0}^{\infty} \frac{\lambda^{n+1}}{(n+1)!} \text{ad}_{W_1}^n \dot{W}_1 + \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{\lambda^{m+2n+2}}{m!(n+1)!} \text{ad}_{W_1}^m \left( \text{ad}_{W_2}^n \dot{W}_2 \right) \\ & + \sum_{k=0}^{\infty} \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{\lambda^{k+2m+3n+3}}{k!m!(n+1)!} \text{ad}_{W_1}^k \text{ad}_{W_2}^m \left( \text{ad}_{W_3}^n \dot{W}_3 \right) + \dots = \lambda A(t), \end{aligned} \quad (4.104)$$

donde hemos denotado  $W_i \equiv W_i(t)$ ,  $i \geq 1$ . Igualando en esta ecuación los coeficientes de las distintas potencias de  $\lambda$  es fácil encontrar la expresión de los primeros términos  $\dot{W}_i$ :

$$\begin{aligned} \dot{W}_1 &= A(t) \\ \dot{W}_2 &= -\frac{1}{2} \text{ad}_{W_1} \dot{W}_1 \\ \dot{W}_3 &= -\frac{1}{6} \text{ad}_{W_1}^2 \dot{W}_1 - \text{ad}_{W_1} \dot{W}_2, \end{aligned}$$

de donde, con un poco de álgebra, se obtiene

$$\begin{aligned} W_1(t) &= \int_0^t A(t_1) dt_1, \\ W_2(t) &= \frac{1}{2} \int_0^t dt_1 \int_0^{t_1} dt_2 [A(t_1), A(t_2)] \\ W_3(t) &= \frac{1}{3} \int_0^t dt_1 \int_0^{t_1} dt_2 \int_0^{t_2} dt_3 ([A(t_1), A(t_2)], A(t_3)) + [A(t_2), [A(t_3), A(t_1)]] \end{aligned} \quad (4.105)$$

Nuestro objetivo consiste ahora en elaborar un procedimiento que nos permita calcular los términos  $W_i(t)$  en el desarrollo (4.102) hasta un orden arbitrario y que sea sistemático. Aunque en principio esto se puede lograr a partir de (4.104) desarrollando las series y aislando los términos relevantes de acuerdo con la potencia de  $\lambda$ , construir términos más allá de  $W_5$  constituye una tarea difícil (en la referencia ([GS02]) se ha llegado hasta  $\dot{W}_7$  en términos de los  $W_i$  precedentes). Dadas las similitudes con la fórmula de Zassenhaus, vamos aplicar aquí el mismo tipo de técnicas.

Teniendo en cuenta la expresión de la derivada de la exponencial de un operador dependiente del tiempo,

$$\frac{d}{dt}e^{\lambda^i W_i} = \lambda^i d \exp_{\lambda^i W_i}(\dot{W}_i) e^{\lambda^i W_i},$$

podemos escribir (4.103) como

$$\lambda d \exp_{\lambda W_1} \dot{W}_1 + \lambda^2 e^{\lambda W_1} d \exp_{\lambda^2 W_2}(\dot{W}_2) e^{-\lambda W_1} + \lambda^3 e^{\lambda W_1} e^{\lambda^2 W_2} d \exp_{\lambda^3 W_3}(\dot{W}_3) e^{-\lambda^2 W_2} e^{-\lambda W_1} + \dots = \lambda A$$

o bien

$$\lambda d \exp_{\lambda W_1} \dot{W}_1 + \lambda^2 e^{\lambda \text{ad}_{W_1}} d \exp_{\lambda^2 W_2}(\dot{W}_2) + \lambda^3 e^{\lambda \text{ad}_{W_1}} e^{\lambda^2 \text{ad}_{W_2}} d \exp_{\lambda^3 W_3}(\dot{W}_3) + \dots + \lambda^n e^{\lambda \text{ad}_{W_1}} e^{\lambda^2 \text{ad}_{W_2}} \dots e^{\lambda^{n-1} \text{ad}_{W_{n-1}}} d \exp_{\lambda^n W_n}(\dot{W}_n) + \dots = \lambda A.$$

Por consiguiente,

$$\lambda A - \lambda d \exp_{\lambda W_1} \dot{W}_1 = \sum_{j=2}^{\infty} \lambda^j e^{\lambda \text{ad}_{W_1}} \dots e^{\lambda^{j-1} \text{ad}_{W_{j-1}}} d \exp_{\lambda^j W_j}(\dot{W}_j). \quad (4.106)$$

En este punto, conviene introducir las series

$$F_1 = \sum_{l=1}^{\infty} f_{1,l} \lambda^l, \quad \text{con} \quad f_{1,1} = A; \quad f_{1,l} = 0, \quad l > 1$$

y

$$G_1 = \lambda d \exp_{\lambda W_1} \dot{W}_1 = \lambda \dot{W}_1 + \sum_{k=1}^{\infty} \frac{\lambda^{k+1}}{(k+1)!} \text{ad}_{W_1}^k \dot{W}_1 \equiv \sum_{l=1}^{\infty} g_{1,l} \lambda^l,$$

de manera que

$$g_{1,1} = \dot{W}_1, \quad g_{1,l} = \frac{1}{l!} \text{ad}_{W_1}^{l-1} \dot{W}_1.$$

Imponiendo  $f_{1,1} - g_{1,1} = 0$  resulta

$$\dot{W}_1 = A, \quad (4.107)$$

con lo que, por una parte, tenemos

$$\tilde{F}_1 \equiv F_1 - G_1 = \sum_{l=2}^{\infty} \tilde{f}_{1,l} = \sum_{l=2}^{\infty} (f_{1,l} - g_{1,l})$$

y por otra parte, de (4.106),

$$\tilde{F}_1 = \lambda^2 e^{\lambda \text{ad}_{W_1}} d \exp_{\lambda^2 W_2}(\dot{W}_2) + \sum_{j=3}^{\infty} \lambda^j e^{\lambda \text{ad}_{W_1}} \dots e^{\lambda^{j-1} \text{ad}_{W_{j-1}}} d \exp_{\lambda^j W_j}(\dot{W}_j). \quad (4.108)$$

Estamos ahora en condiciones que ir al paso siguiente. De (4.108) se tiene

$$e^{-\lambda \text{ad}_{W_1}} \tilde{F}_1 = \lambda^2 d \exp_{\lambda^2 W_2}(\dot{W}_2) + \sum_{j=3}^{\infty} \lambda^j e^{\lambda^2 \text{ad}_{W_2}} \dots e^{\lambda^{j-1} \text{ad}_{W_{j-1}}} d \exp_{\lambda^j W_j}(\dot{W}_j). \quad (4.109)$$

Introducimos a continuación la serie

$$F_2 \equiv e^{-\lambda \text{ad}_{W_1}} \tilde{F}_1. \quad (4.110)$$

Desarrollando esta expresión, podemos escribir

$$F_2 = \sum_{l=2}^{\infty} \lambda^l f_{2,l}, \quad \text{donde} \quad f_{2,l} = \sum_{k=0}^{[(l-1)]-1} \frac{(-1)^k}{k!} \text{ad}_{W_1}^k \tilde{f}_{1,l-k}.$$

Análogamente, definimos

$$G_2 = \lambda^2 d \exp_{\lambda^2 W_2}(\dot{W}_2) \equiv \sum_{l=1}^{\infty} \lambda^{2l} g_{2,2l},$$

siendo

$$g_{2,2} = \dot{W}_2, \quad g_{2,2l} = \frac{1}{l!} \text{ad}_{W_2}^{l-1} \dot{W}_2, \quad g_{2,r} = 0, \quad r \neq 2l.$$

De esta forma, se tiene que

$$\tilde{F}_2 \equiv F_2 - G_2 = \lambda^2 (f_{2,2} - g_{2,2}) + \lambda^3 f_{2,3} + \lambda^4 (f_{2,4} - g_{2,4}) + \dots$$

Comparando esta expresión con (4.109), e imponiendo que

$$f_{2,2} - g_{2,2} = 0,$$

se deduce que

$$\dot{W}_2 = \tilde{f}_{1,2} = -g_{1,2} = -\frac{1}{2} \text{ad}_{W_1} \dot{W}_1.$$

De esta forma

$$\tilde{F}_2 = \sum_{l=3}^{\infty} \lambda^l \tilde{f}_{2,l}, \quad \text{con} \quad \tilde{f}_{2,l} = f_{2,l} - g_{2,l}.$$

Es claro que también se verifica

$$\tilde{F}_2 = \lambda^3 e^{\lambda^2 \text{ad}_{W_2}} d \exp_{\lambda^3 W_3}(\dot{W}_3) + \sum_{j=4}^{\infty} \lambda^j e^{\lambda^2 \text{ad}_{W_2}} \dots e^{\lambda^{j-1} \text{ad}_{W_{j-1}}} d \exp_{\lambda^j W_j}(\dot{W}_j),$$

con lo cual el proceso se puede iniciar de nuevo, esta vez con

$$\begin{aligned} F_3 &\equiv e^{-\lambda^2 \text{ad}_{W_2}} \tilde{F}_2 = \sum_{l=3}^{\infty} \lambda^l f_{3,l} \\ G_3 &= \lambda^3 d \exp_{\lambda^3 W_3}(\dot{W}_3) \equiv \sum_{l=1}^{\infty} \lambda^{3l} g_{3,3l} \\ \tilde{F}_3 &= F_3 - G_3 = \lambda^3 (f_{3,3} - g_{3,3}) + \lambda^4 f_{3,4} + \dots, \end{aligned}$$

de manera que al imponer  $f_{3,3} - g_{3,3} = 0$  podemos obtener la expresión de  $\dot{W}_3$ , etc.

En resumen, la recurrencia general viene dada por las siguientes expresiones ( $n \geq 2$ ):

$$\begin{aligned} F_n &= \sum_{l=2}^{\infty} f_{n,l} \lambda^l, & f_{n,l} &= \sum_{k=0}^{\lfloor \frac{l-1}{n-1} \rfloor} \frac{(-1)^k}{k!} \text{ad}_{W_{n-1}}^k \tilde{f}_{n-1, l-(n-1)k}, & l &\geq n \\ G_n &= \sum_{l=1}^{\infty} g_{n,nl} \lambda^{nl}, & g_{n,l} &= \frac{1}{l!} \text{ad}_{W_n}^{l-1} \dot{W}_n, & g_{n,r} &= 0, r \neq nl \\ \tilde{F}_n &= F_n - G_n = \sum_{l=n+1}^{\infty} \lambda^l \tilde{f}_{n,l}, & \tilde{f}_{n,l} &= f_{n,l} - g_{n,l} \end{aligned} \quad (4.111)$$

$$\dot{W}_n = f_{n,n} = \tilde{f}_{n-1,n} = f_{n-1,n} - g_{n-1,n}. \quad (4.112)$$

De esta forma, los primeros términos son:

$$\begin{aligned} \dot{W}_1 &= A \\ \dot{W}_2 &= -\frac{1}{2} \text{ad}_{W_1} \dot{W}_1 \\ \dot{W}_3 &= \frac{1}{3} \text{ad}_{W_1}^2 \dot{W}_1 \\ \dot{W}_4 &= -\frac{1}{8} \text{ad}_{W_1}^3 \dot{W}_1 + \frac{1}{4} \text{ad}_{W_2} \text{ad}_{W_1} \dot{W}_1. \end{aligned}$$

### 4.7.1. Ejemplos

**Ejemplo 1: Un problema propuesto por Bellman.** En la referencia [Bel65][p. 175], Bellman propone el siguiente ejercicio: dadas dos matrices cuadradas  $A$  y  $B$ , suponiendo que  $e^{A+\varepsilon B}$  se puede escribir en la forma

$$e^{A+\varepsilon B} = e^A e^{\varepsilon C_1} e^{\varepsilon^2 C_2} e^{\varepsilon^3 C_3} \dots, \quad (4.113)$$

determinense las matrices  $C_1, C_2, C_3$ , etc.. Bellman añade que, contrariamente a otros desarrollos perturbativos previamente considerados, éste aproxima la matriz  $e^{A+\varepsilon B}$  (unitaria si las matrices  $A$  y  $B$  son antihermíticas) mediante un producto de matrices que también son unitarias.

Este problema se puede resolver fácilmente con las técnicas desarrolladas en este capítulo. En efecto, consideremos la matriz ligeramente más general  $e^{t(A+\varepsilon B)}$  y escribámosla en la forma

$$U(t) \equiv e^{t(A+\varepsilon B)} = e^{tA} V, \quad (4.114)$$

tal como hicimos al abordar el problema de Cernusca (ecuación (4.31)). Dado que

$$\frac{dU(t)}{dt} = (A + \varepsilon B)U(t), \quad U(0) = I,$$

un sencillo cálculo muestra que  $V$  satisface la ecuación diferencial

$$\frac{dV}{dt} = \varepsilon e^{-tA} B e^{tA} V \equiv \varepsilon \tilde{B}(t)V, \quad V(0) = I. \quad (4.115)$$

Aplicando ahora el desarrollo de Wilcox a esta ecuación, tenemos

$$V(t) = e^{\varepsilon W_1(t)} e^{\varepsilon^2 W_2(t)} e^{\varepsilon^3 W_3(t)} \dots,$$

con lo que finalmente,

$$U(1) = e^{A+\varepsilon B} = e^A e^{\varepsilon W_1(1)} e^{\varepsilon^2 W_2(1)} e^{\varepsilon^3 W_3(1)} \dots.$$

Por consiguiente, calculando las recurrencias (4.111) y efectuando las correspondientes integrales hasta  $t = 1$ , seremos capaces de determinar las matrices  $C_i$  en (4.113) hasta el orden que nos interese. En particular, la expresión de  $C_1$  está dada por una expresión análoga a la (4.34):

$$\begin{aligned} C_1 &= \int_0^1 \tilde{B}(t) dt = \int_0^1 e^{-tA} B e^{tA} dt = \int_0^1 e^{-t \text{ad}_A} B dt = \frac{1 - e^{\text{ad}_A}}{\text{ad}_A} B \\ &= B - \frac{1}{2}[A, B] + \frac{1}{3!}[A, [A, B]] - \frac{1}{4!}[A, [A, [A, B]]] + \dots \end{aligned} \quad (4.116)$$

Resulta interesante notar que la técnica usada para resolver el problema de Cernusca y el de Bellman es similar: se efectúa una transformación previa, se determina la ecuación diferencial satisfecha por la nueva variable y se resuelve ésta mediante el desarrollo de Magnus (Cernusca) y el desarrollo de Wilcox (Bellman) hasta el orden que interese.



**Ejemplo 2.** Vamos a considerar ahora un sistema cuántico simple: se trata de una partícula de spin 1/2 en el seno de un campo magnético oscilatorio. De forma más precisa, se tiene la ecuación

$$i\hbar \frac{d}{d\tau} \psi(\tau) = -\mu \vec{S} \cdot \vec{B}(\tau) \psi(\tau),$$

donde  $\mu$  es constante (con unidades  $\text{sec}^{-1} \cdot \text{Gauss}^{-1}$ ),  $\vec{S} = \frac{\hbar}{2} \vec{\sigma}$  es el operador de spin expresado en términos de las matrices de Pauli  $\sigma_j$ ,  $j = 1, 2, 3$ , y

$$\vec{B}(\tau) = \begin{pmatrix} B_x \cos(\omega\tau) \\ 0 \\ B_z \end{pmatrix}, \quad B_x, B_z = \text{cte}$$

es el campo magnético con tiempo inicial  $\tau_0 = 0$ . En este sistema  $-\mu B_z S_3$  se puede considerar como la parte no perturbada y  $-\mu B_x \cos(\omega\tau) S_1$  como una perturbación cuya intensidad es proporcional a  $B_x$  y que depende de la frecuencia  $\omega$  como un parámetro adicional.

Usando la variable adimensional  $t = \omega\tau$ , y los parámetros (adimensionales)  $a = -\mu B_z / (2\omega)$ ,  $\varepsilon = -\mu B_x / (2\omega)$  y  $\chi(t) \equiv \psi(t/\omega)$ , el sistema se escribe como

$$i\hbar \frac{d\chi(t)}{dt} = (a \sigma_3 + \varepsilon \sigma_1 \cos t) \chi(t), \quad \chi(0) = \chi_0. \quad (4.117)$$

Si tomamos  $\hbar = 1$ , esta ecuación se puede escribir como (4.101) (con  $\lambda = 1$ ), con matriz de coeficientes

$$A(t) = -i(a \sigma_3 + \varepsilon \sigma_1 \cos t).$$

La codificación en el software *Mathematica*<sup>TM</sup> que permite la generación de los términos  $W_k(t)$  de la serie de Wilcox para este ejemplo es la siguiente:

```
Clear[Cmt, ad, ff, cc, hh, gg];
Cmt[a_, a_] := 0;
Cmt[a___, 0, b___] := 0;
Cmt[a___, c_ + d_, b___] := Cmt[a, c, b] + Cmt[a, d, b];
Cmt[a___, n_ c_Cmt, b___] := n Cmt[a, c, b];
Cmt[a___, t^n_ x_, b___] := t^n Cmt[a, x, b];
Cmt[a___, x_ y_, b___] := Cmt[a, Expand[x y], b];
mt[a___, n_ sig[j_Integer], b___] := n *
  Cmt[a, sig[j], b];
Cmt[sig[i_], sig[j_]] := -Cmt[sig[j], sig[i]] /; i > j
```

```

Cmt[z_Cmt, sig[i_]] := -Cmt[sig[i], z];
Cmt[Cmt[sig[i_], sig[j_]], z_Cmt] := -Cmt[Cmt[sig[j], z], sig[i]] -
    Cmt[Cmt[z, sig[i]], sig[j]];
Cmt[z_Cmt, Cmt[sig[i_], sig[
    j_]]] := -Cmt[sig[i], Cmt[sig[j], z]] - Cmt[sig[j], Cmt[z, sig[i]]];
Cmt[sig[1], sig[2]] := 2 I sig[3];
Cmt[sig[1], sig[3]] := -2 I sig[2];
Cmt[sig[2], sig[3]] := 2 I sig[1];
ad[a_, 0, b_] := b;
ad[a_, j_Integer, b_] := Cmt[a, ad[a, j - 1, b]];
A[t_] := -I (a sig[3] + ? Cos[t] sig[1]);
W[1] = Integrate[A[s], {s, 0, t}];
ff[1, 1] := A[t]; ff[1, n_] := 0; gg[n_, m_] := 0;
gg[1, k_] := (1/k!)ad[W[1], k - 1, ff[1, 1]];
ffhat[1, k_] := ff[1, k] - gg[1, k];
ff[n_, l_] := Sum[((-1)^k/k!]
ad[W[n-1], k, ffhat[n-1, l-(n-1)k]], {k, 0, IntegerPart[(1-1)/(n-1)]-1, 1}];
gg[n_, r_] := (1/(r/n)!)ad[W[n], (r/n) - 1, ff[n, n]] /; IntegerQ[r/n];
ffhat[n_, l_] := ff[n, l] - gg[n, l];
R[n_] := Simplify[Integrate[ffhat[n - 1, n], t]];
W[n_] := Collect[Simplify[R[n] - (R[n] /. t -> 0)], {sig[1], sig[2], sig[3]};

```

Los primeros términos de la serie de Wilcox, generados de esta manera son

$$\begin{aligned}
W_1(t) &= -i(at\sigma_3 + \varepsilon\sigma_1 \sin(t)) \\
W_2(t) &= ia\varepsilon\sigma_2(-2 + 2\cos(t) + t\sin(t)) \\
W_3(t) &= \frac{4}{3}ia\varepsilon\sigma_1(3at\cos(t) + a(-3 + t^2)\sin(t)) \\
&\quad + \frac{4}{3}ia\varepsilon\sigma_3\left(\frac{t\varepsilon}{2} + \frac{1}{4}t\varepsilon\cos(2t) - \frac{3}{8}\varepsilon\sin(2t)\right).
\end{aligned}$$

Al realizar cálculos numéricos con **Matlab**, para las constantes  $a = 1$ ,  $\varepsilon = 1$  y  $\lambda = 1$ , tomando como intervalo de integración  $t = 1$ , obtenemos

$$\chi(t = 1) \approx \prod_{i=1}^4 e^{\lambda^i W_i(1)} = \begin{pmatrix} -0.2798 + 0.7033i & 0.1009 - 0.6457i \\ -0.1009 - 0.6457i & 0.2798 + 0.7033i \end{pmatrix}$$

Comparando estos resultados con los obtenidos por una versión de Magnus de orden 4, usando como referencia un método de RK de orden 6, con  $h =$

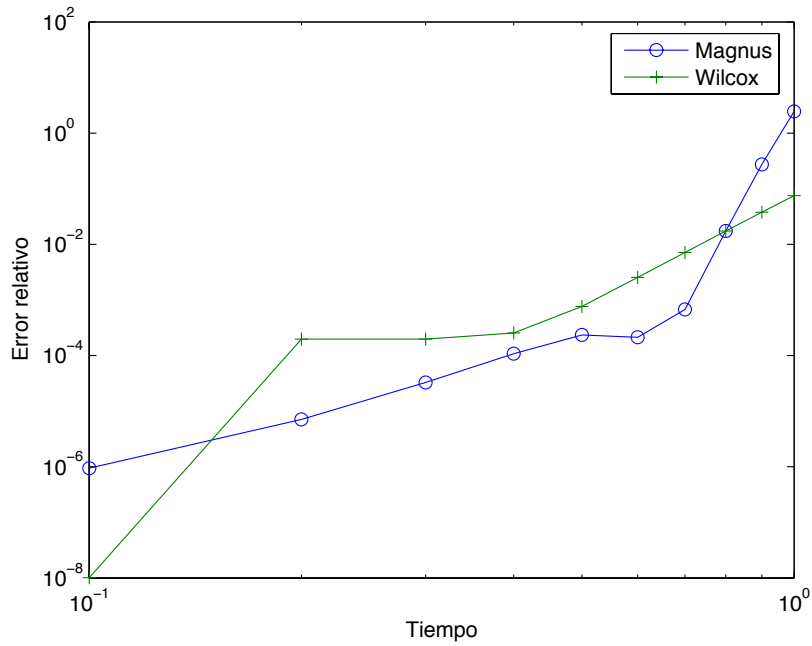


Figura 4.9: Comportamiento del error relativo de la norma de Frobenius para la matriz de la solución aproximada dada por el método de Wilcox, evolucionando el sistema de una partícula de spin  $1/2$  en el seno de un campo magnético oscilatorio, cuando el tiempo  $t$  se toma en el intervalo  $[0,1]$ .

$10^{-6}$ , figura 4.9, se aprecia que el algoritmo de Wilcox entrega una buena aproximación, superando en este caso del problema de la partícula de spin  $1/2$ , al método de Magnus de cuarto orden, en varias de las evaluaciones tomadas del tiempo  $t, t \in [0, 1]$ .



# Capítulo 5

## Conclusiones

A lo largo de este trabajo se fue creando un cúmulo de experiencias que adecuadamente canalizadas permitieron dar solución a algunos de los desafíos planteados en esta tesis. Las principales conclusiones se refieren a continuación en el orden a los temas tratados.

- Se dio una exposición detallada de la teoría necesaria para el desarrollo de nuevos métodos de integración numérica en el contexto de los grupos y álgebras de Lie. Esto se logró presentando una síntesis de las definiciones, teoremas, proposiciones y demás temas relacionados, con ello se armó la estructura abstracta que sustenta los métodos clásicos y nuevos, presentados a lo largo del texto. Un punto de partida clave de esta exposición fue el concepto de un grupo de Lie, como una variedad diferenciable [Gil74], [Var84], así como la de que un álgebra de Lie es un espacio vectorial, cuyos elementos pueden ser combinados por el corchete de Lie. La definición del operador  $ad_A$ , y la exponencial de éste facilitan el trabajo algebraico, bastante arduo a veces, en el desarrollo de los métodos de integración. Cabe destacar que una adecuada comprensión de los grupos matriciales de Lie facilita el desarrollo de los integradores geométricos [Gil74], [Pos94], [CI00].
- Se confeccionaron y analizaron diversos métodos de integración numérica para ecuaciones diferenciales definidas en grupos de Lie [CI00]. Estos métodos resultan competitivos con los métodos de Magnus [Bur81], [BCR02] para algunos de los problemas tratados. Es el caso del método de splitting de orden 4. Se diseñó e implementó una versión de orden 6, que podría ser utilizada para situaciones donde se requiera una precisión aún mayor. Además se desarrollaron interesantes modificaciones como la del cálculo de  $e^A$  para  $A$  matriz constante con traza nula, cuyo algoritmo **expmS()** se implementó especialmente en

**Matlab** y es una alternativa adecuada a otras rutinas del estándar de este software [ML03] que permiten calcular la exponencial de una matriz.

- Se diseñó una versión algorítmica de paso adaptativo, que permite modificar el tamaño del paso cuando la precisión lo requiere. De esta manera se consiguieron buenos resultados de cotas de error y con un esfuerzo computacional obtensiblemente menor al que requiere su equivalente de paso fijo. El procedimiento desarrollado resulta especialmente propicio para problemas que presentan soluciones en algunas zonas del rango de integración con cambios abruptos, y otras con variaciones menores [HLW06].
- Se obtuvo un esquema numérico basado en el procedimiento propuesto por Voslamber [Vos72], apropiado, en particular, para problemas adiabáticos [Net00]. El esquema numérico desarrollado para el método de Voslamber mantiene interesantes propiedades de convergencia de la solución aproximada, lo que hace adecuado su utilización en diversos problemas. Se expuso, implementó y probó el nuevo esquema, dando resultados que comparados con técnicas clásicas [MK98], como Runge-Kutta, mostraron gran efectividad, tanto en la precisión de los resultados como en el tiempo de CPU requeridos.
- Se elaboraron procedimientos para resolver sistemas no lineales a través de la generalización de la Expansión de Magnus [Mag54], [CI06]. Siendo estos procedimientos variados para la forma de calcular  $\Omega$  y con ello se consiguió un error relativo menor de las soluciones aproximadas que la dada por el esquema basado en Magnus de cuarto orden [CI06], para problemas que involucran ecuaciones clásicas como las ecuaciones de Emden–Fowler, Lane–Emden y Thomas–Fermi [Zwi97], [Won75], [GL07], [SS99]. Estas ecuaciones aparecen en fenómenos de alta oscilación, lo cual suele perturbar a los métodos numéricos tradicionales. Los procedimientos desarrollados son efectivos y eficientes en las experiencias desarrolladas.
- Se estudió profusamente la fórmula de Baker–Campbell–Hausdorff y la de Zassenhaus [CM09], [WM62], [Kur07], consiguiendo procedimientos nuevos y eficientes. En particular, para la fórmula de Zassenhaus se desarrolló un eficiente método [CMN12] que permite calcular los términos  $C_n$  de tal manera que por construcción sólo involucran conmutadores independientes y que permiten mejores cotas de convergencias comparados con otros esquemas.

- Se generalizó el nuevo procedimiento para la factorización  $e^{\lambda A_1 + \lambda^2 A_2 + \dots + \lambda^n A_n + \dots} = e^{\lambda C_1} e^{\lambda^2 C_2} \cdot e^{\lambda^n C_n} \dots$  con un nuevo algoritmo que permite obtener los términos  $C_n$ .
- Se estudió y construyó un procedimiento para la versión continua de la fórmula de Zassenhaus, conocida como fórmula de Wilcox [Wil67]. El nuevo esquema numérico se utilizó para experiencias como la propuesta por Bellman y un problema que involucra un sistema cuántico simple: se trata de una partícula de *spin* 1/2 en el seno de un campo magnético oscilatorio. Los resultados de estas experiencias muestran que el algoritmo desarrollado es eficaz y con una precisión comparable a otros esquemas ya probados.

## 5.1. Futuros trabajos

Las investigaciones y desarrollos que ha producido esta tesis doctoral abren la posibilidad de seguir, en futuros trabajos, el desarrollo de nuevos métodos geométricos y abordar nuevos problemas aplicados, que modelen situaciones de alta complejidad para la física/matemática. Al respecto se puede destacar las siguientes líneas de investigación:

- Ampliar las pruebas del esquema numérico de Voslamber a otros problemas que requieran sus propicias propiedades numéricas y comparar exhaustivamente su rendimiento con otros esquemas.
- Obtener nuevos esquemas numéricos para el método propuesto por Voslamber de mayores órdenes. Realizar experimentos numéricos de eficiencia y de complejidad de cómputo.
- Confeccionar nuevos esquemas adaptativos de orden más alto basados en *splitting* para problemas formulados en el grupo lineal especial y aplicarlos a situaciones conocidas para medir su rendimiento, así como a otros casos de interés.
- Adaptar el método de *splitting* de orden 6 para el cálculo de  $e^A$ , con  $A$  matriz constante de traza nula. Analizar el rendimiento de este nuevo algoritmo, comparándolo con la amplia variedad de métodos existentes [ML03]. Asimismo, generalizaremos la técnica para el caso más general de una matriz cuadrada arbitraria.
- Ampliar el tratamiento de problemas no lineales mediante integradores en grupos de Lie a nuevos problemas de interés en las aplicaciones de

la física-matemática. Analizar las ventajas de utilizar el cálculo exacto o aproximado de  $\Omega^{[1]}$  en cada nueva situación.

- Adaptar el método de splitting de orden 6 para sistemas de ecuaciones diferenciales matriciales no lineales y analizar su desempeño.
- Realizar un estudio de la fórmula BCH para el caso de más de dos variables.
- Empezar el estudio detallado de la convergencia del esquema de Zassenhaus generalizado desarrollado en este trabajo para así obtener cotas de convergencias más ajustadas.
- En la misma línea del ítem anterior, desarrollar un estudio detallado sobre la convergencia del nuevo esquema numérico de Wilcox propuesto en la tesis, obteniendo cotas teóricas y empíricas.
- Aplicar el esquema numérico de Wilcox, en forma exhaustiva, para obtener comparaciones cualitativas y cuantitativas de las soluciones aproximadas obtenidas, versus otras técnicas numéricas, en el contexto de problemas más complejos.



# Apéndice



# Apéndice A

## Cálculo de la exponencial de una matriz

Un gran número de integradores numéricos en grupos de Lie requieren el cálculo de la exponencial de una matriz o, al menos de una aproximación a la misma en el grupo de Lie correspondiente. Este es un problema estándar en el análisis numérico y se puede realizar de varias maneras: aproximaciones racionales (Baker 1975, Iserles y Nørsett 1991), métodos basados en subespacios de Krylov (Hochbruck y Lubich 1997), descomposición de Schur (Golub y Van Loan 1996), entre otros. Sin embargo, en el contexto de los integradores en grupos de Lie, nuestra tarea está sujeta a una restricción importante: la aproximación debe asignar a un elemento en el álgebra de Lie  $\mathfrak{g}$  otro en el grupo de Lie  $\mathcal{G}$  correspondiente.

En el caso de álgebras de Lie de dimensión baja el problema es relativamente fácil y con frecuencia se puede evaluar la exponencial de forma explícita. En particular, los siguientes dos casos son de importancia práctica:

1. Álgebra  $\mathfrak{sl}(2)$ . Teniendo en cuenta que cualquier matriz  $A \in \mathfrak{sl}(2)$  es de la forma  $A = \begin{bmatrix} a & b \\ c & -a \end{bmatrix}$ ,

fácilmente podemos establecer que  $e^A = \cosh \omega I + \frac{\sinh \omega}{\omega} A$  donde  $\omega = \sqrt{a^2 + bc}$ .

2. Álgebra  $\mathfrak{so}(3)$ . En este caso, la exponencial de  $A = \begin{bmatrix} 0 & a & b \\ -a & 0 & c \\ -b & -c & 0 \end{bmatrix} \in \mathfrak{so}(3)$  es  $e^A = I + \frac{\sin \sigma}{\sigma} A + \frac{1 - \cos \sigma}{\sigma^2} A^2$ , donde  $\sigma = \sqrt{a^2 + b^2 + c^2}$ .

Cuando ni una fórmula explícita, ni el cálculo numérico con precisión de la máquina son factibles, tenemos que recurrir a ciertas aproximaciones. Esos procedimientos deben ajustarse a dos condiciones: (i) el resultado debe estar en el grupo de Lie  $\mathcal{G}$  y (ii) la aproximación ha de ser consistente con el grado de aproximación del método de integración en el grupo de Lie.

Una forma estándar de aproximación es reemplazar  $e^z$  por una función  $r(z)$ , analítica en un entorno del origen. La acción de dicha función se puede extender desde  $\mathbb{C}$  a  $\mathfrak{gl}(n)$ , por lo tanto, se extiende a cualquier álgebra de Lie matricial. Las dos cosas que deseamos ahora pueden ser reformuladas, al exigir  $r(z) = e^z + O(z^{p+1})$ , siendo  $p \geq 1$  el orden de consistencia del método de integración y  $r(\mathfrak{g}) \subset \mathcal{G}$ .

Como ha sido demostrado por Feng Kang y Shang (1995)<sup>1</sup>, la única función analítica que mapea  $\mathfrak{sl}(n)$  en  $SL(n)$  para cada  $n \in \mathbb{N}$  y es la identidad en cero, es  $r(z) = e^{\alpha z}$  para  $\alpha \in \mathbb{R}$ . Para una demostración véase [HLW06].

Esto significa que debemos elegir la exponencial exacta. Por otro lado, en un álgebra de Lie cuadrática hay muchas funciones con estas características: dada una función arbitraria  $f$ , analítica alrededor del origen, entonces  $e^{f(g)} \subset \mathcal{G}$  (Celledoni y Iserles 1998). En particular, éste es el caso de todas las aproximaciones de Padé diagonales,

$$r(z) = \frac{p_m(z)}{p_m(-z)},$$

donde los polinomios  $p_m$  se generan de acuerdo a la siguiente recurrencia:

$$\begin{aligned} p_0(z) &= 1, & p_1(z) &= 2 + z, \\ p_m(z) &= 2(2m - 1)p_{m-1}(z) + \lambda^2 p_{m-2}(z). \end{aligned}$$

Entonces se verifica que  $p_{2m}(z) = \exp(z) + O(z^{2m+1})$ .

## A.1. Aproximación de Chebyshev

Otra alternativa válida es el uso de aproximaciones polinómicas a la exponencial de una cierta matriz  $C$ . Supongamos que, en particular, que  $C$  es una matriz de la forma  $C = -i\tau H$ , con  $H$  hermítica y  $\tau > 0$ , como

<sup>1</sup>Este hecho es producto del siguiente resultado:

**Lema:** Sea  $R(z)$  una función diferenciable definida en una venedad de  $z = 0$ , y asumiendo que  $R(0) = 1$  y  $R'(0) = 1$ . Entonces, se tiene para  $n \geq 3$

$$\det(R(A)) = 1, \quad \forall A \in \{X \in M_{\mathbb{K}} : \text{Tr}(X) = 0\} \Leftrightarrow R(z) = e^z \quad \diamond$$

es el caso en la Mecánica Cuántica. En la aproximación de Chebyshev, el operador  $\exp(-i\tau H)$  se desarrolla en serie de polinomios de Chebyshev, en analogía con la aproximación de una función escalar [TEK84]. Como es conocido, dada una función  $F(x)$  en el intervalo  $[-1, 1]$ , las aproximaciones polinómicas de Chebyshev son óptimas, en el sentido de que el máximo error en la aproximación es mínimo en comparación con casi todas las posibles aproximaciones polinómicas [SM03]. Para aplicar este procedimiento, se tiene que disponer de los valores propios extremos  $E_{\min}$  y  $E_{\max}$  de  $H$ . Entonces para una expansión truncada de Chebyshev de  $\exp(-ix)$  en el intervalo  $[\tau E_{\min}, \tau E_{\max}]$  se tendrá

$$\exp(ix) \approx \sum_{n=0}^m c_n P_n(x),$$

donde

$$P_n(x) = T_n \left( \frac{2x - \tau E_{\max} - \tau E_{\min}}{\tau E_{\max} - \tau E_{\min}} \right)$$

con coeficientes  $c_n$  elegidos apropiadamente. Aquí  $T_n(x)$  son los polinomios de Chebyshev en el intervalo  $[-1, 1]$  [AS65], que se pueden determinar a través de la relación de recurrencia

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x); \quad T_1(x) = x; \quad T_0(x) = 1.$$

Por último, se utiliza la aproximación

$$\exp(-i\tau H) \approx \sum_{n=0}^m c_n P_n(\tau H). \quad (\text{A.1})$$

Esta técnica se utiliza con frecuencia en el tratamiento numérico de problemas en Mecánica Cuántica para calcular  $\exp(-i\tau H)\psi_0$  a tiempos muy largos. Esto se puede hacer con  $m$  productos matriz/vector, si en la aproximación (A.1) se considera hasta grado  $m$ . De hecho, el grado  $m$  necesario para lograr una precisión específica depende linealmente del tamaño de paso  $\tau$  y el radio espectral de  $H$  [Net00].

En una implementación práctica,  $m$  puede ser elegido de tal manera que la precisión sea dominada por el error de redondeo [LBC<sup>+</sup>91]. Este enfoque tiene dos inconvenientes principales: (i) la aproximación resultante no es unitaria, y por lo tanto, la norma no se conserva (aunque la desviación unitaria es muy pequeña debido a su extrema exactitud), y (ii) no se obtienen resultados intermedios, ya que por lo general  $\tau$  es muy grande.

## A.2. Métodos basados en espacios de Krylov

Como ya hemos señalado, muy a menudo lo que es realmente necesario, en lugar de la exponencial de la matriz  $C$  en sí, es el cálculo de  $\exp(C)$  aplicado a un vector. En esta situación, la evaluación  $e^C$  es de alguna manera análoga a computar  $C^{-1}$  para obtener el solución del sistema lineal de ecuaciones  $Cx = b$  para distintos  $b$ : otros procedimientos son claramente mucho más deseables. El cálculo de  $e^C v$  se puede hacer de manera eficiente con los métodos de Krylov, en el que aproximaciones a la solución se obtienen en los espacios de Krylov generados por los vectores  $\{v, Cv, C^2v, \dots, C^jv\}$  para algún  $j$  que es típicamente pequeño en comparación con la dimensión de  $C$  [DK95]. El método de Lanczos para resolver iterativa y simétricamente problemas de valores propios es de esta forma [Wat02]. Si, como antes,  $C = -i\tau H$ , el proceso simétrico de Lanczos genera de forma recursiva una base ortonormal  $V_m = [v_1 \cdots v_m]$  del  $m$ -ésimo subespacio de Krylov, que podemos escribir como  $K_m(H, u) = \langle u, Hu, \dots, H^{m-1}u \rangle$  tal que

$$HV_m = V_m L_m + [0 \cdots 0 \beta_m, v_{m+1}],$$

donde la matriz simétrica  $m \times m$  tridiagonal  $L_m = V_m^T H V_m$  es la proyección ortogonal de  $H$  en  $K_m(H, u)$ . Finalmente,

$$\exp(-i\tau H)u \approx V_m \exp(-i\tau L_m) V_m^T u$$

y la matriz exponencial  $\exp(-i\tau L_m)$  se puede calcular diagonalizando  $L_m$ ,  $L_m = Q_m D_m Q_m^T$ , como

$$\exp(-i\tau L_m) = Q_m \exp(-i\tau D_m) Q_m^T,$$

con  $D_m$  una matriz diagonal. Este proceso iterativo se detiene cuando

$$\beta_m \left\| (\exp(-i\tau L_m))_{m,m} \right\| < tol$$

para una tolerancia fija. A menudo se obtienen muy buenas aproximaciones cuando existen valores relativamente pequeños de  $m$  y cotas de error computables para la aproximación. Esta clase de esquemas requieren generalmente  $\mathcal{O}(n^2)$  operaciones en coma flotante para el cálculo de  $e^C v$  si  $C$  es una matriz  $n \times n$ . Más detalles están contenidos en la referencia [HL99].

## A.3. Métodos de splitting

Existe una gran variedad de métodos basados en descomponer la matriz de manera tal que el cálculo de la exponencial de cada una de las partes sea

más simple que el cálculo de la exponencial de la matriz original. Es decir, se aplica la consabida premisa muy usada en la computación numérica, *dividir para vencer*. En las líneas siguientes comentamos con más detalle un esquema de este tipo.

Dado  $A \in \mathfrak{g}$ , queremos aproximar  $e^{tA}$  como  $e^{tA} \approx R(tA) = e^{tB_1} e^{tB_2} \dots e^{tB_s}$ , donde las matrices  $B = \{B_1, B_2, \dots, B_s\}$  están sujetas a los siguientes requisitos:

- Cada  $B_l \in \mathfrak{g}$ .
- Cada exponencial  $\exp(tB_i)$  es fácil de calcular y además el producto de estas exponenciales es relativamente barato.
- El error es suficientemente pequeño,  $R(tA) = e^{tA} + \mathcal{O}(t^{p+1})$ .

Una posibilidad consiste en descomponer la matriz  $A$  de manera que cada  $B_i$  sea una matriz de rango bajo. Sean  $K$  y  $L$  dos matrices  $N \times r$ , con  $r \geq 1$  pequeño. Supongamos que

$$e^{tA} = I + tKD^{-1}(e^{tD} - I)L^T,$$

donde  $D = L^TK$ . Se debe tener en cuenta que  $D$  es de dimensión  $r \times r$  y el coste de evaluar  $e^{tA}$  es moderado para valores pequeños de  $r$ .

Como ejemplo, consideremos  $\mathfrak{g} = \mathfrak{so}(n)$ . Hacemos  $r = 2$ ,  $s = n - 1$ , donde  $e_k \in \mathbb{R}^n$  es el  $k$ -ésimo vector unitario. Dejando  $B^{[1]} = B^{[0]} - B_1$ , observamos que su primera fila y columna desaparecen. Seguimos en una forma similar a la factorización típica LU, dejando  $B^{[i]} = B^{[i-1]} - B_i$  y

$$B_i = b_i^{[i-1]} e_i^T - e_i b_i^{[i-1]} \in \mathfrak{so}(n), \quad i = 1, 2, \dots, n - 1.$$

Celledoni y Iserles [CI00] muestran la estimación precisa de los costos, detalles de la implementación y un ejemplo similar para  $\mathfrak{sl}(n)$ , así como un ejemplo, de nuevo con  $r = 2$ , que elimina dos filas y columnas de  $B \in \mathfrak{so}(n)$  a la vez.

La principal desventaja de los métodos de splitting de bajo rango es la calidad de la aproximación: en general, podemos esperar orden  $p = 1$ . La condición de segundo orden es

$$\sum_{k=1}^{s-1} \sum_{l=k+1}^s [B_k, B_l] = 0$$

y es fácil de verificar que se cumple cuando  $R(tA)$  es precisamente el splitting de Strang, es decir,  $s = 2\bar{s} + 1$  y  $B_{\bar{s}+i} = B_{\bar{s}-i}$ . En principio, es fácil convertir cualquier matriz de rango bajo de modo que se convierta en un

splitting de Strang, en primer lugar una aproximación  $\frac{1}{2}A$  con una división de primer orden  $R(tA)$ , al lado de la aproximación de la misma matriz con la misma división pero con las matrices dispuestas en orden inverso y finalmente agregar el centro de los dos términos. El resultado es

$$S(tA) = e^{tB_1/2} e^{tB_2/2} \dots e^{tB_{s-1}/2} e^{tB_s/2} \dots e^{tB_2/2} e^{tB_1/2}.$$



# Apéndice B

## Métodos de Runge–Kutta

La idea general de los métodos de Runge–Kutta (RK) es producir una fórmula para  $\Phi(t_n, x_n; h)$  en el integrador

$$y_{n+1} = y_n + h\Phi(t_n, y_n; h) \quad (\text{B.1})$$

para la ecuación diferencial

$$\frac{dy}{dt} = f(t, y), \quad y(0) = y_0 \quad (\text{B.2})$$

de manera que (B.1) coincida con el desarrollo de Taylor de la solución exacta  $y(t)$  de (B.2) hasta términos de orden  $\mathcal{O}(h^r)$ ,  $r > 1$ , *sin tener que calcular derivadas de  $f$* , sino sólo reevaluando  $f$  en puntos intermedios entre  $(t_n, y_n)$  y  $(t_{n+1}, y_{n+1})$ . Dos ejemplos sencillos de esta clase de métodos son los siguientes:

- Método de Euler modificado:

$$y_{n+1} = y_n + hf \left( t_n + \frac{1}{2}h, y_n + \frac{1}{2}hf(t_n, y_n) \right). \quad (\text{B.3})$$

- Método de Euler mejorado:

$$y_{n+1} = y_n + \frac{1}{2}h (f(t_n, y_n) + f(t_n + h, y_n + hf(t_n, y_n))). \quad (\text{B.4})$$

Ambos son de orden 2. Tal vez uno de los métodos más usados en la práctica dentro de esta clase es el *método de Runge–Kutta clásico de orden cuatro*. En él la función  $\Phi$  es un promedio con pesos de cuatro evaluaciones de la función  $f$ , donde los argumentos en cada evaluación dependen de la precedente. Específicamente, el esquema es como sigue:

$$y_{n+1} = y_n + \frac{1}{6}h(k_1 + 2k_2 + 2k_3 + k_4), \quad (\text{B.5})$$

con

$$\begin{aligned}
 k_1 &= f(t_n, y_n) \\
 k_2 &= f\left(t_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_1\right) \\
 k_3 &= f\left(t_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_2\right) \\
 k_4 &= f(t_n + h, y_n + hk_3).
 \end{aligned} \tag{B.6}$$

Más en general, dado un entero  $s$  (el número de etapas) y los coeficientes reales  $a_{ij}$ ,  $b_i$ ,  $c_i$  ( $i = 1, \dots, s$ ,  $j = 1, \dots, i - 1$ ), el método

$$\begin{aligned}
 k_1 &= f(t_n, y_n) \\
 k_2 &= f(t_n + c_2h, y_n + ha_{21}k_1) \\
 k_3 &= f(t_n + c_3h, y_n + h(a_{31}k_1 + a_{32}k_2)) \\
 \dots &\quad \dots \\
 k_s &= f(t_n + c_sh, y_n + h(a_{s1}k_1 + \dots + a_{s,s-1}k_{s-1})) \\
 x_{n+1} &= x_n + h(b_1k_1 + \dots + b_s k_s)
 \end{aligned} \tag{B.7}$$

se llama esquema de Runge–Kutta explícito de  $s$  etapas. Usualmente (pero no siempre)  $c_i = \sum_{j=1}^{i-1} a_{ij}$ , de manera que todos los puntos donde se evalúa  $f$  son aproximaciones de primer orden a la solución.

Se pueden considerar métodos todavía más generales (implícitos) para la ecuación diferencial (B.2) permitiendo que  $j = 1, \dots, s$  en la expresión (B.7). Así se tiene la clase general de métodos de Runge–Kutta de  $s$  etapas caracterizado por los números reales  $a_{ij}$ ,  $b_i$  ( $i, j = 1, \dots, s$ ) y  $c_i = \sum_{j=1}^s a_{ij}$  como

$$\begin{aligned}
 k_i &= f\left(t_n + c_ih, y_n + h \sum_{j=1}^s a_{ij}k_j\right), \quad i = 1, \dots, s \\
 y_{n+1} &= y_n + h \sum_{i=1}^s b_i k_i.
 \end{aligned} \tag{B.8}$$

Normalmente, se suele expresar el método a través del llamado *tablero de Butcher* como sigue:

$$\begin{array}{c|ccc}
 c_1 & a_{11} & \dots & a_{1s} \\
 \vdots & \vdots & & \vdots \\
 c_s & a_{s1} & \dots & a_{ss} \\
 \hline
 & b_1 & \dots & b_s
 \end{array} \tag{B.9}$$

Si el método es explícito ( $a_{ij} = 0$ ,  $j \geq i$ ), los coeficientes  $a_{ij}$  nulos (en la parte triangular superior del tablero) se suelen omitir. Con la notación

(B.9), el método de Runge–Kutta de orden 4 (B.5)-(B.6) se puede expresar como

$$\begin{array}{c|cccc}
 0 & & & & \\
 \frac{1}{2} & \frac{1}{2} & & & \\
 \frac{1}{2} & 0 & \frac{1}{2} & & \\
 1 & 0 & 0 & 1 & \\
 \hline
 & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6}
 \end{array} \tag{B.10}$$



# Apéndice C

## Código expmS()

El código **Matlab** aquí incluido es una adaptación del método de *splitting* que permite el cálculo aproximado de  $e^A$ , siendo  $A$  una matriz  $n \times n$  de números reales con traza cero.

```
function ysol4=exp_splitting(A)
% Aproxima exp(A), A de traza cero
% using the new method of ORDER 4 based on
% separability + splitting
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Dimension of the matrix

n=length(A0);
%n=3;
% Initial time
t0=0;
% Final time
tfinal = 1;
% Step size
%h=2e-2;
%h=0.1;
h=0.001;

% Matrices to be used
ysol4 = [];
y0 = [];
error = [];
tot = [];
err = [];
```

```

time = [];
comp1 = [];
comp2 = [];
%A0 = [];
A1 = [];
A2 = [];
LOh = [];
LOh2 = [];
FO = [];
Fh2 = [];
Fh = [];
pro1 = [];
CO0 = [];
COh2 = [];
COh = [];
U0h2 = [];
U0h = [];
Gh2 = [];
Gh = [];
AA10 = [];
AA1h2 = [];
AA1h = [];
pro1 = [];
C10 = [];
C1h2 = [];
C1h = [];
U1h = [];
MP4 = [];
MP4es = [];
MP4re = [];

%number of steps
nsteps=floor((tfinal-t0)/h + eps);

% Initial conditions
ysol4 = eye(n);

h=(tfinal-t0)/nsteps;

```

```

h2 = h/2;
h6= h/6;
h24= h/24;

t = t0;

C00=triu(A0)-diag(diag(A0));
AA10=tril(C00)-diag(diag(C00));
C10=triu(AA10)-diag(diag(AA10));

for i2 = 1:nsteps;

    y0=ysol4;
    %Evaluation of the matrices at the NC quadratures
    t1 = t + h2;
    t2 = t + h;

    A1=A0;
    A2=A0;

    % The iteration begins: L0(h/2) and L0(h)
    for i=1:n;
        L0h(i,i)=exp(A0(i,i)*h);
        L0h2(i,i)=exp(A0(i,i)*h2);
        for j=1:i-1;
            s1=0;
            s2=0;
            for k=j:i-1;
                s1=s1+A1(i,k)*L0h2(k,j);
                s2=s2+A2(i,k)*L0h(k,j);
            end
            Fh2(i,j)=s1*exp(-A0(i,i)*h2);
            Fh(i,j)=s2*exp(-A0(i,i)*h);

            L0h2(i,j)=L0h2(i,i)*h24*(5*A0(i,j)+8*Fh2(i,j)-Fh(i,j));
            L0h(i,j)=L0h(i,i)*h6*(A0(i,j)+4*Fh2(i,j)+Fh(i,j));
        end
    end
end

```

```

% Matrices C0(h/2), C0(h)
pro1=(triu(A1)-diag(diag(A1)))*L0h2;
C0h2=L0h2\pro1;
pro1=(triu(A2)-diag(diag(A2)))*L0h;
C0h=L0h\pro1;

% U0(h/2), U0(h)
for i=n:-1:1;
    CCh=h6*(C00(i,i)+4*C0h2(i,i)+C0h(i,i));
    U0h(i,i)=exp(CCh);
    CCh2=h24*(5*C00(i,i)+8*C0h2(i,i)-C0h(i,i));
    U0h2(i,i)=exp(CCh2);
    for j=i+1:n;
        s1=0;
        s2=0;
        for k=i+1:j;
            s1=s1+C0h2(i,k)*U0h2(k,j);
            s2=s2+C0h(i,k)*U0h(k,j);
        end
        Gh2(i,j)=exp(-CCh2)*s1;
        Gh(i,j)=exp(-CCh)*s2;
        U0h2(i,j)=U0h2(i,i)*h24*(5*C00(i,j)+8*Gh2(i,j)-Gh(i,j));
        U0h(i,j)=U0h(i,i)*h6*(C00(i,j)+4*Gh2(i,j)+Gh(i,j));
    end
end
end

% Matrices A1(0), A1(h/2), A1(h)
pro1=(tril(C0h2)-diag(diag(C0h2)))*U0h2;
AA1h2=U0h2\pro1;
pro1=(tril(C0h)-diag(diag(C0h)))*U0h;
AA1h=U0h\pro1;

%% Second cycle begins...
% L1(h/2) and L1(h)
for i=1:n;
    AMh=h6*(AA10(i,i)+4*AA1h2(i,i)+AA1h(i,i));
    L1h(i,i)=exp(AMh);
    AMh2=h24*(5*AA10(i,i)+8*AA1h2(i,i)-AA1h(i,i));
    L1h2(i,i)=exp(AMh2);

```



```

for j=1:i-1;
    s1=0;
    s2=0;
    for k=j:i-1;
        s1=s1+AA1h2(i,k)*L1h2(k,j);
        s2=s2+AA1h(i,k)*L1h(k,j);
    end
    Fh2(i,j)=exp(-AMh2)*s1;
    Fh(i,j)=exp(-AMh)*s2;
    L1h2(i,j)=L1h2(i,i)*h24*(5*AA10(i,j)+8*Fh2(i,j)-Fh(i,j));
    L1h(i,j)=L1h(i,i)*h6*(AA10(i,j)+4*Fh2(i,j)+Fh(i,j));
end
end

```

%% Es una aproximacion de orden 2

% Matrices C1(h/2), C1(h)

```

pro1=(triu(AA1h2)-diag(diag(AA1h2)))*L1h2;
C1h2=L1h2\pro1;

```

```

pro1=(triu(AA1h)-diag(diag(AA1h)))*L1h;
C1h=L1h\pro1;

```

% U1(h)

```

for i=n:-1:1;
    CCh=h6*(C10(i,i)+4*C1h2(i,i)+C1h(i,i));
    U1h(i,i)=exp(CCh);
    CCh2=h24*(5*C10(i,i)+8*C1h2(i,i)-C1h(i,i));
    U1h2(i,i)=exp(CCh2);
    for j=i+1:n;
        s1=0;
        s2=0;
        for k=i+1:j;
            s1=s1+C1h2(i,k)*U1h2(k,j);
            s2=s2+C1h(i,k)*U1h(k,j);
        end
        Gh2(i,j)=exp(-CCh2)*s1;
        Gh(i,j)=exp(-CCh)*s2;
        U1h(i,j)=U1h(i,i)*h6*(C10(i,j)+4*Gh2(i,j)+Gh(i,j));
    end
end

```

```

        end
    end

% Final Result
    ysol4 =L0h*U0h*L1h*U1h*y0;

%   ym4=y0;

% t = t +h;

C00=triu(A0)-diag(diag(A0));
    AA10=tril(C00)-diag(diag(C00));
    C10=triu(AA10)-diag(diag(AA10));

end %%nsteps

    format long; ysol4
return

end

```

# Bibliografía

- [AS65] M. Abramowitz and I.A. Stegun, *Handbook of Mathematical Functions*, Dover, 1965.
- [Bak05] H.F. Baker, *Alternant and continuous groups*, Proc. London Math. Soc. (Second series) **3** (1905), 24–47.
- [Bak02] A. Baker, *Introduction to Lie Group Theory*, Springer, 2002.
- [Bau81] H.J. Baues, *Commutator Calculus and Groups of Homotopy Classes*, Cambridge University Press, 1981.
- [Bay79] F. Bayen, *On the convergence of the Zassenhaus formula*, Lett. Math. Phys. **3** (1979), 161–167.
- [BC04] S. Blanes and F. Casas, *On the convergence and optimization of the Baker–Campbell–Hausdorff formula*, Linear Algebra Appl. **378** (2004), 135–158.
- [BCOR09] S. Blanes, F. Casas, J.A. Oteo, and J. Ros, *The Magnus expansion and some of its applications*, Phys. Rep. **470** (2009), 151–238.
- [BCR02] S. Blanes, F. Casas, and J. Ros, *High order optimized geometric integrators for linear differential equations*, BIT **42** (2002), 262–284.
- [Bel65] R. Bellman, *Introducción al Análisis Matricial*, Editorial Reverté, 1965.
- [Bos89] A. Bose, *Dynkin’s method of computing the terms of the Baker–Campbell–Hausdorff series*, J. Math. Phys. **30** (1989), 2035–2037.
- [Bou89] N. Bourbaki, *Lie groups and Lie algebras*, Chapters 1–3, Springer, 1989.

- [Bur81] D.P. Burum, *Magnus expansion generator*, Phys. Rev. B **24** (1981), 3684–3692.
- [Cam97] J.E. Campbell, *On a law of combination of operators bearing on the theory of continuous transformation groups*, Proceedings of the London Mathematical Society **28** (1897), 381–390.
- [Cam98] ———, *On a law of combination of operators (second paper)*, Proceedings of the London Mathematical Society **29** (1898), 14–32.
- [Cas07] F. Casas, *New numerical integrators based on solvability and splitting*, J. Comput. Appl. Math. **205** (2007), no. 2, 802 – 813.
- [Cer74] M. Cernusca, *Lösung der formalen Potenzreihen  $e^Ae^B = e^X$  und  $e^{A+Bt} = e^Ae^X$* , Monat. Math. **78** (1974), 4–14.
- [CI00] E. Celledoni and A. Iserles, *Approximating the exponential from a Lie algebra to a Lie group*, Math. Comput. **69** (2000), 1457–1480.
- [CI06] F. Casas and A. Iserles, *Explicit Magnus expansions for nonlinear equations*, J. Phys. A: Math. Gen. **39** (2006), 5445–5461.
- [CL69] D. Chang and J.C. Light, *Exponential solution of the Schrödinger equation: potential scattering*, J. Chem. Phys. **50** (1969), 2517–2525.
- [CM09] F. Casas and A. Murua, *An efficient algorithm for computing the Baker–Campbell–Hausdorff series and some of its applications*, J. Math. Phys. **50** (2009), 033513.
- [CMN12] F. Casas, A. Murua, and M. Nadinic, *Efficient computation of the Zassenhaus formula*, Comp. Phys. Commun. **183** (2012), no. 11, 2386 – 2391.
- [C.R93] C.Reutenauer, *Free Lie Algebras*, Oxford University Press, 1993.
- [Dev85] P.L. Devries, *A high order method for the solution of the single channel Schrödinger equation*, Molecular Physics **54** (1985), 733–740.
- [DF76] A. J. Dragt and J. M. Finn, *Lie series and invariant functions for analytic symplectic map*, J. Math. Phys. **17** (1976), 2215–2227.

- [DK95] V.L. Druskin and L.A. Knizhnerman, *Krylov subspace approximation of eigenpairs and matrix functions in exact and computer arithmetics*, Numer. Linear Algebra **2** (1995), 205–217.
- [Dyn47] E.B. Dynkin, *Evaluation of the coefficients of the Campbell–Hausdorff formula*, Dokl. Akad. Nauk. SSSR **57** (1947), 323–326.
- [Dys49] F.J. Dyson, *The radiation theories of Tomonaga, Schwinger, and Feynman*, Phys. Rev. **75** (1949), 486–502.
- [Gil74] R. Gilmore, *Lie groups, Lie algebras, and some of their Applications*, Dover, 1974.
- [GL96] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, 1996.
- [GL07] K. S. Govinder and P. G. L. Leach, *Integrability analysis of the Emden–Fowler equation*, J. Nonlinear Math. Phys. **14** (2007), 443–461.
- [Gol56] K. Goldberg, *The formal power series for  $\log(e^x e^y)$* , Duke Math. J. **23** (1956), 13–21.
- [GOV97] V.V. Gorbatsevich, A.L. Onishchik, and E.B. Vinberg, *Foundations of Lie Theory and Lie Transformation Groups*, Springer, 1997.
- [GS02] J.P. García-Seguí, *Grups i àlgebres de Lie en la integració d'equacions diferencials*, Master's thesis, Universitat Jaume I, 2002.
- [Hau06] F. Hausdorff, *Die symbolische exponential formed in der gruppen theorie*, Leipziger Ber. **58** (1906), 19–48.
- [HJ85] R.A. Horn and C.R. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.
- [HL99] M. Hochbruck and C. Lubich, *Exponential integrators for quantum-classical molecular dynamics*, BIT **39** (1999), 620–645.
- [HLW06] E. Hairer, Ch. Lubich, and G. Wanner, *Geometric Numerical Integration: Structure Preserving Algorithms for ordinary differential equations*, second ed., Springer, 2006.

- [IN99] A Iserles and S. P. Nørsett, *On the solution of linear differential equations in Lie groups*, Phil. Trans. R. Soc. London A: Math. Phys. Eng. Sciences **357** (1999), no. 1754, 983–1019.
- [Jac79] N. Jacobson, *Lie algebras*, Dover, 1979.
- [Kol93] M. Kolsrud, *Maximal reductions in the Baker-Hausdorff formula*, J. Math. Phys. **34** (1993), no. 1, 270–285.
- [Kos93] P. V. Koseleff, *Calcul formel pour les méthodes de Lie en mécanique hamiltonienne*, Ph.D. thesis, École Polytechnique, 1993.
- [Kum65] K. Kumar, *On expanding the exponential*, J. Math. Phys **6** (1965), 1928–1934.
- [Kur07] V. Kurlin., *The Baker–Campbell–Hausdorff formula in the free metabelian Lie algebra*, J. Lie Theory **17** (2007), 525–538.
- [Las88] F. M. Lasagni, *Canonical Runge-Kutta methods*, ZAMP **39** (1988), 952–953.
- [LBC<sup>+</sup>91] C. Leforestier, R.H. Bisseling, C. Cerjan, M.D. Feit, R. Friesner, A. Guldberg, A. Hammerich, G. Jolicard, W. Karrlein, H.-D. Meyer, N. Lipkin, O. Roncero, and R. Kosloff, *A comparison of different propagation schemes for the time dependent Schrödinger equation*, J. Comput. Phys. **94** (1991), 59–80.
- [Mag54] W. Magnus, *On the exponential solution of differential equations for a linear operator*, Comm. Pure Appl. Math. **7** (1954), no. 4, 649–673.
- [Mic74] J. Michel, *Bases des algèbres de Lie et série de Hausdorff*, Séminaire Dubreil. Algèbre **27** (1974), 1–9.
- [MK98] H. Munthe-Kaas, *Runge–Kutta methods on Lie groups*, BIT Numerical Mathematics **38** (1998), 92–111.
- [ML03] C. Moler and C. Van Loan, *Nineteen dubious to compute the exponential of a matrix, twenty-five years later*, SIAM Rev. **45** (2003), 5074–5076.
- [Mur06] A. Murua, *The Hopf algebra of rooted trees, free Lie algebras, and Lie series*, Found. Comput. Math. **6** (2006), 387–426.

- [Net00] P. Nettesheim, *Mixed quantum-classical dynamics: a unified approach to mathematical modelling and numerical simulation*, Ph.D. thesis, Freie Universität Berlin, 2000.
- [NT87] M. Newman and R.C. Thompson, *Numerical values of Goldberg's coefficients in the series for  $\log(e^x e^y)$* , *Math. Comput.* **48** (1987), 265–271.
- [Ote91] J.A. Oteo, *The Baker-Campbell-Hausdorff formula and nested commutator identities*, *J. Math. Phys.* **32** (1991), no. 2, 419–424.
- [Pos94] M. Postnikov, *Lie Groups and Lie Algebras. Semester V of Lectures in Geometry*, URSS Publishers, 1994.
- [Rei00] M.W. Reinsch, *A simple expression for the terms in the Baker–Campbell–Hausdorff series*, *J. Math. Phys.* **41** (2000), 2434–2442.
- [RG65] R.D. Richtmyer and S. Greenspan, *Expansion of the Campbell–Baker–Hausdorff formula by computer*, *Comm. Pure Appl. Math.* **18** (1965), 107–108.
- [Sal87] W.R. Salzman, *New criterion for convergence of exponential perturbation theory in the Schrödinger representation*, *Phys. Rev. A* **36** (1987), 5074–5076.
- [SM03] E. Süli and D. Mayers, *An Introduction to Numerical Analysis*, Cambridge University Press, 2003.
- [SS88] J. M. Sanz-Serna, *Runge-Kutta schemes for Hamiltonian systems*, *BIT* **28** (1988), 877–883.
- [SS99] A.T. Sornborger and E.D. Stewart, *Higher-order methods for simulations on quantum computers*, *Phys. Rev. A* **60** (1999), 1965–1999.
- [Sur88] Y. B. Suris, *Preservation of symplectic structure in the numerical solution of Hamiltonian systems. en Numerical Solution of Differential Equations*, Filippov, S. S., ed. Akad. Nauk.SSSR, Ins. Prikl. Mat. Moscow, (1988), 148–160.
- [Suz77] M. Suzuki, *On the convergence of exponential operators the Zassenhaus formula, BCH formula and systematic approximants*, *Comm. Math. Phys.* **57** (1977), 193–200.

- [SW06] D. Scholz and M. Weyrauch, *A note on the Zassenhaus product formula*, J. Math. Phys. **47** (2006), 033505.
- [TEK84] H. Tal-Ezer and R. Kosloff, *An accurate and efficient scheme for propagating the time dependent Schrödinger equation*, J. Chem. Phys. **81** (1984), 3967–3971.
- [Tho82] R.C. Thompson, *Cyclic relations and the Goldberg coefficients in the Campbell–Baker–Hausdorff formula*, Proc. Amer. Math. Soc. **86** (1982), 12–14.
- [Tho89] ———, *Convergence proof for Goldberg’s exponential series*, Linear Algebra Appl. **121** (1989), 3–7.
- [TTM05] M. Torres-Torriti and H. Michalska, *A software package for Lie algebraic computations*, SIAM Rev. **47** (2005), 722–745.
- [Tyr97] E.E. Tyrtysnikov, *A Brief Introduction to Numerical Analysis*, Birkhäuser, 1997.
- [Var84] V. S. Varadarajan, *Lie Groups, Lie Algebras, and Their Representations*, Springer-Verlag, 1984.
- [VC04] L.M.K. Vandersypen and I. L. Chuang, *NMR techniques for quantum control and computation*, Rev. Mod. Phys. **76** (2004), 1037–1069.
- [Vie78] X.G. Viennot, *Algèbres de Lie libres et monoïdes libres*, Springer, 1978.
- [Vol68] H.C. Volkin, *Iterated commutators and functions of operators*, Tech. Report D-4857, NASA, 1968.
- [Vos72] D. Voslamber, *On exponential approximations for the evolution operator*, J. Math. Anal. Appl. **37** (1972), no. 2, 403 – 411.
- [War77] R. C. Ward, *Numerical computation of the matrix exponential with accuracy estimate*, SIAM J. Numer. Anal. **14** (1977), 600–610.
- [Wat02] D.S. Watkins, *Fundamentals of Matrix Computations*, 2nd ed., John Wiley & Sons, 2002.
- [Wil67] R. M. Wilcox, *Exponential operators and parameter differentiation in quantum physics*, J. Math. Phys. **8** (1967), no. 4, 962–982.



- [Wit75] W. Witschel, *Ordered operator expansions by comparison*, J. Phys. A: Math. Gen. **8** (1975), 143–155.
- [WM62] G.H. Weiss and A.A. Maradudin, *The Baker–Hausdorff formula and a problem in Crystal Physics*, J. Math. Phys. **3** (1962), 771–777.
- [Won75] J.S.W. Wong, *On the generalized Emden–Fowler equation*, SIAM Rev. **17** (1975), 339–360.
- [WS09] M. Weyrauch and D. Scholz, *Computing the Baker–Campbell–Hausdorff series and the Zassenhaus product*, Comput. Phys. Comm. **180** (2009), 1558–1565.
- [Yos90] H. Yoshida, *Construction of higher order symplectic integration*, Phys Lett. A **150** (1990), 262–268.
- [Zwi97] D. Zwillinger, *Handbook of Differential Equations*, Academic Press, 1997.