Integración numérica de sistemas diferenciales rígidos y de tipo oscilatorio

Rogel Rafael Rojas Bello

6 de octubre de 2008

AGRADECIMIENTO

A Dios por darme la vida, a mi tierna esposa, a mis hijos, a mis padres y hermanos por el amor que siempre me han mostrado y que me nutre.

Agradezco profundamente a mi director de tesis, el Doctor Severiano González Pinto (Seve), por sus enseñanzas y por dedicar sus conocimientos y su valioso tiempo que han hecho posible la culminación de esta memoria.

Deseo expresar también mi agradecimiento a todos los miembros del Departamento de Análisis Matemático y a su secretario Pablo, que siempre me han animado y me han prestado su desinteresada ayuda.

Agradezco también con especial cariño, a mis compañeros los venezolanos, a Domingo Hernández Abreu (Mingo) y a Francisco Perdomo Pío, por aquellos buenos ratos que pasamos.

Índice general

1. Introducción **6**

1.1. Implementación de métodos RK en problemas de primer orden 9

1.2. Implementación de métodos Runge-Kutta Nyström (RKN) en problemas especiales de segundo orden 11

2. Aceleración de métodos iterativos tipo Newton para problemas Stiff **15**

2.1. Introducción 15

2.2. Iteración de Cooper y Butcher para métodos con dos y tres etapas 19

2.2.1. El caso $s = 2$ 19

2.2.2. El caso de 3 etapas implícitas ($s = 3$) 22

2.3. Aceleración de la convergencia 25

2.4. Experimentos numéricos 31

3. Métodos iterativos para problemas especiales de segundo orden **38**

3.1. Introducción 38

3.2. Alternativas a la iteración Quasi–Newton para métodos RKN 40

3.2.1. Análisis de los errores globales de los métodos al usar los esquemas iterativos Quasi–Newton y Single Newton tras μ iteraciones por paso de integración 45

3.3. Selección del esquema iterativo para los métodos de Gauss 48

3.3.1. El método de Gauss de dos etapas ($\mathbf{s = 2}$) 49

3.3.2. El método de Gauss de tres etapas ($\mathbf{s = 3}$) 51

3.3.3. El método de Gauss de cuatro etapas ($\mathbf{s = 4}$) 53

3.4. Predictores para los esquemas iterativos aplicados a los métodos de Gauss . . 54

3.5. VOS: Estrategia del Orden Variable para seleccionar los predictores 59

3.6. Experimentos numéricos 60

4. Un código basado en el método de Gauss de 2 etapas para problemas de segundo orden 68

4.1. Introducción 68
4.2. Solución de las ecuaciones de etapas 70
4.2.1. Predictores para las etapas internas 73
4.3. Estimadores del Error Local para el método de Gauss de dos etapas 74
4.3.1. Los Estimadores del Error Local sobre problemas lineales 79
4.4. Selección del tamaño de paso inicial para Gauss de 2 etapas 90
4.5. La estimación del error global y la salida densa 93
4.6. Código a paso variable 97
4.7. Experimentos numéricos 103
4.8. Conclusiones 124

A. Conclusiones e Investigación Futura 126

B. Anexo sobre Problemas especiales de segundo orden 129

Capítulo 1

Introducción

En esta memoria abordaremos dentro del contexto de las Ecuaciones Diferenciales Ordinarias, aspectos relativos a la resolución numérica de Problemas de Valor Inicial. En el caso de problemas de primer orden consideraremos algunos aspectos relacionados con la resolución de problemas rígidos o de tipo Stiff y dentro de la clase de problemas de segundo orden, consideraremos principalmente problemas de tipo oscilatorio o vibratorio provenientes de la Ingeniería, Mecánica Clásica, Astronomía, Ondas, Mecánica Cuántica, etc. En cuanto a los métodos numéricos a considerar nos centraremos esencialmente en aquellos de la clase Runge-Kutta en caso de problemas de primer orden y en aquellos de tipo Runge-Kutta-Nyström para el caso de los problemas de segundo orden. Además, como queremos contemplar la opción de que el tipo de problema a tratar pueda comportar rigidez (Stiffness), nos centraremos principalmente en ciertas clases de métodos implícitos con buenas propiedades de estabilidad y que posean un orden de convergencia relativamente alto. Así para el caso de problemas de primer orden, consideramos fundamentalmente fórmulas de la familias RK–Gauss , Radau y Lobatto [8, 11, 25, 57], mientras que para el caso de los problemas de segundo orden nos centraremos esencialmente en los métodos de Gauss en versión Runge-Kutta-Nyström [55, 56, 76].

Los sistemas diferenciales de primer orden de tipo rígido (Stiff) describen importantes fenómenos en las Ciencias Aplicadas, tales como [57, 66]: circuitos eléctricos, combinaciones de reacciones violentas y suaves en Química Cinética, Ecuaciones de Advección-Difusión-Reacción en Ecuaciones en Derivadas Parciales, Problemas con Perturbaciones Singulares, etc.

La clase de problema Stiff no es fácil de caracterizar mediante una definición Matemática

rigurosa, sino más bien se va pasando de la clase no-Stiff a la clase Stiff de forma gradual del mismo modo que al contar vamos pasando de los números más pequeños a los más grandes. La clase de problemas en cuestión, está más bien relacionada con el comportamiento de los métodos numéricos cuando se aplican sobre la clase. Así entre los analistas numéricos un problema comporta Stiffness cuando los métodos numéricos con carácter explícito no lo integran satisfactoriamente en comparación con cierta clase de métodos implícitos. De un modo más Matemático podemos decir que los problemas de tipo Stiff, se caracterizan por admitir una solución *suave* (sin picos ni cambios bruscos) en la mayor parte del intervalo de integración y por amortiguar rápidamente las soluciones vecinas al perturbar las condiciones iniciales, esto es las soluciones próximas tienden rápidamente a la solución *suave* del problema, después de una fase de transición de longitud muy pequeña. Esto se suele traducir (de forma no totalmente rigurosa) en que la matriz Jacobiana de la ecuación variacional del problema, posee autovalores con parte real de tamaño moderado (no hay ningún autovalor con parte real positiva grande) y que alguno de ellos tiene parte real fuertemente negativa, para una discusión más detallada sobre esta clase de problemas, véase por ejemplo los capítulos 1 de los textos [24, 57].

La mayoría de los códigos existentes en la actualidad para la resolución numérica de problemas Stiff o de carácter Oscilatorio están basados esencialmente en métodos Multipaso, métodos de Extrapolación o métodos de tipo Runge–Kutta.

Métodos Multipaso. Los métodos Lineales Multipaso han sido quizás los métodos más populares para la integración de problemas Stiff y también para Ecuaciones Diferenciales Algebraicas, gracias a los trabajos de Curtiss y Hirschfelder (1952) y esencialmente de Gear [35], quienes descubrieron las fórmulas BDF (Backward Differentiation Formulae) y sus excelentes propiedades de estabilidad lineal, dentro de la familia de las fórmulas Lineales Multipaso. Basado en estas fórmulas, Gear [35] diseña el código DIFSUB (1971), el cual ha sido, junto con sus sucesores, el más popular para la integración de problemas Stiff y Ecuaciones Diferenciales Algebraicas. Entre sus sucesores más notables, destacamos las variantes: LSODE de Hindmarsh [59] y VODE de Brown et al. [5]. Estos códigos han probado ser excelentes integradores, en precisiones medias y bajas, pero no son muy eficientes cuando el espectro de la matriz Jacobiana del sistema diferencial está próximo al eje imaginario, y posee componentes imaginarias de tamaño medio o significativo. *Este es el caso de los problemas de segundo orden de tipo oscilatorio, cuyo tratamiento numérico consideraremos en profundidad en esta*

Memoria en los capítulos 3 y 4.

Métodos de extrapolación. Uno de los códigos de este tipo, pioneros para problemas Stiff es METANI (1983) de Bader y Deuflhard [2]. El código está basado en la técnica de Extrapolación Local de *la regla implícita del Punto Medio* linealizada. También en la década de los 90, Hairer y Wanner [57] construyeron los códigos SODEX y SEULEX basados en extrapolación local de ciertas linealizaciones de las reglas implícitas del: *Punto Medio* y *Euler Regresivo*, respectivamente. Sin embargo estos métodos no son muy eficientes para problemas con carácter oscilatorio y por esta razón para este último tipo de problemas, Hairer, Nørsett and Wanner [55] construyeron en la década de los años 90, el código ODEX2, el cual está basado en extrapolación local de la regla de Störmer. Esta regla es un método explícito de orden dos.

Veremos en el capítulo 4 de esta memoria, que este código no es adecuado para alguna clase de problemas de tipo vibratorio, especialmente aquellos provenientes de ciertas Ecuaciones en Derivadas Parciales (EDP) importantes en la práctica.

Métodos Runge–Kutta. Esta clase de métodos de un paso han sido históricamente bastante populares para la integración de sistemas diferenciales de tipo no-stiff, debido a la simplicidad de su programación y a las buenas prestaciones dadas por algunas fórmulas de este tipo, desde las fórmulas pioneras de Runge (1895) de orden 2 y de Kutta (1905) con orden 4, hasta los códigos más recientes basados en Pares Encajados de diferentes órdenes (desde orden 3 hasta orden 8) dados por Fehlberg (1967) y de Dormand-Prince (1980). Más tarde las fórmulas dadas por los últimos autores han sido equipadas con salidas densas más eficientes, por Hairer et al. [55], ver también algunas de las referencias allí dadas. Sin embargo, los métodos explícitos de esta familia no son eficientes ni para problemas de tipo Stiff en general, ni para problemas con carácter oscilatorio dominados por las bajas frecuencias pero poseyendo frecuencias altas no significativas. Esto se debe a las relativamente pobres propiedades de estabilidad de estos métodos cuando se comparan con ciertos métodos implícitos dentro de la familia Runge-Kutta, tales como los métodos de colocación Singly-Implicit Runge-Kutta (SIRK) dados por Burrage et al. (1980) en [7], o bien los métodos de colocación basados en las cuadraturas de Gauss, Radau o Lobatto, véase por ejemplo [12, 57]. Estas últimas familias de métodos poseen además de excelentes propiedades de estabilidad (son A-estables o L-estables) altos órdenes de convergencia, lo que las hace especial-

mente atractivas para construir códigos robustos basados en esta clase de métodos. El principal inconveniente para su uso, es que son altamente implícitos, lo cual conlleva la resolución de sistemas de ecuaciones no lineales (si el problema es no lineal), a menudo de dimensión elevada, en cada paso de la integración.

En esta memoria, vamos a tratar con cierta extensión los aspectos de resolución de las ecuaciones de etapa de los métodos Runge-Kutta altamente implícitos. En particular, en el capítulo 2 vamos a considerar un tipo de iteración inspirada en los trabajos pioneros de Butcher et al. [10, 22], para resolver las ecuaciones de etapa en el caso de problemas de primer orden de tipo Stiff. En los capítulos 3 y 4 consideraremos la resolución de las ecuaciones de etapa en el caso de los problemas de segundo orden de carácter oscilatorio.

A continuación damos unas pinceladas introductorias, de los diversos aspectos a considerar en la implementación de fórmulas Runge-Kutta altamente implícitas en problemas de tipo Stiff o de tipo Oscilatorio.

1.1. Implementación de métodos RK en problemas de primer orden

A efectos de realizar la integración de Problemas de Valor Inicial (PVI) en sistemas diferenciales de primer orden

$$y'(t) = f(t, y), \quad y(t_0) = y_0, \quad y, f \in \mathbb{R}^m, \quad t \in [t_0, t_{end}],$$

un método Runge–Kutta de s etapas avanza la solución numérica un tamaño de paso h, desde el punto (t_n, y_n) hasta el punto $(t_{n+1} = t_n + h, y_{n+1})$ mediante la fórmula

$$y_{n+1} = y_n + h \sum_{j=1}^{s} b_i f(t_n + c_i h, Y_i), \tag{1.1.1}$$

donde las etapas intermedias Y_i se calculan mediante el sistema algebraico,

$$Y_i = y_n + h \sum_{j=1}^{s} a_{ij} f(t_n + c_j h, Y_j), \quad i = 1, \ldots, s. \tag{1.1.2}$$

La matriz $A = (a_{ij})$ y el vector $b = (b_i)$ se denominan los coeficientes del método, pues el vector c de nodos, verifica $Ae = c$, para los métodos de interés, donde $e = (1, \ldots, 1) \in \mathbb{R}^s$.

Usando el producto de Kronecker de matrices $A \otimes B = (a_{ij}B)$, denotando por $X_n = e \otimes y_n$, por I_m la matriz identidad de orden m y definiendo los vectores de etapa y sus derivadas respectivamente por

$$Y^T := (Y_1^T, \ldots, Y_s^T), \quad F^T := \big(f_1^T(t_n + c_1 h, Y_1), \ldots, f_s^T(t_n + c_s h, Y_s)\big),$$

entonces las ecuaciones (1.1.1) y (1.1.2) puede escribirse en forma más compacta mediante

$$y_{n+1} = y_n + h(b^T \otimes I_m)F(Y), \quad Y = X_n + h(A \otimes I_m)F(Y). \tag{1.1.3}$$

Puesto que f es en general no lineal, debemos resolver en cada paso de la integración un sistema no lineal de sm ecuaciones con otras tantas incógnitas.

El procedimiento más sencillo para abordar el sistema (1.1.2) es la Iteración Funcional Simple, pero no siempre es el más conveniente, ya que para el caso de problemas que comporten rigidez, la convergencia de la Iteración Funcional Simple conlleva tamaños de paso h excesivamente pequeños en comparación con los que de forma natural daría el integrador para obtener la exactitud demandada por el usuario. Sin embargo, los métodos iterativos de tipo Newton dan resultados satisfactorios en general sin tener que reducir los tamaños de paso en las integraciones. Un tipo de iteración de tipo Newton que da buenos resultados en general es la denominada Quasi-Newton (o Iteración de Newton Modificada). Esta fue primeramente considerada en el año 1973 por Chipman [19] para resolver las ecuaciones de etapa en (1.1.2), y puede expresarse en la forma siguiente,

$$\begin{cases} \big(I_{ms} - h(A \otimes J)\big)\Delta^k = D^k \equiv e \otimes y_n - Y^k + h(A \otimes I_m)F(Y^k) \\ Y^{k+1} = Y^k + \Delta^k, \quad k = 0, 1, \ldots, \end{cases} \tag{1.1.4}$$

donde $J \simeq \dfrac{\partial f}{\partial y}(t_n, y_n)$, es una aproximación a la matriz Jacobiana de f en el punto actual de integración.

Cada paso de la iteración ($k = 0, 1, \ldots$), conlleva la solución de un sistema lineal de dimensión sm. Para reducir el costo computacional del sistema lineal (1.1.4) Butcher [9] y Bickart [3] de forma independiente, explotan la estructura especial de la matriz $(I_{ms} - h(A \otimes J))$, introduciendo una transformación de semejanza para la submatriz A. Así, suponiendo que la matriz A^{-1} tenga un solo autovalor real y pares de autovalores complejos conjugados (esta es la situación típica de los métodos altamente implícitos), se desacoplan los ms sistemas lineales reales en $[s/2]$ sistemas lineales complejos ($[x]$ denota la parte entera del número real x) de dimensión m más un sistema lineal real de dimensión m. Estos sistemas se resuelven

normalmente usando factorizaciones LU y reduciendo cada sistema lineal a dos sistemas triangulares. De esta forma cada factorización LU puede ser usada en todas las iteraciones de un paso de integración y también sobre distintos pasos de integración consecutivos cuando la convergencia es rápida y el tamaño de paso no cambia. Esta técnica es la usada en el código RADAU5 de Hairer y Wanner [57] (1990), para resolver las ecuaciones de etapa de la fórmula Runge–Kutta Radau IIA de tres etapas y orden cinco.

Por otra parte, en la búsqueda de esquemas iterativos alternativos a la iteración Quasi-Newton arriba mencionada y que puedan ser más eficientes, Cooper y Butcher [22] propusieron en 1983 una iteración, que evita el uso de aritmética compleja y que permite reducir gastos en factorizaciones LU. A menudo estos gastos son los más significativos en problemas de dimensión media o elevada. Esta iteración puede escribirse mediante la formulación:

$$\begin{cases} \big(I_{ms} - h\gamma(I_m \otimes J)\big)E^k = (BS^{-1} \otimes I_m)D^k + (L \otimes I_m)E^k \\ Y^{k+1} = Y^k + (S \otimes I_m)E^k, \quad k = 0, 1, \ldots, \end{cases} \tag{1.1.5}$$

donde la constante $\gamma > 0$, las matrices B y S, y la matriz triangular inferior estricta L son los parámetros del esquema, que deben fijarse de antemano mediante algún criterio de optimización. El criterio usado por estos autores fue el de exigir que la iteración alcanzara una velocidad de convergencia lo más alta posible sobre ciertos problemas lineales de tipo Stiff.

En el capítulo 2 de esta memoria, trataremos ampliamente este tipo de iteración y daremos criterios de elección de los parámetros óptimos para familias de métodos Runge-Kutta altamente implícitos que son de gran interés y que no fueron considerados por Cooper y Butcher [22]. Además introduciremos algunas variaciones en el proceso iterativo, las cuales permiten acelerar la convergencia de forma sustancial.

1.2. Implementación de métodos Runge-Kutta Nyström (RKN) en problemas especiales de segundo orden

Muchos modelos en Mecánica, Astronomía e Ingeniería vienen expresados de forma natural como Problemas de Valor Inicial asociados a sistemas diferenciales de segundo orden en los que el campo de fuerzas involucradas no depende de las velocidades, es decir presentan

una formulación del tipo,

$$\begin{cases} y'' = f\big(t, y(t)\big), & t \in [t_0, t_{end}] \\ y(t_0) = y_0, \quad y'(t_0) = y_0', & y, y', f \in \mathbb{R}^m. \end{cases} \tag{1.2.1}$$

Estos problemas se pueden transformar en sistemas diferenciales de primer orden sin más que considerar como nueva variable el vector $z^T = (y, y')^T$, lo que hace que la dimensión del problema se duplique, y sea expresado como

$$z' = \begin{pmatrix} y \\ y' \end{pmatrix}' = F(t, z) := \begin{pmatrix} y' \\ f\big(t, y\big) \end{pmatrix}, \quad z(t_0) = \begin{pmatrix} y(t_0) \\ y'(t_0) \end{pmatrix} = \begin{pmatrix} y_0 \\ y_0' \end{pmatrix}. \tag{1.2.2}$$

Si aplicamos a este último problema de primer orden un método RK(A, b), se obtiene tras unas manipulaciones simples, que las fórmulas de avance están dadas por,

$$\begin{aligned} y_{n+1} &= y_n + hy_n' + h^2 \sum_{j=1}^{s} \bar{b}_j f(t_n + c_j h, Y_{nj}) \\ y_{n+1}' &= y_n' + h \sum_{j=1}^{s} b_j f(t_n + c_j h, Y_{nj}), \end{aligned} \tag{1.2.3}$$

y las etapas intermedias Y_{nj} se calculan del sistema de ecuaciones,

$$Y_{nj} = y_n + hc_j y_n' + h^2 \sum_{l=1}^{s} \bar{a}_{jl} f(t_n + c_l h, Y_{nl}), \quad j = 1, \ldots, s, \tag{1.2.4}$$

con

$$\bar{a}_{ij} = \sum_{k=1}^{s} a_{ik} a_{kj}, \qquad \bar{b}_i = \sum_{j=1}^{s} b_j a_{ji}, \qquad c_i = \sum_{j=1}^{s} a_{ij}. \tag{1.2.5}$$

Las ecuaciones (1.2.3) y (1.2.4) definen un método denominado *Runge–Kutta–Nyström* (RKN$(\bar{A}, c, b, \bar{b})$), ya que fue Nyström en 1925 el primero en considerar métodos directos de este tipo para resolver problemas de segundo orden. Estos métodos tienen ventajas computacionales para integrar el problema (1.2.1), sobre los métodos Runge-Kutta convencionales aplicados directamente sobre el problema de primer orden equivalente (1.2.2), en virtud de que al usar los métodos RKN se evita duplicar la dimensión.

En lo sucesivo asumiremos las condiciones (1.2.5) para los métodos Runge-Kutta Nyström a considerar, donde se supone que la matriz A y el vector columna b son los coeficientes de un método Runge–Kutta original. Este tipo de métodos se denominan *métodos RKN inducidos por un método Runge–Kutta*, y se representan mediante la tabla de Butcher

$$\begin{array}{c|c} c & \bar{A} \\ \hline & \bar{b}^T \\ \hline & b^T \end{array} \quad \equiv \quad \begin{array}{c|ccc} c_1 & \bar{a}_{11} & \dots & \bar{a}_{1s} \\ \vdots & \vdots & \ddots & \vdots \\ c_s & \bar{a}_{s1} & \dots & \bar{a}_{ss} \\ \hline & \bar{b}_1 & \dots & \bar{b}_s \\ \hline & b_1 & \dots & b_s \end{array}$$

Las ecuaciones de avance (1.2.3) pueden escribirse en forma más compacta

$$\begin{aligned} y_{n+1} &= y_n + hy'_n + h^2(\bar{b}^T \otimes I_m)F(Y_n), \\ y'_{n+1} &= y'_n + h(b^T \otimes I_m)F(Y_n), \end{aligned} \tag{1.2.6}$$

junto con las ecuaciones de etapa (1.2.4),

$$Y_n = e \otimes y_n + h(c \otimes y'_n) + h^2(\bar{A} \otimes I_m)F(Y_n), \tag{1.2.7}$$

$$Y_n^T = (Y_{n,1}^T, \dots, Y_{n,s}^T), \qquad F^T(Y_n) = \big(f^T(t_n + c_1 h, Y_{n,1}), \dots, f^T(t_n + c_s h, Y_{n,s})\big) \in \mathbb{R}^{ms}.$$

La iteración Quasi-Newton para resolver las ecuaciones de etapa puede escribirse del modo siguiente,

$$\big(I_{ms} - h^2(A^2 \otimes J)\big)(Y_n^\nu - Y_n^{\nu-1}) = D(Y_n^{\nu-1}, y_n, hy'_n), \quad \nu = 1, 2, \dots, \tag{1.2.8}$$

para un valor inicial dado Y_n^0, donde $J \simeq \dfrac{\partial f}{\partial y}(t_n, y_n)$ y el residual $D(.)$ está definido por

$$D(Z, u, v) := -Z + e \otimes u + (c \otimes v) + h^2(A^2 \otimes I)F(Z), \quad u, v \in \mathbb{R}^m, Z \in \mathbb{R}^{ms}. \tag{1.2.9}$$

Si el método inducido RKN proviene de un método Runge-Kutta SDIRK (*Singly Diagonally Implicit RK*), la iteración Quasi-Newton permite resolver las etapas de forma secuencial (una después de otra) en cada iteración del proceso iterativo, con lo que se reducen bastante los costos computacionales (denotamos esta clase de métodos por SDIRKN). Esto hace que los métodos SDIRKN sean especialmente atractivos para la computación, ya que entre ellos pueden encontrarse métodos con buenas propiedades de estabilidad y órdenes medianamente altos. El tipo de métodos de la clase SDIRKN fue ampliamente estudiado en la Tesis realizada por I. Gómez en el año 2002 [38], con el objetivo principal de buscar métodos de altos órdenes en la fase y con intervalos largos de P-estabilidad. También se investigaron métodos simétricos y simplécticos con el objetivo de buscar sub-familias candidatas a ser buenos integradores en precisión media, para sistemas Hamiltonianos y para problemas de vibraciones.

Sin embargo, para obtener la misma precisión en términos del orden de convergencia, los métodos SDIRKN requieren muchas más etapas internas que los métodos RKN altamente implícitos, tales como aquellos inducidos por los RK–Gauss o Lobatto IIIA en versión RKN, dados por ejemplo en [57, pp. 72–75], los cuales son también P–estables. Por esta razón, la búsqueda de implementaciones de bajo costo computacional para estos últimos métodos es una alternativa interesante y posiblemente más eficiente que la de los métodos SDIRKN.

Los argumentos dados anteriormente nos han motivado a desarrollar en los Capítulos 3 y 4 de esta Memoria, una investigación exhaustiva encaminada a una implementación eficiente de los métodos de Gauss para problemas de segundo orden. En particular, en el Capítulo 3 nos centraremos en desarrollar esquemas iterativos eficientes, que denominaremos de tipo Single-Newton, para resolver las ecuaciones de etapa de los métodos de Gauss (en versión Nyström), y obtendremos esquemas optimizados para los casos de los métodos con 2,3 y 4 etapas. En el capítulo 4, seremos algo más ambiciosos y desarrollaremos un código de integración a paso variable, que está basado en el método de Gauss de 2 etapas y en los procesos iterativos previamente desarrollados a efectos de resolver sus ecuaciones de etapa. Este código está pensado para integrar cualquier tipo de problema de segundo orden (PVI), independientemente de si el sistema diferencial posee rigidez o no. Mostraremos que el código es especialmente útil para la integración de Ecuaciones en Derivadas Parciales (EDP) que modelen fenómenos de tipo ondulatorio. En este caso, no se precisa un orden muy alto en la integración temporal, pues se introducen ya errores en el sistema de EDOs a ser integrado, debido a discretización en las variables espaciales de la EDP. Por tanto, un método de orden 4 como es el de Gauss de 2 etapas, puede ser un buen candidato para la integración temporal del sistema de EDOs resultante. Además, este método posee importantes ventajas para la integración de problemas Hamiltonianos, pues es un método Simétrico y Simpléctico, para mayor detalle sobre sus excelentes propiedades véase por ejemplo [56, 76]. A efectos de desarrollar dicho código, trataremos en los capítulos 3 y 4, diversos detalles de sumo interés tales como: elección de predictores para las iteraciones, selección del paso inicial de integración, estimadores de error local, estrategias de cambio de paso, salida densa, estimaciones de error global, programación optimizada del código (con una selección adecuada de parámetros usados por defecto), experimentos numéricos y comparación con otros códigos. El código ha sido programado en FORTRAN 77-90 y puede descargarse de cualquiera de los portales [47, 50]. La versión en [47] de 2008, es más reciente que la preliminar puesta en 2006 en el famoso portal Netlib [71].

Capítulo 2

Aceleración de métodos iterativos tipo Newton para problemas Stiff

2.1. Introducción

La implementación de métodos Runge–Kutta altamente implícitos para problemas stiff y problemas oscilatorios ha sido un tema de investigación exhaustiva en las últimas décadas, entre los trabajos más relevantes mencionamos [3, 9, 10, 12, 19, 22, 23, 34, 36, 37, 40, 41, 42, 43, 57, 65]. Cuando los métodos Runge–Kutta bajo consideración son altamente implícitos tales como aquellos de la familias Gauss, Radau IA, Radau IIA o Lobatto IIIA–B–C (véase por ejemplo [57, pp. 72–76]) con $s \geq 2$ etapas internas, la matriz de coeficientes A tiene un espectro multi–puntual con $[s/2]$ pares de autovalores complejos conjugados y un autovalor real en el caso que s es impar. En este caso, la iteración de Newton simplificada [3, 9, 57] en la forma normalmente usada (véase por ejemplo [57, pp. 121–122]) tiene el inconveniente de requerir aritmética compleja en las factorizaciones LU. Así, para $s = 2$ esto implica que el costo de la factorización LU es cuatro veces mayor que en el caso real. También los sistemas lineales resultantes de las iteraciones conllevan aritmética compleja y el costo computacional se multiplica aproximadamente por un factor de 4 con respecto al caso real. Para $s > 2$ una discusión detallada acerca del costo computacional de diversas iteraciones de tipo Newton, incluyendo la iteración de Newton simplificada, puede verse en [43, Sec.2]. Como una alternativa de la iteración de Newton simplificada, distintos esquemas iterativos de tipo Newton, que evitan la aritmética compleja, han sido propuestos para el cálculo secuencial de las etapas internas [10, 22, 23, 34, 37, 41, 43], y también usando la

computación en paralelo [64, 65]. En este capítulo, nos limitaremos al cálculo secuencial de las etapas, pero en general nuestros resultados están pensados para acelerar la velocidad de convergencia de una iteración dada y aplicarla a muchas de las situaciones que aparecen en la práctica cuando el esquema iterativo converge linealmente a la solución del problema y el espectro de la matriz de amplificación del error (de las iteraciones) se encuentra en algún subconjunto conocido del plano complejo.

En el caso de los métodos Runge–Kutta con s etapas implícitas, los sistemas algebraicos a ser resueltos en el paso actual de integración están dados por un sistema algebraico del tipo,

$$Y = X_n + h(A \otimes I)F(Y), \tag{2.1.1}$$

donde $A \otimes B = (a_{ij}B)$ denota el producto de Kronecker de matrices. Hemos supuesto que el método se aplica a sistemas diferenciales de la forma

$$y'(t) = f(t, y), \quad y(t_0) = y_0, \quad y, f \in \mathbb{R}^m, \quad t \in \mathbb{R}.$$

En (2.1.1), $Y^T = (Y_1^T, \ldots, Y_s^T)$ denota el vector de las etapas en el paso de integración actual $t_n \to t_{n+1} = t_n + h$,

$$F^T(Y) = (f^T(t_n + c_1 h, Y_1), \ldots, f^T(t_n + c_s h, Y_s))$$

es el super–vector de las derivadas, h es el tamaño de paso tomado, I es la matriz identidad de dimensión apropiada y X_n representa un vector conocido que normalmente es $X_n = e \otimes y_n$, $e = (1, \ldots, 1) \in \mathbb{R}^s$, pero podría diferir para algunos métodos Runge–Kutta tales como los de la familia Lobatto IIIA.

Para resolver estas ecuaciones algebraicas en el caso de métodos implícitos de alto orden, el tipo de iteración propuesto por Cooper y Butcher [22] ha probado ser bastante eficiente en la práctica, ya que evita la aritmética compleja, tiene una buena ratio de convergencia y la dimensión de los sistemas lineales a resolver es igual a la del sistema diferencial original. Esta iteración viene dada por la formulación siguiente [22]:

$$\begin{aligned} [I \otimes I - h\gamma I \otimes J]\Delta^\nu &= (BS^{-1} \otimes I)D(Y^\nu) + (L \otimes I)\Delta^\nu, \\ Y^{\nu+1} &= Y^\nu + (S \otimes I)\Delta^\nu, \quad \nu = 0, 1, 2, \ldots, \end{aligned} \tag{2.1.2}$$

donde $D(Z) := X_n - Z + h(A \otimes I)F(Z)$ es el residual para el super–vector Z, $J \simeq \partial f/\partial y(t_n, y_n)$, B y S son $s \times s$–matrices no–singulares , L es una $s \times s$–matriz triangular inferior estricta y $\gamma > 0$ es una constante. Cuando esta iteración es aplicada a sistemas

diferenciales con coeficientes constantes,

$$y' = f(y) \equiv Jy + b, \tag{2.1.3}$$

no es difícil ver que los errores en las iteraciones están dados por,

$$Y - Y^{\nu+1} = M(hJ)(Y - Y^{\nu}), \quad \nu = 0, 1, 2, \ldots, \tag{2.1.4}$$

donde

$$\begin{aligned} M(hJ) &= (S \otimes I)\bar{M}(hJ)(S^{-1} \otimes I), \\ \bar{M}(hJ) &= I \otimes I - ((I - L) \otimes I - \gamma I \otimes hJ)^{-1}(B \otimes I)(I \otimes I - \bar{A} \otimes hJ) \\ \bar{A} &= S^{-1}AS. \end{aligned} \tag{2.1.5}$$

De (2.1.4) se sigue que,

$$Y - Y^{\nu} = M(hJ)^{\nu}(Y - Y^{0}), \quad \nu = 1, 2, \ldots ,$$

donde Y^0 denota el predictor o aproximación inicial tomada para iniciar las iteraciones.

Un enfoque usado frecuentemente para seleccionar los parámetros libres del esquema (es decir, S, B, L, γ), consiste en minimizar el radio espectral de $M(hJ)$ cuando J posee autovalores arbitrarios contenidos en la parte izquierda del plano complejo $\mathbb{C}^- := \{z, \operatorname{Re} z \leq 0\}$, ya que deseamos una velocidad de convergencia alta sobre la clase de problemas stiff. Observe que los autovalores de $M(hJ)$ están dados por los autovalores de la más simple $(s \times s)$–matriz

$$\bar{M}(z) = I - ((I - L) - z\gamma I)^{-1}B(I - z\bar{A}), \tag{2.1.6}$$

donde $z = h\lambda$, y λ varía en el espectro de J.

En [22] se da una técnica para minimizar el radio espectral $\rho(z)$ de $\bar{M}(z)$, para los métodos de dos etapas cuya matriz A es de la forma (2.1.7). luego se aplica a los métodos de Gauss de dos etapas haciendo $S = I$. Cuando $s > 2$ y los métodos considerados son altamente implícitos (tales como los de las familias, Gauss, Radau IA–IIA, Lobatto C), existe una matriz S tal que $S^{-1}AS = \overline{A}$, donde

$$\bar{A} = A_1 \oplus A_2 \oplus \ldots \oplus A_r,$$

es una matriz real diagonal por bloques, y las submatrices A_i pueden ser elegidas del modo siguiente,

$$A_i = \begin{pmatrix} a_i & a_i - b_i \\ a_i + b_i & a_i \end{pmatrix}, \quad i = 1, \ldots, r. \tag{2.1.7}$$

En el caso en que s es impar, el último bloque es más simple y tiene la forma $A_r = [a_r]$, donde a_r es el único autovalor real de A. Butcher y Cooper [22] dan pautas precisas para obtener buenos esquemas iterativos (matrices B, L y γ) para los métodos de Gauss con $s = 2, 3, 4$ etapas, usando las sumas directas

$$L = L_1 \oplus L_2 \oplus \ldots \oplus L_r, \quad B = B_1 \oplus B_2 \oplus \ldots \oplus B_r,$$

y eligiendo de modo apropiado las matrices L_i y B_i y la constante $\gamma > 0$.

Una técnica de sobrerrelajación sucesiva, consiste en la introducción de un parámetro adecuado ω es considerada también en [22]. Esto permite acelerar la convergencia de una iteración dada en algunos casos, principalmente cuando los autovalores de J están sobre el semieje real negativo o en algunos conjuntos específicos contenidos en la parte izquierda del plano complejo.

Gladwell y Thomas [36, 37] recomiendan para problemas de segundo orden $y''(t) = f(t, y)$, al comparar diversos tipos de métodos tipo Newmark, Híbridos, Multipaso, etc., la versión RK–Nyström del método de Gauss de 2 etapas implementado con el esquema propuesto por Cooper y Butcher [22] usando la técnica de sobrerrelajación sucesiva. Este esquema está dado por (2.1.2) tomando $S = I$,

$$L = \begin{pmatrix} 0 & 0 \\ 2\omega & 0 \end{pmatrix}, \quad B = \begin{pmatrix} \omega & \omega(7 - 4\sqrt{3}) \\ -\omega(1+\omega) & \omega - \omega^2(7 - 4\sqrt{3}) \end{pmatrix}, \tag{2.1.8}$$

y parámetro de sobrerrelajación $\omega = 2(\sqrt{6} - \sqrt{2} + 1)^{-1}$ y $\gamma = \sqrt{3}/6$.

Es un hecho bien conocido que para integrar problemas stiff de primer orden, los métodos de Gauss no son los mejores candidatos (ya que no son fuertemente A–estable, pues $|R(iy)| = 1,\ \forall y \in \mathbb{R}$), sin embargo para problemas oscilatorios, problemas Hamiltonianos y problemas de segundo orden en general $y''(t) = f(t, y)$, estos métodos son bastante eficientes. Esto se debe a su relativo alto orden de convergencia y a sus buenas propiedades de estabilidad, simetría y simplecticidad. Para problemas stiff de primer orden y para cierto tipos de DAEs de Indices 1 y 2, así como para problemas de Perturbaciones Singulares, otros métodos tales como los de las familias Radau IA, Radau IIA, Lobatto IIIA, Lobatto IIIC, son más convenientes, véase por ejemplo [57]. Por este motivo dedicaremos la sección 2.2 a dar un proceso sistemático basado en el trabajo de Cooper y Butcher [22], para obtener buenos esquemas de tipo (2.1.2) para métodos Runge–Kutta con $s = 2, 3$ etapas implícitas, que poseen un espectro multipuntual en su matriz de coeficientes A. Luego derivaremos esquemas

particulares para métodos de las familias de: Gauss, Radau IA, Radau IIA, Lobatto IIIA, Lobatto IIIB y Lobatto IIIC. En la sección 2.3, consideraremos la aceleración de los esquemas propuestos usando la información de las dos últimas iteraciones dadas. De modo más preciso, proponemos la introducción de dos parámetros fijos $\alpha, \beta \neq 0$ en la iteración original de Cooper y Butcher (2.1.2) del siguiente modo,

$$\begin{aligned} [I \otimes I - h\gamma I \otimes J]\Delta^{\nu} &= \beta(BS^{-1} \otimes I)D(Y^{\nu} + \alpha(Y^{\nu} - Y^{\nu-1})) + (L \otimes I)\Delta^{\nu}, \\ Y^{\nu+1} &= Y^{\nu} + (S \otimes I)\Delta^{\nu}, \quad \nu = 0, 1, 2, \ldots, \end{aligned} \tag{2.1.9}$$

comenzando con $Y^{-1} = Y^0$. Esta nueva iteración esta motivada por tres hechos; primero el análisis de convergencia sobre problemas lineales del nuevo esquema iterativo está estrechamente relacionado con la convergencia de la iteración original, esto simplifica bastante el estudio. Segundo, la nueva iteración puede usarse para acelerar muchas iteraciones de tipo (2.1.2) o similares, que aparecen en la literatura, haciendo una elección cuidadosa de los parámetros α y β en (2.1.9). En tercer lugar, el costo computacional de la nueva iteración es prácticamente el mismo que el de la iteración (2.1.2), sólo deben añadirse ms productos extra en el caso de (2.1.9). Además, veremos que en problemas prácticos no lineales la nueva iteración también acelera la convergencia de la iteración original siempre que ésta última converja de forma satisfactoria. Los resultados presentados aquí están publicados esencialmente en [45].

2.2. Iteración de Cooper y Butcher para métodos con dos y tres etapas

Primero, analicemos el caso de dos etapas implícitas, y luego los resultados serán aplicados para el caso de tres etapas implícitas.

2.2.1. El caso $s = 2$

En este caso escogemos $S = I$, y denotamos la correspondiente matriz Runge–Kutta para las dos etapas implícitas por

$$A = \begin{pmatrix} \bar{a}_{11} & \bar{a}_{12} \\ \bar{a}_{21} & \bar{a}_{22} \end{pmatrix}. \tag{2.2.1}$$

Entonces, según (2.1.5) tenemos que $\bar{A} = A$. De esta manera, haciendo

$$a_1 = \mathrm{tr}(\bar{A}) = \bar{a}_{11} + \bar{a}_{22}, \qquad a_2 = \det(\bar{A}),$$

de (2.1.5) y (2.1.6) se sigue que

$$I - \bar{M}(z) = ((I - L) - z\gamma I)^{-1} B(I - z\bar{A}). \tag{2.2.2}$$

Asumiendo por el momento que $\bar{M}(z)$ tiene un autovalor nulo, entonces de (2.2.2) tenemos que el otro autovalor esta dado por

$$\phi(z) = 1 - \mu \frac{\det(I - z\bar{A})}{(1 - \gamma z)^2}, \qquad \mu = \det(B). \tag{2.2.3}$$

La idea de Cooper y Butcher para minimizar $\phi(z)$ en el eje imaginario cuando $\bar{A}$ es de la forma (2.1.7), fue elegir δ, μ y $\gamma > 0$, con $|\delta|$ mínimo, tal que

$$\phi(z) = \delta \frac{(1 + \gamma z)^2}{(1 - \gamma z)^2} \ . \tag{2.2.4}$$

Observe que esto implicaría que $|\phi(z)| = |\delta|$ en el eje imaginario y por el Teorema del Módulo Máximo $|\phi(z)| \leq |\delta|$ en $\mathbb{C}^-$. Siguiendo el enfoque de Cooper y Butcher, teniendo en cuenta que

$$\det(I - z\bar{A}) = 1 - \mathrm{tr}(\bar{A})z + \det(\bar{A})z^2 = 1 - a_1 z + a_2 z^2,$$

de (2.2.3) y (2.2.4) deducimos que

$$\delta = 1 - \mu, \qquad 2\gamma\delta = \mu a_1 - 2\gamma, \qquad \delta\gamma^2 = \gamma^2 - \mu a_2. \tag{2.2.5}$$

De aquí eliminando γ y μ es fácil ver que δ satisface

$$(4a_2 - a_1^2)\delta^2 + (8a_2 + 2a_1^2)\delta + (4a_2 - a_1^2) = 0, \tag{2.2.6}$$

y además que

$$\gamma = \frac{(1 - \delta)a_1}{2(1 + \delta)}, \qquad \mu = \det(B) = 1 - \delta. \tag{2.2.7}$$

De (2.2.6) elegimos el $|\delta|$ más pequeño de las dos posibilidades resultantes y de (2.2.7) los valores correspondientes para γ y μ.

A continuación, damos una condición para asegurar que $\bar{M}(z)$ tiene un autovalor nulo independientemente de z. Para este propósito, es suficiente exigir la existencia de un vector constante no nulo $v = (v_1, v_2)^T$ tal que $\bar{M}(z)v = 0, \ \forall z$. En virtud de (2.2.2) esto nos lleva a

$$((I-L)-z\gamma I)v = B(I-z\bar{A})v,$$

y de aquí obtenemos

$$(I-L)v = Bv, \qquad \gamma v = B\bar{A}v. \tag{2.2.8}$$

Eligiendo $v = e_1 = (1,0)^T$, se tiene que la matriz $\bar{M}(z)$ resulta triangular superior. En este caso de (2.2.6)–(2.2.8) obtenemos explícitamente los coeficientes de las matrices L, B como sigue

$$B = \begin{pmatrix} 1 & \dfrac{\gamma - \bar{a}_{11}}{\bar{a}_{21}} \\ -\dfrac{\bar{a}_{21}(1-\delta)}{\gamma} & \dfrac{\bar{a}_{11}(1-\delta)}{\gamma} \end{pmatrix}, \qquad L = \begin{pmatrix} 0 & 0 \\ \dfrac{\bar{a}_{21}(1-\delta)}{\gamma} & 0 \end{pmatrix}. \tag{2.2.9}$$

Además, la matriz $\bar{M}(z) = M(z)$ esta dada por

$$\bar{M}(z) = \begin{pmatrix} 0 & \psi(z) \\ 0 & \phi(z) \end{pmatrix} \quad \text{con} \qquad \psi(z) = \frac{-b_{12} + z(\bar{a}_{12} + b_{12}\bar{a}_{22})}{(1-\gamma z)^2}, \tag{2.2.10}$$

y $\phi(z)$ satisfaciendo (2.2.4). Otras elecciones son posibles para el vector v, pero sugerimos esta opción ya que $v = e_2 = (0,1)^T$, es incompatible con el hecho que L es una matriz triangular inferior estricta. Además $v = e_1$ da lugar a una matriz $\bar{M}(z)$ muy simple con $\max_{x\le 0}|\psi(x)| \simeq \max_{x\le 0}|\phi(x)|$ para los métodos de interés.

Abajo, damos las matrices B y L y los parámetros δ y γ, para los métodos de 2 etapas implícitas más importantes. Observe que de (2.2.9)

$$l_{21} = -b_{21} \quad \text{y} \quad b_{11} = 1$$

en todos los casos.

Coeficientes para el método de Gauss de dos etapas.

$$\bar{a}_{11} = \bar{a}_{22} = 1/4, \quad \bar{a}_{12} = 1/4 - \sqrt{3}/6, \quad \bar{a}_{21} = 1/4 + \sqrt{3}/6,$$
$$b_{12} = 7 - 4\sqrt{3}, \qquad b_{21} = -2, \qquad b_{22} = -6 + 4\sqrt{3},$$
$$\gamma = \sqrt{3}/6, \qquad \delta = -7 + 4\sqrt{3} = -0{,}071796\ldots .$$

Observe que estos parámetros coinciden con los dados en (2.1.8) cuando $\omega = 1$, es decir, el caso de no considerar sobrerrelajación.

Coeficientes para el método Radau IA de dos etapas

$$\bar{a}_{11} = \bar{a}_{21} = 1/4, \quad \bar{a}_{12} = -1/4, \quad \bar{a}_{22} = 5/12,$$

$$b_{12} = -1 + 2\sqrt{6}/3, \qquad b_{21} = 16 - 32/\sqrt{3}, \qquad b_{22} = -3 + 3\sqrt{6}/2,$$
$$\gamma = 1/\sqrt{6}, \qquad \delta = -5 + 2\sqrt{6} = -0{,}10102\ldots .$$

Coeficientes para el método Radau IIA de dos etapas

$$\bar{a}_{11} = 5/12, \quad \bar{a}_{12} = -1/12, \quad \bar{a}_{21} = 3/4, \quad \bar{a}_{22} = 1/4,$$
$$b_{12} = (-5 + 2\sqrt{6})/9, \qquad b_{21} = 3 - 3\sqrt{6}/2, \qquad b_{22} = -5 + 5\sqrt{6}/2,$$
$$\gamma = 1/\sqrt{6}, \qquad \delta = -5 + 2\sqrt{6} = -0{,}10102\ldots .$$

Coeficientes para el método Lobatto IIIA de tres etapas

Ya que la primera etapa es explícita, los coeficientes con nuestra notación están dados por,

$$\bar{a}_{11} = 1/3, \quad \bar{a}_{12} = -1/24, \quad \bar{a}_{21} = 2/3, \quad \bar{a}_{22} = 1/6,$$
$$b_{12} = -1/2 + \sqrt{3}/4, \qquad b_{21} = 3 - 3\sqrt{6}/2, \qquad b_{22} = -8 + 16\sqrt{3},$$
$$\gamma = \sqrt{3}/6, \qquad \delta = -7 + 4\sqrt{3} = -0{,}071796\ldots .$$

Coeficientes para el método Lobatto IIIB de tres etapas

Ya que la tercera etapa es combinación lineal de Y_1 y Y_2, los coeficientes para las etapas implícitas están dados por,

$$\bar{a}_{11} = \bar{a}_{21} = 1/6, \quad \bar{a}_{12} = -1/6, \quad \bar{a}_{22} = 1/3,$$
$$b_{12} = -1 + \sqrt{3}, \qquad b_{21} = 4 - 8/\sqrt{3}, \qquad b_{22} = -4 + 8\sqrt{3},$$
$$\gamma = \sqrt{3}/6, \qquad \delta = -7 + 4\sqrt{3} = -0{,}071796\ldots .$$

Coeficientes para el método Lobatto IIIC de tres etapas

$$\bar{a}_{11} = \bar{a}_{21} = \bar{a}_{22} = 1/2, \quad \bar{a}_{12} = -1/2,$$
$$b_{12} = -1 + \sqrt{2}, \qquad b_{21} = 2 - 2\sqrt{2}, \qquad b_{22} = -2 + 2\sqrt{2},$$
$$\gamma = \sqrt{2}/2, \qquad \delta = -3 + 2\sqrt{2} = -0{,}17157\ldots .$$

2.2.2. El caso de 3 etapas implícitas ($s = 3$)

Aplicando una transformación apropiada de semejanza (matriz S) a la matriz original A (correspondiente a las etapas internas) obtenemos que

$$S^{-1}AS = \bar{A}_1 \oplus [\lambda_3], \tag{2.2.11}$$

donde $\bar{A}_1$ es una matriz de dimensión dos y λ_3 es el autovalor real de la matriz original A. En este caso nuestra elección para las matrices B y L viene dada por

$$B = B_1 \oplus [b_3], \qquad L = L_1 \oplus [0], \tag{2.2.12}$$

y tendremos en cuenta los resultados anteriores para el caso de $s = 2$. Así, B_1, L_1 y γ se obtienen como se indica en (2.2.9) con $\bar{A}_1$ reemplazando a A en (2.2.1), δ se calcula de (2.2.6) (la raíz más pequeña en módulo) y γ es dado por (2.2.7).

Ahora, según (2.1.6) la matriz de amplificación está dada por

$$\bar{M}(z) = \bar{M}_1(z) \oplus [m_3(z)],$$

con

$$\begin{array}{rcl} \bar{M}_1(z) & = & I - ((I - L_1) - z\gamma I)^{-1} B_1 (I - z\bar{A}_1), \\ m_3(z) & = & 1 - b_3(1 - \gamma z)^{-1}(1 - z\lambda_3). \end{array} \tag{2.2.13}$$

La elección de la constante b_3 en (2.2.12) se hará como se explica a continuación. En virtud de que el parámetro δ resulta ser negativo para los métodos considerados (Gauss, Radau, Lobatto), entonces estamos interesados en que se verifique

$$\delta \leq m_3(z) \leq 0, \quad \forall z \in \mathbb{R}^-, \tag{2.2.14}$$

ya que en esta situación, la iteración resultante puede ser acelerada de manera más eficaz según veremos en la próxima sección. Puesto que estos métodos verifican que $\lambda_3 > \gamma$, entonces proponemos tomar b_3 tal que $m_3(0) = 1 - b_3 = 0$. De este modo se sigue que

$$b_3 = 1. \tag{2.2.15}$$

Observe que con esta elección se satisface (2.2.14) siempre que

$$1 - \lambda_3/\gamma \geq \delta. \tag{2.2.16}$$

La condición (2.2.16) se cumple para los métodos de las familias Gauss, Radau y Lobatto.

A continuación daremos los coeficientes (con 16 cifras significativas) para los métodos Runge–Kutta de tres etapas implícitas. Se puede observar que como en el caso de dos etapas implícitas también aquí se tiene que :

$$l_{21} = -b_{21} \quad \text{y} \quad b_{11} = 1.$$

- Gauss de tres etapas

$$\bar{A} = \begin{pmatrix} \frac{5}{36} & \frac{2}{9} - \frac{\sqrt{15}}{15} & \frac{5}{36} - \frac{\sqrt{15}}{30} \\ \frac{5}{36} + \frac{\sqrt{15}}{24} & \frac{2}{9} & \frac{5}{36} - \frac{\sqrt{15}}{24} \\ \frac{5}{36} + \frac{\sqrt{15}}{30} & \frac{2}{9} + \frac{\sqrt{15}}{15} & \frac{5}{36} \end{pmatrix},$$

$$B = \begin{pmatrix} 1 & -0{,}4005025672960664 & 0 \\ 0{,}8010051345921328 & 0{,}8395976935892598 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$S = \begin{pmatrix} -0{,}045611384659161056 & 0{,}04414658392664155 & 0{,}07215185205520017 \\ -0{,}13958959990794589 & -0{,}13958959990794589 & 0{,}1188325787412779 \\ 1 & -0{,}2440147553957837 & 1 \end{pmatrix},$$

$$\gamma = 0{,}1967310073266746, \qquad \delta = -0{,}1604023064107402.$$

- Radau IIA de tres etapas

$$\bar{A} = \begin{pmatrix} \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\ \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\ \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \end{pmatrix},$$

$$B = \begin{pmatrix} 1 & -0{,}4524329709908831 & 0 \\ 0{,}9048659419817663 & 0{,}7953044067603627 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$S = \begin{pmatrix} -0{,}1412552950209542 & -0{,}030029194105147424 & 0{,}09443876248897524 \\ 0{,}2041293522937999 & 0{,}3829421127572619 & 0{,}2502131229653333 \\ 1 & 0 & 1 \end{pmatrix},$$

$$\gamma = 0{,}2462327575264407, \qquad \delta = -0{,}2046955932396373.$$

- Lobatto IIIA de cuatro etapas

  Ya que para este método la primera etapa es explícita, tenemos que:

$$\bar{A} = \begin{pmatrix} \frac{25-\sqrt{5}}{120} & \frac{25-13\sqrt{5}}{120} & \frac{-1+\sqrt{5}}{120} \\ \frac{25+13\sqrt{5}}{120} & \frac{25+\sqrt{5}}{120} & \frac{-1-\sqrt{5}}{120} \\ \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \end{pmatrix},$$

$$S = \begin{pmatrix} -0{,}07776129960563076 & 0{,}006043307469475509 & 0{,}05303036326129938 \\ 0{,}2193839918662961 & 0{,}3198765142300936 & 0{,}2637242522173698 \\ 1 & 0 & 1 \end{pmatrix},$$

$$\gamma = 0{,}1967310073266746, \qquad \delta = -0{,}1604023064107402.$$

La matriz B para este método coincide con la matriz B del Gauss de tres etapas.

- Lobatto IIIC de tres etapas

$$\bar{A} = \begin{pmatrix} \frac{1}{6} & -\frac{1}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{5}{12} & -\frac{1}{12} \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{pmatrix},$$

$$B = \begin{pmatrix} 1 & -0{,}5326013262245936 & 0 \\ 1{,}065202652449187 & 0{,}7163358273038041 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$S = \begin{pmatrix} -0{,}6027050205505142 & -0{,}4309321229203226 & 0{,}4554100411010285 \\ 0{,}1775508472321798 & 0{,}5194499080011395 & 0{,}2073983055356404 \\ 1 & 0 & 1 \end{pmatrix},$$

$$\gamma = 0{,}3307703646387769, \qquad \delta = -0{,}2836641726961959.$$

2.3. Aceleración de la convergencia

En esta sección estudiamos como acelerar la convergencia de la iteración (2.1.2) mediante la introducción de dos parámetros α y β tal como se expresa en (2.1.9).

Teniendo en cuenta que sobre sistemas lineales (2.1.3), el residual satisface (véase (2.1.2))

$$D(Y) - D(Z) = (-I \otimes I + A \otimes hJ)(Y - Z), \quad \forall\, Y, Z \in \mathbb{R}^{ms},$$

Entonces, denotando por Y la solución exacta de (2.1.1) en el caso lineal, se sigue de (2.1.9) que

$$[(I-L)\otimes I-\gamma I\otimes hJ]\Delta^{\nu}=\beta(BS^{-1}\otimes I)(D(Y^{\nu}+\alpha(Y^{\nu}-Y^{\nu-1}))-D(Y))$$
$$=\beta(BS^{-1}\otimes I)(-I\otimes I+A\otimes hJ)(Y^{\nu}-Y+\alpha(Y^{\nu}-Y^{\nu-1})).$$

usando (2.1.2) y (2.1.5) deducimos que

$$\begin{aligned}&[(I-L)\otimes I-\gamma I\otimes hJ](S^{-1}\otimes I)(Y^{(\nu+1)}-Y^{(\nu)})\\&=\beta(BS^{-1}\otimes I)(-I\otimes I+S\bar{A}S^{-1}\otimes hJ)(Y^{\nu}-Y+\alpha(Y^{\nu}-Y^{\nu-1})).\end{aligned}\tag{2.3.1}$$

Denotando por

$$E^{\nu}=(S^{-1}\otimes I)(Y-Y^{\nu}),\qquad \nu=0,1,\ldots,$$

obtenemos de (2.3.1) después de algunas manipulaciones algebraicas sencillas que

$$\begin{aligned}&[(I-L)\otimes I-\gamma I\otimes hJ](E^{\nu+1}-E^{\nu})=\\&\beta(B\otimes I)(-I\otimes I+\bar{A}\otimes hJ)((1+\alpha)E^{\nu}-\alpha E^{\nu-1}).\end{aligned}\tag{2.3.2}$$

Usando la segunda ecuación de (2.1.5), tras unos cálculos directos se sigue de (2.3.2) que,

$$E^{\nu+1}=(a\bar{M}(hJ)+(1-a)I)E^{\nu}+c(\bar{M}(hJ)-I)E^{\nu-1},\ \nu=0,1,2,\ldots,\tag{2.3.3}$$

con

$$c=-\alpha\beta,\qquad a=\beta(1+\alpha).\tag{2.3.4}$$

De acuerdo con (2.3.3) y (2.3.4), el análisis de propagación de errores puede realizarse examinando el radio espectral de la matriz de dimensión $2ms$

$$\bar{N}(hJ)=\begin{pmatrix}a\bar{M}(hJ)+(1-a)I & c(\bar{M}(hJ)-I)\\ I & 0\end{pmatrix},$$

pues (2.3.3) puede ser re-escrito como

$$(E^{\nu+1},E^{\nu})^{T}=\bar{N}(hJ)(E^{\nu},E^{\nu-1})^{T},\quad \nu=0,1,2\ldots\ .$$

Los autovalores de $\bar{N}(hJ)$ son aquellos de la matriz más simple de dimensión $2s$

$$\bar{N}(z)=\begin{pmatrix}a\bar{M}(z)+(1-a)I & c(\bar{M}(z)-I)\\ I & 0\end{pmatrix},$$

donde $z=h\lambda$ y λ varía en el espectro de J.

Sin embargo, preferimos hacer el análisis sobre la ecuación en diferencias (2.3.3), debido a que resulta más directo y natural. Así, asumiendo que (λ,v) es un par autovalor–autovector

de J, y haciendo $E^0 \equiv (S^{-1} \otimes I)(Y - Y^0) = e_i \otimes v$ (recordando que $Y^{-1} = Y^0$, con e_i denotando el i–vector de la base canónica de $\mathbb{R}^s$), obtenemos de (2.3.3) que

$$E^\nu = M_\nu(z)(e_i \otimes v), \tag{2.3.5}$$

donde $z = \lambda h$ y $M_\nu(z)$ satisface la relación de recurrencia,

$$\begin{gathered} M_{\nu+1}(z) = (a\bar{M}(z) + (1-a)I)M_\nu(z) + c(\bar{M}(z) - I)M_{\nu-1}(z), \quad \nu = 0, 1, 2, \ldots, \\ M_{-1}(z) = M_0(z) = I. \end{gathered} \tag{2.3.6}$$

Ya que $M_\nu(z)$ es una matriz polinomial en $\bar{M}(z)$ de grado ν, con coeficientes reales, que tiene el mismo sistema de autovectores independientemente del ν considerado, entonces el tamaño del error en la iteración ν-ésima depende esencialmente de los autovalores $\tau_\nu(z)$ de $M_\nu(z)$. Estos autovalores satisfacen la ecuación lineal en diferencias

$$\begin{gathered} \tau_{\nu+1}(z) = (a\tau(z) + (1-a))\tau_\nu(z) + c(\tau(z) - 1)\tau_{\nu-1}(z), \quad \nu = 0, 1, 2, \ldots, \\ \tau_{-1}(z) = \tau_0(z) = 1, \end{gathered} \tag{2.3.7}$$

con $\tau(z)$ denotando un autovalor de $\bar{M}(z)$ (o de $M(z)$).

A continuación, discutimos como realizar una elección óptima de los parámetros a y c (y consecuentemente de α y β según la relación (2.3.4)), bajo la hipótesis de que los autovalores $\tau(z)$ de $M(z)$ pertenecen a algún intervalo real acotado cuando z varía en algún conjunto Ω de $\mathbb{C}^-$, típicamente para problemas de tipo stiff supondremos que $\Omega = \mathbb{R}^-$, es decir,

$$-\rho_1 \leq \tau(z) \leq \rho_0 < 1, \quad \forall z \in \Omega, \quad \rho_j \geq 0 \ (j = 0, 1). \tag{2.3.8}$$

Ya que el polinomio característico asociado a (2.3.7) es

$$\xi^2(z) - (a\tau(z) + (1-a))\xi(z) + c(1 - \tau(z)) = 0,$$

el problema se reduce a encontrar a y c tal que las raíces $\xi = \xi_\pm(t)$ de

$$\xi^2 - (at + (1-a))\xi + c(1-t) = 0, \quad t \in [-\rho_1, \rho_0],$$

satisfacen que su máximo en módulo resulta ser mínimo, es decir,

$$\rho := \text{máx}\{|\xi_\pm(t)|, \ t \in [-\rho_1, \rho_0]\} \tag{2.3.9}$$

es mínimo.

Usando el criterio de Schur–Cohn [70], se prueba que éste mínimo se alcanza para

$$a = \frac{2\rho + 1}{1 + \rho_1}, \quad c = \frac{\rho^2}{1 + \rho_1}, \quad \rho = \frac{\rho_1 + \rho_0}{4 + \rho_1 - 3\rho_0}. \tag{2.3.10}$$

De (2.3.4) obtenemos

$$\beta = \frac{(\rho + 1)^2}{1 + \rho_1}, \quad \alpha = -\left(\frac{\rho}{1 + \rho}\right)^2. \tag{2.3.11}$$

Para aplicar estos resultados a los métodos de la sección 2.2, considerando $\Omega = \mathbb{R}^-$ en (2.3.8), se sigue de (2.2.4), teniendo en cuenta que en todos los casos $\delta < 0$ que

$$\rho_1 = -\delta, \qquad \rho_0 = 0. \tag{2.3.12}$$

En tal situación, de (2.3.10) y (2.3.11) se deduce que

$$\rho = -\frac{\delta}{4 - \delta} \ . \tag{2.3.13}$$

Es conveniente aclarar que ρ da la razón de convergencia de la nueva iteración cuando $\Omega = \mathbb{R}^-$. Observe que ρ es el radio máximo espectral de la matriz $\bar{N}(z)$ cuando z varía en $\mathbb{R}^-$.

Las tablas 2.3.1 y 2.3.2 recogen (con 3 cifras significativas) los parámetros óptimos α, β, ρ y otros parámetros relacionados $\bar{\rho}, \rho^*, \delta$, para los métodos de dos y tres etapas implícitas de la sección precedente. Aquí, ρ^* representa el radio máximo espectral de $\bar{N}(z)$ sobre el eje imaginario, y $\bar{\rho}$ es el radio máximo espectral de $\bar{N}(z)$ cuando z varía en el conjunto $\bar{\Omega} := \{z = (-1 + i)y, \ y \in \mathbb{R}^+\}$.

De las tablas 2.3.1 y 2.3.2 se puede observar que la nueva iteración tiene una razón de convergencia cuatro veces más pequeña que la iteración original ($|\delta|$) en problemas lineales cuando los autovalores de J se encuentran en el semieje negativo. Además la razón de la nueva iteración es aproximadamente la mitad de $|\delta|$ cuando z varía en $\bar{\Omega}$, y es sólo un poco mayor que $|\delta|$ sobre el eje imaginario.

Nota 2.1 Es importante resaltar que algunas iteraciones no pueden ser aceleradas. Por ejemplo el esquema (de sobrerrelajación) propuesto por Cooper y Butcher para el método de Gauss de dos etapas, véase (2.1.8), tiene una razón de convergencia $r = 1 - \omega = 0{,}0173\ldots$, en problemas lineales cuando z varía en $\mathbb{R}^-$ (véase [22] para más detalles). Esta iteración no puede ser acelerada usando nuestra técnica, ya que los autovalores de la matriz $M(z)$ se mueven en el círculo (centrado en el origen) de radio $r = 0{,}0173...$, cuando z varía en $\mathbb{R}^-$. En este caso se puede probar que los mejores parámetros para acelerar la iteración son $\beta = 1$ y $\alpha = 0$, lo cual no supone ningún cambio en la iteración original.

Tabla 2.3.1: Algunos parámetros de la iteración (2.1.9) para métodos de 2 etapas implícitas.

Método	α	β	ρ	$\bar{\rho}$	ρ^*	δ
Gauss, Lob IIIA-B	$-3{,}06 \cdot 10^{-4}$	0,966	0,0176	0,0388	0,101	−0,0718
Rad IA-IIA	$-5{,}92 \cdot 10^{-4}$	0,954	0,0246	0,0524	0,139	−0,101
Lob IIIC	$-1{,}62 \cdot 10^{-3}$	0,925	0,0411	0,0850	0,229	−0,172

Tabla 2.3.2: Algunos parámetros de la iteración (2.1.9) para métodos de 3 etapas implícitas.

Método	α	β	ρ	$\bar{\rho}$	ρ^*	δ
Gauss, Lob IIIA	$-1{,}38 \cdot 10^{-3}$	0,930	0,0386	0,0832	0,217	−0,1604
Rad IIA	$-2{,}16 \cdot 10^{-3}$	0,913	0,0487	0,205	0,205	−0,205
Lob IIIC	$-3{,}86 \cdot 10^{-3}$	0,886	0,0662	0,284	0,284	−0,284

Nota 2.2 El análisis precedente puede extenderse a un contexto más general. Para tal fin, consideremos un esquema iterativo tipo,

$$P(y^{\nu+1} - y^{\nu}) = QD(y^{\nu}), \quad \nu = 0, 1, \ldots \tag{2.3.14}$$

donde $D(x) \in \mathbb{R}^m$ representa el residual para un vector $x \in \mathbb{R}^m$ (queremos resolver el sistema $D(x) = 0$), con P, Q dos matrices cuadradas de dimensión m de coeficientes constantes. Entonces, nuestro esquema modificado encajaría en el formato

$$\begin{gathered} P(y^{\nu+1} - y^{\nu}) = \beta QD(y^{\nu} + \alpha(y^{\nu} - y^{\nu-1})), \quad \nu = 0, 1, \ldots, \\ y^{-1} = y^0. \end{gathered} \tag{2.3.15}$$

Asumiendo que $D(\bar{x}) = 0$ y $D(x)$ admite un desarrollo de Taylor alrededor de la solución exacta $\bar{x}$ tenemos que

$$D(x) = D(\bar{x}) + \frac{1}{1!}D'[\bar{x}](x - \bar{x}) + \frac{1}{2!}D''[\bar{x}](x - \bar{x}, x - \bar{x}) + \ldots,$$

donde $D^{(j)}[\bar{x}](\cdot, \ldots, \cdot)$ denota la j–derivada de Fréchet de $D(x)$.

Observe que el error de las iteraciones $e^{\nu} = \bar{x} - y^{\nu}$, satisface

$$e^{\nu+1} \simeq M(\bar{x})e^{\nu}, \quad \nu = 0, 1, \ldots, \quad M(\bar{x}) = I - P^{-1}QD'[\bar{x}].$$

Esta relación sería exacta si $D(x)$ fuera lineal. Asumiendo que el espectro de $M(\bar{x})$ varía en algún intervalo real conocido $[-\rho_1, \rho_0]$, entonces el análisis lineal previo puede aplicarse

a esta situación y las fórmulas (2.3.10)–(2.3.11) suministran los parámetros adecuados para acelerar la nueva iteración.

Para ilustrar este caso en particular, consideremos un método BDF (Backward Differential Formulae) de k pasos

$$D_h(y_{n+1}) := -y_{n+1} + x_n + \gamma h f(y_{n+1}) = 0, \tag{2.3.16}$$

aplicado a un problema stiff $y' = f(y)$. Aquí x_n es un vector conocido y γ un parámetro dado por el método BDF. Supongamos que y_{n+1} ha sido calculado usando una iteración Quasi–Newton

$$(I - h\gamma J)(y_{n+1}^{\nu+1} - y_{n+1}^{\nu}) = D_h(y_{n+1}^{\nu}), \quad \nu = 0, 1, \ldots, \nu_{\text{máx}}, \tag{2.3.17}$$

con $J \simeq f'(y_n)$. Asumimos que $(I - h\gamma J)$ ha sido factorizado en la forma LU, y que un nuevo tamaño de paso $h^* = rh$ ha sido propuesto; entonces tenemos que resolver

$$D_{h^*}(y_{n+2}) := -y_{n+2} + x_{n+1} + \gamma h^* f(y_{n+2}) = 0, \tag{2.3.18}$$

donde x_{n+1} es un vector conocido.

Si el Jacobiano J se mantiene para el siguiente paso de longitud $h^* = rh$, en vez de resolver (2.3.18) por la iteración Quasi–Newton (la cual implica una nueva factorización para $(I - h^*\gamma J)$), podemos usar en su lugar la iteración modificada

$$\begin{gathered}(I - h\gamma J)(y_{n+2}^{\nu+1} - y_{n+2}^{\nu}) = \beta D_{h^*}(y_{n+2}^{\nu} + \alpha(y_{n+2}^{\nu} - y_{n+2}^{\nu-1})), \quad \nu = 0, 1, \ldots, \\ y_{n+1}^{-1} = y_{n+1}^{0}.\end{gathered} \tag{2.3.19}$$

En este caso, para problemas lineales (2.1.3), si el espectro de J varía en el eje negativo, se sigue que $M(\bar{x}) = (r-1)\gamma h J(I - \gamma h J)^{-1}$. Entonces ρ_1, ρ_0 en (2.3.8) están dados por

$$-\rho_1 = \text{mín}\{1 - r, 0\}, \qquad \rho_0 = \text{máx}\{1 - r, 0\}.$$

De (2.3.10) obtenemos que

$$\rho = \begin{cases} (3 + r)^{-1}(r - 1), & \text{si } r \geq 1, \\ (1 + 3r)^{-1}(1 - r), & \text{en otro caso.} \end{cases}$$

Los parámetros α y β deben ser calculados de (2.3.11). Esta alternativa puede usarse en la práctica en problemas lineales cuando el valor de $|\rho|$ es pequeño. También puede usarse en problemas no–lineales cuando la matriz Jacobiana no cambia demasiado en dos pasos de integración consecutivos y el valor de $|\rho|$ es pequeño. □

2.4. Experimentos numéricos

En esta sección presentamos resultados numéricos usando los métodos iterativos desarrollados a lo largo de este capítulo, con el propósito de mostrar que los nuevos esquemas propuestos en las secciones anteriores, son una buena alternativa a las iteraciones propuestas por Cooper y Butcher [22].

Hemos llevado a cabo numerosos experimentos numéricos aplicados a un conjunto de problemas test que aparecen en la literatura por ejemplo en [28, 69], los cuales difieren en que los autovalores de su matriz Jacobiana $J = \partial f/\partial y(y_0)$ pertenecen a conjuntos diferentes del semiplano complejo negativo (con parte real no positiva). Los resultados obtenidos sobre la mayoría de los problemas [28, 69] son similares a aquellos expuestos en las tablas al final de este capítulo.

Con este objetivo en mente, hemos implementado los códigos basados en las correspondientes fórmulas Runge–Kutta y resolviendo sus ecuaciones de etapa mediante:

(a) La iteración de Cooper y Butcher (2.1.2) basada en los métodos Runge–Kutta Gauss de 2 y 3 etapas, con parámetros dados en [22] para el caso simple, $\omega = 1$.

(b) La iteración propuesta en (2.1.9) con los valores de γ y las matrices B, L y S dados explícitamente en la sección 2.2 para los métodos de 2 y 3 etapas respectivamente, y los parámetros α y β de (2.3.11) mostrados en las tablas 2.3.1 y 2.3.2.

(c) La iteración de Cooper y Butcher (2.1.2) basada en los métodos Runge–Kutta Gauss de 2 y 3 etapas, con parámetros de sobrerrelajación $\omega = 2(\sqrt{6} - \sqrt{2} + 1)^{-1}$ y $\omega =$ 0,962825449, respectivamente, véase [22] para más detalles.

En todos los casos hemos dado solamente un paso de longitud razonable para el problema stiff considerado. Las aproximaciones iniciales tomadas en todos los casos fueron $Y_j^{(0)} = y_0, j = 1, 2, \ldots, s$. Naturalmente, en la práctica después del primer paso de integración existen mejores aproximaciones iniciales (o predictores) usando extrapolación de los pasos previos, véase por ejemplo [44, 49, 74] para un análisis detallado en el caso stiff.

Sólo se presentan resultados numéricos con tres problemas no–lineales seleccionados de la literatura, de dimensiones baja, media y alta respectivamente, los cuales aparecen a menudo como prototipos de la clase de problemas stiff.

Problema 2.1 *Es el Problema (D4) en [28, p. 21] (véase también [22, p. 138]). Se ha tomado como tamaño de paso inicial* $h_0 = 0{,}1$. *Los autovalores de* $J = \partial f/\partial y(t_0, y_0)$ *son* 0, $-0{,}0093$ *y* -3500.

$$\begin{aligned} y_1' &= -0{,}013y_1 - 1000y_1y_3, & y_1(0) &= 1, \\ y_2' &= -2500y_2y_3, & y_2(0) &= 1, \\ y_3' &= -0{,}013y_1 - 1000y_1y_3 - 2500y_2y_3, & y_3(0) &= 0, \end{aligned}$$

Problema 2.2 *Problema "Ring Modulator" (dimensión 15) [69]. El valor inicial es* $y_0 = 0^T$, *y el paso inicial* $h_0 = 10^{-6}$. *Los autovalores de* $J = \partial f/\partial y(t_0, y_0)$ *son* $-3{,}78 \cdot 10^5 \pm 3{,}16 \cdot 10^7 i$ *(con multiplicidad 2),* $-7{,}39 \cdot 10^5 \pm 3{,}16 \cdot 10^7 i$, $-1{,}73 \cdot 10^5 \pm 3{,}16 \cdot 10^7 i$, $-1{,}60 \cdot 10^5 \pm 7{,}96 \cdot 10^4 i$, $-2{,}28 \cdot 10^4 \pm 1{,}76 \cdot 10^5 i$, $-2{,}00 \cdot 10^6, -1{,}39 \cdot 10^2$, $-19{,}3$.

Problema 2.3 *El CUSP (dimensión 96, véase [57, p. 147]). Los autovalores de* $J = \partial f/\partial y(t_0, y_0)$ *son todos complejos con parte real negativa y sus normas están en el intervalo* $[0{,}39; 2 \cdot 10^8]$, *alcanzado en ambos extremos.*

$$\begin{aligned} y_i' &= -(1/\varepsilon)(y_i^3 + a_i y_i + b_i) + D(y_{i-1} - 2y_i + y_{i+1}) \\ a_i' &= b_i + 0{,}07v_i + D(a_{i-1} - 2a_i + a_{i+1}) \qquad i = 1, \ldots, N \\ b_i' &= (1 - a_i^2)b_i - a_i - 0{,}4y_i + 0{,}035v_i + D(b_{i-1} - 2b_i + b_{i+1}) \end{aligned}$$

donde

$$v_i = \frac{u_i}{u_i + 0{,}1}, \quad u_i = (y_i - 0{,}7)(y_i - 1{,}3), \quad D = \frac{N^2}{144}, \quad \varepsilon = 10^{-8}, \quad N = 32$$

y

$$y_0 = y_N, \quad a_0 = a_N, \quad b_0 = b_N, \quad y_{N+1} = y_1, \quad a_{N+1} = a_1, \quad b_{N+1} = b_1.$$

Se han tomado los valores iniciales

$$y_0 = 0, \quad a_i(0) = -2\cos(2i\pi/N), \quad b_i(0) = 2\sin(2i\pi/N), \quad i = 1, \ldots, N,$$

y tamaño de paso inicial $h_0 = 10^{-6}$.

Para ilustrar más claramente el estudio numérico de las iteraciones (a), (b) y (c), hemos dispuesto los resultados en tablas, las cuales se han agrupados en dos bloques; un primer bloque basado en los métodos de 2 etapas, y el otro bloque basadas en métodos de 3 etapas.

Para la iteración (2.1.2) con $S = I$ y L, B y γ dados en la sección 2.2, los métodos Gauss, Radau IIA, Lobatto IIIA y Lobatto IIIC de dos etapas, se han denotados por *Gauss2*, *RadIIA2*, *LobIIIA2* y *LobIIIC2* respectivamente. A la iteración (2.1.9), Se han tomado los correspondientes valores de α y β en (2.3.11) y (2.3.13) (véase también la Tabla 2.3.1), y las iteraciones son denotadas respectivamente por *Gauss2N*, *RadIIA2N*, *LobIIIA2N* y *LobIIIC2N*. También, se ha empleado la iteración al método de Gauss de dos etapas con parámetro de sobrerrelajación $\omega = 2(\sqrt{6} - \sqrt{2} + 1)^{-1}$, en este caso la iteración ha sido denotada por *Gauss2ω*.

Los esquemas iterativos de tipo (2.1.2) basados en los métodos Gauss, Radau IIA, Lobatto IIIA y Lobatto IIIC de tres etapas, las hemos denotados por *Gauss3*, *RadIIA3*, *LobIIIA3* y *LobIIIC3* respectivamente con S, L, B y γ dados en la sección 2.2. Para la nueva iteración (2.1.9), hemos tomado los valores correspondientes de α y β dados por (2.3.11) y (2.3.13) (véase también la Tabla 2.3.2), en este caso las iteraciones se denotan respectivamente por *Gauss3N*, *RadIIA3N*, *LobIIIA3N* y *LobIIIC3N*. También, hemos comparado con la iteración denotada por *Gauss3ω* propuesta por Cooper y Butcher [22] para el método de Gauss de tres etapas con parámetro de sobrerrelajación $\omega = 0{,}962825449....$

En las tablas dadas a continuación se muestran los errores de las iteraciones (usando la norma Euclídea). Además se ha medido la razón de convergencia por iteración mediante la fórmula $r_i = \| Y - Y^i \|_2 \; / \; \| Y - Y^{i-1} \|_2$.

En los resultados numéricos relativos a los Problemas 2.1 y 2.3 mostrados en la Tablas 2.4.1, 2.4.5 y 2.4.7 respectivamente, puede observarse que los valores r_i confirman el valor de la razón de convergencia $|\delta|$ predicha en la teoría (véase las Tablas 2.3.1 y 2.3.2) para la iteración (2.1.2) basada en los métodos de dos y tres etapas *Gauss2* y *Gauss3* respectivamente. Sin embargo, para el Problema 2.2 los valores de $|\delta|$ son un poco más pequeños que r_i (véase Tablas 2.4.2 y 2.4.6), pero cuando reducimos h suficientemente ($h \leq 10^{-7}$), se puede constatar que los valores de r_i son próximos a $|\delta|$. Resultados similares se obtuvieron con los tres problemas considerados cuando usamos la iteración (2.1.2) basada en los métodos Radau IIA, Lobatto IIIA y Lobatto IIIC de dos y tres etapas implícitas.

En las Tablas 2.4.1–2.4.9 también se puede apreciar que los esquemas *Gauss2N*, *Gauss3N*, *RadIIA2N*, *RadIIA3N*, *LobIIIA2N*, *LobIIIA3N*, *LobIIIC2N* y *LobIIIC3N* convergen mucho más rápido que sus correspondiente contrapartidas *Gauss2*, *Gauss3*, *LobIIIA2*, *LobIIIA3*, *RadIIA2*, *RadIIA3*, *LobIIIC2* y *LobIIIC3*, respectivamente. Solamente en el Problema 2.2 los esquemas *Gauss2N* y *Gauss3N* presentan una velocidad de convergencia similar a los

Tabla 2.4.1: Errores en las iteraciones ($i = 0, 1, \ldots, 6$) para el Problema 2.1 y método de Gauss de 2 etapas.

i	$Gauss2$	r_i	$Gauss2N$	r_i	$Gauss2\omega$	r_i
0	$0{,}11 \cdot 10^{-2}$	–	$0{,}11 \cdot 10^{-2}$	–	$0{,}11 \cdot 10^{-2}$	–
1	$0{,}10 \cdot 10^{-3}$	$0{,}10 \cdot 10^{0}$	$0{,}72 \cdot 10^{-4}$	$0{,}68 \cdot 10^{-1}$	$0{,}84 \cdot 10^{-4}$	$0{,}78 \cdot 10^{-1}$
2	$0{,}75 \cdot 10^{-5}$	$0{,}71 \cdot 10^{-1}$	$0{,}99 \cdot 10^{-6}$	$0{,}14 \cdot 10^{-1}$	$0{,}26 \cdot 10^{-5}$	$0{,}31 \cdot 10^{-1}$
3	$0{,}53 \cdot 10^{-6}$	$0{,}71 \cdot 10^{-1}$	$0{,}42 \cdot 10^{-7}$	$0{,}42 \cdot 10^{-1}$	$0{,}64 \cdot 10^{-7}$	$0{,}24 \cdot 10^{-1}$
4	$0{,}38 \cdot 10^{-7}$	$0{,}71 \cdot 10^{-1}$	$0{,}45 \cdot 10^{-9}$	$0{,}11 \cdot 10^{-1}$	$0{,}13 \cdot 10^{-8}$	$0{,}21 \cdot 10^{-1}$
5	$0{,}27 \cdot 10^{-8}$	$0{,}71 \cdot 10^{-1}$	$0{,}17 \cdot 10^{-10}$	$0{,}37 \cdot 10^{-1}$	$0{,}26 \cdot 10^{-10}$	$0{,}19 \cdot 10^{-1}$
6	$0{,}19 \cdot 10^{-9}$	$0{,}71 \cdot 10^{-1}$	$0{,}16 \cdot 10^{-12}$	$0{,}95 \cdot 10^{-2}$	$0{,}46 \cdot 10^{-12}$	$0{,}18 \cdot 10^{-1}$

Tabla 2.4.2: Errores en las iteraciones ($i = 0, 1, \ldots, 6$) para el Problema 2.2 y método de Gauss de 2 etapas.

i	$Gauss2$	r_i	$Gauss2N$	r_i	$Gauss2\omega$	r_i
0	$0{,}49 \cdot 10^{-3}$	–	$0{,}49 \cdot 10^{-3}$	–	$0{,}49 \cdot 10^{-3}$	–
1	$0{,}49 \cdot 10^{-3}$	$0{,}10 \cdot 10^{1}$	$0{,}49 \cdot 10^{-3}$	$0{,}10 \cdot 10^{1}$	$0{,}49 \cdot 10^{-3}$	$0{,}10 \cdot 10^{1}$
2	$0{,}29 \cdot 10^{-3}$	$0{,}58 \cdot 10^{0}$	$0{,}27 \cdot 10^{-3}$	$0{,}54 \cdot 10^{0}$	$0{,}28 \cdot 10^{-3}$	$0{,}57 \cdot 10^{0}$
3	$0{,}42 \cdot 10^{-4}$	$0{,}15 \cdot 10^{0}$	$0{,}21 \cdot 10^{-4}$	$0{,}78 \cdot 10^{-1}$	$0{,}28 \cdot 10^{-4}$	$0{,}10 \cdot 10^{0}$
4	$0{,}51 \cdot 10^{-5}$	$0{,}12 \cdot 10^{0}$	$0{,}13 \cdot 10^{-5}$	$0{,}65 \cdot 10^{-1}$	$0{,}21 \cdot 10^{-5}$	$0{,}72 \cdot 10^{-1}$
5	$0{,}60 \cdot 10^{-6}$	$0{,}12 \cdot 10^{0}$	$0{,}10 \cdot 10^{-6}$	$0{,}75 \cdot 10^{-1}$	$0{,}16 \cdot 10^{-6}$	$0{,}79 \cdot 10^{-1}$
6	$0{,}69 \cdot 10^{-7}$	$0{,}12 \cdot 10^{0}$	$0{,}74 \cdot 10^{-8}$	$0{,}72 \cdot 10^{-1}$	$0{,}13 \cdot 10^{-7}$	$0{,}81 \cdot 10^{-1}$

respectivos *Gauss2* y *Gauss3* (véase Tablas 2.4.2 y 2.4.6). Esto se explica por el hecho de que cuando el espectro de J es cercano al eje imaginario ambas iteraciones presentan una razón de convergencia similar (véase Tablas 2.3.1 y 2.3.2).

Las iteraciones con parámetros de sobrerrelajación $Gauss2\omega$ y $Gauss3\omega$ muestran también un buen comportamiento. Para el Gauss de dos etapas la razón de convergencia lineal es $r = 0{,}0173$ cuando los autovalores del Jacobiano están sobre $\mathbb{R}^-$. Esta iteración funciona de manera similar a la iteración acelerada que proponemos en casi todos los problemas stiff que hemos considerado.

Como conclusión se puede decir que la nueva iteración presentada aquí puede ser vista como una alternativa a la iteración óptima de sobrerrelajación, que además es fácil de implementar y que es aplicable para acelerar una iteración dada cuando el espectro de su matriz

Tabla 2.4.3: Errores en las iteraciones ($i = 0, 1, \ldots, 6$) para el Problema 2.3 y método de Gauss de 2 etapas.

i	$Gauss2$	r_i	$Gauss2N$	r_i	$Gauss2\omega$	r_i
0	$0{,}68 \cdot 10^{0}$	–	$0{,}68 \cdot 10^{0}$	–	$0{,}68 \cdot 10^{0}$	–
1	$0{,}69 \cdot 10^{0}$	$0{,}10 \cdot 10^{0}$	$0{,}48 \cdot 10^{-1}$	$0{,}71 \cdot 10^{-1}$	$0{,}55 \cdot 10^{-1}$	$0{,}81 \cdot 10^{-1}$
2	$0{,}53 \cdot 10^{-2}$	$0{,}78 \cdot 10^{-1}$	$0{,}11 \cdot 10^{-2}$	$0{,}23 \cdot 10^{-1}$	$0{,}21 \cdot 10^{-2}$	$0{,}38 \cdot 10^{-1}$
3	$0{,}43 \cdot 10^{-3}$	$0{,}80 \cdot 10^{-1}$	$0{,}57 \cdot 10^{-4}$	$0{,}52 \cdot 10^{-1}$	$0{,}83 \cdot 10^{-4}$	$0{,}39 \cdot 10^{-1}$
4	$0{,}35 \cdot 10^{-4}$	$0{,}82 \cdot 10^{-1}$	$0{,}21 \cdot 10^{-5}$	$0{,}38 \cdot 10^{-1}$	$0{,}35 \cdot 10^{-5}$	$0{,}42 \cdot 10^{-1}$
5	$0{,}30 \cdot 10^{-5}$	$0{,}84 \cdot 10^{-1}$	$0{,}10 \cdot 10^{-6}$	$0{,}47 \cdot 10^{-1}$	$0{,}16 \cdot 10^{-6}$	$0{,}45 \cdot 10^{-1}$
6	$0{,}25 \cdot 10^{-6}$	$0{,}85 \cdot 10^{-1}$	$0{,}46 \cdot 10^{-8}$	$0{,}45 \cdot 10^{-1}$	$0{,}74 \cdot 10^{-8}$	$0{,}46 \cdot 10^{-1}$

Tabla 2.4.4: Errores en las iteraciones ($i = 0, 1, \ldots, 6$) para Problema 2.1 con métodos de Radau IIA y Lobatto IIIC de 2 etapas.

i	$RadIIA2$	$RadIIA2N$	$LobIIIC2$	$LobIIIC2N$
0	$0{,}14 \cdot 10^{-2}$	$0{,}14 \cdot 10^{-2}$	$0{,}13 \cdot 10^{-2}$	$0{,}13 \cdot 10^{-2}$
1	$0{,}13 \cdot 10^{-3}$	$0{,}73 \cdot 10^{-4}$	$0{,}59 \cdot 10^{-3}$	$0{,}51 \cdot 10^{-3}$
2	$0{,}13 \cdot 10^{-4}$	$0{,}23 \cdot 10^{-5}$	$0{,}10 \cdot 10^{-3}$	$0{,}71 \cdot 10^{-5}$
3	$0{,}14 \cdot 10^{-5}$	$0{,}89 \cdot 10^{-7}$	$0{,}17 \cdot 10^{-4}$	$0{,}16 \cdot 10^{-5}$
4	$0{,}14 \cdot 10^{-6}$	$0{,}25 \cdot 10^{-8}$	$0{,}29 \cdot 10^{-5}$	$0{,}19 \cdot 10^{-7}$
5	$0{,}14 \cdot 10^{-7}$	$0{,}80 \cdot 10^{-10}$	$0{,}49 \cdot 10^{-6}$	$0{,}34 \cdot 10^{-8}$
6	$0{,}14 \cdot 10^{-8}$	$0{,}21 \cdot 10^{-11}$	$0{,}83 \cdot 10^{-7}$	$0{,}39 \cdot 10^{-10}$

Tabla 2.4.5: Errores en las iteraciones ($i = 0, 1, \ldots, 6$) para el Problema 2.1 y método de Gauss de 3 etapas.

i	$Gauss3$	r_i	$Gauss3N$	r_i	$Gauss3\omega$	r_i
0	$0{,}13 \cdot 10^{-2}$	–	$0{,}13 \cdot 10^{-2}$	–	$0{,}13 \cdot 10^{-2}$	–
1	$0{,}29 \cdot 10^{-3}$	$0{,}21 \cdot 10^{0}$	$0{,}26 \cdot 10^{-3}$	$0{,}19 \cdot 10^{0}$	$0{,}26 \cdot 10^{-3}$	$0{,}19 \cdot 10^{0}$
2	$0{,}28 \cdot 10^{-4}$	$0{,}14 \cdot 10^{0}$	$0{,}46 \cdot 10^{-5}$	$0{,}18 \cdot 10^{-1}$	$0{,}67 \cdot 10^{-5}$	$0{,}70 \cdot 10^{-1}$
3	$0{,}52 \cdot 10^{-5}$	$0{,}16 \cdot 10^{0}$	$0{,}69 \cdot 10^{-6}$	$0{,}15 \cdot 10^{0}$	$0{,}74 \cdot 10^{-6}$	$0{,}82 \cdot 10^{-1}$
4	$0{,}80 \cdot 10^{-6}$	$0{,}16 \cdot 10^{0}$	$0{,}94 \cdot 10^{-8}$	$0{,}14 \cdot 10^{-1}$	$0{,}17 \cdot 10^{-7}$	$0{,}60 \cdot 10^{-1}$
5	$0{,}13 \cdot 10^{-6}$	$0{,}16 \cdot 10^{0}$	$0{,}14 \cdot 10^{-8}$	$0{,}14 \cdot 10^{0}$	$0{,}15 \cdot 10^{-8}$	$0{,}64 \cdot 10^{-1}$
6	$0{,}20 \cdot 10^{-7}$	$0{,}16 \cdot 10^{0}$	$0{,}17 \cdot 10^{-10}$	$0{,}12 \cdot 10^{-1}$	$0{,}29 \cdot 10^{-10}$	$0{,}51 \cdot 10^{-1}$

Tabla 2.4.6: Errores en las iteraciones ($i = 0, 1, \ldots, 6$) para el Problema 2.2 y método de Gauss de 3 etapas.

i	$Gauss3$	r_i	$Gauss3N$	r_i	$Gauss3\omega$	r_i
0	$0{,}38 \cdot 10^{-3}$	–	$0{,}38 \cdot 10^{-3}$	–	$0{,}38 \cdot 10^{-3}$	–
1	$0{,}38 \cdot 10^{-3}$	$0{,}10 \cdot 10^{1}$	$0{,}38 \cdot 10^{-3}$	$0{,}10 \cdot 10^{1}$	$0{,}38 \cdot 10^{-3}$	$0{,}10 \cdot 10^{1}$
2	$0{,}24 \cdot 10^{-4}$	$0{,}25 \cdot 10^{0}$	$0{,}32 \cdot 10^{-4}$	$0{,}83 \cdot 10^{-1}$	$0{,}29 \cdot 10^{-4}$	$0{,}27 \cdot 10^{0}$
3	$0{,}14 \cdot 10^{-4}$	$0{,}33 \cdot 10^{0}$	$0{,}11 \cdot 10^{-4}$	$0{,}36 \cdot 10^{0}$	$0{,}13 \cdot 10^{-4}$	$0{,}32 \cdot 10^{0}$
4	$0{,}29 \cdot 10^{-5}$	$0{,}30 \cdot 10^{0}$	$0{,}50 \cdot 10^{-5}$	$0{,}44 \cdot 10^{0}$	$0{,}46 \cdot 10^{-5}$	$0{,}33 \cdot 10^{0}$
5	$0{,}22 \cdot 10^{-6}$	$0{,}22 \cdot 10^{0}$	$0{,}12 \cdot 10^{-5}$	$0{,}24 \cdot 10^{0}$	$0{,}94 \cdot 10^{-6}$	$0{,}30 \cdot 10^{0}$
6	$0{,}34 \cdot 10^{-7}$	$0{,}21 \cdot 10^{0}$	$0{,}18 \cdot 10^{-6}$	$0{,}15 \cdot 10^{0}$	$0{,}12 \cdot 10^{-6}$	$0{,}26 \cdot 10^{0}$

Tabla 2.4.7: Errores en las iteraciones ($i = 0, 1, \ldots, 6$) para el Problema 2.3 y método de Gauss de 3 etapas.

i	$Gauss3$	r_i	$Gauss3N$	r_i	$Gauss3\omega$	r_i
0	$0{,}85 \cdot 10^{0}$	–	$0{,}85 \cdot 10^{0}$	–	$0{,}85 \cdot 10^{0}$	–
1	$0{,}18 \cdot 10^{0}$	$0{,}22 \cdot 10^{0}$	$0{,}16 \cdot 10^{0}$	$0{,}19 \cdot 10^{0}$	$0{,}16 \cdot 10^{0}$	$0{,}19 \cdot 10^{0}$
2	$0{,}19 \cdot 10^{-1}$	$0{,}15 \cdot 10^{0}$	$0{,}29 \cdot 10^{-2}$	$0{,}18 \cdot 10^{-1}$	$0{,}52 \cdot 10^{-2}$	$0{,}78 \cdot 10^{-1}$
3	$0{,}37 \cdot 10^{-2}$	$0{,}16 \cdot 10^{0}$	$0{,}54 \cdot 10^{-3}$	$0{,}19 \cdot 10^{0}$	$0{,}64 \cdot 10^{-3}$	$0{,}91 \cdot 10^{-1}$
4	$0{,}61 \cdot 10^{-3}$	$0{,}16 \cdot 10^{0}$	$0{,}16 \cdot 10^{-4}$	$0{,}29 \cdot 10^{-1}$	$0{,}35 \cdot 10^{-4}$	$0{,}80 \cdot 10^{-1}$
5	$0{,}11 \cdot 10^{-3}$	$0{,}17 \cdot 10^{0}$	$0{,}22 \cdot 10^{-5}$	$0{,}14 \cdot 10^{0}$	$0{,}33 \cdot 10^{-5}$	$0{,}83 \cdot 10^{-1}$
6	$0{,}19 \cdot 10^{-4}$	$0{,}17 \cdot 10^{0}$	$0{,}13 \cdot 10^{-6}$	$0{,}58 \cdot 10^{-1}$	$0{,}24 \cdot 10^{-6}$	$0{,}81 \cdot 10^{-1}$

Tabla 2.4.8: Errores en las iteraciones ($i = 0, 1, \ldots, 6$) para Problema 2.1 con métodos de Radau IIA, Lobatto IIIA y Lobatto IIIC de 3 etapas.

i	$RadIIA3$	$RadIIA3N$	$LobIIIA3$	$LobIIIA3N$	$LobIIIC3$	$LobIIIC3N$
0	$0{,}16 \cdot 10^{-2}$	$0{,}16 \cdot 10^{-2}$	$0{,}16 \cdot 10^{-2}$	$0{,}16 \cdot 10^{-2}$	$0{,}16 \cdot 10^{-2}$	$0{,}16 \cdot 10^{-2}$
1	$0{,}52 \cdot 10^{-3}$	$0{,}66 \cdot 10^{-3}$	$0{,}65 \cdot 10^{-3}$	$0{,}66 \cdot 10^{-3}$	$0{,}15 \cdot 10^{-3}$	$0{,}53 \cdot 10^{-3}$
2	$0{,}82 \cdot 10^{-4}$	$0{,}61 \cdot 10^{-5}$	$0{,}83 \cdot 10^{-4}$	$0{,}36 \cdot 10^{-5}$	$0{,}24 \cdot 10^{-4}$	$0{,}11 \cdot 10^{-4}$
3	$0{,}18 \cdot 10^{-4}$	$0{,}28 \cdot 10^{-5}$	$0{,}14 \cdot 10^{-4}$	$0{,}18 \cdot 10^{-5}$	$0{,}77 \cdot 10^{-5}$	$0{,}40 \cdot 10^{-5}$
4	$0{,}36 \cdot 10^{-5}$	$0{,}17 \cdot 10^{-7}$	$0{,}22 \cdot 10^{-5}$	$0{,}66 \cdot 10^{-8}$	$0{,}21 \cdot 10^{-5}$	$0{,}53 \cdot 10^{-7}$
5	$0{,}75 \cdot 10^{-6}$	$0{,}91 \cdot 10^{-8}$	$0{,}36 \cdot 10^{-6}$	$0{,}37 \cdot 10^{-8}$	$0{,}60 \cdot 10^{-6}$	$0{,}24 \cdot 10^{-7}$
6	$0{,}15 \cdot 10^{-6}$	$0{,}56 \cdot 10^{-10}$	$0{,}58 \cdot 10^{-7}$	$0{,}14 \cdot 10^{-10}$	$0{,}17 \cdot 10^{-6}$	$0{,}28 \cdot 10^{-9}$

Tabla 2.4.9: Errores en las iteraciones ($i = 0, 1, \ldots, 6$) para Problema 2.3 con métodos de Radau IIA, Lobatto IIIA y Lobatto IIIC de 3 etapas.

i	$RadIIA3$	$RadIIA3N$	$LobIIIA3$	$LobIIIA3N$	$LobIIIC3$	$LobIIIC3N$
0	$0{,}99 \cdot 10^{0}$	$0{,}99 \cdot 10^{0}$	$0{,}94 \cdot 10^{0}$	$0{,}94 \cdot 10^{0}$	$0{,}93 \cdot 10^{0}$	$0{,}93 \cdot 10^{0}$
1	$0{,}33 \cdot 10^{0}$	$0{,}42 \cdot 10^{0}$	$0{,}41 \cdot 10^{0}$	$0{,}42 \cdot 10^{0}$	$0{,}92 \cdot 10^{-1}$	$0{,}33 \cdot 10^{0}$
2	$0{,}54 \cdot 10^{-1}$	$0{,}74 \cdot 10^{-2}$	$0{,}54 \cdot 10^{-1}$	$0{,}48 \cdot 10^{-2}$	$0{,}15 \cdot 10^{-1}$	$0{,}10 \cdot 10^{-1}$
3	$0{,}13 \cdot 10^{-1}$	$0{,}25 \cdot 10^{-2}$	$0{,}98 \cdot 10^{-2}$	$0{,}15 \cdot 10^{-2}$	$0{,}50 \cdot 10^{-2}$	$0{,}39 \cdot 10^{-2}$
4	$0{,}28 \cdot 10^{-2}$	$0{,}14 \cdot 10^{-3}$	$0{,}16 \cdot 10^{-2}$	$0{,}53 \cdot 10^{-4}$	$0{,}15 \cdot 10^{-2}$	$0{,}39 \cdot 10^{-3}$
5	$0{,}64 \cdot 10^{-3}$	$0{,}22 \cdot 10^{-4}$	$0{,}29 \cdot 10^{-3}$	$0{,}67 \cdot 10^{-5}$	$0{,}49 \cdot 10^{-3}$	$0{,}84 \cdot 10^{-4}$
6	$0{,}15 \cdot 10^{-3}$	$0{,}21 \cdot 10^{-5}$	$0{,}50 \cdot 10^{-4}$	$0{,}43 \cdot 10^{-6}$	$0{,}16 \cdot 10^{-3}$	$0{,}13 \cdot 10^{-4}$

de amplificación del error es conocida. Se ha estudiado en detalle solamente el caso cuando el espectro varía en algún intervalo real conocido. En la práctica, podría desarrollarse algún mecanismo para detectar el conjunto donde se mueve el espectro de la matriz Jacobiana del problema a integrar, y luego usar esto para obtener valores apropiados para los parámetros α y β los cuales no tendrían que ser constantes en todas las iteraciones y que podrían elegirse dinámicamente con las iteraciones.

Capítulo 3

Métodos iterativos para problemas especiales de segundo orden

3.1. Introducción

Consideramos la solución numérica de problemas oscilatorios para Problemas de Valor Inicial en sistemas diferenciales de segundo orden de tipo especial

$$\begin{cases} y'' = f\big(t, y(t)\big), & t \in [t_0, t_f] \\ y(t_0) = y_0, \quad y'(t_0) = y'_0, & y, y', f \in \mathbb{R}^m. \end{cases} \tag{3.1.1}$$

Este tipo de problemas describen usualmente fenómenos vibratorios o quasi-periódicos y sus soluciones $y(t)$ combinan a menudo componentes dominantes de baja frecuencia con componentes menos dominantes en las frecuencias medias y altas.

La mayoría de los métodos numéricos propuestos para integrar (3.1.1) se clasifican en tres grupos: los métodos Lineales Multipaso (LM), los métodos Directos Híbridos y los métodos de un Paso. Los métodos Lineales Multipaso tienen el inconveniente de que el orden de convergencia está limitado a dos si se requiere la propiedad de P–Estabilidad, véase [68] y [62, p. 417]. La propiedad de P–Estabilidad indica que el método no amplifica ni amortigua las amplitudes de onda en problemas lineales, es decir conserva la amplitud de onda asociada a cada frecuencia. Para superar la barrera de la P–Estabilidad Cash [14, 15], Chawla et al. [17, 18] y Thomas [80] han considerado variantes híbridas de Métodos Multipaso (LM), introduciendo puntos adicionales fuera del paso natural en la fórmula LM, de este modo estos autores han derivado métodos Híbridos P–Estables de ordenes $4, 5, \ldots, 8$. Por otro

lado, es posible obtener fórmulas P–Estables de ordenes arbitrariamente grandes basadas en los métodos Runge–Kutta–Nyström (RKN), tal como se indica en [54]. Además, Van der Houwen y Sommeijer [62], Sharp et al. [78] y Franco et al. [33] entre otros, han obtenidos métodos Runge–Kutta–Nyström diagonalmente implícitos (DIRKN) P–estables de ordenes medios (4, 5, 6 y 7) y con altos ordenes de convergencia en la fase al considerar problemas lineales.

El atractivo de los métodos DIRKN dentro de la clase de métodos Runge–Kutta Nyström implícitos estriba en que la estructura especial de su matriz de coeficientes permite reducir el costo algebraico al resolver sus ecuaciones de etapa mediante alguna iteración de tipo Newton. Sin embargo, para obtener la misma precisión en términos del orden de convergencia, los métodos DIRKN requieren más etapas internas que los métodos RKN altamente implícitos, tales como aquellos derivados de la familia Gauss (o Lobatto IIIA, véase [57, pp. 72–75]), en versión RKN los cuales son también P–estables. Por esta razón, la búsqueda de implementaciones de bajo costo computacional para estos últimos métodos es una alternativa interesante y posiblemente más eficiente que la de los métodos DIRKN.

En este capítulo se pretenden alcanzar varios objetivos que finalmente conducirán a implementaciones eficientes de métodos RKN altamente implícitos. Dirigiremos nuestra atención principalmente a la familia de métodos Runge–Kutta Gauss, aunque los resultados pueden extenderse sin ninguna dificultad a otros métodos altamente implícitos. Para ello, nuestro primer trabajo será analizar con cierta profundidad la Iteración Quasi–Newton (QNI) también conocida en la literatura como iteración de Newton Simplificada o iteración de Newton Modificada [57, Ch. IV], [3, 9]. En las secciones 3.2 y 3.3, se desarrollará un esquema iterativo alternativo, que denominaremos Single-Newton (SNI), el cual reduce el costo computacional por iteración respecto a la iteración Quasi–Newton, además se prueban dos teoremas relacionados con el orden de convergencia global de los métodos de Gauss cuando se da un número fijo de iteraciones con cualquiera de los esquemas iterativos considerados anteriormente. En las secciones 3.4 y 3.5 se suministrarán algoritmos de arranque para las etapas internas (predictores) de varios ordenes con el objetivo de seleccionar el mejor de ellos en cada paso de la integración. En la sección 3.5 se explicitará la estrategia de orden variable (VOS) para los predictores. Por último, en la sección 3.6 se presentarán algunos experimentos numéricos con el objetivo de comparar los distintos esquemas iterativos considerados. Además, se corroborará la teoría desarrollada en las secciones anteriores. Los resultados presentados aquí están publicados esencialmente en [46] y en menor medida se encuentran parcialmente recogidos o

aplicados en [48], [51] y [75].

3.2. Alternativas a la iteración Quasi–Newton para métodos RKN

Un método Runge–Kutta–Nyström $(\bar{A}, c, b, \bar{b})$ de s etapas, avanza la solución numérica desde (t_n, y_n, y'_n) a $(t_{n+1} = t_n + h, y_{n+1}, y'_{n+1})$ por medio de las fórmulas

$$\begin{aligned} y_{n+1} &= y_n + hy'_n + h^2(\bar{b}^T \otimes I)F(Y_n), \\ y'_{n+1} &= y'_n + h(b^T \otimes I)F(Y_n), \end{aligned} \tag{3.2.1}$$

donde las etapas internas $Y_n^T = (Y_{n,1}^T, \ldots, Y_{n,s}^T) \in \mathbb{R}^{ms}$ se calculan en cada paso mediante la solución del sistema algebraico,

$$Y_n = e \otimes y_n + h(c \otimes y'_n) + h^2(\bar{A} \otimes I)F(Y_n), \tag{3.2.2}$$

con $F^T(Y_n) := \big(f^T(t_n + c_1 h, Y_{n,1}), \ldots, f^T(t_n + c_s h, Y_{n,s})\big) \in \mathbb{R}^{ms}$, $e = (1, \ldots, 1)^T \in \mathbb{R}^s$, $\otimes$ denota el producto de Kronecker de matrices $(A \otimes B = (a_{ij}B))$ e I es la matriz identidad con dimensión m (otras veces la dimensión se deducirá fácilmente del contexto). La matriz $\bar{A} = (\bar{a}_{i,j})_{i,j=1,s}$ y los vectores $c^T = (c_j)_{j=1,s}$, $b^T = (b_j)_{j=1,s}$ y $\bar{b}^T = (\bar{b}_j)_{j=1,s}$ representan los coeficientes del método RKN en cuestión.

Al aplicar un método Runge–Kutta al problema de segundo orden (3.1.1) se obtiene un método RKN que satisface

$$\bar{A} = A^2, \qquad \bar{b}^T = b^T A, \quad c = Ae, \tag{3.2.3}$$

donde la matriz A, y los vectores b^T y c son los coeficientes de un método Runge–Kutta que se aplica a sistemas diferenciales de primer orden. En nuestro caso el método original es el método Runge–Kutta Gauss, el cual posee una matriz llena no-singular, es decir $\det A \neq 0$.

La iteración Quasi–Newton computa las etapas internas por la fórmula

$$\big(I - h^2(A^2 \otimes J)\big)(Y_n^{(\nu)} - Y_n^{(\nu-1)}) = D(Y_n^{(\nu-1)}, y_n, hy'_n), \quad \nu = 1, 2, \ldots, \tag{3.2.4}$$

donde $Y_n^{(0)}$ es un valor inicial dado por algún predictor, $J \simeq \dfrac{\partial f}{\partial y}(t_n, y_n)$ y el residual $D(.)$ está definido mediante la fórmula

$$D(Z, u, v) := -Z + e \otimes u + (c \otimes v) + h^2(A^2 \otimes I)F(Z), \quad u, v \in \mathbb{R}^m, Z \in \mathbb{R}^{ms}. \tag{3.2.5}$$

Para el método de Gauss de s etapas el costo algebraico de la iteración (3.2.4) puede reducirse desacoplando los ms sistemas lineales reales en $[s/2]$ sistemas lineales complejos (aquí $[x]$ denota la parte entera del número real x) de dimensión m más un sistema lineal real de dimensión m cuando s es impar. Estos sistemas lineales usualmente se resuelven por medio de la factorización LU, reduciendo de este modo la resolución de cada sistema lineal a dos sistemas triangulares. Además la misma factorización LU puede usarse para todas las iteraciones en cada paso de integración y también en varios pasos consecutivos de integración cuando la convergencia del esquema iterativo es rápida y el tamaño de paso no cambia. Para más detalles sobre esta técnica cuando se aplica al método de Gauss de 2 etapas, véase [37, pp. 187–190].

Ya que la convergencia en (3.2.4) se alcanza normalmente después de μ iteraciones, entonces reemplazaríamos $Y_n \simeq Y_n^{(\mu)}$, y la solución de avance se calcula usando las ecuaciones (3.2.1). Este modo de actuación presenta dos inconvenientes. Primero se requieren s evaluaciones extras de la derivada, es decir, la actualización del supervector $F(Y_n^{(\mu)})$, y en segundo lugar (y más importante) el error de las aproximaciones $F(Y_n) - F(Y_n^{(\mu)})$ puede amplificarse de forma sustancial si la función f posee una constante de Lipschitz grande respecto de y. Para solventar ambos inconvenientes y siguiendo a varios autores [37, 77] recomendamos el uso de la formulación alternativa para la computación de y_{n+1} y de hy'_{n+1}

$$\begin{aligned} y_{n+1} &= r^* y_n + (b^T A^{-1} \otimes I) Y_n, \\ v_{n+1} &= r' y_n + r^* v_n + (b^T A^{-2} \otimes I) Y_n. \end{aligned} \tag{3.2.6}$$

con

$$\begin{aligned} r^* &= 1 - b^T A^{-1} e, \quad r' = -b^T A^{-2} e, \\ v_n &:= hy'_n, \quad v_{n+1} := hy'_{n+1}. \end{aligned} \tag{3.2.7}$$

La cual se sigue de (3.2.3) sin más que tener en cuenta que

$$\big((\bar{A})^{-1} \otimes I\big)\big(Y_n - e \otimes y_n - h(c \otimes y'_n)\big) = h^2 F(Y_n), \tag{3.2.8}$$

e insertar esta expresión en (3.2.1).

A efectos de analizar la propagación de errores en el esquema iterativo Quasi–Newton (QNI), es interesante notar que, como veremos a continuación, sobre sistemas diferenciales no lineales en general, el error $Y_n - Y_n^{(\nu)}$ se comporta como

$$Y_n - Y_n^{(\nu)} = \mathcal{O}(h^3)(Y_n - Y_n^{(\nu-1)}), \quad h \to 0, \quad \nu = 1, 2, \ldots, \tag{3.2.9}$$

siempre que asumamos que

$$J_n - \frac{\partial f}{\partial y}(t_n, y_n) = \mathcal{O}(h) \quad \text{y} \quad Y_n - Y_n^{(0)} = \mathcal{O}(h), \quad \text{cuando } h \to 0. \tag{3.2.10}$$

Para ver esto, téngase en cuenta (3.2.4)-(3.2.5) y úsese que

$$D(Y_n^{(\nu-1)}, y_n, v_n) - D(Y_n, y_n, v_n) = \big(-I + h^2(A^2 \otimes J_n) + \mathcal{O}(h^3)\big)(Y_n^{(\nu-1)} - Y_n), \quad \nu = 1, 2, \ldots$$

De (3.2.9) se deduce ahora que

$$Y_n - Y_n^{(\nu)} = h^{3\nu}\big(C_\nu + \mathcal{O}(h)\big)(Y_n - Y_n^{(0)}), \quad \nu = 0, 1, \ldots, \quad (h \to 0), \tag{3.2.11}$$

donde C_ν es una matriz de coeficientes constantes que podría variar con las iteraciones $\nu = 1, 2, 3 \ldots$.

Siguiendo a Gladwell y Thomas [37], podemos decir que la Iteración QNI para problemas de segundo orden, presenta algunas desventajas que han conducido a varios autores a buscar mejores alternativas, véase por ejemplo [37, 72]. Una de los principales inconvenientes es que requiere el uso de la aritmética compleja para una implementación eficiente de la misma en los métodos altamente implícitos, tales como los de la familia Gauss de $s \geq 2$ etapas. El uso de la aritmética compleja conlleva normalmente a que los costos operacionales en cada iteración (factorizaciones LU, solución de sistemas triangulares, multiplicación de una matriz por un vector) son cuatro veces mayores que en el caso real. Por otra parte, con respecto al almacenamiento y la actualización de la matriz, se puede observar que el QNI involucra $[s/2]$ factorizaciones LU complejas y $2(s/2 - [s/2])$ factorizaciones reales por paso de integración. También vale la pena mencionar las conclusiones de Gladwell y Thomas [37, p. 205], donde para el método de Gauss de dos etapas ellos recomiendan usar, en lugar de la iteración QNI, la iteración propuesta por Cooper y Butcher en [22]. Sin embargo, la iteración de Cooper y Butcher en [22] cuando se aplica a (3.1.1) presenta algunos inconvenientes. En primer lugar, fue diseñada para problemas de primer orden y requiere una cantidad no despreciable de transformaciones intermedias que involucran productos de matriz por vector, véase [37, pp. 192–193] para el caso del método de Gauss de dos etapas. En segundo lugar y más importante, el orden de aproximación en potencias de h no aumenta con las iteraciones sucesivas. Por lo tanto, independientemente del número de iteraciones dadas por paso de integración, al final tendríamos un método del mismo orden que el predictor usado, aunque eso sí, con coeficientes de error muy pequeños.

Para superar las desventajas de la iteración [22] sobre problemas de segundo orden, proponemos una iteración directa de tipo Newton para resolver las ecuaciones de etapas (3.2.2), la

cual tiene cierta similitud a la iteración denominada "de cuadrado perfecto" (*perfect–square iteration*, véase [72]). Nuestra iteración puede considerarse como la versión para problemas de segundo orden, de la iteración Single–Newton en [41, véase (1.8)]. La nueva iteración propuesta es

$$\begin{gathered}\big(I - h^2(T \otimes J_n)\big)(Y_n^{(\nu)} - Y_n^{(\nu-1)}) = D(Y_n^{(\nu-1)}, y_n, hy'_n), \quad \nu = 1, 2, \ldots, \\ J_n \simeq \frac{\partial f}{\partial y}(t_n, y_n),\end{gathered} \tag{3.2.12}$$

con el residual $D(.)$ definido en (3.2.5) y $T \in \mathbb{R}^{s\times s}$ es una matriz constante con espectro unipuntual $\sigma(T) = \{\gamma\}, \quad \gamma > 0$. Esta matriz T será optimizada apropiadamente.

Ya que la matriz T tiene espectro simple, podemos descomponerla como

$$T = \gamma S(I - L)^{-1}S^{-1}, \tag{3.2.13}$$

donde L es una matriz triangular inferior estricta y S es una matriz no singular. De esta manera, sustituyendo (3.2.13) en (3.2.12) y realizando algunas operaciones matriciales elementales podemos reescribir (3.2.12) como

$$\begin{cases} \big(\xi I - (I \otimes J_n)\big)\Delta^{(\nu)} = (\xi P \otimes I)D(Y_n^{(\nu-1)}, y_n, hy'_n) + (\xi L \otimes I)\Delta^{(\nu)}, \quad \nu = 1, 2, \ldots, \\ Y^{(\nu)} = Y^{(\nu-1)} + (S \otimes I)\Delta^{(\nu)}, \quad \text{con } \xi = (\gamma h^2)^{-1} \text{ y } \quad P = (I - L)S^{-1}. \end{cases} \tag{3.2.14}$$

Para propósitos computacionales la formulación (3.2.14) es preferible a (3.2.12), ya que (3.2.14) permite desacoplar los ms sistemas lineales de cada iteración en s sistemas lineales de dimensión m. Además, sólo una factorización real LU (de dimensión m) es necesaria en el caso de (3.2.14). Sin embargo, para los fines del análisis trabajaremos con la formulación algebraica equivalente dada por (3.2.12).

Es importante recalcar que cuando se alcanza la convergencia (numérica) en (3.2.14), digamos después de μ iteraciones, usualmente reemplazamos $Y_n \simeq Y_n^{(\mu)}$ y calculamos la solución de avance por medio de las fórmulas (3.2.6), la cual es más estable como mencionamos anteriormente.

El error de las iteraciones en (3.2.12) sobre sistemas diferenciales satisface la recursión

$$Y_n - Y_n^{(\nu)} = \mathcal{O}(h^2)(Y_n - Y_n^{(\nu-1)}), \quad h \to 0, \quad \nu = 1, 2, \ldots, \tag{3.2.15}$$

presuponiendo que se verifica (3.2.10). Esto implica que

$$Y_n - Y_n^{(\nu)} = h^{2\nu}\big(C_\nu^* + \mathcal{O}(h)\big)(Y_n - Y_n^{(0)}), \quad h \to 0, \quad \nu = 0, 1, \ldots, \tag{3.2.16}$$

Tabla 3.2.1: Costo de la Iteración Quasi–Newton y (3.2.12)

Esquema Iterativo	*Iter*	*LS*	f–eval	flops	Precisión
Quasi–Newton	ν	ν (Complejo)	2ν	$6\nu m^2$	$\mathcal{O}(h^{3\nu+q})$
(3.2.12)	$3\nu/2$	3ν (Real)	3ν	$6\nu m^2$	$\mathcal{O}(h^{3\nu+q})$

donde C_ν^* es una matriz de coeficientes constantes que depende de ν. El término $\mathcal{O}(h)$ en (3.2.16) puede ser grande para problemas con grandes constantes de Lipschitz.

Para obtener la misma precisión en las etapas cuando comparamos la iteración Quasi–Newton (3.2.4) (versión compleja) y la iteración (3.2.12), digamos la misma potencia de h, tenemos en virtud de (3.2.11) y (3.2.16), que $\nu_T \simeq \frac{3}{2}\nu_{\bar{A}}$. Donde ν_T y $\nu_{\bar{A}}$ denotan el número de iteraciones cuando usamos las fórmulas (3.2.14) y (3.2.4) respectivamente. Para el método de Gauss de s etapas ($s = 2, 4, 6, \ldots$) comparamos en la Tabla 3.2.1 el costo computacional de ambas iteraciones para alcanzar una precisión similar. Aquí, una evaluación de la derivada se computa como $2m^2$ flops (sumas más productos), y supondremos que conlleva el mismo costo que la solución de los dos sistemas triangulares en la forma LU. En la Tabla 3.2.1, *Iter* denota el número de iteraciones dadas, LS el número de sistemas lineales reales m–dimensionales con matriz LU factorizada, fn el número de evaluaciones de la derivada. Además se asume que $Y_n - Y_n^{(0)} = \mathcal{O}(h^q)$. De la Tabla 3.2.1 resulta claro que ambas iteraciones alcanzan la misma precisión (en términos de potencias de h) con el mismo costo. Sin embargo debe mencionarse que el costo para las factorizaciones LU no ha sido tenido en cuenta en la Tabla 3.2.1. Ya que las factorizaciones contribuyen sustancialmente al costo total de los métodos (y la factorización LU es proporcional al cubo de la dimensión de la ODE), se debe recalcar que la QNI (3.2.4) requiere $s/2$ descomposiciones LU complejas resultando aproximadamente en $4sm^3/3$ flops, mientras que la iteración Single–Newton (3.2.14) sólo demanda una factorización real LU, aproximadamente de $2m^3/3$ flops. Por lo tanto es de esperar que el proceso iterativo (3.2.14) sea más eficiente que la iteración Quasi–Newton, especialmente para problemas de dimensión media y grande.

3.2.1. Análisis de los errores globales de los métodos al usar los esquemas iterativos Quasi–Newton y Single Newton tras μ iteraciones por paso de integración

Un aspecto interesante es el análisis de los errores globales del método Runge-Kutta Nyström cuando usamos las iteraciones (3.2.4) o (3.2.12) para resolver las ecuaciones de etapa, dando un número fijo de iteraciones μ en cada paso de la integración. De forma mas precisa, estamos interesados en el tamaño de los errores

$$\varepsilon_n := y_n - y_n{}^{(\mu)}, \qquad \tau_n := h(y'_n - y'_n{}^{(\mu)}), \qquad n = 1, 2, \ldots, N, \tag{3.2.17}$$

donde (y_n, y'_n) denota la solución exacta Runge–Kutta después de n pasos consecutivos de tamaño $h = (t_f - t_0)/N$ y $(y_n^{(\mu)}, y'_n{}^{(\mu)})$ es la solución numérica después de n pasos consecutivos y μ iteraciones por paso de integración. También asumimos que el mismo predictor es usado en cada paso de la integración para comenzar las iteraciones, excepto posiblemente en el primer paso.

Para hacer el análisis necesitamos algunas notaciones extras. Así, considerando que hemos seleccionado la iteración Single Newton (3.2.12). $Y_n^{(0)}$ representará el predictor para las etapas a efectos de avanzar de $(t_n, y_n^{(\mu)}, y'_n{}^{(\mu)})$ a $(t_{n+1}, y_{n+1}^{(\mu)}, y'_{n+1}{}^{(\mu)})$; $Y_n^{(\nu)}$ $(\nu = 1, \ldots, \mu)$ denota la ν–ésima iteración , $Y_n^{(\infty)}$ representa la solución exacta de las ecuaciones de etapas $D(Y, y_n^{(\mu)}, hy'_n{}^{(\mu)}) = 0$ y Y_n la solución exacta Runge–Kutta de las etapas después de n pasos de integración. Asumiremos que la aproximación inicial es de orden q, es decir,

$$Y_n^{(\infty)} - Y_n^{(0)} = \big(K(t_n) + \mathcal{O}(h)\big)h^q. \tag{3.2.18}$$

Donde $Y_n^{(0)}$ denota una aproximación inicial basada en la información obtenida en pasos previos $(y_{n-1}^{(\mu)}, y'_{n-1}{}^{(\mu)}, Y_{n-1}^{(\mu)})$ y $K(t) = \mathcal{O}(1)$ se supone que es una función "suave" de t (vector de dimensión ms), la cual depende de las diferenciales elementales en la q–ésima derivada de la solución exacta local $y(t; t_{n-1}, y_{n-1}^{(\mu)}, y'^{(\mu)}_{n-1})$.

Por ejemplo, el predictor,

$$Y_n^{(0)} := e \otimes y_n^{(\mu)} + h(c \otimes y'_n{}^{(\mu)})$$

es una aproximación inicial de segundo orden, ya que por (3.2.5), tenemos que

$$D(Y_n^{(\infty)}, y_n^{(\mu)}, y'^{(\mu)}_n) = 0,$$

o bien

$$Y_n^{(\infty)} = e \otimes y_n^{(\mu)} + c \otimes h{y'}_n^{(\mu)} + h^2(A \otimes I)F(Y_n^{(\infty)}),$$

y de aquí se deduce que

$$Y_n^{(\infty)} - Y_n^{(0)} = \big((A^2 \otimes I)F(e \otimes y_n^{(\mu)}) + \mathcal{O}(h)\big)h^2.$$

Con la notación anterior, de (3.2.17), (3.2.6) y (3.2.7) se sigue que

$$\begin{aligned} \varepsilon_{n+1} &= r^*\varepsilon_n + (b^T A^{-1} \otimes I)(Y_n - Y_n^{(\mu)}) \\ \tau_{n+1} &= r'\varepsilon_n + r^*\tau_n + (b^T A^{-2} \otimes I)(Y_n - Y_n^{(\mu)}). \end{aligned} \qquad n = 0, 1, \ldots \tag{3.2.19}$$

Ahora, separamos

$$Y_n - Y_n^{(\mu)} = (Y_n - Y_n^{(\infty)}) + (Y_n^{(\infty)} - Y_n^{(\mu)}), \tag{3.2.20}$$

entonces, en virtud de que

$$D(Y_n, y_n, hy'_n) = D(Y_n^{(\infty)}, y_n^{(\mu)}, {hy'}_n^{(\mu)}) = 0,$$

asumiendo que f es suave, se sigue tras unos cálculos simples que

$$Y_n - Y_n^{(\infty)} = \big(I + h^2\mathcal{O}(1)\big)(e \otimes \varepsilon_n + c \otimes \tau_n). \tag{3.2.21}$$

Para estudiar el segundo sumando en (3.2.20) hacemos

$$E^{(\nu)} := Y_n^{(\infty)} - Y_n^{(\nu)}, \quad \nu = 0, 1, \ldots.$$

De (3.2.12) se sigue que

$$\big(I - h^2(T \otimes J_n)\big)(-E^{(\nu)} + E^{(\nu-1)}) = D(Y_n^{(\nu-1)}, y_n^{(\mu)}, {hy'}_n^{(\mu)}) - D(Y_n^{(\infty)}, y_n^{(\mu)}, {hy'}_n^{(\mu)}) =$$
$$\big(-I + h^2(A^2 \otimes J_n) + \mathcal{O}(h^3)\big)(-E^{(\nu-1)}), \quad \nu = 1, 2, \ldots,$$

donde hemos usado que

$$J_n = \frac{\partial f}{\partial y}(t_n, y_n^{(\mu)}) + \mathcal{O}(h).$$

De aquí, teniendo en cuenta (3.2.18) llegamos a que

$$\begin{aligned} E^{(\mu)} &= h^{2\mu}\big((A^2 - T)^\mu \otimes (J_n)^\mu + \mathcal{O}(h)\big)E^{(0)} = \\ &h^{2\mu+q}\Big(\big((A^2 - T)^\mu \otimes (J_n)^\mu\big)K(t_n) + \mathcal{O}(h)\Big). \end{aligned} \tag{3.2.22}$$

Ahora, calculando $Y_n - Y_n^{(\mu)}$ de (3.2.20), usando (3.2.21) y (3.2.22), y llevando esto a (3.2.19), tenemos para $\mu \geq 1$ que

$$\begin{aligned} \varepsilon_{n+1} = \;& (1 + \mathcal{O}(h^2))\varepsilon_n + (1 + \mathcal{O}(h^2))\tau_n + \\ & h^{2\mu+q}\Big(\big(b^T A^{-1}(A^2 - T)^\mu \otimes (J_n)^\mu\big)K(t_n) + \mathcal{O}(h)\Big), \end{aligned} \tag{3.2.23}$$

$$\begin{aligned}\tau_{n+1} = \quad & (1+\mathcal{O}(h^2))\tau_n + \mathcal{O}(h^2)\varepsilon_n + \\ & h^{2\mu+q}\Big(\big(b^T A^{-2}(A^2-T)^\mu \otimes (J_n)^\mu\big)K(t_n) + \mathcal{O}(h)\Big), \\ & n = 0,1,2,\ldots \quad (\mu \text{ fijo, } \mu \in \{1,2,\ldots,\mu_{\text{máx}}\}).\end{aligned} \tag{3.2.24}$$

De (3.2.23) y (3.2.24), tomando las partes principales del error se deduce que

$$\tau_n = h^{2\mu+q-1}\Big(\big(b^T A^{-2}(A^2-T)^\mu \otimes I\big)M(t_n) + \mathcal{O}(h)\Big), \qquad n = 0,1\ldots, \tag{3.2.25}$$

donde

$$M(t_n) = h\sum_{j=0}^{n-1}\big(I \otimes (J_n)^\mu\big)K(t_n) = \mathcal{O}(1).$$

Por lo tanto, si asumimos que

$$0^T = b^T A^{-2}(A^2-T) = b^T(I - A^{-2}T), \tag{3.2.26}$$

se concluye de (3.2.23) y (3.2.24) que

$$\begin{aligned}\varepsilon_n = y_n - y_n^{(\mu)} &= \mathcal{O}(h^{2\mu+q-1}), \\ h^{-1}\tau_n = y'_n - y'^{\,(\mu)}_n &= \mathcal{O}(h^{2\mu+q-1})\end{aligned} \qquad n = 0,1,\ldots,N, \quad \mu \geq 1 \text{ fijo.} \tag{3.2.27}$$

De acuerdo con los desarrollos anteriores tenemos los siguientes resultados

Teorema 3.1 *Empleando las fórmulas de avance (3.2.6) y (3.2.7) junto con la iteración Single Newton (3.2.12), dando μ iteraciones por paso con un esquema iterativo que satisface la relación (3.2.26), y usando un predictor de orden q (ver (3.2.18)), se tiene que los errores globales tras N pasos de integración, satisfacen la relación (3.2.27) para $h = (t_f - t_0)/N \quad (h \to 0^+)$.*

□

En general para un algoritmo de arranque arbitrario de orden q, si la condición (3.2.26) no es satisfecha, entonces sólo puede garantizarse orden $2\mu + q - 2$ en la solución de avance.

Por otra parte, llevando a cabo un análisis similar al previo, acerca del orden de convergencia global después de μ iteraciones (por paso de integración) para la iteración Quasi–Newton (3.2.4), permite demostrar el siguiente teorema para problemas no lineales en general.

Teorema 3.2 *Bajo la hipótesis del Teorema 3.1, pero considerando la iteración Quasi–Newton, para los esquemas (3.2.4)–(3.2.7) tenemos que los errores globales satisfacen:*

$$\begin{array}{l} y_n - y_n^{(\mu)} = \mathcal{O}(h^{3\mu+q-2}), \\ y'_n - y'^{(\mu)}_n = \mathcal{O}(h^{3\mu+q-2}) \end{array} \qquad n = 0, 1, \ldots, N. \tag{3.2.28}$$

□

De (3.2.28) se deduce para $\mu = 1$, que ambos procesos iterativos (QNI y (3.2.14)) dan la misma precisión para la solución de avance, es decir, orden $q + 1$, pero nuestra iteración (3.2.14) es menos costosa computacionalmente. Sin embargo, en la práctica cuando integramos un problema con códigos basados en métodos Runge–Kutta–Nyström, normalmente se da más de una iteración en la mayoría de los pasos de integración. En la Tabla 3.2.1 se muestran los errores globales frente a los costos computacionales para ambos esquemas iterativos aplicados al método de Gauss de s etapas (s par).

De los resultados presentados en la Tabla 3.2.1 y tal como indican los Teoremas 3.1 y 3.2, no es difícil deducir que para el mismo esfuerzo computacional (y sin considerar las factorizaciones LU) la iteración (3.2.14) gana un orden más en la solución de avance que la iteración QNI.

Se debe notar además que la iteración (3.2.14) tiene la ventaja adicional que sólo requiere una descomposición LU en aritmética real, independientemente del número de etapas del método considerado, mientras que la iteración Quasi–Newton requiere s factorizaciones LU usando aritmética compleja. En la sección 3.6, ilustraremos numéricamente los Teoremas 3.1 y 3.2.

3.3. Selección del esquema iterativo para los métodos de Gauss

Estamos interesados en iteraciones del tipo (3.2.12) que en general pueden resolver satisfactoriamente sistemas no lineales en general. Por lo tanto, un requerimiento mínimo será que estas iteraciones deben también ser convergentes para problemas lineales de la forma,

$$y''(t) = Jy, \quad y(t_0) = y_0,\ y'(t_0) = y'_0, \quad t \in [t_0, t_f], \quad J \in \mathbb{R}^{m,m}, \tag{3.3.1}$$

independientemente del tamaño de los autovalores de J siempre que éstos sean negativos, lo que se traduce en convergencia independiente de la magnitud de la frecuencia. Así, la

iteración debe ser convergente sobre el problema test lineal

$$y''(t) = -\omega^2 y, \qquad \forall \omega \in \mathbb{R}. \tag{3.3.2}$$

A menudo en muchos problemas prácticos, las frecuencias $\{\omega\}$ involucradas en el sistema lineal original combinan pequeñas amplitudes para las frecuencias altas y amplitudes de menor magnitud cuando las frecuencias son pequeñas (frecuencias dominantes). Es importante para una integración efectiva que la solución de avance no quede dramáticamente afectada por los errores del esquema iterativo cuando las frecuencias altas posean amplitudes pequeñas. Si esto se consigue, entonces un número moderado de iteraciones μ con el esquema iterativo Single–Newton (3.2.12) dará lugar a una solución de avance bastante aproximada y estable $(y_n^{(\mu)}, y_n'^{(\mu)})$.

A efectos de analizar con más detalles la estabilidad y convergencia del esquema iterativo, haciendo $z = h\omega$ y aplicando la iteración (3.2.12) con $J_n = -\omega^2$, al problema test (3.3.2), obtenemos el error en las iteraciones

$$Y_n - Y_n^{(\nu)} = N(z)(Y_n - Y_n^{(\nu-1)}) = N(z)^\nu (Y_n - Y_n^{(0)}), \quad \nu = 1, 2 \ldots, \tag{3.3.3}$$

donde

$$N(z) := z^2(I + z^2 T)^{-1}(T - A^2). \tag{3.3.4}$$

Está claro que la convergencia de la iteración (3.2.12) para todo z es equivalente a asumir que el radio espectral de $N(z)$ satisfaga

$$\rho\big(N(z)\big) < 1, \quad \forall z \in \mathbb{R}.$$

En la construcción de nuestros esquemas iterativos impondremos también la condición (3.2.26), por las razones dadas anteriormente en el Teorema 3.1. Esto implica que un autovalor de $N(z)$ se anula, pues de (3.2.26) se deduce que $(A - T)$ posee un autovalor nulo. También debemos tener en cuenta que la matriz T debe poseer un espectro unipuntual, $\sigma(T) = \{\gamma\}, \quad \gamma > 0$.

3.3.1. El método de Gauss de dos etapas (s = 2)

Por la condición anterior, tenemos que para el método de Gauss de 2 etapas un autovalor de $N(z)$ es siempre cero y el otro autovalor está dado por $\phi(z) = 1 - \det(I - N(z))$. Lo que nos lleva a que

$$\phi(z) = 1 - \big(\det(I + z^2 T)\big)^{-1} \det(I + z^2 A^2), \tag{3.3.5}$$

y de aquí, se sigue que

$$\phi(z) = \frac{z^2(2\gamma - \mathrm{tr}A^2) + z^4\big(\gamma^2 - (\det A)^2\big)}{(1+\gamma z^2)^2},$$

donde $\mathrm{tr}A^2$ denota la traza de la matriz A^2.

Para aumentar el orden de convergencia en el origen ($z \to 0$), debemos asumir que $\mathrm{tr}A^2 - 2\gamma = 0$, pero en este caso tenemos que $\phi(\infty) = -3$ y la iteración no convergería para las frecuencias altas. Debemos encontrar una elección óptima de γ de modo que el esquema iterativo tenga una buena razón de convergencia en toda la recta real $\omega h = z \in \mathbb{R}$, (y especialmente en el caso $z = \infty$), conjugado con el hecho de proporcionar un valor pequeño para $|\delta|$; $\delta := \mathrm{tr}A^2 - 2\gamma$.

Una opción podría ser considerar $\phi(\infty) = 0$, entonces tendríamos que $\gamma = \det(A) = 1/12$, $\delta = -1/12$ con lo que $\phi_1 := \mathrm{máx}_{Re(z)\leq 0}\,|\phi(z)| = 0{,}25$.

Otra opción podría ser elegir γ tal que

$$g(\gamma) := \max_{z\in\mathbb{R}} |\phi(z)| \quad \text{es mínimo.}$$

Este mínimo se alcanza para $\gamma^* = (12x^*)^{-1}$, donde x^* es la única raíz positiva de

$$4x^4 - 4x^3 - x^2 + 8x - 8 = 0.$$

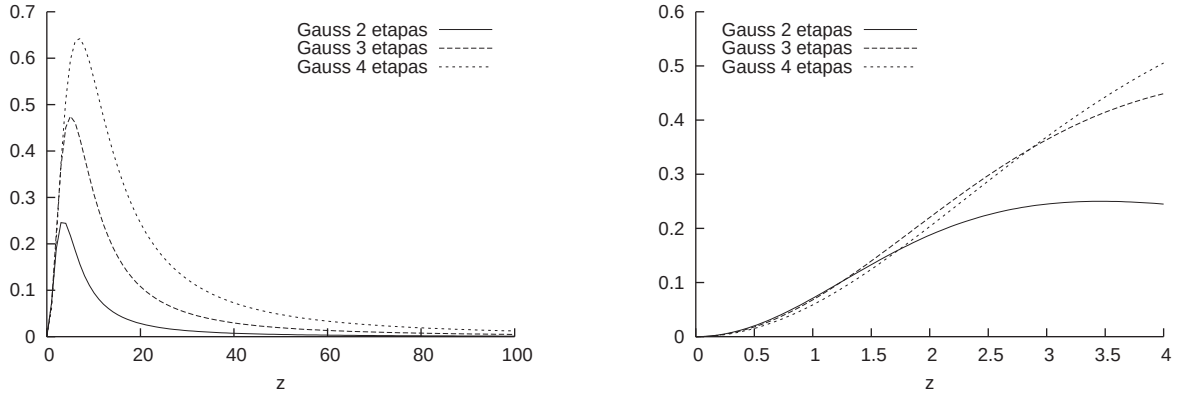
Así, obtenemos que

$$\gamma^* = 0{,}07644181307872985 03\ldots. \tag{3.3.6}$$

Para esta elección $\phi_2 := g(\gamma^*) = 0{,}18...$, el argumento de que el Máximo del radio espectral sea mínimo es favorable para esta segunda opción, pero en este caso tenemos que $\phi(\infty) = -\phi_2$, y por tanto en la mayor parte de la recta real el radio espectral para la segunda opción es mayor que para la primera opción, también la diferencia $\phi_1 - \phi_2$ es relativamente pequeña, además para un número fijo de iteraciones los errores globales asociados a las altas frecuencias, del método implementado con el esquema iterativo Single-Newton, se amortiguan bastante mejor cuando se considera la primera opción. Este hecho juega un papel importante en la elección de predictores de altos ordenes, los cuales funcionan mejor con la primera opción, tal como veremos en la próxima sección.

Por los argumentos anteriores, y por razones prácticas tras la comparación de ambas opciones sobre muchos problemas test de la literatura, nos inclinamos por la primera opción, ya que normalmente ésta funciona un poco mejor que la segunda en la mayoría de los

Figura 3.3.1: Radio espectral de $N(z)$ para los métodos de Gauss de s etapas ($s = 2, 3, 4$).



problemas, y tiene la ventaja adicional de ser más robusta. En conclusión, tomamos $\gamma = \frac{1}{12}$, y esto nos da una única matriz T cuya matriz inversa T^{-1} satisface

$$b^T(A^{-2} - T^{-1}) = 0^T, \quad tr(T^{-1}) = 2\gamma^{-1} = 24, \quad \det(T^{-1}) = \det(A^{-2}) = 12^2.$$

La matriz T puede ser descompuesta de acuerdo a (3.2.13), pero debe tenerse en cuenta que tal descomposición no es única. Imponiendo que la matriz S sea triangular superior con "1" en su diagonal por motivos de ahorro computacional. Obtenemos que L y S son respectivamente la parte triangular inferior estricta y la parte triangular superior de la siguiente matriz denotada por $L \boxplus S$,

$$L \boxplus S = \begin{pmatrix} 1 & -7 + 4\sqrt{3} \\ (12 + 7\sqrt{3})/6 & 1 \end{pmatrix}, \quad \gamma = \frac{1}{12}. \tag{3.3.7}$$

3.3.2. El método de Gauss de tres etapas (s = 3)

En este caso, la matriz T suministra nueve parámetros. La condición (3.2.26) impone tres ecuaciones lineales y a la vez implica un autovalor nulo para $N(z)$. Por otra parte el requisito $\sigma(T) = \{\gamma\}$ nos lleva a tres ecuaciones (una lineal y dos no lineales) y suministra un nuevo parámetro γ. A efectos de simplificar el problema y obtener una velocidad de convergencia todavía razonable, vamos a exigir que la matriz $N(z)$ tenga un sólo autovalor no nulo $\phi(z)$. En este caso $\phi(z)$ está dado por (3.3.5).

Como en el caso de dos etapas, asumiendo que $\phi(\infty) = 0$, obtenemos que

$$\gamma = (\det A)^{2/3} = \left(\frac{1}{120}\right)^{2/3} = 0{,}041103534\ldots.$$

Además, la condición (3.2.26) es equivalente a

$$0^T = b^T N(\infty),$$

o bien usando (3.3.4)

$$0^T = b^T(A^{-2} - T^{-1}).$$

Además, si imponemos la nueva condición

$$0^T = e_3^T N(\infty),$$

con $e_3^T = (0, 0, 1)$, se deduce inmediatamente que

$$0^T = e_3^T(A^{-2} - T^{-1}). \tag{3.3.8}$$

De aquí deducimos que $N(z)$ tiene un autovalor nulo de multiplicidad dos, en virtud de que $A^{-2} - T^{-1}$ tiene dos autovectores por la izquierda correspondientes al autovalor nulo, y el otro autovalor está dado por $\phi(z)$ en (3.3.5). Debemos recordar que este era uno de los objetivos trazados. Además esta elección para T^{-1} tiene la ventaja de que su última fila coincide con una fila de A^{-2}, y este hecho implica una propagación pequeña de los errores en las altas frecuencias para las iteraciones en la tercera componente $Y_{n,3}^{(\nu)}(\nu = 1, 2, \ldots)$, $(z \to \infty)$, pues $N(\infty) = (A^{-2} - T^{-1})A^2$. También debe observarse que $Y_{n,3}$ es la etapa más lejana para ser aproximada por predictores basados en los pasos previos de integración, así la elección de (3.3.8) es conveniente para amortiguar los errores cometidos por los predictores al aproximar la etapa más lejana. Con estos requisitos (nueve ecuaciones y nueve incógnitas) la matriz T está unívocamente determinada. Nuevamente omitimos la escritura de T y damos la descomposición $L \boxplus S$ con 16 dígitos significativos.

$$L \boxplus S = \begin{pmatrix} 1 & -0{,}34134805819933375 & 0{,}08060287745941966 \\ 3{,}0972763877611617 & 1 & 0{,}09100037186032114 \\ -6{,}3351370812325647397 & 4{,}3129085842580614 & 1 \end{pmatrix}. \tag{3.3.9}$$

3.3.3. El método de Gauss de cuatro etapas (s = 4)

En este caso y por argumentos similares a los casos de $s = 2$ y 3 etapas, exigiremos $\sigma(T) = \{\gamma\}$, la condición (3.2.26), $e_3^T N(\infty) = 0^T$ y $e_4^T N(\infty) = 0^T$. Esto nos lleva a 16 ecuaciones con 16 incógnitas suministradas por la matriz T^{-1}. Estas ecuaciones son:

$$\begin{aligned} &b^T(A^{-2} - T^{-1}) = 0^T, \qquad \sigma(T^{-1}) = \{\gamma^{-1}\}, \\ &e_3^T(A^{-2} - T^{-1}) = 0^T, \quad e_4^T(A^{-2} - T^{-1}) = 0^T. \end{aligned} \tag{3.3.10}$$

En (3.3.10), e_3 y e_4 denotan el tercer y cuarto vector de la base canónica de $\mathbb{R}^4$ respectivamente, y el valor para γ se obtiene exigiendo que el autovalor no nulo de $N(z)$ satisfaga que $\phi(\infty) = 0$. Esto implica que

$$\gamma = (\det A)^{1/2} = \frac{\sqrt{105}}{420} = 0{,}0243975018\ldots.$$

De las condiciones en (3.3.10) obtenemos una única matriz T de coeficientes reales. Esta matriz como en los casos anteriores también puede ser descompuesta en la forma $L \boxplus S$ con

$$L = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 2{,}8903693478178977 & 0 & 0 & 0 \\ -3{,}880435892734769 & 3{,}1457588737925907 & 0 & 0 \\ 10{,}056003837018828 & -10{,}386602189612067 & 5{,}349223474607041 & 0 \end{pmatrix}, \tag{3.3.11}$$

$$S = \begin{pmatrix} 1 & -0{,}6259618003648055 & 0{,}28004508015579187 & -0{,}026021682011884171 \\ 0 & 1 & 0{,}04262904477976469 & 0{,}04746745319522649 \\ 0 & 0 & 1 & 0{,}16883422865076189 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{3.3.12}$$

En la Figura 3.3.1 se muestra la gráfica del valor absoluto del autovalor no nulo $\phi(z)$ de la matriz $N(z)$ en (3.3.5) para cada método de Gauss de s etapas con $s = 2, 3, 4$. Se puede apreciar que la iteración es siempre convergente y que la velocidad de convergencia decrece cuando el número de etapas s aumenta. También es interesante considerar la razón de convergencia en media, para una iteración dada. Este promedio está definido en [65, p. 45] mediante

$$\rho_j(z) := \sqrt[j]{\|N(z)^j\|_\infty}, \quad j = 1, 2, \ldots.$$

Tabla 3.3.1: Promedio de las razones de convergencia para el método de Gauss.

j	Gauss de dos etapas	Gauss de tres etapas	Gauss de cuatro etapas
1	$\rho_j^* = 0{,}577$; $\rho_j' = 0{,}577$	$\rho_j^* = 1{,}72$; $\rho_j' = 1{,}72$	$\rho_j^* = 3{,}72$; $\rho_j' = 3{,}72$
2	$\rho_j^* = 0{,}280$; $\rho_j' = 0$	$\rho_j^* = 0{,}830$; $\rho_j' = 0$	$\rho_j^* = 1{,}17$; $\rho_j' = 0$
3	$\rho_j^* = 0{,}261$; $\rho_j' = 0$	$\rho_j^* = 0{,}675$; $\rho_j' = 0$	$\rho_j^* = 0{,}881$; $\rho_j' = 0$
4	$\rho_j^* = 0{,}256$; $\rho_j' = 0$	$\rho_j^* = 0{,}613$; $\rho_j' = 0$	$\rho_j^* = 0{,}803$; $\rho_j' = 0$

Para nuestra iteración (3.2.14), tenemos que

$$\lim_{j\to\infty} \rho_j(z) = \rho(N(z)) = |\phi(z)|.$$

En [65] se señala la importancia que tiene que $\rho_j(z) \simeq \rho(N(z))$ para pequeños valores de j, ya que esto implicaría una razón de convergencia uniforme próxima al radio espectral de $N(z)$, desde la primera iteración. En la Tabla 3.3.1 recogemos los números $\rho_j^* = \text{máx}_{z\in\mathbb{R}}\, \rho_j(z)$ y $\rho_j^{'} \equiv \rho_j(\infty)$ para los casos $j = 1, 2, 3, 4$, y $s = 2, 3, 4$.

3.4. Predictores para los esquemas iterativos aplicados a los métodos de Gauss

Un predictor $Y_{n,i}^{(0)} (i = 1, 2, \ldots, s)$ para computar las etapas $Y_{n,i}$ se dirá que tiene orden q si su error es de orden h^q, es decir, si verifica:

$$Y_{n,i} - Y_{n,i}^{(0)} = \mathcal{O}(h^q), \quad i = 1, 2, \ldots, s \qquad h \to 0.$$

Los predictores a considerar resultan de combinaciones lineales de las soluciones de avance y de las etapas del último paso dado, es decir, se computarán mediante,

$$Y_{n,i}^{(0)} = a_i y_{n-1} + h d_i y'_{n-1} + \sum_{j=1}^{s} b_{ij} Y_{n-1,j}, \quad i = 1, 2, \ldots, s. \tag{3.4.1}$$

Predictores de este tipo han sido considerados previamente en la literatura, entre otros por P. Laburta [67] a efectos de optimizar el costo computacional en esquemas iterativos de tipo Iteración Funcional Simple en sistemas de tipo Hamiltoniano, principalmente. También han sido usados por I. Gómez [38] y I. Higueras et al. [39] en el caso de implementar esquemas iterativos de tipo Newton (Quasi-Newton o Newton Modificado) sobre Métodos de colocación

de orden relativamente bajo y también para métodos del tipo SDIRKN (Single Diagonally Implicit Runge Kutta Nyström methods).

Su orden se deduce usando la *teoría especial de árboles de Nyström* (véase por ejemplo [55, Ch. II.14]). En este caso, las condiciones de orden vienen dadas por las siguientes ecuaciones lineales, sobre los parámetros a determinar (a_i, d_i, b_{ij}):

1. Orden 1: $\quad a_i + \sum_{j=1}^{s} b_{ij} = 1, \quad i = 1, \ldots, s.$

2. Orden 2: $\quad$ (Orden 1 y $d_i + \sum_{j=1}^{s} b_{ij} c_j = 1 + \tau c_i, \quad i = 1, \ldots, s$).

   Teniendo en cuenta que los métodos de Gauss de s–etapas satisfacen la condición simplificadora de orden $C(s)$ (abajo, la potencia de un vector se entenderá como tomar la potencia en cada componente),

   $C(s): \qquad Ac^{j-1} = \frac{1}{j} c^j, \qquad j = 1, \ldots, s.$

   Entonces, tras unos cálculos no complicados pero algo tediosos, se llega a que las condiciones de orden $(q \geq 3)$ vienen dadas por:

3. Orden 3 (para $s \geq 2$): $\quad$ (Orden 2 y $\sum_{j=1}^{s} b_{ij} (c_j)^2 = (1 + \tau c_i)^2, \quad i = 1, \ldots, s$).

4. Orden 4 (para $s \geq 2$): $\quad$ (Orden 3 y

   $$\sum_{j=1}^{s} b_{ij} \kappa_j = \frac{1}{6} \left((1 + \tau c_i)^3 - (\tau c_i)^3 \right) + \kappa_i \tau^3, \quad i = 1, \ldots, s. \quad \text{donde } \kappa = (\kappa_i) = A^2 c).$$

   Para $s \geq 3$, la última condición es equivalente a

   $$\sum_{j=1}^{s} b_{ij} (c_j)^3 = (1 + \tau c_i)^3, \quad i = 1, \ldots, s.$$

5. Orden 5 (para $s \geq 3$): $\quad$ (Orden 4 y

   $$\sum_{j=1}^{s} b_{ij} \zeta_j = \frac{1}{12} \left((1 + \tau c_i)^4 - (\tau c_i)^4 \right) + \zeta_i \tau^4, \quad i = 1, \ldots, s, \quad \text{donde } \zeta = (\zeta_i) = A^2 c^2).$$

Es de gran interés analizar como propaga cada predictor los errores globales acumulados, especialmente cuando se consideran frecuencias altas, debido a que en estos casos los errores suelen amplificarse de forma más catastrófica y necesitan ser bien amortiguados por el esquema iterativo. Así, para problemas lineales

$$y''(t) = f(t, y) \equiv Jy + g(t). \tag{3.4.2}$$

haciendo

$$\varepsilon_{n-1} := y_{n-1} - \hat{y}_{n-1}, \quad \eta_{n-1} := hy'_{n-1} - h\hat{y}'_{n-1},$$

denotando por $y(t)$ y $\hat{y}(t)$ las soluciones locales que pasan por $(t_{n-1}, y_{n-1}, y'_{n-1})$ y $(t_{n-1}, \hat{y}_{n-1}, \hat{y}'_{n-1})$ respectivamente, entonces la diferencia $\xi(t) = y(t) - \hat{y}(t)$ satisface el sistema lineal

$$\xi''(t) = J\xi(t), \quad \xi(t_{n-1}) = \varepsilon_{n-1}, \quad \xi'(t_{n-1}) = h^{-1}\eta_{n-1}.$$

De aquí, resulta evidente que el comportamiento de un predictor sobre un problema test simple

$$y''(t) = -\omega^2 y, \quad y(t_{n-1}) = \varepsilon_{n-1}, \quad y'(t_{n-1}) = h^{-1}\eta_{n-1}, \tag{3.4.3}$$

es significativo para conocer como los errores acumulados $(\varepsilon_{n-1}, \eta_{n-1})$ en el paso t_{n-1} se propagan al paso siguiente, especialmente en el caso de considerar frecuencias altas.

Aplicando el método de Gauss de s etapas al problema test (3.4.3) y haciendo $z = \omega h$ tenemos que

$$Y_{n-1} = (I - zA^2)^{-1}(e \otimes y_{n-1} + c \otimes \varepsilon_{n-1}).$$

Por otra parte de (3.2.6)

$$y_n = r^* y_{n-1} + (b^T A^{-1} \otimes I)Y_{n-1},$$

$$v_n = r' y_{n-1} + r^* v_{n-1} + (b^T A^{-2} \otimes I)Y_{n-1}.$$

Con lo cual, haciendo $z \to \infty$ (frecuencias altas) se tiene que

$$Y_{n-1} = Y_n = 0 \in \mathbb{R}^s, \quad y_n = r^* \varepsilon_{n-1} \quad \text{y} \quad hy'_n = r' \varepsilon_{n-1} + r^* \eta_{n-1}. \tag{3.4.4}$$

Por otra parte, para el predictor (3.4.1), haciendo $z = \infty$, se sigue que

$$Y^{(0)}_{n,i} = a_i \varepsilon_{n-1} + d_i \eta_{n-1}, \qquad i = 1, \ldots, s. \tag{3.4.5}$$

Ya que el error está dado por $Y^{(0)}_n - Y_n = Y^{(0)}_n$, entonces para cada orden $q = 1, \ldots,$ los factores de amplificación de error de un predictor vendrán dados por vectores $a = (a_i)$ y

$d = (d_i)$. Sería muy conveniente elegir predictores con factores de amplificación de error pequeños. Pues estos errores deben ser amortiguados luego por las iteraciones del esquema iterativo.

Los predictores de ordenes $q = 1, 2, \ldots, s$ con factor de amplificación 0 en $z = \infty$, son dados respectivamente por (la interpolación polinómica en las etapas de los pasos previos),

$$Y_{n,i}^{(0),q} = P_{s,q}(1 + \tau c_i), \quad i = 1, \ldots, s, \quad q = 1, \ldots, s, \tag{3.4.6}$$

donde $P_{s,q}(t)$ es un polinomio de grado $q - 1$ a lo sumo, el cual satisface

$$P_{s,q}(c_{s+1-i}) = Y_{n-1,s+1-i}, \quad i = 1, \ldots, q. \tag{3.4.7}$$

El único predictor de orden $s + 1$ que satisface $d^T = (d_i) = 0^T$, está dado por

$$Y_{n,i}^{(0),s+1} = P_{s,s+1}(1 + \tau c_i), \quad P_{s,s+1}(0) = y_{n-1}, \quad P_{s,s+1}(c_i) = Y_{n-1,i}, \quad (i = 1, \ldots, s). \tag{3.4.8}$$

Aquí, $P_{s,s+1}(t)$ es un polinomio de grado s a lo sumo.

Para el caso $s = 2$ y $s = 3$, los predictores $\left(Y_{n,i}^{(0),s+2}\right)$ de orden $s + 2$ pueden calcularse fácilmente de las ecuaciones lineales dadas al comienzo de esta sección. Además, predictores de ordenes más altos pueden calcularse usando la *teoría de árboles especiales de Nyström* junto con la condición $C(s)$.

Para ilustrar el tamaño de los factores de amplificación de error asociados con predictores de altos ordenes, hemos calculado los factores de amplificación para el método de Gauss de dos etapas y su predictor natural de cuarto orden, ya mencionado anteriormente. Los valores para $a_2(\tau)$ y $d_2(\tau)$ (véase (3.4.5)) para distintas razones τ de tamaño de paso $\tau = 1, 2, 3$, están dadas a continuación. Estos valores de τ pueden ser representativos para los casos en los cuales el tamaño de paso es constante o aumenta en la integración.

$$\begin{aligned} a_2(1) &= 31{,}86, \\ d_2(1) &= 3{,}732, \\ a_2(2) &= 131{,}4, \\ d_2(2) &= 17{,}66, \\ a_2(3) &= 338{,}3, \\ d_2(3) &= 48{,}25. \end{aligned}$$

De lo expuesto anteriormente, podemos deducir, que las amplificaciones de error asociadas con las altas frecuencias en el esquema iterativo no serán amortiguadas en pocas iteraciones a menos que el radio espectral de la matriz $N(\infty)$ (véase sección 3.3) sea bastante pequeño.

Tabla 3.4.1: solución numérica en la componente y, obtenida con el método Gauss de dos etapas en el punto final $t = 4$, después de $N = 40$ pasos de tamaño $h = 0{,}1$, usando nuestra iteración y_{sni} y también usando la Iteración Quasi–Newton y_{qni}, para resolver sus ecuaciones de etapa, tomando μ iteraciones por paso y el predictor $Y_{n,i}^{(0),q}$.

Iter	$Y_{n,i}^{(0),1}$	$Y_{n,i}^{(0),2}$	$Y_{n,i}^{(0),3}$	$Y_{n,i}^{(0),4}$
$\mu = 1$	$y_{sni} = -2{,}27e\text{–}9$	$y_{sni} = 5{,}12e{+}2$	$y_{sni} = -4{,}61e{+}22$	$y_{sni} = -1{,}16e{+}33$
	$y_{qni} = 1{,}09e\text{–}8$	$y_{qni} = 1{,}10e\text{–}8$	$y_{qni} = 1{,}76e\text{–}3$	$y_{qni} = 1{,}88e{+}7$
$\mu = 2$	$y_{sni} = 8{,}11e\text{–}9$	$y_{sni} = 8{,}33e\text{–}9$	$y_{sni} = -2{,}03e\text{–}13$	$y_{sni} = 1{,}09e{+}0$
	$y_{qni} = 1{,}00e\text{–}8$	$y_{qni} = 1{,}00e\text{–}8$	$y_{qni} = 1{,}94e\text{–}8$	$y_{qni} = 2{,}91e\text{–}5$
$\mu = 3$	$y_{sni} = 1{,}10e\text{–}8$	$y_{sni} = 1{,}00e\text{–}8$	$y_{sni} = 5{,}02e\text{–}8$	$y_{sni} = 6{,}52e\text{–}3$
	$y_{qni} = 9{,}96e\text{–}9$	$y_{qni} = 9{,}96e\text{–}9$	$y_{qni} = 1{,}03e\text{–}8$	$y_{qni} = 2{,}30e\text{–}8$

Para ilustrar lo expuesto anteriormente sobre la influencia de los predictores en el comportamiento del error global de un método cuando se usa un determinado método iterativo para resolver sus ecuaciones de etapa y se da un número fijo de iteraciones por paso de integración, consideremos el problema simple,

$$\begin{cases} y''(t) = -\eta(1+t)^{-1}y, \\ y(0) = 10^{-8},\ y'(0) = 0, \\ \eta = 10^{10},\ t \in [0,4]. \end{cases}$$

En la Tabla 3.4.1 se refleja como el predictor elegido, puede tener una influencia catastrófica en la precisión alcanzada en el punto final de integración, cuando se da un número pequeño de iteraciones por paso de integración con un esquema prefijado. Aquí, las integraciones se llevaron a cabo usando el método de Gauss de dos etapas y las iteraciones empleadas fueron la QNI y la iteración Single-Newton (SNI) dada por (3.2.14) y (3.3.7). En todos los casos se tomó para el primer paso: el predictor $Y_n^0 = e \otimes y_0$ y se dieron 5 iteraciones con el esquema iterativo correspondiente. Debe tenerse en cuenta que la solución exacta en la componente y, para todos los puntos del intervalo de integración es de magnitud $\mathcal{O}(10^{-8})$.

3.5. VOS: Estrategia del Orden Variable para seleccionar los predictores

De los estudios previos no podemos inferir que los algoritmos de arranque de orden más alto sean los mejores. Así, para conjugar en los predictores un orden alto con pequeños factores de amplificación de error para las altas frecuencias, podemos adoptar una estrategia de orden variable para seleccionar el mejor predictor en cada paso de la integración. Denotaremos esta estrategia por VOS: *Variable Order Strategy* para predictores. Esta técnica ha resultado ser muy efectiva en sistemas diferenciales de primer orden de tipo stiff integrados mediante métodos Runge–Kutta altamente implícitos [44, pp. 89–93]. La idea es elegir el predictor que tenga el error más pequeño (en alguna norma). Teniendo en cuenta que los algoritmos de arranque desarrollados previamente $Y_{n,i}^{(0),q}$, tienen ordenes consecutivos $q = 1, 2, 3, \ldots, q_{\text{máx}}$, el error de $Y_{n,i}^{(0),q}$ puede estimarse mediante la fórmula

$$E_{n,i}^{(0),q} = \| \, Y_{n,i}^{(0),q} - Y_{n,i}^{(0),q+1} \, \|, \quad (i = 1, \ldots, s), \quad (q = 1, \ldots, q_{\text{máx}} - 1).$$

La selección puede hacerse (excepto para el primer paso) de la siguiente manera:
Para $q = 1, 2, \ldots, q_{\text{máx}} - 1$, se toma el primer q (y el predictor $Y_{n,i}^{(0),q}, i = 1, \ldots, s$) tal que

$$E_{n,s}^{(0),q+1} \geq \kappa E_{n,s}^{(0),q}, \quad \text{tipicamente } \kappa = 0{,}5. \tag{3.5.1}$$

Si (3.5.1) no es satisfecha, entonces se elige $q = q_{\text{máx}}$ (el algoritmo de arranque de más alto orden $Y_{n,i}^{(0),q_{\text{máx}}}$) si

$$E_{n,s}^{(0),q+1} \leq \mu\kappa E_{n,s}^{(0),q}, \quad \text{tipicamente } \mu = 0{,}2,$$

y $q = q_{\text{máx}} - 1$ en otro caso.

Se puede observar que sólo medimos el error del predictor en la componente s-ésima, pues es la que se supone que presentará un error más grande, al estar más alejada. Aunque no hemos hecho una estimación del error para el predictor de más alto orden, seleccionamos éste, cuando su diferencia con el predictor de un orden menor no sea significativa. Conviene resaltar que para dos predictores con errores similares preferiremos aquel que tenga orden más bajo, véase (3.5.1), ya que éste tendrá factores de amplificación de error más pequeños generalmente.

Para el primer paso, podemos adoptar la misma estrategia anterior, pero considerando

como algoritmos de arranque:

$$\begin{aligned} Y_{0,i}^{(0),1} &= y_0 \quad \text{(orden 1)}, \\ Y_{0,i}^{(0),2} &= y_0 + h_0 c_i y_0' \quad \text{(orden 2)}, \\ Y_{0,i}^{(0),3} &= y_0 + h_0 c_i y_0' + \tfrac{1}{2}(h_0 c_i)^2 f(t_0, y_0) \quad \text{(orden 3)} \quad \text{para } i = 1, \ldots, s. \end{aligned} \tag{3.5.2}$$

En el predictor de orden 3, hemos incluido una evaluación adicional de la derivada $f(t_0, y_0)$. De todos modos su computación se hace necesaria para estimar el error local del primer paso, y por tanto no supone coste adicional alguno.

3.6. Experimentos numéricos

En esta sección presentamos algunos experimentos numéricos con el propósito de confirmar la teoría del orden (Teorema 3.1 y Teorema 3.2, sección 3.2) para la solución de avance (3.2.6) implementando la iteración Single Newton y la Iteración Quasi–Newton (QNI), usando los predictores desarrollados en la sección precedente y dando un número fijo μ de iteraciones por paso de integración. Además mostraremos que el esquema iterativo Single Newton es más eficiente en general que el de tipo Quasi–Newton y que otras iteraciones propuestas en la literatura. Para el primer objetivo, hemos implementado las iteraciones Single Newton y Quasi–Newton, en códigos a paso fijo basados en las fórmulas de Gauss de 2, 3 y 4 etapas, evaluando la matriz Jacobiana (y actualizando las correspondientes factorizaciones LU) en cada paso de integración (excepto en el primer paso, donde hemos realizado $\mu + 2$ iteraciones para compensar el bajo orden de los predictores en el primer paso). Para nuestra iteración Single Newton hemos desarrollado tres códigos (uno para cada número de etapas $s = 2, 3, 4$), cuyas ecuaciones de etapa han sido resueltas usando la iteración propuesta en (3.2.14), con los valores de los parámetros γ y las matrices S y L dadas en la sección 3.3. También construimos otros tres códigos (para $s = 2, 3, 4$) basados en la Iteración Quasi–Newton (3.2.4).

En todos los problemas los errores absolutos fueron medidos usando la norma Euclídea ponderada,

$$\| x \| := m^{-1/2} \, \| x \|_2, \quad x \in \mathbb{R}^m.$$

Para cada iteración y cada predictor considerado ($Y_{n,i}^{(0),q}$, véase (3.4.6), (3.4.7) y (3.4.8)) hemos calculado los ordenes de convergencia del método con respecto a la solución Runge-Kutta en el punto final. Estos órdenes han sido denotados por $p(h)$ y $p'(h)$ en las Tablas

3.6.1 y 3.6.2, donde además, entre corchetes se han incluido los errores globales con respecto a la solución Runge–Kutta. Estos órdenes $p(h)$ y $p'(h)$ se han estimado mediante la técnica de extrapolación global del modo siguiente,

$$p(h) = \frac{\ln\big(e(h)\big) - \ln(e(h/2))}{\ln 2}, \quad p'(h) = \frac{\ln\big(e'(h)\big) - \ln(e'(h/2))}{\ln 2},$$

donde

$$e(h) = \| \, y_{RK}(t_f, h) - y_N^{(\mu)} \, \|, \quad e'(h) = \| \, y'_{RK}(t_f, h) - y'^{(\mu)}_N \, \|, \tag{3.6.1}$$

con $y_{RK}(t_f, h)$ y $y'_{RK}(t_f, h)$ denotando la solución exacta Runge–Kutta en el punto final después de N pasos de tamaño $h = (t_f - t_0)/N$ y $\{y_n^{(\mu)},\ y'^{(\mu)}_n\}$ $(n = 0, 1, \ldots, N)$ denota la solución de avance después de n pasos de integración y μ iteraciones por paso de integración. Para el primer paso, el predictor usado en todos los casos fue

$$Y_0^{(0)} = e \otimes y_0 \quad \text{(orden 1)},$$

y se dieron dos iteraciones más que en los otros pasos a efectos de compensar el bajo orden del predictor usado en el primer paso.

Presentaremos aquí los resultados numéricos sobre tres problemas test que aparecen frecuentemente en la literatura y que han sido usados como prototipos de problemas oscilatorios para probar la eficiencia de los métodos numéricos.

Problema 3.1 *([37, pp. 201], Apéndice Prob. B.14)*

$$y''(t) + \sinh(y(t)) = 0$$

$$y(0) = 1, \quad y'(0) = 0, \quad t \in [0, 4].$$

Problema 3.2 *([56, pp. 10–12], Apéndice Prob. B.24) es el problema de los planetas externos del sistema solar (outer solar system) de dimensión $m = 18$ y $t \in [0, 500000]$. En este caso la componente de y más pequeña en el punto final es $y_4 = -5{,}56\ldots$ y la más grande es $y_{14} = 38{,}6\ldots$. Este problema presenta sólo combina bajas frecuencias en las componentes de su solución. Por ejemplo, los autovalores (ω) de la matriz Jacobiana $J_0 = \partial f/\partial y(t_0, y_0)$ en el punto inicial son muy pequeños. Un cálculo numérico muestra que todos ellos son reales y satisfacen $|\omega| \leq 10^{-5}$.*

Problema 3.3 *([63, Prob. 4.5, p. 610], Apéndice Prob. B.32) es la ecuación de la onda unidimensional con fricción no lineal dada por,*

$$\begin{cases} \dfrac{\partial^2 u}{\partial t^2} = gd(x)\dfrac{\partial^2 u}{\partial x^2} + \dfrac{g^2 u^3}{C^4 d^2}, \quad 0 \le x \le l, \quad t \in [0,10], \\ u_x(0,t) = u_x(l,t) = 0, \\ u(0,t) = \sin\left(\dfrac{\pi x}{l}\right), \quad u_t(0,t) = -\dfrac{\pi}{l}\sqrt{gd}\cos\left(\dfrac{\pi x}{l}\right). \end{cases}$$

donde $d = d_0(2 + \cos(2l^{-1}\pi x))$, g es la aceleración de la gravedad y C es el coeficiente de Chezy dado más abajo. Se han usado diferencias simétricas de cuarto orden para discretizar $u_{xx}(t,x)$ sobre la variable x en $x_j = j\Delta x, \quad j = 2,3,\ldots,m-1 \quad (\Delta x = l/(m+1))$, y también diferencias de cuarto orden para discretizar sobre la línea x_1 (involucrando los valores de u desde x_0 hasta x_5) y sobre la línea x_m. Las condiciones de fronteras $u_x(t,0)$ y $u_x(t,l)$, son respectivamente discretizadas usando diferencias progresivas y regresivas de quinto orden, con el propósito que los errores de discretización se mantengan dentro del mismo orden cuando se pase de la ecuación en derivadas parciales al sistema diferencial ordinario de segundo orden mediante el método de líneas (MOL). Las componentes de la solución u_0 y u_{m+1} son eliminadas mediante las condiciones de frontera. Los parámetros que hemos tomado son

$$m = 41, \qquad l = 100, \qquad d_0 = 10, \qquad C = 50, \qquad g = 9{,}81.$$

Este problema es medio–stiff, ya que los 41 autovalores de $J_0 = \partial f/\partial y(t_0, y_0)$ son reales y están casi uniformemente distribuidos en el intervalo $[-276{,}3,\ldots,0]$. No hay muchos cambios cuando la matriz Jacobiana es evaluada en otros puntos $(t, y(t))$ con $t > 0$. En $t = 10$, la componente de la solución más pequeña es $y_{39} = 0{,}0550\ldots$ y la más grande es $y_1 = 1{,}956\ldots$.

Para el problema 3.1, en la Tabla 3.6.1 se puede ver que los ordenes de convergencia obtenidos en ambas componente (y e y') se ajustan a lo establecido en el Teorema 3.1, independientemente del predictor elegido. También los resultados en la Tabla 3.6.2 confirman numéricamente los resultados del Teorema 3.2 para la iteración Quasi–Newton.

Considerando problemas más prácticos, hemos aplicado los esquemas iterativos aplicados a los Problemas 3.2 y 3.3, usando los métodos de Gauss de s etapas ($s = 2,3,4$), dando un número fijo de iteraciones por paso de integración, excepto en el primer paso donde hemos tomando como predictor $Y_0 = e \otimes y_0$ y hemos dado $\mu + 2$ iteraciones para compensar el bajo orden de este predictor. Para los restantes pasos de integración, hemos usado la estrategia de orden variable (VOS) para los predictores, tanto en el caso del esquema Quasi–Newton

como para nuestra iteración Single–Newton. En el caso del método de Gauss de dos etapas hemos incluido los resultados obtenidos usando la iteración de Cooper y Butcher, tal como ésta fue modificada en [37, pp. 198–199] y con los predictores que allí se proponen. En este caso, también hemos dado dos iteraciones de más en el primer paso de integración a efectos de estar en las mismas condiciones que con los otros esquemas iterativos.

En las Tablas 3.6.3 y 3.6.5 se aprecia que para el método de Gauss de dos etapas, nuestra iteración Single-Newton (3.3.7)–(3.2.14), se comporta de manera análoga a la iteración Quasi-Newton, aún dando el mismo número de iteraciones, es decir, ambas iteraciones proporcionan errores globales (denotados por ϵ) similares en el punto final de integración. También se observa en las tablas que, ambas iteraciones seleccionan prácticamente los mismos predictores al hacer uso de la estrategia VOS. Entre corchetes mostramos el número de veces que cada predictor fue elegido, así, $[0, 0, 187, 12]$ significa que los predictores de orden 1 y 2 nunca fueron elegidos, que el predictor de orden tres fue elegido 187 veces y el predictor de orden cuatro fue tomado 12 veces. También se aprecia en la Tabla 3.6.3 (Problema 3.2) que la iteración de Cooper y Butcher necesita más iteraciones que nuestro esquema iterativo para alcanzar la misma precisión. Esto puede explicarse porque la iteración de Cooper y Butcher no gana ordenes en potencias de h con las iteraciones, sino que mantiene el orden del predictor, y no alcanza por tanto el orden del corrector si se empieza con predictores de bajo orden, lo cual supone una gran desventaja respecto a nuestra iteración Single-Newton. En el Problema 3.3 (caso de frecuencias más altas) la iteración de Cooper y Butcher tiene un comportamiento similar a la nuestra. También puede verse que la estrategia VOS funciona adecuadamente tanto en el Problema 3.2 como en el Problema 3.3. También se observa que los mejores predictores para los problemas que solo posean frecuencias bajas (Problema 3.2) son aquellos de ordenes más alto, sin embargo cuando el problema posee frecuencias altas no despreciables (Problema 3.3) entonces los predictores de orden medio son elegidos más a menudo por el código. Esto está en concordancia con la teoría de amplificación de errores a través de los predictores, expuesta anteriormente.

En el caso de los método de Gauss de tres y de cuatro etapas, cuando comparamos nuestra iteración Single-Newton con la iteración Quasi-Newton, se puede apreciar en las Tablas 3.6.4 y 3.6.6 que ambas iteraciones prácticamente alcanzan precisiones similares en el mismo número de iteraciones. En el peor de los casos, nuestra iteración necesita una iteración más que la Quasi-Newton para obtener errores globales similares. Esto muestra una gran ventaja a favor de nuestra iteración cuando comparamos los esfuerzos computacionales realizados.

También se muestra que cuando ambas iteraciones son implementadas con la estrategia VOS, ambas toman prácticamente los mismos predictores y en el caso del Problema 3.3 los predictores de orden medio son los preferidos nuevamente, observe que el predictor de orden más alto no fue elegido casi nunca. Por otra parte, para el caso de los problemas con todos sus modos de frecuencia de tamaño moderado, los predictores de ordenes mas altos son los elegidos por el código. En estas tablas también puede observarse, un incremento en orden alcanzado por los métodos cuando el número de etapas crece, como es de esperar. Además, se aprecia que el proceso iterativo converge prácticamente tras 4 ó 5 iteraciones. Sin embargo, para un número bajo de iteraciones (digamos $\mu = 2, 3$) por paso de integración, la precisión alcanzada es prácticamente independiente del método de Gauss usado, es decir es similar para los casos $s = 2, 3, 4$, véanse Tablas 3.6.3–3.6.6.

De los resultados anteriores y de otros muchos experimentos numéricos no presentados aquí, concluimos que la iteración Single–Newton dada por (3.2.14), con las matrices L, S y los parámetros $\gamma = (\det A)^{2/s}$ elegidos tal cual se indica en la sección 3.3, parece ser una alternativa más eficaz que la iteración Quasi-Newton, cuando se implementa sobre los métodos de Gauss y posiblemente sobre cualquier método Runge–Kutta–Nyström altamente implícito. También parece ser más eficiente que la iteración propuesta por Cooper y Butcher [22], ya que esta última fue diseñada para problemas de primer orden y además presenta el inconveniente de no ganar orden con las iteraciones, sino que mantiene el orden del predictor. Por otra parte, es importante recalcar que la estrategia VOS para los predictores parece ser más atractiva que la opción de un determinado predictor fijo, ya que la alternativa VOS es más flexible y se ajusta mejor a cada paso de la integración.

En el caso de los métodos Lobatto IIIA, que son también potencialmente atractivos para problemas de segundo orden, podemos adaptar fácilmente la iteración de tipo Single–Newton desarrollada en este capítulo. Teniéndose en cuenta que para esos métodos la matriz T del Single–Newton (3.2.12) debe optimizarse sólo para las etapas implícitas de dichos métodos, en virtud de que la primera etapa es explícita.

Tabla 3.6.1: Ordenes de convergencia y errores en el punto final: $p(h)[e(h)], p'(h)[e'(h)]$, para el Problema 3.1 ($t_{end} = 4$, $h = 0{,}4$), usando nuestra iteración Single-Newton (dando μ iteraciones por paso de integración) sobre el método de Gauss de dos etapas. El predictor usado fue el denotado por $Y_{n,i}^{(0),q}$.

Gauss 2 etapas	$Y_{n,i}^{(0),1}$	$Y_{n,i}^{(0),2}$	$Y_{n,i}^{(0),3}$	$Y_{n,i}^{(0),4}$
$\mu = 1$	$p = 1{,}9[4{,}6e{-}2]$	$p = 2{,}7[5{,}9e{-}3]$	$p = 3{,}8[3{,}1e{-}3]$	$p = 5{,}1[2{,}3e{-}4]$
	$p' = 2{,}2[2{,}7e{-}2]$	$p' = 3{,}6[3{,}1e{-}3]$	$p' = 4{,}0[2{,}4e{-}3]$	$p' = 6{,}6[1{,}4e{-}4]$
$\mu = 2$	$p = 3{,}9[1{,}1e{-}3]$	$p = 5{,}1[2{,}3e{-}4]$	$p = 5{,}7[4{,}3e{-}5]$	$p = 7{,}1[4{,}3e{-}6]$
	$p' = 4{,}0[7{,}5e{-}4]$	$p' = 5{,}6[6{,}1e{-}5]$	$p' = 6{,}0[3{,}7e{-}5]$	$p' = 9{,}2[1{,}6e{-}6]$
$\mu = 3$	$p = 5{,}8[1{,}8e{-}5]$	$p = 7{,}1[4{,}7e{-}6]$	$p = 7{,}6[6{,}9e{-}7]$	$p = 9{,}1[8{,}3e{-}8]$
	$p' = 6{,}1[1{,}3e{-}5]$	$p' = 7{,}4[1{,}1e{-}6]$	$p' = 8{,}0[6{,}4e{-}7]$	$p' = 10{,}5[2{,}9e{-}8]$

Tabla 3.6.2: Ordenes de convergencia y errores en el punto final: $p(h)[e(h)], p'(h)[e'(h)]$, para el Problema 3.1 ($t_{end} = 4$, $h = 0{,}4$), usando la iteración Quasi-Newton (dando μ iteraciones por paso de integración) sobre el método de Gauss de dos etapas. El predictor usado fue el denotado por $Y_{n,i}^{(0),q}$.

Gauss 2 etapas	$Y_{n,i}^{(0),1}$	$Y_{n,i}^{(0),2}$	$Y_{n,i}^{(0),3}$	$Y_{n,i}^{(0),4}$
$\mu = 1$	$p = 1{,}8[9{,}2e\text{-}3]$	$p = 4{,}7[4{,}1e\text{-}3]$	$p = 3{,}9[1{,}6e\text{-}3]$	$p = 5{,}0[8{,}8e\text{-}5]$
	$p' = 2{,}2[1{,}4e\text{-}2]$	$p' = 2{,}6[1{,}1e\text{-}2]$	$p' = 3{,}8[1{,}4e\text{-}3]$	$p' = 5{,}2[3{,}4e\text{-}4]$
$\mu = 2$	$p = 5{,}2[6{,}6e\text{-}7]$	$p = 5{,}6[1{,}2e\text{-}5]$	$p = 7{,}2[8{,}5e\text{-}7]$	$p = 7{,}6[1{,}3e\text{-}7]$
	$p' = 5{,}0[1{,}8e\text{-}5]$	$p' = 5{,}9[1{,}6e\text{-}5]$	$p' = 6{,}7[2{,}3e\text{-}6]$	$p' = 8{,}4[3{,}1e\text{-}7]$
$\mu = 3$	$p = 8{,}7[1{,}2e\text{-}9]$	$p = 9{,}6[1{,}1e\text{-}8]$	$p = 10{,}1[3{,}6e\text{-}10]$	$p = 10{,}5[2{,}1e\text{-}11]$
	$p' = 8{,}1[2{,}5e\text{-}9]$	$p' = 8{,}7[8{,}2e\text{-}9]$	$p' = 10{,}0[2{,}7e\text{-}10]$	$p' = 10{,}9[2{,}6e\text{-}11]$

Tabla 3.6.3: Método de Gauss de dos etapas en el Problema 3.2.

Gauss 2 etapas	Nuestra iteración	Iter. Quasi–Newton	Iter. Cooper-Butcher.
$\mu = 1$	$\epsilon = 2{,}26e{+}1[0, 0, 0, 3999]$	$\epsilon = 2{,}23e{+}1[0, 0, 0, 3999]$	$\epsilon = 2{,}85e{+}1$
$\mu = 2$	$\epsilon = 1{,}89e{-}1[0, 0, 0, 3999]$	$\epsilon = 1{,}78e{-}2[0, 0, 0, 3999]$	$\epsilon = 2{,}92e{+}1$
$\mu = 3$	$\epsilon = 1{,}60e{-}2[0, 0, 0, 3999]$	$\epsilon = 1{,}62e{-}2[0, 0, 0, 3999]$	$\epsilon = 2{,}37e{+}0$
$\mu = 4$	$\epsilon = 1{,}50e{-}2[0, 0, 0, 3999]$	$\epsilon = 1{,}50e{-}2[0, 0, 0, 3999]$	$\epsilon = 4{,}69e{-}1$

Errores globales en el punto final de integración sobre la componente y para $t_{end} = 5 \cdot 10^5$ ($h = 125$), usando nuestra iteración Single-Newton , la iteración Quasi–Newton y la iteración de Cooper y Butcher. Se dieron μ iteraciones por paso de integración y se usó la estrategia de orden variable VOS para predictores. Entre corchetes se recoge el número de veces que cada predictor fue elegido para el caso de las iteraciones Single–Newton y Quasi–Newton.

Tabla 3.6.4: Métodos de Gauss de tres etapas (orden seis) y de cuatro etapas (orden 8), sobre el Problema 3.2.

Gauss 3 etapas	Nuestra iteración	Iter. Quasi–Newton
$\mu = 1$	$\epsilon = 9{,}90e{+}0[0, 0, 0, 3527, 472]$	$\epsilon = 3{,}48e{+}0[0, 0, 0, 3316, 683]$
$\mu = 2$	$\epsilon = 8{,}29e{-}2[0, 0, 0, 2635, 1364]$	$\epsilon = 5{,}69e{-}3[0, 0, 0, 2632, 1367]$
$\mu = 3$	$\epsilon = 5{,}78e{-}4[0, 0, 0, 2630, 1369]$	$\epsilon = 4{,}11e{-}4[0, 0, 0, 2630, 1369]$
$\mu = 4$	$\epsilon = 4{,}87e{-}6[0, 0, 0, 2630, 1369]$	$\epsilon = 3{,}10e{-}6[0, 0, 0, 2630, 1369]$
$\mu = 5$	$\epsilon = 2{,}93e{-}6[0, 0, 0, 2630, 1369]$	$\epsilon = 2{,}94e{-}6[0, 0, 0, 2630, 1369]$
Gauss 4 etapas		
$\mu = 1$	$\epsilon = 1{,}96e{+}1[0, 0, 0, 498, 3501]$	$\epsilon = 1{,}74e{+}1[0, 0, 0, 436, 3563]$
$\mu = 2$	$\epsilon = 9{,}23e{-}3[0, 0, 0, 535, 3464]$	$\epsilon = 4{,}65e{-}3[0, 0, 0, 535, 3464]$
$\mu = 3$	$\epsilon = 1{,}75e{-}4[0, 0, 0, 535, 3464]$	$\epsilon = 1{,}46e{-}4[0, 0, 0, 535, 3464]$
$\mu = 4$	$\epsilon = 2{,}69e{-}7[0, 0, 0, 535, 3464]$	$\epsilon = 4{,}84e{-}8[0, 0, 0, 535, 3464]$
$\mu = 5$	$\epsilon = 2{,}07e{-}8[0, 0, 0, 535, 3464]$	$\epsilon = 4{,}01e{-}8[0, 0, 0, 535, 3464]$

Errores globales en el punto final $t_{end} = 5 \cdot 10^5$, para la componente y con tamaño de paso fijo $h = 125$. Hemos usado nuestra iteración y la iteración Quasi–Newton. Se dieron μ iteraciones por paso de integración y se usa la estrategia VOS para los predictores. Entre corchetes se recoge el número de veces que cada predictor fue elegido.

Tabla 3.6.5: Método de Gauss de dos etapas pare el Problema 3.3.

Gauss 2 etapas	Nuestra iteración	Iter. Quasi–Newton	Iter. Cooper–Butcher
$\mu = 1$	$\epsilon = 3{,}65e\text{–}3[0, 0, 187, 12]$	$\epsilon = 3{,}40e\text{–}3[0, 0, 176, 23]$	$\epsilon = 4{,}92e\text{–}3$
$\mu = 2$	$\epsilon = 1{,}22e\text{–}4[0, 0, 161, 38]$	$\epsilon = 4{,}66e\text{–}5[0, 0, 158, 41]$	$\epsilon = 2{,}22e\text{–}4$
$\mu = 3$	$\epsilon = 2{,}20e\text{–}5[0, 0, 158, 41]$	$\epsilon = 1{,}81e\text{–}5[0, 0, 158, 41]$	$\epsilon = 2{,}18e\text{–}5$
$\mu = 4$	$\epsilon = 1{,}85e\text{–}5[0, 0, 158, 41]$	$\epsilon = 1{,}83e\text{–}5[0, 0, 158, 41]$	$\epsilon = 1{,}96e\text{–}5$

Errores globales en el punto final $t_{end} = 10$, en la componente y usando tamaño de paso constante $h = 0{,}05$. Hemos usado nuestra iteración Single-Newton, la iteración Quasi–Newton y la iteración de Cooper y Butcher. Se dieron μ iteraciones por paso de integración y se usó la estrategia VOS para predictores. Entre corchetes el número de veces que cada predictor fue elegido para el caso de las iteraciones Single–Newton y Quasi–Newton.

Tabla 3.6.6: Método de Gauss de tres etapas y método de Gauss de cuatro etapas en Problema 3.3.

Gauss 3 etapas	Nuestra iteración	Iter. Quasi–Newton
$\mu = 1$	$\epsilon = 3{,}16e\text{–}4[0, 0, 1, 198, 0]$	$\epsilon = 3{,}27e\text{–}4[0, 0, 1, 198, 0]$
$\mu = 2$	$\epsilon = 5{,}42\text{–}6[0, 0, 1, 198, 0]$	$\epsilon = 3{,}68e\text{–}6[0, 0, 1, 198, 0]$
$\mu = 3$	$\epsilon = 1{,}20e\text{–}7[0, 0, 1, 198, 0]$	$\epsilon = 3{,}52e\text{–}8[0, 0, 1, 198, 0]$
$\mu = 4$	$\epsilon = 3{,}03e\text{–}8[0, 0, 1, 198, 0]$	$\epsilon = 3{,}16e\text{–}8[0, 0, 1, 198, 0]$
Gauss 4 etapas		
$\mu = 1$	$\epsilon = 3{,}40e\text{–}4[0, 0, 1, 197, 1]$	$\epsilon = 3{,}43e\text{–}4[0, 0, 1, 197, 1]$
$\mu = 2$	$\epsilon = 4{,}83\text{–}6[0, 0, 1, 198, 0]$	$\epsilon = 4{,}25e\text{–}6[0, 0, 1, 198, 0]$
$\mu = 3$	$\epsilon = 7{,}77e\text{–}8[0, 0, 1, 198, 0]$	$\epsilon = 4{,}01e\text{–}8[0, 0, 1, 198, 0]$
$\mu = 4$	$\epsilon = 1{,}48e\text{–}9[0, 0, 1, 198, 0]$	$\epsilon = 2{,}43e\text{–}10[0, 0, 1, 198, 0]$
$\mu = 5$	$\epsilon = 5{,}25e\text{–}11[0, 0, 1, 198, 0]$	$\epsilon = 4{,}22e\text{–}11[0, 0, 1, 198, 0]$

Errores globales en la componente y para $t_f = 10$ ($h = 0{,}05$) con nuestra iteración y con la iteración Quasi–Newton. Se dieron μ iteraciones por paso y se usó la estrategia de orden variable para predictores (en corchetes el número de veces que cada algoritmo de arranque fue elegido).

Capítulo 4

Un código basado en el método de Gauss de 2 etapas para problemas de segundo orden

4.1. Introducción

Es un hecho bien conocido que las fórmulas Runge–Kutta Gauss poseen excelentes propiedades de orden y de estabilidad. Además son P–estables para problemas de valor inicial de segundo orden, es decir, los métodos preservan las amplitudes de las frecuencias en problemas lineales, véase [62] para una definición precisa de P–estabilidad en métodos de un paso. También poseen los ordenes más altos de convergencia en relación al número de etapas dentro de la clase de métodos Runge–Kutta y pertenecen a la la clase de métodos simétricos y simplécticos, véase por ejemplo [55, 56, 76]. Esto supone una gran ventaja para la integración de problemas Hamiltonianos. El principal inconveniente de estos métodos es que son altamente implícitos. Para solventar este handicap, diversos autores han propuestos iteraciones adecuadas de tipo Newton para resolver las ecuaciones de etapas en el caso de problemas de primer orden de tipo stiff [22, 23, 41] y para problemas oscilatorios cuyas soluciones involucran frecuencias altas [37, 46, 60]. Sin embargo, no hay en la literatura códigos disponibles de carácter general incorporando estos procesos iterativos. En este capítulo pretendemos cubrir parte de este vacío. Abordaremos aquí la construcción de un código de propósito general basado en el método de Gauss de dos etapas, para integrar satisfactoriamente problemas del tipo especial (4.1.1) en precisión baja y media, de tal manera que éste

pueda cubrir adecuadamente la clase de problemas oscilatorios en general, y especialmente aquellos en que las frecuencias altas de soluciones próximas a la solución exacta $y(t)$ del PVI no sean significativas, es decir, lleven asociadas amplitudes pequeñas. En este caso, los métodos explícitos no son apropiados, ya que toman un elevado número de pasos debido a sus pobres propiedades de estabilidad. Nuestro código podría ser de especial interés para la solución de importantes clases de Ecuaciones Diferenciales Parciales (PDE) que surgen de fenómenos vibratorios, las cuales se discretizan en las variables espaciales por alguna técnica estándar tal como el Método de Líneas (MOL) . Este capítulo está estrechamente ligado a las investigaciones llevadas a cabo en el capítulo anterior y también recogidas en [46] y [48]. En estos trabajos, se desarrolló una iteración Single–Newton para resolver las ecuaciones implícitas de las etapas de los métodos de Gauss. También se probó por medio de argumentos teóricos y prácticos que la iteración Single–Newton es más eficiente computacionalmente que la iteración Quasi–Newton.

Cuando un método Runge–Kutta de s etapas con matrices de coeficientes (A, b, c) se aplica a un sistema diferencial de segundo orden de tipo especial

$$\begin{aligned} &y''(t) = f\big(t, y(t)\big), \quad t \in [t_0, t_{fin}], \\ &y(t_0) = y_0, \quad y'(t_0) = y'_0, \quad y, y', f \in \mathbb{R}^m, \end{aligned} \tag{4.1.1}$$

las etapas internas Y_n verifican la ecuación (3.2.2) (véase capítulo anterior) y la solución de avance se computa mediante las fórmulas (3.2.6)–(3.2.7) por las razones expuestas en la sección 3.2.

La versión Runge–Kutta–Nyström del método de Gauss de 2 etapas viene dada por la siguiente tabla de Butcher [76, p. 37],

$$\begin{array}{c|c} c & A^2 \\ \hline & b^T A \\ \hline & b^T \end{array} \quad \equiv \quad \begin{array}{c|cc} \frac{1}{2} - \frac{\sqrt{3}}{6} & \frac{1}{24} & \frac{1}{8} - \frac{\sqrt{3}}{12} \\ \frac{1}{2} + \frac{\sqrt{3}}{6} & \frac{1}{8} + \frac{\sqrt{3}}{12} & \frac{1}{24} \\ \hline & \frac{1}{4} + \frac{\sqrt{3}}{12} & \frac{1}{4} - \frac{\sqrt{3}}{12} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \tag{4.1.2}$$

En este caso la solución de avance para las etapas toma la forma:

$$\begin{aligned} &y_{n+1} = y_n + (b^T A^{-1} \otimes I) Y_n, && b^T A^{-1} = (-\sqrt{3}, \sqrt{3}), \\ &hy'_{n+1} = 12y_n + hy'_n + (b^T A^{-2} \otimes I) Y_n, && b^T A^{-2} = (-6(1+\sqrt{3}), 6(-1+\sqrt{3})). \end{aligned} \tag{4.1.3}$$

El resto del capítulo se organiza de la siguiente manera: en la sección 4.2 siguiendo la línea de investigación realizada en [46], se describirá un tipo de esquema iterativo de tipo

Single–Newton para el método de Gauss de dos etapas, además se explicitarán predictores para empezar las iteraciones de las etapas internas con el objetivo de usar una estrategia de orden variable la cual será implementada en el código. En la sección 4.3, a efectos de usar una estrategia de paso variable, se diseñarán estimadores del error local asintóticamente correctos y se recomendará la elección de uno de ellos. También se suministrará una predicción del paso inicial en la sección 4.4. El pseudocódigo y el algoritmo se describirán con detalle en la sección 4.5. Finalmente en la sección 4.6 se presentan algunos experimentos numéricos con el propósito de tasar los méritos y deméritos del código al compararse con otros códigos de propósito general considerados en la literatura. Los resultados teóricos y prácticos de este capítulo están recogidos esencialmente en los trabajos [47, 49, 51, 75].

4.2. Solución de las ecuaciones de etapas

En el capítulo 3 (véase también [46]) se desarrolló una iteración eficiente tipo Newton para resolver las etapas internas de los métodos Runge–Kutta–Nyström altamente implícitos. Para los métodos de Gauss de s etapas con $s = 2, 3$ y 4. Esta iteración resulta más ventajosa que la iteración Quasi–Newton, pues para errores globales similares en las soluciones de avance comporta costos computacionales notablemente inferiores. Además la iteración Single–Newton evita el uso de aritmética compleja (una implementación eficiente de la Iteración Quasi–Newton requiere aritmética compleja para los métodos de Gauss [37, Sec. 3]) y también evita el incremento de la dimensión en el problema algebraico asociado para el cálculo de los valores de las etapas. Además, la iteración Single-Newton posee una razón de convergencia pequeña (alta velocidad de convergencia) sobre problemas lineales,

$$y''(t) = -\omega^2 y, \quad \omega \in \mathbb{R}, \quad \omega \text{ arbitrario.} \tag{4.2.1}$$

Así, la nueva iteración sacrifica alcanzar convergencia en la primera iteración en problemas lineales

$$y''(t) = Jy + g(t), \quad J \in \mathbb{R}^{m,m}, g(t) \in \mathbb{R}^m, \tag{4.2.2}$$

a costa de reducir sensiblemente el costo algebraico por iteración dada.

La iteración propuesta en el capítulo anterior está dada por las fórmulas,

$$\begin{cases} \big(\xi I - (I \otimes J_n)\big)\Delta^{(\nu)} = (\xi P \otimes I)D(Y_n^{(\nu-1)}) + (\xi L \otimes I)\Delta^{(\nu)} \\ Y^{(\nu)} = Y^{(\nu-1)} + (S \otimes I)\Delta^{(\nu)}, \quad \nu = 1, 2, \ldots, \nu_{\text{máx}}, \end{cases} \tag{4.2.3}$$

con

$$\xi = \gamma^{-1}h^{-2} \text{ y } P = (I - L)S^{-1}.$$

Donde

$$D(Z) := e \otimes y_n + hc \otimes y'_n - Z + h^2(A^2 \otimes I)F(Z)$$

representa el residual en el paso t_n, $J \simeq \dfrac{\partial f}{\partial y}(t_n, y_n)$ y los coeficientes en la iteración para el caso del método de Gauss de dos etapas vienen dados por

$$\gamma = 1/12, \quad L = \begin{pmatrix} 0 & 0 \\ (12 + 7\sqrt{3})/6 & 0 \end{pmatrix}, \qquad S = \begin{pmatrix} 1 & -7 + 4\sqrt{3} \\ 0 & 1 \end{pmatrix}. \tag{4.2.4}$$

El error de las iteraciones (4.2.3)–(4.2.4) sobre el test lineal (4.2.1) satisface

$$Y_n - Y_n^{(\nu)} = N(z)(Y_n - Y_n^{(\nu-1)}),$$

donde la matriz iteración

$$N(z) := z^2(I + z^2T)^{-1}(T - A^2), \quad T = \gamma S(I - L)^{-1}S^{-1}, \; z = \omega h,$$

tiene un único autovalor distinto de cero, el cual para el caso del método de Gauss de 2 etapas viene dado por

$$\phi(z) = 1 - \det(I - N(z)) = \frac{z^2/12}{(1 + z^2/12)^2} \; .$$

Aunque esta iteración en cierto sentido fue *optimizada* en problemas lineales (capítulo 3), la razón de convergencia para $z \simeq \sqrt{12}$ no es lo suficientemente buena ya que su máximo es $\phi(\sqrt{12}) = 1/4$. Proponemos hacer aquí una mejora acelerando el esquema iterativo en la iteración ν^*–ésima. Para este propósito consideremos la iteración más general (abajo, $\beta_\nu = 1$ para cada $\nu \neq \nu^*$ y $\beta_{\nu^*} = \beta^*$, ν^* y β^* han de elegirse adecuadamente)

$$\begin{array}{c} (\xi I - I \otimes J)\Delta^{(\nu)} = (\xi\beta_\nu P \otimes I)D(Y_n^{(\nu-1)}) + (\xi L \otimes I)\Delta^{(\nu)}, \;\; \nu = 1, 2, \ldots, \nu_{\text{máx}} \\ Y_n^{(\nu)} = Y_n^{(\nu-1)} + (S \otimes I)\Delta^{(\nu)}, \quad \text{con} \;\; P = (I - L)S^{-1} \quad \text{y} \quad \xi = \gamma^{-1}h^{-2}. \end{array} \tag{4.2.5}$$

Ahora, el error de las iteraciones en problemas lineales (4.2.1) satisface

$$Y_n - Y_n^{(\nu)} = N_\nu(z)(Y_n - Y_n^{(\nu-1)}), \quad \nu = 1, 2 \ldots, \quad z = \omega h,$$

con

$$N_\nu(z) = \begin{cases} N(z)^\nu, & \text{si } \nu < \nu^*, \\ N(z)^{\nu-1}((1-\beta^*)I + \beta^* N(z)), & \text{en otro caso.} \end{cases}$$

De aquí, se sigue que *la mejor aceleración* para la iteración ν^*–ésima se consigue cuando β^* se elige de tal modo que

$$K(\nu^*, \beta^*) := \max_{z\in\mathbb{R}} |\phi(z)^{\nu^*-1}((1-\beta^*) + \beta^*\phi(z))| \qquad \text{es mínimo.} \tag{4.2.6}$$

Para el método de Gauss de dos etapas, hemos seleccionado $\nu^* = 4$, con lo cual,

$$\nu^* = 4, \quad \beta_4 = \beta^* = 1{,}30011107084447842..., \quad K(\nu^*, \beta^*) \simeq 3{,}9 \cdot 10^{-4}. \tag{4.2.7}$$

De hecho $\beta^* = (1-\delta^*)^{-1}$, donde δ^* es la única raíz positiva del polinomio $27\delta^4 + 4\delta - 1$. Debe recalcarse que con la elección standard de $\beta_4 = 1$ se obtiene $K(4,1) \simeq 3{,}9 \cdot 10^{-3}$. Esto nos dice que la opción (4.2.7) permite reducir el error en la cuarta iteración por un factor de 10. Además, la opción de *aceleración en* la iteración $\nu^* = 4$, prácticamente preserva todas las buenas propiedades de orden y convergencia exhibida por la iteración original (4.2.3)–(4.2.4), las cuales fueron ampliamente estudiadas en el capítulo anterior.

Para problemas lineales del tipo (4.2.2), el costo de la iteración (4.2.5) puede reducirse. De hecho, el número de evaluaciones de la derivada puede reducirse a s (el número de etapas del método) independientemente del número de iteraciones tomadas para alcanzar la convergencia. Esto es debido a que en esta situación especial, el residual ν puede ser actualizado desde la iteración anterior usando el hecho de que

$$D(Y_n^{(\nu)}) - D(Y_n^{(\nu-1)}) = (-I + h^2A^2 \otimes J)(Y_n^{(\nu)} - Y_n^{(\nu-1)}), \qquad \nu = 1, 2, \dots . \tag{4.2.8}$$

Ahora el cálculo de $(h^2A^2 \otimes J)(Y_n^{(\nu)} - Y_n^{(\nu-1)})$ en el lado derecho de (4.2.8) puede evitarse usando (4.2.5). Así, después de algunas manipulaciones algebraicas sencillas pero algo tediosas se prueba que sobre la clase de problemas lineales (4.2.2), la iteración (4.2.5) es equivalente a,

$$\left.\begin{array}{c} R^{(0)} = D(Y_n^{(0)}), \qquad \xi = \gamma^{-1}h^{-2}, \\ \big(\xi I - (I \otimes J)\big)E^{(\nu)} = (\xi P \otimes I)R^{(\nu-1)} + (\xi L \otimes I)E^{(\nu)}, \\ Y^{(\nu)} = Y^{(\nu-1)} + (S \otimes I)E^{(\nu)}, \\ R^{(\nu)} = \beta_{\nu+1}\left[(Q \otimes I)(Y^{(\nu)} - Y^{(\nu-1)} - R^{(\nu-1)}) + (\beta_\nu^{-1} - 1)R^{(\nu-1)}\right], \\ \nu = 1, \dots, \nu_{\text{máx}}. \end{array}\right\} \tag{4.2.9}$$

Donde la matriz Q calcula en general mediante la fórmula

$$Q := \gamma^{-1} A^2 S(I - L)S^{-1} - I.$$

Para el método de Gauss de dos etapas tenemos que,

$$Q = \begin{pmatrix} \frac{\sqrt{3}}{3} & 1 - \frac{2\sqrt{3}}{3} \\ 1 + \frac{2\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} \end{pmatrix}. \tag{4.2.10}$$

En conclusión, para el método de Gauss de dos etapas, proponemos el esquema iterativo (4.2.4)–(4.2.5) para problemas no lineales y el esquema (4.2.4), (4.2.9) y (4.2.10) para problemas lineales del tipo (4.2.2), usando parámetros $\beta_\nu = 1$ para $\nu \neq 4$ y con β_4 dado en (4.2.7).

4.2.1. Predictores para las etapas internas

Para los métodos de Gauss de s etapas, distintos predictores del tipo

$$Y_{n,i}^{(0)} = a_i y_{n-1} + h d_i y'_{n-1} + \sum_{j=1}^{s} b_{ij} Y_{n-1,j}, \quad i = 1, 2, \ldots, s, \tag{4.2.11}$$

para comenzar las iteraciones en el paso $t_n \to t_{n+1} = t_n + \tau h$, fueron analizados en detalle en el capítulo 3, sección 3.4 de esta memoria. Asumíamos allí que el paso previo de t_{n-1} a $t_n = t_{n-1} + h$ había sido completado.

Recordemos además que un predictor se dice de orden q, si

$$Y_{n,i} - Y_{n,i}^{(0)} = \mathcal{O}(h^q),\ i = 1, \ldots, s,\ (h \to 0).$$

En particular, de lo expuesto en el capítulo anterior se deduce inmediatamente que para el método de Gauss de dos etapas los siguientes predictores tienen ordenes consecutivos de 1 a 4. Además poseen factores de amplificación de error relativamente pequeños, siendo más estables los de menor orden.

Orden 1. $Y_{n,i}^{(0),1} = Y_{n-1,2}, \quad i = 1, 2.$

Orden 2. $Y_{n,i}^{(0),2} = p(1 + \tau c_i),\ i = 1, 2$, donde $p(t)$ es un polinomio de grado 1 a lo sumo que satisface, $p(c_i) = Y_{n-1,i},\ i = 1, 2.$

Orden 3. $Y_{n,i}^{(0),3} = p(1 + \tau c_i),\ i = 1, 2$, donde $p(t)$ es un polinomio de grado 2 que a lo sumo

satisface, $p(0) = y_{n-1},\ p(c_i) = Y_{n-1,i},\ i = 1, 2.$

Orden 4. $\{Y_{n,i}^{(0),4}\}$ de acuerdo a (4.2.11) con los siguientes coeficientes

$$\begin{aligned}
a_1 &= -(1+\tau)(-1+(-5+2\sqrt{3})\tau+(-3+2\sqrt{3})\tau^2),\\
d_1 &= -6^{-1}\tau(1+\tau)(-3+\sqrt{3}+(-3+2\sqrt{3})\tau),\\
a_2 &= (1+\tau)(1+(5+2\sqrt{3})\tau+(3+2\sqrt{3})\tau^2),\\
d_2 &= 6^{-1}\tau(1+\tau)(3+\sqrt{3}+(3+2\sqrt{3})\tau),\\
b_{12} &= \sqrt{3}+(-6+4\sqrt{3})\tau+(-17/2+5\sqrt{3})\tau^2+(-7/2+2\sqrt{3})\tau^3,\\
b_{21} &= -\sqrt{3}-(6+4\sqrt{3})\tau-(17/2+5\sqrt{3})\tau^2-(7/2+2\sqrt{3})\tau^3,\\
b_{11} &= 2^{-1}(1+\tau)(-2\sqrt{3}-2\sqrt{3}\tau+\tau^2),\\
b_{22} &= 2^{-1}(1+\tau)(2\sqrt{3}+2\sqrt{3}\tau+\tau^2).
\end{aligned}$$

En cada paso de integración el predictor seleccionado es aquel que proporciona un error más pequeño tal cual fue explicado en capítulo 3, sección 3.4 usando la estrategia de orden variable VOS. Para el primer paso los predictores elegidos son:

$$\begin{aligned}
Y_{0,i}^{(0),1} &= y_0 \text{ (orden 1)},\\
Y_{0,i}^{(0),2} &= y_0 + h_0 c_i y_0' \text{ (orden 2)},\\
Y_{0,i}^{(0),3} &= y_0 + h_0 c_i y_0' + 2^{-1}(h_0 c_i)^2 f(t_0, y_0) \text{ (orden 3)}, \quad \text{para } i = 1, 2.
\end{aligned}$$

4.3. Estimadores del Error Local para el método de Gauss de dos etapas

Aunque muchos problemas oscilatorios de tipo (4.1.1) podrían ser integrados usando una estrategia de paso fijo, esta opción ha sido comúnmente descartada en los integradores usados en el software moderno. A raíz de las investigaciones llevadas a cabo por varios autores [4, 27, 29], [55, Sect. II.14], se prefieren las estrategias de paso variable, pues generalmente proporcionan integraciones más efectivas con más bajos costos computacionales. Además una estrategia de paso fijo conlleva muchas integraciones de tipo "ensayo y error", al no conocerse de antemano el paso adecuado que debe usarse para una integración satisfactoria para el usuario. En integraciones adaptativas los tamaños de paso son controlados usando un estimador del error local junto con la fórmula de avance, normalmente la mayoría de los autores se inclinan por un estimador del error local por paso antes que por un estimador del error local por unidad de paso. Para problemas oscilatorios de tipo (4.1.1), un estimador local que requiera precisiones locales similares en ambas componentes y y y' puede conducir

a integraciones ineficientes, debido a que la presencia de altas frecuencias afecta en mayor medida la componente y', que a la componente y. Así, Addison [1] y W. Enright [27] sugieren o bien exigir una precisión local similar en ambas componentes y y hy', o bien estimar el error local solamente en la componente y, véase también [37, Sect. 5], [55, p. 224], [29, 31] y las referencias que allí se dan. Para la constitución de nuestro código GAUSS2 [47] hemos preferido esta segunda opción ya que con el estimador local descrito más adelante, el código funciona de forma aceptable en todos los problemas que hemos integrado y más importante aún, un estimador de error global basado en la propiedad *Proporcionalidad respecto a la Tolerancia* [58] es suministrado con el código. Este estimador de error global ha sido ampliamente discutido en [13] y parece ser robusto en el sentido que los errores globales en ambas componentes se ajustan al tamaño del error global estimado y también conservan cierta proporcionalidad con las tolerancias usadas en la integración.

Cuando se implementan las fórmulas implícitas de alto orden (por ejemplo, los métodos de Gauss) no existen estimadores del error local asintóticamente correctos que sean aceptablemente baratos computacionalmente. Por esa razón en muchos casos se buscan métodos de orden más bajo para estimar el error local, véase [37], y el código avanza con la solución de orden más alto. Esta estrategia tiene la desventaja de requerir muchos más pasos de los necesitados por el método original para alcanzar una precisión predeterminada δ. Observe que un alto número de pasos no necesariamente corresponde a una alta precisión, como en el caso de los métodos explícitos, ya que las ecuaciones de etapas se resuelven aproximadamente por algún esquema iterativo de tipo Newton, hasta que una condición del tipo

$$\|Y_n^{(\mu+1)} - Y_n^{(\mu)}\| \leq \kappa_1 \cdot \delta, \quad (\text{para algún } \mu = 0, 1, \ldots, \mu_{\text{máx}})$$

es satisfecha. La constante κ_1 típicamente varía entre 0,01 y 0,1. Así que independientemente del número de pasos tomados por el código, el cual depende en gran medida del estimador de error local, se espera que los errores globales tengan tamaños $\mathcal{O}(\delta)$ cuando la integración avanza. Tal razón justifica en parte la búsqueda de estimadores asintóticamente correctos del error local para el método de Gauss de dos etapas.

Nuestra primera meta será deducir una fórmula de orden 5 para la componente y, ya que el error local será sólo estimado en esa componente. Así, para el paso $t_n \to t_{n+1} = t_n + h$,

consideraremos una fórmula del tipo,

$$\tilde{y}_{n+1} = y_n + hy'_n + h^2 \sum_{j=0}^{3} \delta_j f(t_n + c_j h, Y_{n,j}), \quad Y_{n,0} \equiv y_n,\ c_0 = 0,\ c_3 = 1, \tag{4.3.1}$$

donde $Y_{n,j},\ j = 1, 2$, son los valores de las etapas del método de Gauss de dos etapas, c_j $(j = 1, 2)$ denota sus nodos y $Y_{n,3}$ es una etapa (explícita) adicional,

$$Y_{n,3} = y_n + hy'_n + h^2 \sum_{j=1}^{2} a_j f(t_n + c_j h, Y_{n,j}). \tag{4.3.2}$$

El nuevo método puede ser escrito usando una tabla de Butcher como

$$\begin{array}{c|c} \tilde{c} & \tilde{A} \\ \hline & \tilde{\delta}^T \end{array} \quad \equiv \quad \begin{array}{c|ccc} 0 & 0 & \mathbf{0}^T & 0 \\ c & \mathbf{0} & A^2 & \mathbf{0} \\ 1 & 0 & a^T & 0 \\ \hline & \delta_0 & \delta^T & \delta_3 \end{array},$$

donde $\mathbf{0} = (0, 0)$, $c^T = (c_1, c_2)$, $a^T = (a_1, a_2)$, y $\delta^T = (\delta_1, \delta_2)$ son vectores en $\mathbb{R}^2$.

En virtud de que se verifica

$$\tilde{A}\tilde{e} = \frac{1}{2}\tilde{c}^2, \tag{4.3.3}$$

entonces para que $\tilde{y}_{n+1}$ alcance orden cinco, las siguientes condiciones deben satisfacerse, véase [55, Ch. II.14],

$$\tilde{\delta}^T\tilde{e} = \frac{1}{2}, \quad \tilde{\delta}^T\tilde{c} = \frac{1}{6}, \quad \tilde{\delta}^T\tilde{c}^2 = \frac{1}{12}, \quad \tilde{\delta}^T\tilde{c}^3 = \frac{1}{20}, \quad \tilde{\delta}^T(\tilde{A}\tilde{c}) = \frac{1}{120}. \tag{4.3.4}$$

Aquí, $\tilde{e} = (1, 1, 1, 1)^T$ y las potencias de $\tilde{c}$ se entienden como tomar potencias en sus componentes.

Las condiciones (4.3.3) y (4.3.4), dan lugar a un sistema de seis ecuaciones con seis incógnitas las cuales poseen una solución única ($\tilde{\delta}$ se calcula de las 4 primeras ecuaciones en (4.3.4), que son lineales y a^T de (4.3.3) y la última ecuación de (4.3.4)),

$$a^T = \left(\frac{2+\sqrt{3}}{8}, \frac{2-\sqrt{3}}{8}\right), \quad \tilde{\delta}^T = \left(\frac{1}{30}, \frac{5+\sqrt{3}}{20}, \frac{5-\sqrt{3}}{20}, -\frac{1}{30}\right). \tag{4.3.5}$$

Desde aquí, teniendo en cuenta (4.3.1), (4.3.2) y (4.1.2), se obtiene tras unos cálculos directos el siguiente estimador de error local para la componente y,

$$d(t_n, h) := \tilde{y}_{n+1} - y_{n+1} = w_n + \frac{h^2}{30}\big(f(t_n, y_n) - f(t_n + h, Y_{n,3})\big), \tag{4.3.6}$$

donde $Y_{n,3}$ está dado por (4.3.2) y (4.3.5), y puede ser re-escrito después de eliminar las combinaciones en la derivada segunda $f(\cdot)$, como

$$Y_{n,3} = -2y_n + \frac{3}{2}\left(Y_{n,1} + Y_{n,2}\right) - \frac{h}{2}y'_n \tag{4.3.7}$$

y

$$w_n = \frac{12}{5}y_n - \frac{6+4\sqrt{3}}{5}Y_{n,1} + \frac{-6+4\sqrt{3}}{5}Y_{n,2} + \frac{2h}{5}y'_n. \tag{4.3.8}$$

El estimador del error $\|d(t_n, h)\|$ presenta tres inconvenientes. Primero, puede dar grandes errores en la estimación en presencia de altas frecuencias como veremos en la próxima sección. En segundo lugar los errores en el esquema iterativo podrían amplificarse considerablemente debido a la presencia de $f(t_n, y_n)$ y de $f(t_n + h, Y_{n,3})$, y tercera, se necesitan dos evaluaciones extras de la función derivada por paso de integración. Para evitar los dos primeros inconvenientes podemos filtrar el estimador en el modo recomendado por Shampine para problemas stiff [77], es decir, proponemos un nuevo estimador del tipo,

$$\epsilon_0(t_n, h, \tilde{h}) = (I - \gamma\tilde{h}^2 J)^{-1} d(t_n, h). \tag{4.3.9}$$

Aquí, J representa la última matriz jacobiana computada y $((\gamma\tilde{h}^2)^{-1}I - J)$ denota la última factorización LU empleada en el proceso iterativo (4.2.5). Observe que desde el punto de vista computacional la forma $((\gamma\tilde{h}^2)^{-1}I - J)$ es mejor que $(I - \gamma\tilde{h}^2 J)$, ya que supone algún ahorro en operaciones elementales.

La técnica de filtrado en (4.3.9) también puede aplicarse a iteraciones tipo Newton tales como las propuestas en [22, 37, 46], sin embargo cuando la Iteración Quasi–Newton se implementa en métodos tales como las fórmulas de Gauss de dos etapas, el uso de (4.3.9) presenta algunos inconvenientes. En primer lugar, la constante anterior γ es un número complejo [37, Sec. 3], y resulta algo dificultoso formar el estimador. En segundo lugar, la solución del sistema lineal complejo es aproximadamente cuatro veces más costosa que en el caso real, ya que cada producto complejo involucra cuatro productos y dos sumas reales, y una suma compleja involucra dos sumas reales.

Una forma de evitar una evaluación extra de la función derivada por paso de integración en (4.3.9) (véase 4.3.6), es proceder de la siguiente manera: de las condiciones de orden (4.3.3), (4.3.4) y de (4.3.7), se deduce tras algunos cálculos directos pero algo tediosos que

$$\tilde{w}_n := Y_{n,3} - y_{n+1} = -3y_n - \frac{h}{2}y'_n + \left(\frac{3}{2} + \sqrt{3}\right)Y_{n,1} + \left(\frac{3}{2} - \sqrt{3}\right)Y_{n,2} \tag{4.3.10}$$

Entonces, suponiendo que la razón entre el nuevo paso y el anterior es de orden 1, es decir,

$$r^{-1} := \frac{h}{\tilde{h}} = \mathcal{O}(1) \quad \text{y} \quad J = \frac{\partial f}{\partial y}(t_n, y_n) + \mathcal{O}(h),$$

no es difícil probar que

$$\begin{aligned}&(I - \gamma\tilde{h}^2 J)^{-1}\big(h^2 f(t_n + h, Y_{n,3})\big) = \\ &(I - \gamma\tilde{h}^2 J)^{-1}\left(h^2 f(t_n + h, y_{n+1}) + (r^2\gamma)^{-1}\tilde{w}_n\right) - (r^2\gamma)^{-1}\tilde{w}_n + \mathcal{O}(h^6).\end{aligned}$$

Llevando esto último a (4.3.9) (véase también (4.3.6)) se obtiene un nuevo estimador de error local dado por

$$\begin{aligned}\epsilon_1(t_n, h, \tilde{h}) = (30r^2\gamma)^{-1}\tilde{w}_n + \\ (I - \gamma\tilde{h}^2 J)^{-1}\left(w_n - (30r^2\gamma)^{-1}\tilde{w}_n + 30^{-1}h^2(f(t_n, y_n) - f(t_{n+1}, y_{n+1}))\right).\end{aligned} \tag{4.3.11}$$

Observe que para problemas lineales (4.2.2), ambos estimadores ϵ_0 y ϵ_1 coinciden, pero ϵ_1 ahorra una evaluación de la derivada por paso de integración. Razones prácticas y teóricas explicitadas más adelante, nos han llevado a considerar también la versión filtrada del estimador ϵ_1 y que denotamos por ϵ_2.

$$\epsilon_2(t_n, h, \tilde{h}) = (I - \gamma\tilde{h}^2 J)^{-1}\epsilon_1(t_n, h, \tilde{h}), \tag{4.3.12}$$

Además por sus propiedades de exactitud para las frecuencias altas, el estimador ϵ_3 obtenido usando la media geométrica de los dos estimadores anteriores resultará de gran interés práctico,

$$\varepsilon_3(t_n, h, \tilde{h}) = \sqrt{\epsilon_1(t_n, h, \tilde{h}) \cdot \epsilon_2(t_n, h, \tilde{h})}. \tag{4.3.13}$$

A modo de resumen de esta sección, podemos decir que los cinco estimadores del error local aquí estudiados, tienen diferentes costos computacionales y sirven para diferentes propósitos. Así, para cada paso de integración aceptado tenemos que $\|d(t_n, h)\|$ hace uso de dos evaluaciones extras de derivadas y esto sólo es útil para problemas con modos de frecuencias moderado, $\|\epsilon_0(t_n, h, \tilde{h})\|$ necesita dos evaluaciones de derivadas y una solución de un sistema lineal, $\|\epsilon_1(t_n, h, \tilde{h})\|$ requiere una evaluación extra de derivada y una solución de un sistema lineal, $\|\epsilon_2(t_n, h, \tilde{h})\|$ y $\|\epsilon_3(t_n, h, \tilde{h})\|$ necesitan una evaluación adicional de la derivada y la solución de dos sistemas lineales. Para problemas con frecuencias altas sólo son de interés los estimadores $\epsilon_j(t_n, h, \tilde{h})$, $j = 0, 1, 2, 3$. A continuación, discutiremos con más detalle las cualidades de cada estimador ϵ_j para este tipo de problemas.

4.3.1. Los Estimadores del Error Local sobre problemas lineales

Si consideramos el problema test

$$y'' = -\omega^2 y, \quad y(0) = y_0,\ y'(0) = y'_0, \tag{4.3.14}$$

tenemos que tras de un paso de tamaño h y escribiendo $z = \omega h$, la solución exacta toma la forma

$$y(h) = y_0 \cos z + h y'_0 \frac{\sin z}{z},$$

mientras que para el método de Gauss de dos etapas tenemos

$$\begin{gathered} y_1 = y_0 R_1(z) + h y'_0 R_2(z), \\ R_1(z) = 1 + b^T A^{-1}(I + z^2 A^2)^{-1} e, \quad R_2(z) = b^T A^{-1}(I + z^2 A^2)^{-1} c, \end{gathered}$$

con los valores de las etapas $Y_0 = (Y_{01}, Y_{02})^T$ satisfaciendo

$$Y_0(z) = y_0 (I + z^2 A^2)^{-1} e + h y'_0 (I + z^2 A^2)^{-1} c.$$

Por tanto,

$$\begin{gathered} y(h) - y_1 = y_0 d_1(z) + h y'_0 d_2(z), \quad \text{con} \\ d_1(z) = \cos z - R_1(z) \quad \text{y} \quad d_2(z) = \frac{\sin z}{z} - R_2(z). \end{gathered} \tag{4.3.15}$$

Por otra parte, considerando el estimador del error local $\epsilon_1(t_0, h, \tilde{h})$ en (4.3.11), tomando $J = -\omega^2$, y además usando (4.3.8) y (4.3.10) se deduce después de algunos cálculos directos que

$$\begin{aligned} &\epsilon_1(t_0, h, \tilde{h}) = y_0 \eta_1(z) + h y'_0 \eta_2(z), \\ &\eta_1(z) = \frac{\alpha_1(z)}{30\varrho} + (1 + \varrho z^2)^{-1} \left(\beta_1(z) - \frac{\alpha_1(z)}{30\varrho} + \frac{z^2}{30} (b^T A^{-1}(I + z^2 A^2)^{-1} e) \right), \\ &\eta_2(z) = \frac{\alpha_2(z)}{30\varrho} + (1 + \varrho z^2)^{-1} \left(\beta_2(z) - \frac{\alpha_2(z)}{30\varrho} + \frac{z^2}{30} (b^T A^{-1}(I + z^2 A^2)^{-1} c) \right), \end{aligned} \tag{4.3.16}$$

donde $\varrho = \gamma r^2$, $r = h^{-1}\tilde{h}$, con

$$\begin{aligned} \alpha_1(z) &= -3 + \tilde{u}^T (I + z^2 A^2)^{-1} e, \quad \beta_1(z) = \frac{12}{5} + u^T (I + z^2 A^2)^{-1} e, \\ \alpha_2(z) &= -\frac{1}{2} + \tilde{u}^T (I + z^2 A^2)^{-1} c, \quad \beta_2(z) = \frac{2}{5} + u^T (I + z^2 A^2)^{-1} c, \end{aligned}$$

$$u^T = \left(-\frac{6 + 4\sqrt{3}}{5},\ \frac{-6 + 4\sqrt{3}}{5} \right), \quad \tilde{u}^T = \left(\frac{3}{2} + \sqrt{3},\ \frac{3}{2} - \sqrt{3} \right).$$

En virtud de que el estimador del error local es asintóticamente correcto (de orden 5), entonces se tiene que

$$\epsilon_1(t_0, h, \tilde{h}) = \widetilde{y}_1 - y_1 = y_1(h) - y_1 + \mathcal{O}(h^6), \quad h \to 0.$$

Ahora, usando el hecho de que sólo aparecen potencias pares de z cuando desarrollemos $d_j(z)$ y $\eta_j(z)$ para $j = 1, 2$, entonces se deduce de (4.3.15), (4.3.16), sin más que considerar los casos a) $y_0 = 1$, $y_0' = 0$, b) $y_0 = 0$, $y_0' = 1$, que:

$$\text{a) } d_1(z) - \eta_1(z) = \mathcal{O}(z^6), \quad \text{b) } d_2(z) - \eta_2(z) = \mathcal{O}(z^6), \quad \text{cuando } z \to 0.$$

Por otro lado, para las altas frecuencias que satisfacen $z = \omega h \gg 1$, tenemos que

$$d_1(z) = \cos z - 1 + \mathcal{O}(z^{-2}), \qquad d_2(z) = \frac{\sin z}{z} + \mathcal{O}(z^{-2}), \quad z \to \infty, \tag{4.3.17}$$

mientras que para los coeficientes del estimador $\epsilon_1(t_0, h, \tilde{h})$ se obtiene que

$$\eta_1(z) = \frac{2}{25\gamma r^2} + \mathcal{O}(z^{-2}), \qquad \eta_2(z) = -\frac{1}{60\gamma r^2} + \mathcal{O}(z^{-2}), \quad z \to \infty. \tag{4.3.18}$$

De aquí (véase también Fig. 4.3.1 y Fig. 4.3.2), vemos que el estimador $\epsilon_1(t_0, h, \tilde{h})$ es bastante aproximado cuando $\omega h \in (0, \kappa]$ para algún $\kappa = \mathcal{O}(1)$ y es también estable en el sentido que está acotado por una constante moderada para cualquier frecuencia ($\omega h \in \mathbb{R}$).

Por otra parte, el estimador $\|d(t, h)\| = \|(I - \gamma\tilde{h}^2 J)\epsilon_0(t_0, h, \tilde{h})\|$ en (4.3.9) es inadecuado en caso que $z = \omega h \gg 1$, ya que

$$\|d(t, h)\| = \|(1 + \varrho z^2)\epsilon_0(t_0, h, \tilde{h})\| = \|(1 + \varrho z^2)\epsilon_1(t_0, h, \tilde{h})\| \to \infty, \quad z \to \infty.$$

Esto justifica el *filtrado* realizado para $d(t, h)$ en la sección previa.

Por otro lado, para $\omega h \gg 1$ de (4.3.17)–(4.3.18) se puede apreciar que ϵ_1 sobreestima el error local para la componente y'. Una forma de evitar este inconveniente es hacer un nuevo filtrado del estimador ϵ_1. En este caso obtenemos el nuevo estimador ϵ_2 dado en (4.3.12) que aplicado al problema test lineal (4.3.14), nos da

$$|\epsilon_2(t_0, h, \tilde{h})| = \left| \frac{\epsilon_1(t_0, h, \tilde{h})}{1 + \varrho z^2} \right| \to 0, \quad z \to \infty.$$

Además, $|\epsilon_2|$ también da una estimación asintóticamente correcta para $h \to 0$.

En la Figura 4.3.1 se representan para $r = h^{-1}\tilde{h} = 1$ las gráficas de los errores locales (denotado por *Error Local*), del estimador $|\epsilon_1|$ (denotado por *Est1*), del estimador $|\epsilon_2|$ (denotado por *Est2*) y el estimador $|\epsilon_3|$ dado en (4.3.13) y denotado por *Est3* para el problema

Figura 4.3.1: Gráfica de la norma de estimadores de error local para (4.3.14) con $y_0 = 1,\ y_0' = 0$.

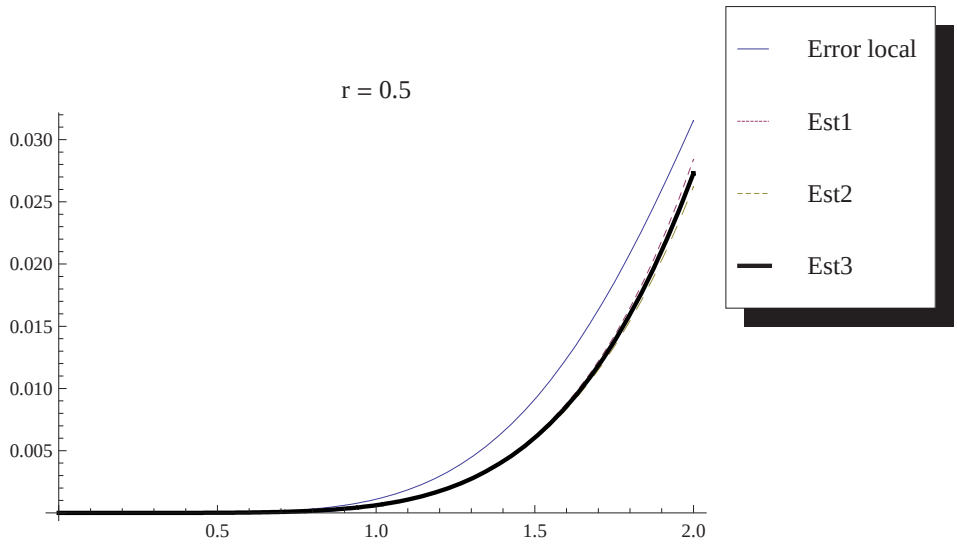


Figura 4.3.2: Gráfica de la norma de estimadores de error local para (4.3.14) con $y_0 = 1,\ y_0' = 0$.

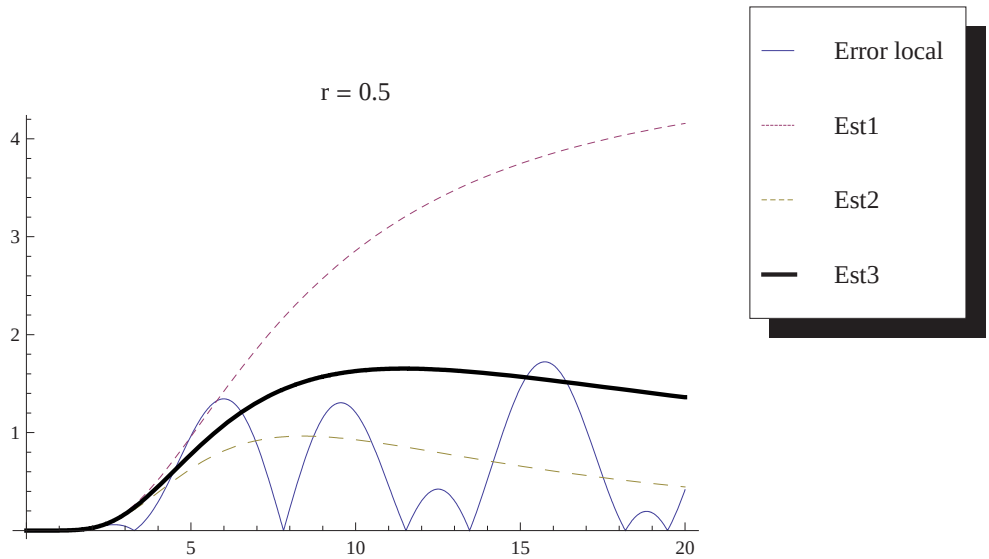


Figura 4.3.3: Gráfica de la norma de estimadores de error local para (4.3.14) con $y_0 = 1,\ y_0' = 0$.

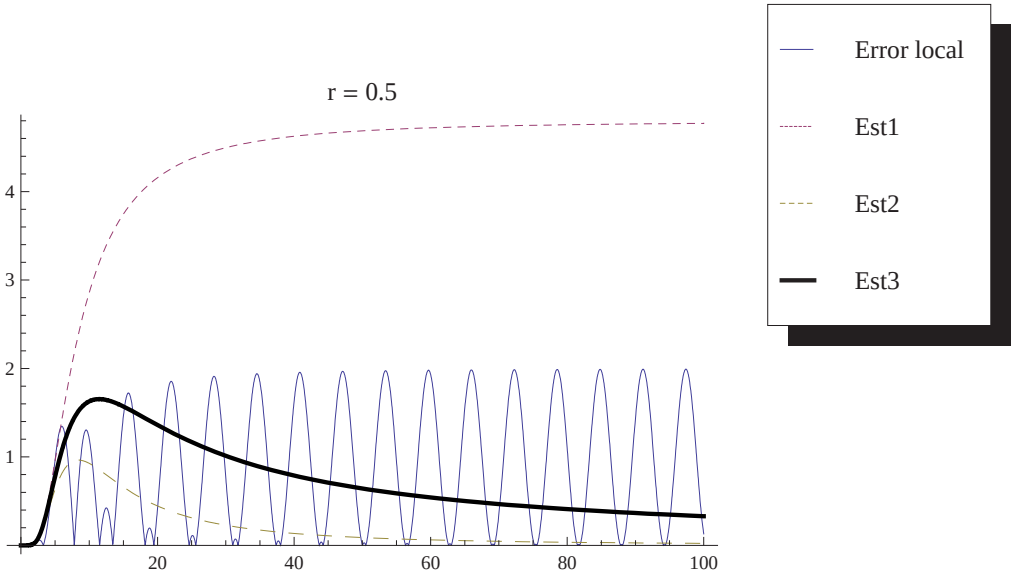


Figura 4.3.4: Gráfica de la norma de estimadores de error local para (4.3.14) con $y_0 = 1,\ y_0' = 0$.

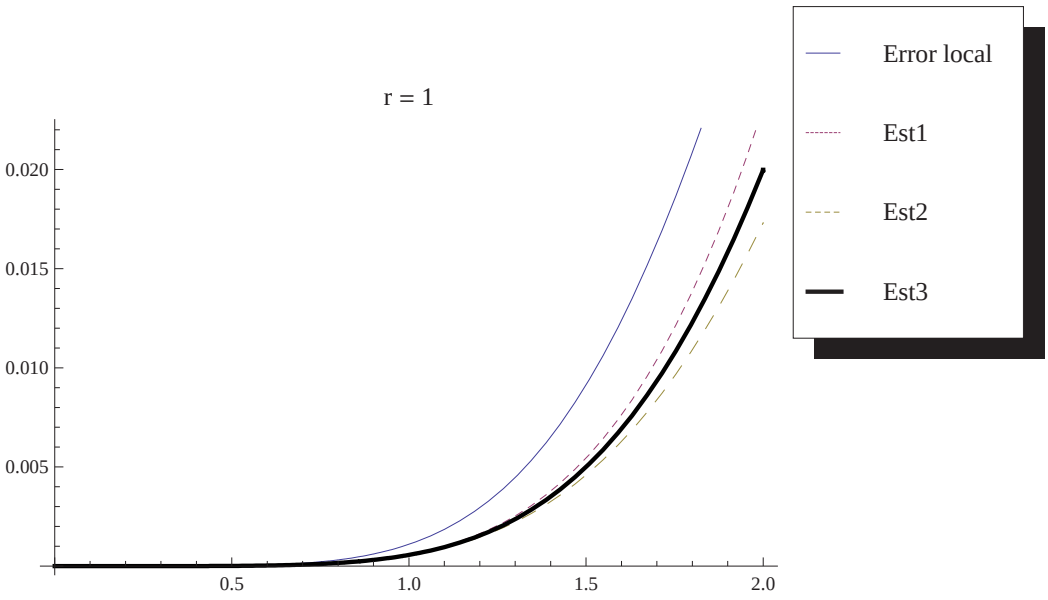


Figura 4.3.5: Gráfica de la norma de estimadores de error local para (4.3.14) con $y_0 = 1,\ y_0' = 0$.

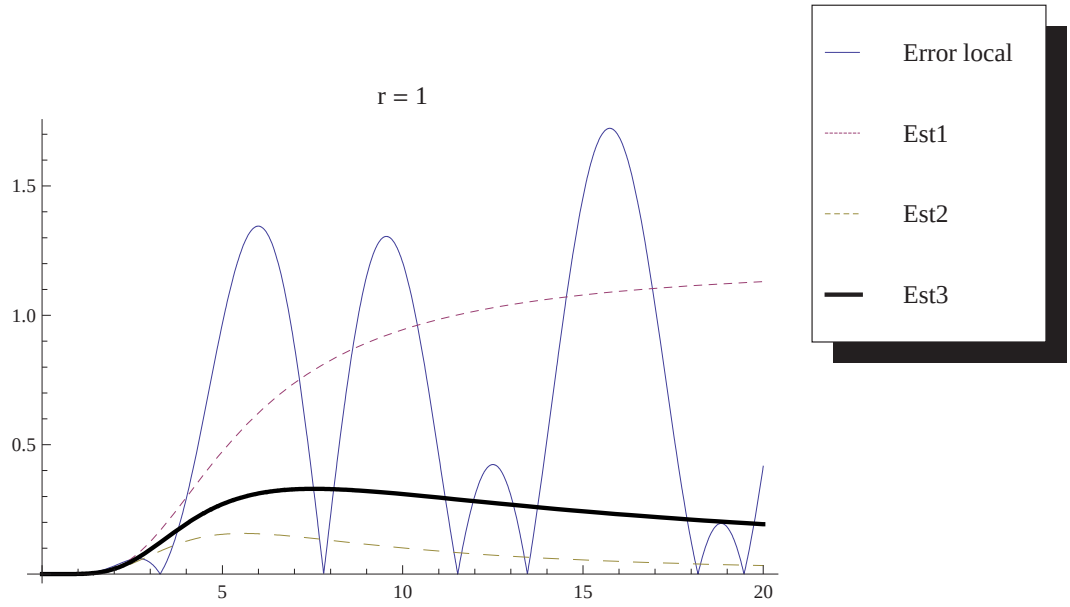


Figura 4.3.6: Gráfica de la norma de estimadores de error local para (4.3.14) con $y_0 = 1,\ y_0' = 0$.

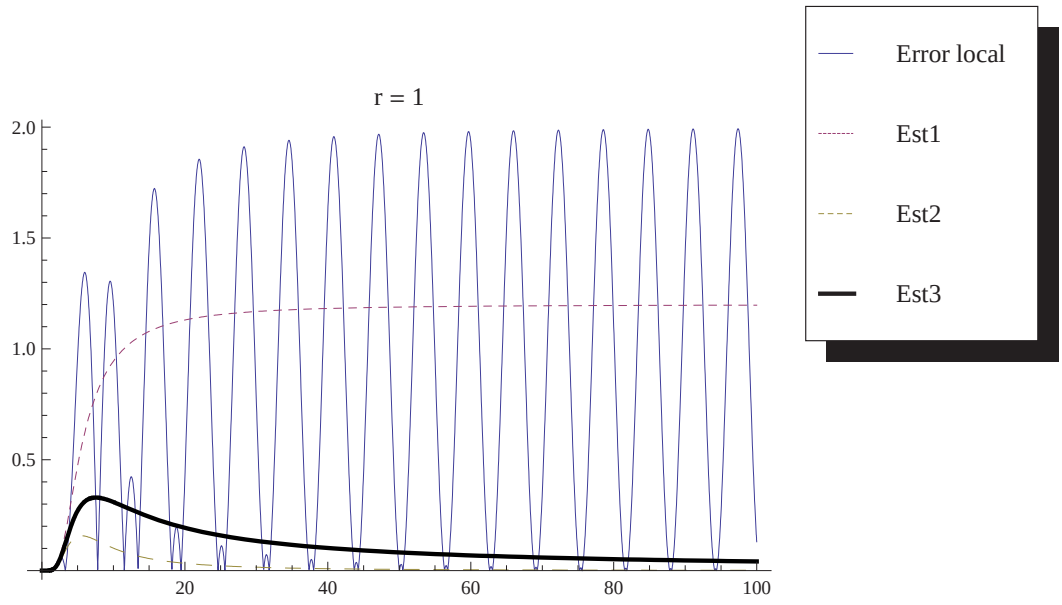


Figura 4.3.7: Gráfica de la norma de estimadores de error local para (4.3.14) con $y_0 = 1,\ y_0' = 0$.

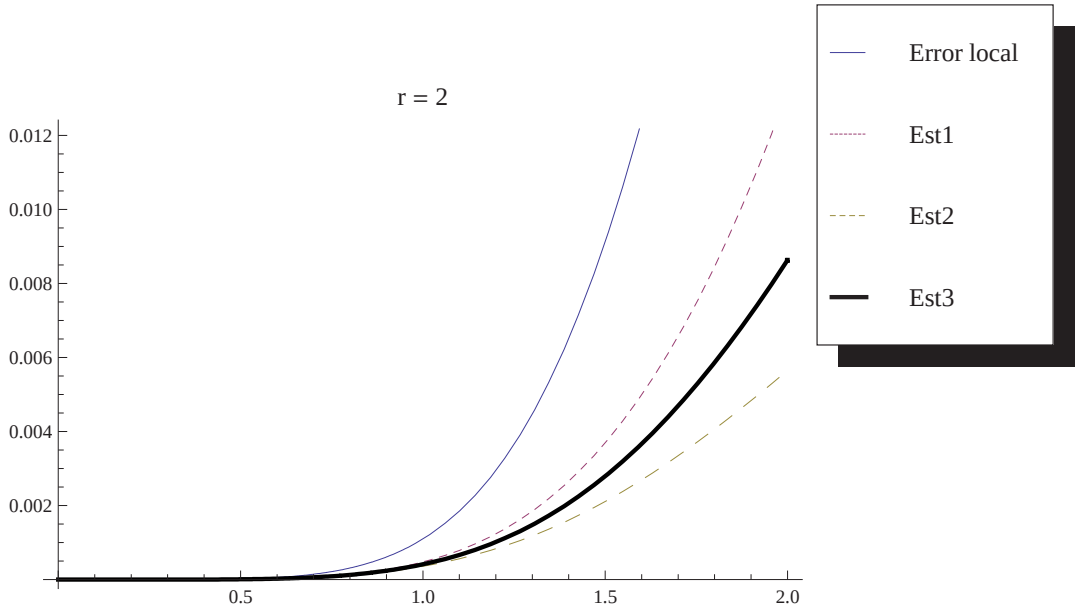


Figura 4.3.8: Gráfica de la norma de estimadores de error local para (4.3.14) con $y_0 = 1,\ y_0' = 0$.

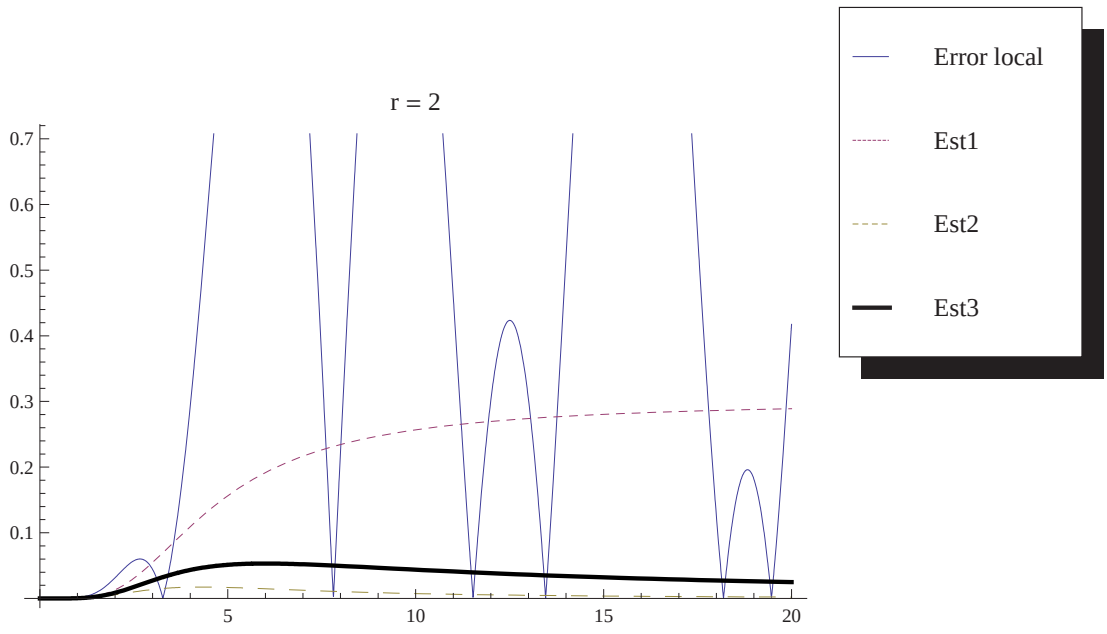


Figura 4.3.9: Gráfica de la norma de estimadores de error local para (4.3.14) con $y_0 = 1,\ y_0' = 0$.

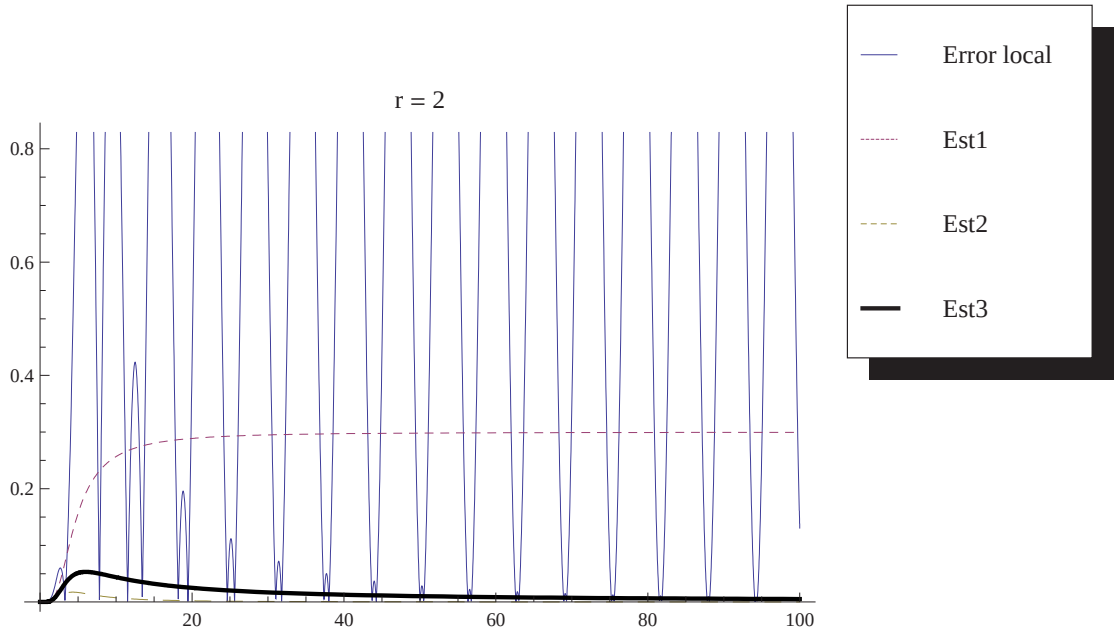


Figura 4.3.10: Gráfica de la norma de estimadores de error local para (4.3.14) con $y_0 = 0,\ y_0' = 1$.

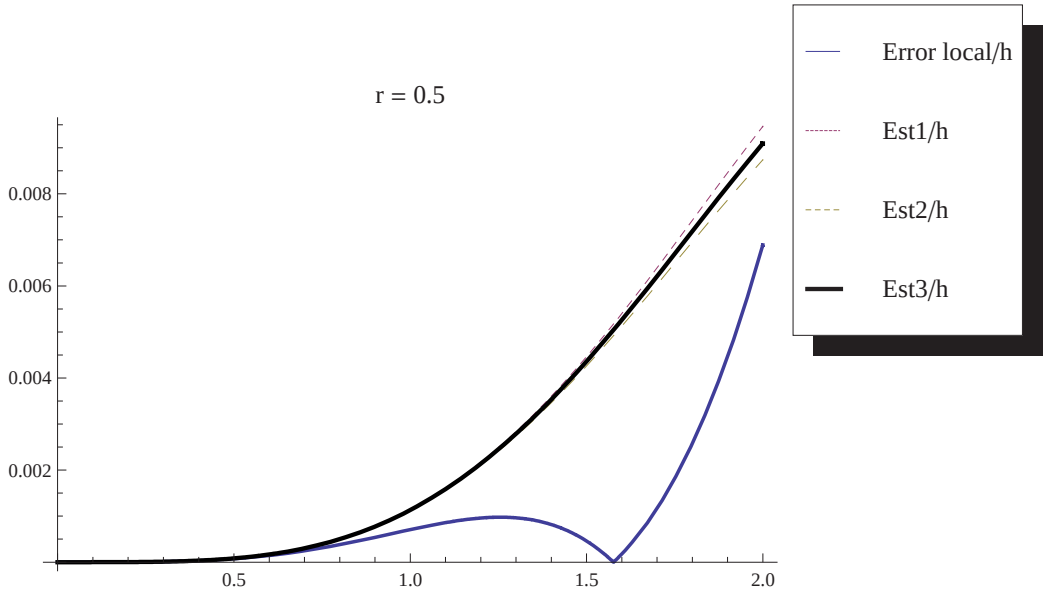


Figura 4.3.11: Gráfica de la norma de estimadores de error local para (4.3.14) con $y_0 = 0,\ y_0' = 1$.

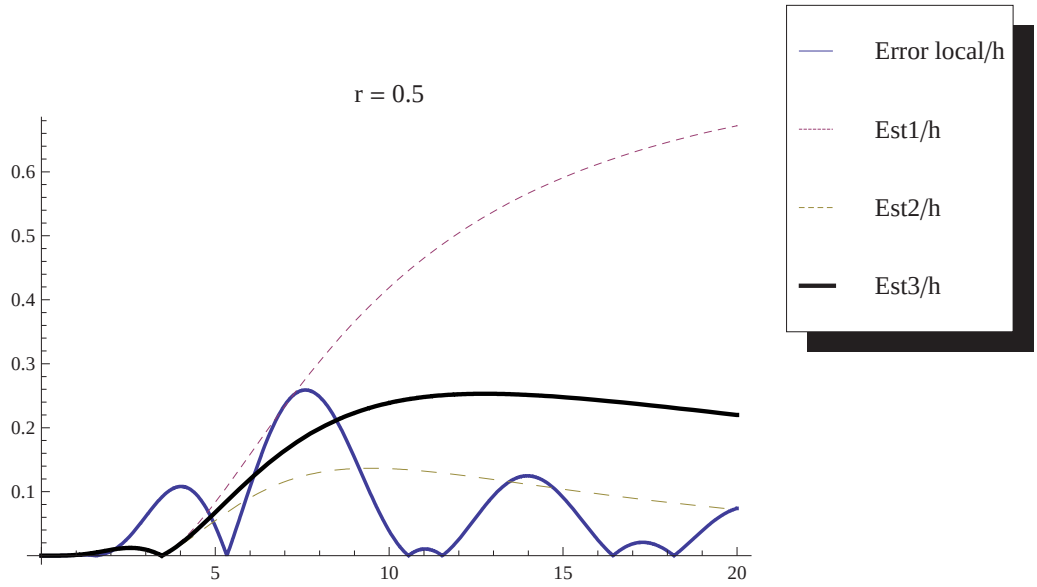


Figura 4.3.12: Gráfica de la norma de estimadores de error local para (4.3.14) con $y_0 = 0,\ y_0' = 1$.

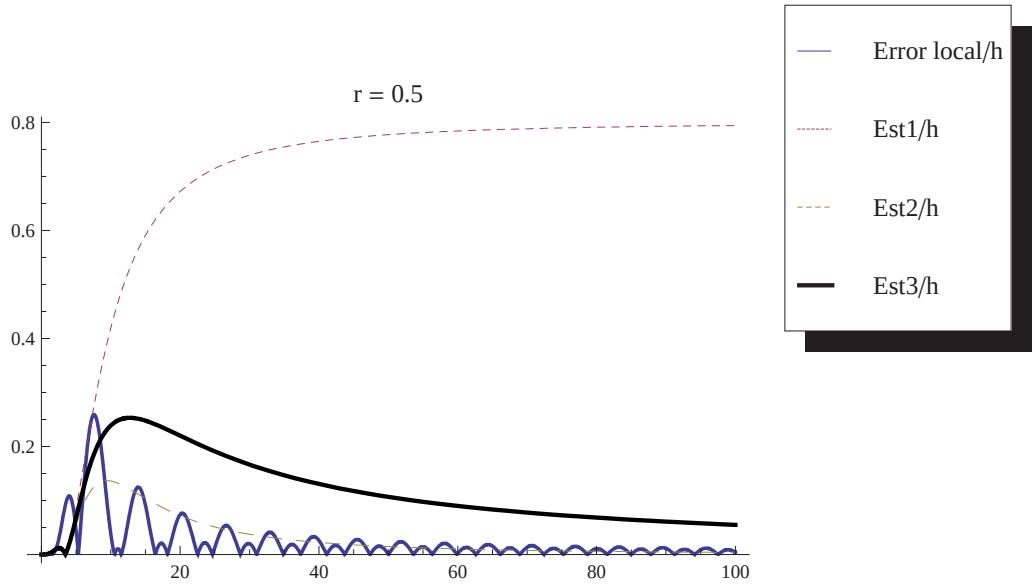


Figura 4.3.13: Gráfica de la norma de estimadores de error local para (4.3.14) con $y_0 = 0,\ y_0' = 1$.

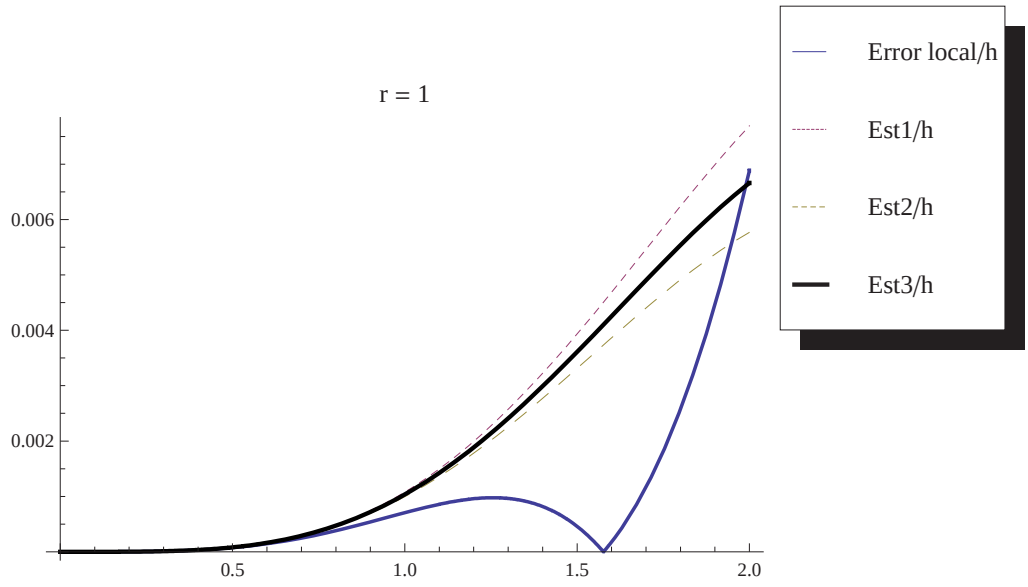


Figura 4.3.14: Gráfica de la norma de estimadores de error local para (4.3.14) con $y_0 = 0,\ y_0' = 1$.

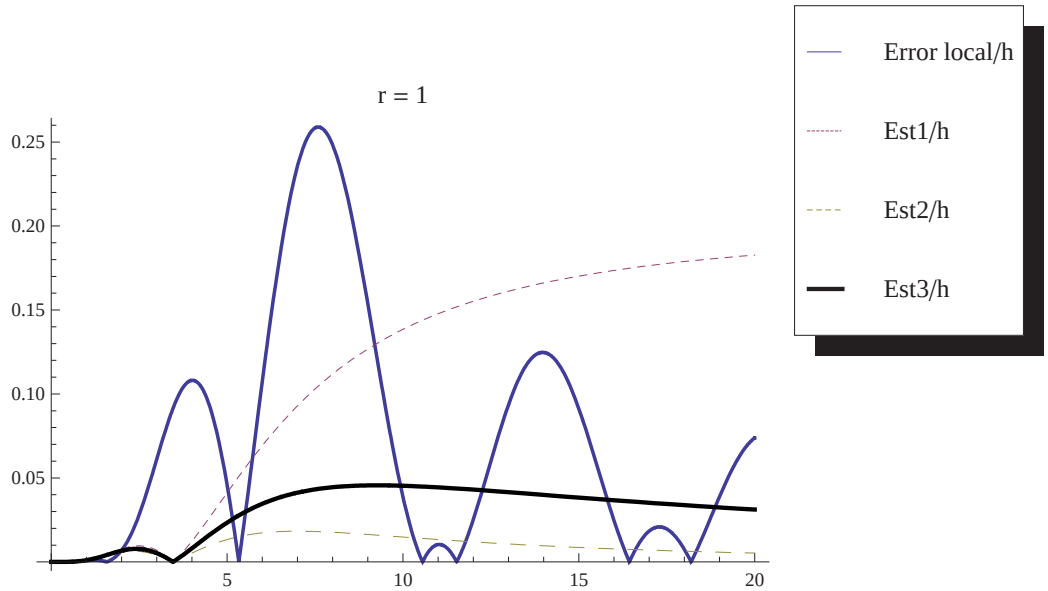


Figura 4.3.15: Gráfica de la norma de estimadores de error local para (4.3.14) con $y_0 = 0,\ y_0' = 1$.

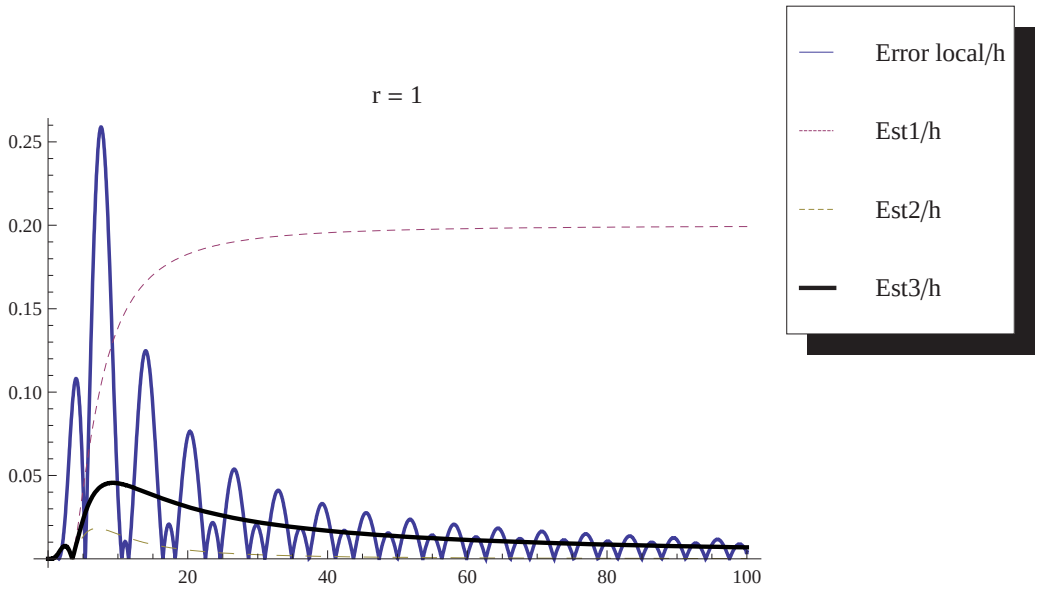


Figura 4.3.16: Gráfica de la norma de estimadores de error local para (4.3.14) con $y_0 = 0,\ y_0' = 1$.

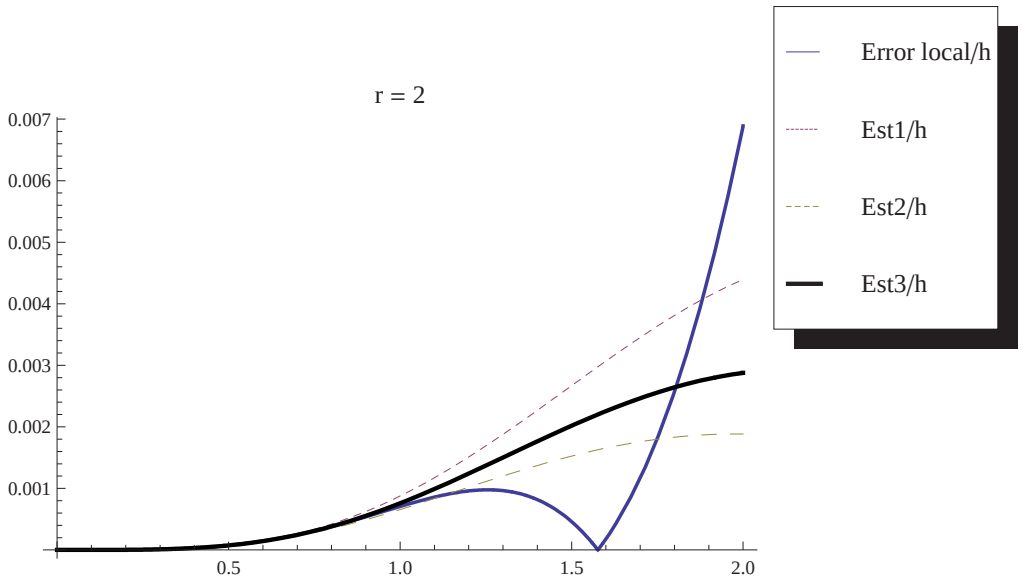


Figura 4.3.17: Gráfica de la norma de estimadores de error local para (4.3.14) con $y_0 = 0,\ y_0' = 1$.

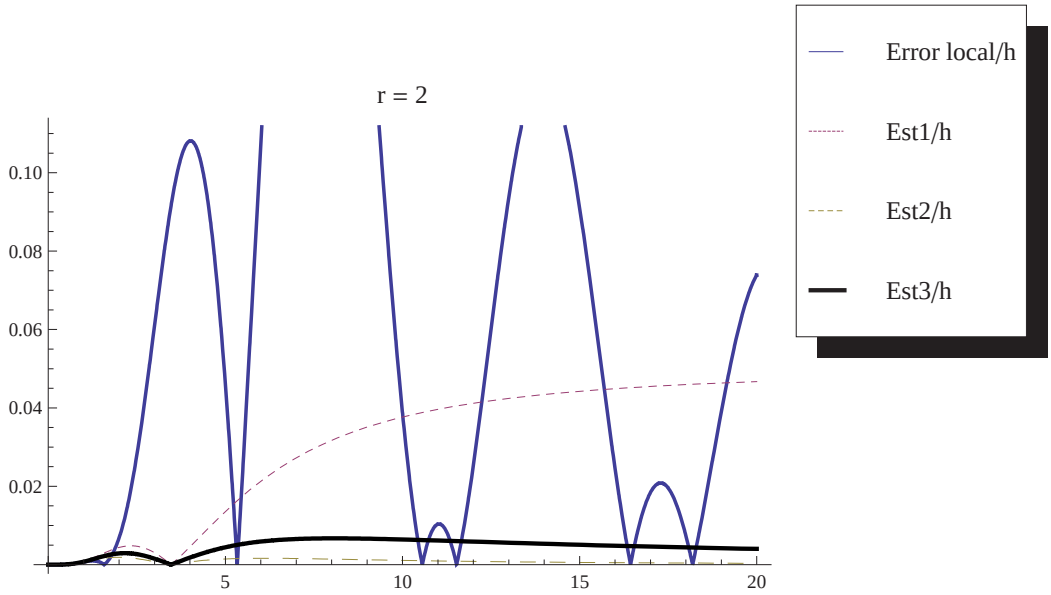
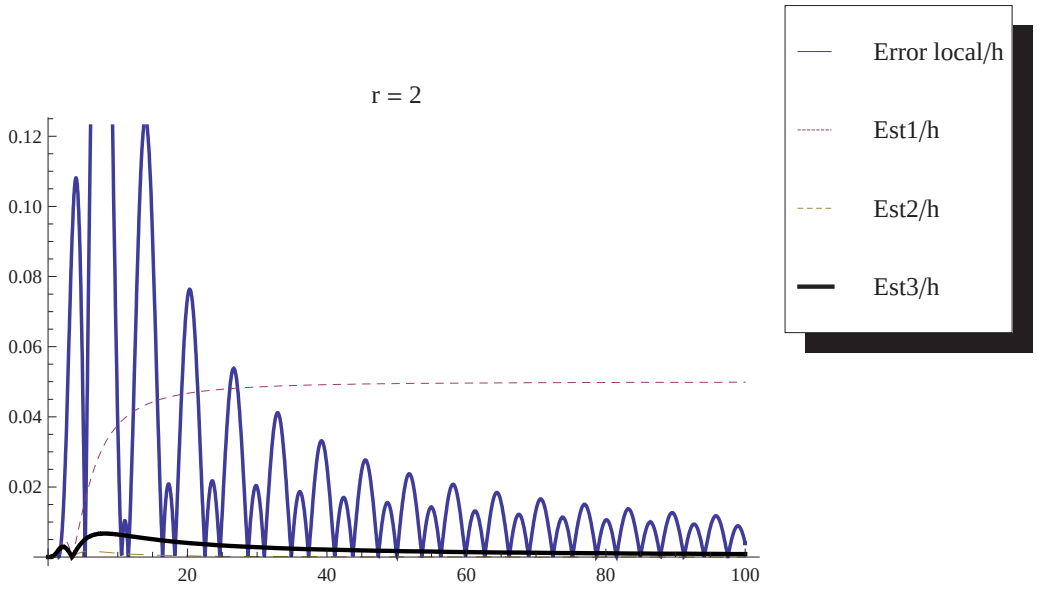


Figura 4.3.18: Gráfica de la norma de estimadores de error local para (4.3.14) con $y_0 = 0,\ y_0' = 1$.



test (4.3.14) con valores iniciales $y_0 = 1,\ y_0' = 0$. En la Figura 4.3.2 se muestran las mismas variables (divididas por h) pero cambiando las condiciones iniciales a $y_0 = 0,\ y_0' = 1$.

El uso de $\|\epsilon_2\|$ en un código tiene el efecto positivo de requerir normalmente menos pasos para completar las integraciones para una tolerancia dada, pero tiene el importante hándicap de ignorar las componentes de las altas frecuencias cuando sus amplitudes asociadas están al nivel de la tolerancia. Estas componentes serán detectadas por el estimador sólo cuando la tolerancia es disminuida con respecto a la amplitud asociada por algún factor η, típicamente, $\eta = \mathcal{O}(10^{-3})$. De lo mostrado en las Figuras que van desde la 4.3.1 hasta la 4.3.18, parece que el estimador que muestra una mejor correlación con los errores locales verdaderos es el $\|\epsilon_1\|$ para el caso $y_0 = 1,\ y_0' = 0$, pues además de ser asintóticamente correcto para el caso $z \to 0$, da también una estimación promediada de los verdaderos errores locales para el caso $z \to \infty$. En el caso de tomar valores iniciales $y_0 = 0,\ y_0' = 1$, el estimador de error local mas adecuado es el denotado como ε_3, pues además de ser asintóticamente correcto para $z \to 0$, también da una estimación casi asintóticamente correcta para el caso $z \to \infty$. En general, como la estimación del error local se va a hacer sobre la componente y solamente, y teniendo en cuenta que los problemas tendrán componente y no nula en general, parece que la opción $\|\epsilon_1\|$ es algo mas conservadora y conveniente. De todos modos la opción ε_3 es perfectamente plausible.

4.4. Selección del tamaño de paso inicial para Gauss de 2 etapas

La selección adecuada del tamaño de paso al iniciar la integración de un problema es una cuestión importante, ya que una elección ajustada del paso inicial supone que el código no rechace pasos. Además para el tipo de problemas considerado aquí las integraciones suelen realizarse con pasos casi constantes en subintervalos más o menos largos del intervalo de integración. Por lo tanto, una buena elección redunda en un número bajo de factorizaciones LU, pues el tamaño de paso no cambia en ciertos subintervalos, y esto juega un papel importante en el ahorro computacional.

Usando las condiciones de orden para métodos RKN [55, Ch. II.14], no es difícil ver que

los errores locales en un paso de tamaño h, están dados por

$$le(t_0, h) := y(t_0 + h) - y_1 = \frac{h^5}{5} \sum_{j=1}^{3} \alpha(\tau_{4j})\kappa(\tau_{4j})F(\tau_{4j}) + \mathcal{O}(h^6), \tag{4.4.1}$$

donde $\tau_{4j},\ j = 1, 2, 3$, denota el *árbol especial de Nyström* de cuarto orden (SN–trees, véase por ejemplo [55, p. 292])

$\tau_{41} =$  $\tau_{42} =$  $\tau_{43} =$

con $\alpha(\tau)$ y $F(\tau)$ denotando respectivamente el cardinal y las diferenciales elementales asociadas a los árboles, y $\kappa(\tau)$ es una aplicación real que depende de los árboles y de los coeficientes del método en cuestión (véase [55, Ch. II.14] para más detalles). De hecho,

$$\kappa(\tau) := 1 - 5\gamma(\tau)b^T\Phi(\tau),$$

donde $b^T = (1/2, 1/2)$ es el vector peso del método de Gauss de 2 etapas, $\gamma(\tau)$ es una aplicación que actúa sobre los árboles y con recorrido en $\mathcal{R}^m$, y $\Phi(\tau)$ es una aplicación asociada a los árboles y a los coeficientes del método.

Para el método de Gauss de 2 etapas se tiene que,

$$\alpha(\tau_{41}) = \alpha(\tau_{43}) = 1,\ \alpha(\tau_{42}) = 3, \quad \kappa(\tau_{41}) = \kappa(\tau_{42}) = -\frac{1}{9},\ \kappa(\tau_{43}) = \frac{1}{6}\ .$$

Por otra parte, las diferenciales elementales en notación de derivada de Frechet vienen dadas por,

$$F(\tau_{41}) = f'''[y_0', y_0', y_0'], \quad F(\tau_{42}) = f''[f(y_0), y_0'], \quad F(\tau_{43}) = f[f[y_0']].$$

La estrategia que seguimos consiste en asumir que el problema a ser integrado es autónomo y lineal, es decir,

$$f(t, y) = Jy + v, \qquad J \in \mathbb{R}^{m,m}, \quad v \in \mathbb{R}^m \text{ constantes},$$

en este caso

$$F(\tau_{41}) = 0, \quad F(\tau_{42}) = 0 \quad \text{y} \quad F(\tau_{43}) = J^2 y_0'.$$

Insertando esto en (4.4.1) se sigue que

$$le(t_0, h) = \frac{1}{720}(J^2 y_0')h^5 + \mathcal{O}(h^6). \tag{4.4.2}$$

De aquí, para lograr que $\|le(t_0,h)\| \simeq tol$, necesitamos que

$$h = \left(\frac{720 \cdot tol}{\| J^2 y_0' \|}\right)^{1/5}.$$

Para evitar posibles cancelaciones de $J^2 y_0'$ y teniendo en cuenta que los argumentos anteriores están basados sobre problemas lineales y en un desarrollo truncado del error local, preferimos usar la fórmula más conservativa

$$h_0 = \text{mín}\left\{t_f - t_0, \quad 0{,}8 \cdot \left(\frac{720 \cdot tol_0}{1+\| J^2 y_0' \|}\right)^{1/5}\right\}, \tag{4.4.3}$$

donde la tolerancia en cada paso t_n (tomando $n = 0$ para este caso) serán calculadas del modo usual

$$tol_n := atol + \| y_n \| \cdot rtol, \quad n = 0, 1, 2, \ldots, \tag{4.4.4}$$

con *atol* y *rtol* denotando las tolerancias absolutas y relativas respectivamente. Especialmente para problemas de altas dimensiones, es conveniente tomar como norma el promedio,

$$\| x \| := \left(\frac{1}{m} \| x \|_2^2\right)^{1/2}, \quad \text{si} \quad x \in \mathbb{R}^m. \tag{4.4.5}$$

Debemos resaltar que para problemas no lineales J sería reemplazado por $J_0 = \partial f/\partial y(t_0, y_0)$, y que para el cálculo de $J_0^2 y_0' = J_0(J_0 y_0')$ sólo se requieren $2m^2$ productos (m es la dimensión del problema original). Observe que el cálculo de J_0 es necesario para aplicar el proceso iterativo en el primer paso. Así, sólo $2m^2$ productos extras son necesarios para obtener una estimación aceptable (como veremos más adelante) del tamaño de paso inicial.

Otra alternativa para la elección del tamaño de paso inicial, la cual ha sido incorporada en nuestro código debido a que resulta ligeramente más efectiva que la anterior sobre problemas no lineales, es la siguiente: teniendo en cuenta que para problemas lineales (4.2.2) la cantidad

$$\alpha(\epsilon) := J_0^2 y_0',$$

satisface

$$\begin{array}{rcll} \alpha(\epsilon) & = & \epsilon^{-1}\big(f\big(t_0, y_0 + \epsilon\beta(\epsilon)\big) - f(t_0, y_0)\big), & \text{con} \\ \beta(\epsilon) & := & \epsilon^{-1}\big(f(t_0, y_0 + \epsilon y_0') - f(t_0, y_0)\big), & \epsilon \neq 0. \end{array} \tag{4.4.6}$$

Entonces, proponemos reemplazar $J_0^2 y_0'$ en (4.4.3) por $\alpha(\epsilon)$ y tomar $\epsilon \simeq \sqrt{\theta_0}$, siendo θ_0 la unidad de redondeo del ordenador. De ésta manera sólo dos evaluaciones extras de derivadas deben ser calculadas, ya que $f(t_0, y_0)$ también se necesita para los predictores al comenzar el esquema iterativo en el primer paso. En los experimentos numéricos presentados en la sección 3.6 se corroborará que esta técnica funciona bien en general.

4.5. La estimación del error global y la salida densa

A efectos de probar que los errores globales obtenidos por un método basado en la fórmula de Gauss de dos etapas en la que los tamaños de paso se controlan a través de estimadores de error local (tales como los desarrollados en las secciones previas) poseen la propiedad de *Proporcionalidad respecto a la Tolerancia*, seguiremos lo esencial del estudio realizado en [13, sec. 3–4]. Explicaremos con cierto detalle los resultados allí expuestos, y extenderemos los resultados para el interpolante continuo suministrado por la interpolación de Hermite, el cual proporciona un interpolante continuamente diferenciable sobre todo el intervalo de integración.

El error local para el método de Gauss de 2 etapas en su versión RK Nyström, esta dado mediante la expresión siguiente cuando $h \to 0$,

$$LE(t,h;y,y') := \begin{pmatrix} y(t+h;t,y,y') - y_{RK}(t+h;t,y,y') \\ y'(t+h;t,y,y') - y'_{RK}(t+h;t,y,y') \end{pmatrix} = h^5 \begin{pmatrix} \psi_1(t,y,y') \\ \psi_2(t,y,y') \end{pmatrix} + \mathcal{O}(h^6). \tag{4.5.1}$$

Por tanto un estimador de error local asintóticamente correcto para la y-componente estaría dado por $\|Est(t,h,y,y')\|$ donde,

$$Est(t,h;y,y') = h^5 \begin{pmatrix} \psi_1(t,y,y') \\ \mathbf{0} \end{pmatrix} + \mathcal{O}(h^6). \tag{4.5.2}$$

Necesitamos ahora las siguientes hipótesis para obtener nuestros resultados:

(H1) $f(t,y)$ es cuatro veces diferenciable en algún entorno de la solución exacta $\{(t,y(t)),\ t \in [t_0,t_{end}]\}$.

(H2) $\phi(t) := \|\psi_1(t,y(t),y'(t))\|$ es una función continua y positiva en el intervalo $[t_0,t_{end}]$ y continuamente Lipschitz respecto de y y y' en un entorno de ambas variables. Además se satisface la hipótesis **(H5)**, dada mas abajo.

Dada una tolerancia $\delta > 0$ suficientemente pequeña se tiene que,

(H3) El paso inicial $h_0 = t_1 - t_0$, verifica $(h_0)^5\phi(t_0) < \delta$.

(H4) El cambio de tamaño de paso se hace *esencialmente* de acuerdo con la fórmula siguiente

$$h_{n+1} = \widehat{r}_n h_n, \quad \text{con} \quad \widehat{r}_n = \begin{cases} 1 & \text{si} \quad r_n \in (\varrho_1,\varrho_2), \\ r_n & \text{en otro caso,} \end{cases} \tag{4.5.3}$$

con $r_n = \eta(\delta_n/e_n)^{1/5}$, $e_n = \|Est(t_n, h; y_n, y'_n)\|$, $\delta_n = (1 + \kappa\|y_n\|)\delta$ ($\kappa > 0$ es una constante), $\eta \in [0{,}8; 0{,}95]$ es un coeficiente de seguridad y $0 < \varrho_1 < 1 < \varrho_2$ son dos constantes prefijadas.

Nota. Naturalmente el punto final conlleva un ajuste especial, para que no sea rebasado en la integración. Hemos empleado arriba la palabra *esencialmente*, en virtud de que la ratio de tamaños de paso $\widehat{r}_n$ no se permite ser menor que un umbral tal como $r_{mín} = 0{,}2$, ni tampoco aumentar sobre alguna constante tal como $r_{máx} = 2, 3$, etc. Esta estrategia que resulta ser discontinua para los cambios de tamaño de paso, ha sido considerada a efectos de ahorrar en costos computacionales de evaluaciones de matriz Jacobiana y de factorizaciones LU. Otras estrategias mas sofisticadas para el cambio paso y que han sido incorporadas como opciones en algunos códigos tales como RADAU5, pueden verse en [52]. Nosotros en este primera versión del código hemos preferido usar la estrategia clásica indicada arriba.

(H5) $\phi(t)$ es una función estrictamente monótona en cada punto del conjunto $\mathcal{S}$, que se define a continuación. Además la función inversa $\phi^{-1}(t)$ se supone que es continuamente Lipschitz en algún entorno de cada punto $t = \phi(\xi_j)$, $\xi_j \in \mathcal{S}$. El conjunto $\mathcal{S}$ se define como,

$$\mathcal{S} = \{\xi_j,\ (j = 0, 1, \ldots, r), \xi_0 = t_0\},$$

donde ξ_j es el primer punto por la derecha de ξ_{j-1} que satisface

$$\phi(\xi_{j-1})/\phi(\xi_j) \notin \left((\varrho_1)^5, (\varrho_2)^5\right),\ j = 1, 2, \ldots, r,$$

donde ϱ_1 y ϱ_2 son dos constantes previamente dadas en **(H4)**. Naturalmente, podría darse la situación de que $\mathcal{S} = \{t_0\}$.

Asumiendo las **(H)**-hipótesis, se demuestra en [13, sect.3-4], que existe una tolerancia crítica $\delta_0 > 0$, tal que para cualquier tolerancia $0 < \delta \leq \delta_0$, no habrá pasos rechazados (el paso actual de integración $t_n \to t_n + h$ será rechazado cuando se verifique $e_n > \delta_n$) y los errores globales en cualquier punto de red $\bar{t}$ satisfacen,

$$y_\delta(\bar{t}) - y(\bar{t}) = u_1(\bar{t})\delta^{4/5} + \mathcal{O}(\delta), \quad y'_\delta(\bar{t}) - y'(\bar{t}) = u_2(\bar{t})\delta^{4/5} + \mathcal{O}(\delta), \quad \delta \to 0^+. \tag{4.5.4}$$

Aquí, $u_1(t)$ y $u_2(t)$ son funciones continuas en t que tienen derivadas continuas a trozos y además dichas funciones son independientes de la tolerancia δ. El par $(y_\delta(\bar{t}), y'_\delta(\bar{t}))$ denota

la solución numérica obtenida en el punto de red $\bar{t}$, cuando la integración se realiza con tolerancia δ.

Usando la fórmula (4.5.4), podemos realizar extrapolación global después de llevar a cabo dos integraciones numéricas con distintas tolerancias. Así considerando una primera integración con tolerancias $rtol = \delta,\ atol = \kappa\delta$ y una segunda integración con tolerancias $rtol = \tau\delta,\ atol = \kappa\tau\delta$ ($\kappa > 0$ y $\tau \neq 1$ son dos constantes) entonces en un punto de red común $\tilde{t}$ a ambas integraciones, por ejemplo el punto final, el error global puede ser estimado por la fórmula

$$\begin{aligned} \|y_\delta(\tilde{t}) - y(\tilde{t})\| &\simeq |(1-\tau^{4/5})|^{-1}\|y_\delta(\tilde{t}) - y_{\tau\delta}(\tilde{t})\|, \\ \|y'_\delta(\tilde{t}) - y'(\tilde{t})\| &\simeq |(1-\tau^{4/5})|^{-1}\|y'_\delta(\tilde{t}) - y'_{\tau\delta}(\tilde{t})\|. \end{aligned} \tag{4.5.5}$$

A efectos de suministrar una salida densa con el código, consideramos la interpolación de Hermite, ya que éste es un interpolante continuamente diferenciable en todo el intervalo de integración. Además posee cuarto orden de aproximación en la componente y, es decir el error es de tamaño $\mathcal{O}(h^4)$ sobre subintervalos de longitud h. Este error de interpolación es de magnitud similar al error global del método de Gauss de dos etapas. También tiene este interpolante la ventaja de ser fácil de implementar y ampliamente usado en la literatura.

Asumiendo que un paso desde t_n a $t_{n+1} = t_n + h_n$ ha sido completado, entonces el interpolante de Hermite está dado por,

$$H_n(t_n+\theta h_n) = y_n+\theta^2(3\theta-2)(y_{n+1}-y_n)+h_n\theta(\theta-1)\big((\theta-1)y'_n+\theta y'_{n+1}\big), \quad 0 \leq \theta \leq 1. \tag{4.5.6}$$

Computamos ahora la salida densa para ambas componentes, (y, y') mediante el polinomio interpolador de Hermite y su derivada, respectivamente. De este modo, para cada $0 \leq \theta \leq 1$ obtenemos el siguiente interpolante

$$\begin{aligned} y_{RK}(t_n + \theta h_n) = H_n(t_n + \theta h_n), \quad y'_{RK}(t_n + \theta h_n) = H'_n(t_n + \theta h_n) = \\ 6\theta(1-\theta)(h_n)^{-1}(y_{n+1} - y_n) + (\theta-1)(3\theta-1)y'_n + \theta(3\theta-2)y'_{n+1}. \end{aligned} \tag{4.5.7}$$

A continuación establecemos un teorema que nos da una estimación del error global cometido cuando usamos el interpolante de Hermite descrito anteriormente para aproximar la solución exacta del sistema diferencial sobre cualquier punto del intervalo de integración y no sólo sobre el punto final de integración.

Teorema 4.1 *Asumamos las* **(H)***-hipótesis para el PVI (4.1.1), para el estimador de error local (4.5.2) y para la estrategia de cambio de tamaño de paso. Entonces, para una tolerancia dada, el interpolante de Hermite dado por (4.5.7) satisface,*

$$\left.\begin{array}{l} y(t) - y_{RK}(t) = \mathcal{O}(\delta^{4/5}) \\ y'(t) - y'_{RK}(t) = \mathcal{O}(\delta^{3/5}) \end{array}\right\}, \quad t \in [t_0, t_{end}], \quad \delta \to 0^+. \tag{4.5.8}$$

Demostración. Dada una tolerancia cualquiera $\delta > 0$, entonces de los resultados presentados de la sección 4 del artículo [13] se deduce que la igualdad expresada en (4.5.4) es cierta sobre cualquier punto $\bar{t}$ que pertenezca a la red $\{t_n\}_{n=0}^{N_\delta}$, la cual se genera al avanzar la integración con el método dado y con cualquiera de los estimadores de error local anteriormente propuestos. También se probó en [13, Teorema 3], que los tamaños de paso $h_n = t_{n+1} - t_n$ satisfacen la relación,

$$h_n = \varphi(t_n)\delta^{1/5} + \mathcal{O}(\delta^{2/5}), \quad n = 0, 1, 2, \ldots \tag{4.5.9}$$

siendo $\varphi(t)$ una función constante a trozos definida en todo el intervalo de integración y que depende exclusivamente de $\phi(t)$ de las constantes ϱ_1 y ϱ_2 y del propio intervalo de integración.

El polinomio de Hermite que interpola la solución exacta $y(t)$ de (4.1.1) en el subintervalo $[t_n, t_n + h_n]$ esta dado por $\big($debajo $\widehat{y}_n \equiv y(t_n)$ y $\widehat{y}' \equiv y'(t_n)\big)$

$$\widehat{H}_n(t_n+\theta h_n) = \widehat{y}_n+\theta^2(3\theta-2)(\widehat{y}_{n+1}-\widehat{y}_n)+h_n\theta(\theta-1)\big((\theta-1)\widehat{y}'_n+\theta\widehat{y}'_{n+1}\big), \quad 0 \le \theta \le 1. \tag{4.5.10}$$

Luego, para $t = t_n + \theta h_n$, $0 \le \theta \le 1$, se sigue que

$$y(t) - y_{RK}(t) = \big(y(t) - \widehat{H}_n(t)\big) + \big(\widehat{H}_n(t) - H_n(t)\big). \tag{4.5.11}$$

Tomando en cuenta que el error para la interpolación de Hermite esta dado por

$$y(t) - \widehat{H}_n(t) = y[t_n, t_{n+1}, t_{n+1}, t_n, t](t-t_n)^2(t-t_{n+1})^2 = \frac{y^{(4)}(t'_n)}{4!}(t-t_n)^2(t-t_{n+1})^2, \tag{4.5.12}$$

donde $t_n < t'_n < t_{n+1}$ y $y[., ..., .]$ denota una diferencia dividida, entonces de (4.5.9) se sigue que $y(t) - \widehat{H}(t) = \mathcal{O}(\delta^{4/5})$. Adicionalmente, sustrayendo $H_n(t)$ de $\widehat{H}_n(t)$ y teniendo en cuenta (4.5.4), (4.5.10) y (4.5.9), resulta que $\widehat{H}_n(t) - H_n(t) = \mathcal{O}(\delta^{4/5})$. Esto completa la primera parte de la prueba.

A efectos de concluir la parte relativa a la aproximación de la primera derivada, tenemos en cuenta que de (4.5.11) se sigue que

$$y'(t) - y'_{RK}(t) = \big(y'(t) - \widehat{H}'_n(t)\big) + \big(\widehat{H}'_n(t) - H'_n(t)\big).$$

Usando (4.5.12) y (4.5.9), no es difícil mostrar que

$$y'(t) - \widehat{H}'_n(t) = \mathcal{O}\big((h_n)^3\big) = \mathcal{O}(\delta^{3/5}).$$

Por otra parte, de considerar la derivada en (4.5.7) y tener en cuenta (4.5.4) se deduce que

$$\widehat{H}'_n(t) - H'_n(t) = \mathcal{O}(\delta^{4/5}).$$

Esto completa la prueba.

□

El teorema nos dice que en la componente y, el error global para el interpolante de Hermite $y_{RK}(t)$ es de tamaño similar al error global del método sobre los puntos de red. También el error del interpolante es similar al error global del método para las componentes $h_n y'(t_n + \theta h_n)$, pero el error del interpolante es algo mayor que el error global del método al aproximar las componentes de la primera derivada.

Este teorema también permite hacer extrapolación global para estimar los errores globales sobre cualquier punto de integración t. Esto se haría mediante la técnica de extrapolación global descrita anteriormente. Esto es, se realizan dos integraciones con tolerancias distintas hasta el punto final de integración y entonces para cualquier punto t, mediante la fórmula (4.5.5), se obtiene un estimador asintóticamente correcto para la componente y, mientras que para la estimación del error en la componente y' debe reemplazarse en la segunda ecuación de (4.5.5), la cantidad $\tau^{4/5}$ por $\tau^{3/5}$, para todos aquellos puntos que sean no comunes a ambas integraciones.

4.6. Código a paso variable

En esta sección se describirá un pseudo código para el método de Gauss de dos etapas para la integración de problemas de tipo (4.1.1), El código está basado en los estudios llevados a cabo anteriormente, en los resultados presentados en [46] y en los códigos y artículos de investigación de distintos autores, [5, 6, 53, 55, 57, 63, 77, 83]. En su diseño se intenta ahorrar, tanto como sea posible, en evaluaciones de Jacobianos y en factorizaciones LU. Este hecho es extremadamente importante para problemas de dimensión media–alta. Hemos integrado muchos problemas propuestos en la literatura, en particular todos los propuestos en [33, 37, 46, 55, 56, 57, 62, 63], y el código funciona adecuadamente con los parámetros

(n_j, θ_j) indicados a continuación (θ_0 denota la unidad de redondeo del ordenador).

$$\begin{array}{lll} n_1 = 10, & n_2 = 6, & n_3 = n_2 - 2, \\ \theta_1 = 0{,}8, & \theta_2 = 1{,}5, & \theta_3 = 0{,}85, \\ \theta_4 = 2, & \theta_5 = 0{,}01, & \theta_6 = 0{,}6, \\ \theta_7 = 0{,}7, & \theta_8 = 0{,}2. & \end{array} \tag{4.6.1}$$

En el desarrollo del código hemos seguido las siguientes pautas:

(I) El usuario debe indicar si el problema es lineal ($ilin = 1$), es decir, de tipo (4.2.2), o no lineal ($ilin = 0$). Esto es importante a la hora del ahorro en las actualizaciones de la matriz Jacobiana y en evaluaciones de la función derivada segunda.

(II) Para problemas lineales la matriz Jacobiana se evalúa sólo una vez en el primer paso t_0. Para problemas no lineales ésta se actualiza cada vez que el esquema iterativo toma más de n_2 iteraciones para alcanzar convergencia (convergencia lenta), o bien la convergencia no se alcanza en n_1 iteraciones, y por tanto hay que reducir el tamaño de paso y hacer un nuevo intento.

A continuación asumiremos que el paso de t_{n-1} a $t_n = t_{n-1} + h_{n-1}$ con tamaño de paso h_{n-1} ha sido completado, es decir, para algún estimador del error local tenemos que

$$est(t_{n-1}, h_{n-1}) \leq tol_{n-1} := atol + \parallel y_{n-1} \parallel \cdot rtol.$$

Nos proponemos ahora dar una estimación h_n, para el paso siguiente (de t_n a $t_{n+1} = t_n + h_n$) y ver cuando éste es finalmente aceptado.

(III) Calculamos

$$\delta_{n-1} := \text{mín} \left\{ \theta_4, \theta_1 \left(\frac{tol_{n-1}}{\theta_0 + est(t_{n-1}, h_{n-1})} \right)^{1/5} \right\}. \tag{4.6.2}$$

Hacemos ahora la predicción siguiente:

1) Si el tamaño de paso no fue rechazado (por el estimador del error local o por el esquema iterativo) en el paso previo aceptado (de tamaño h_{n-1}), y la convergencia de las iteraciones no fue lenta (no se necesitan más de n_2 iteraciones), entonces tomamos

$$h_n = \begin{cases} h_{n-1}, & \text{si} \quad \theta_3 \leq \delta_{n-1} \leq \theta_2 \\ \delta_{n-1} h_{n-1}, & \text{en otro caso.} \end{cases}$$

Esta técnica está pensada para el ahorro en factorizaciones LU. Está claro, que el punto final conlleva un ajuste especial.

2) Cuando hubo algún rechazo en el paso previo de $t_{n-1} \to t_n$, debido al estimador del error local o por la falta de convergencia en la iteración, entonces proponemos para el paso actual la predicción $h_n = \delta^* h_{n-1}$ (con $\delta^* = \text{mín}\{1, \delta_{n-1}\}$), siempre que la convergencia con el tamaño de paso exitoso h_{n-1} no fuera lenta (más de n_2 iteraciones). En el caso de convergencia lenta para h_{n-1}, aparte de actualizarse la matriz Jacobiana (véase (II)), el tamaño de paso propuesto vendrá dado por la fórmula

$$h_n = \big(\text{mín}\{\delta_{n-1}, r^*_{n-1}, \theta_4\}\big)h_{n-1}, \quad \text{donde} \quad r^*_{n-1} = \left(\frac{\theta_3\theta_5 tol_{n-1}}{q_{n-1,n_3+1}}\right)^{1/(2n_3-2)}, \tag{4.6.3}$$

con $q_{j,\nu}$ definido como se indica abajo en (4.6.4). Con ésta predicción para h_n se espera que el esquema iterativo converja en n_3 iteraciones a lo sumo.

(IV) Cuando ocurre un rechazo por el estimador de error local en el paso propuesto $t_n \to t_{n+1}$, es decir, $est(t_n, h_n) > tol_n$, entonces la nueva predicción para el tamaño de paso es $h_{nuevo} = \big(\text{máx}\{\theta_8, \delta_n\}\big)h_{viejo}$, donde δ_n es calculado de (4.6.2) con el subíndice n reemplazando al subíndice $n-1$ y tomando h_{viejo} en vez de h_n.

(V) La convergencia de la iteración en el paso $t_n \to t_{n+1}$ (con tamaño de paso h_n) es alcanzada en la primera iteración $Y_n^{(\nu)}$ tal que

$$q_{n,\nu} := \| Y_n^{(\nu)} - Y_n^{(\nu-1)} \| \leq \theta_5 tol_n, \quad \nu = 1, 2, \ldots, n_1. \tag{4.6.4}$$

Diremos que el esquema iterativo no converge cuando la inecuación anterior no se satisface tras n_1 iteraciones o bien cuando

$$\tau_{n,\nu} := \frac{q_{n,\nu}}{q_{n,\nu-1}} > s_n := \text{máx}\left\{\theta_6,\ \left(\frac{\theta_1\theta_5 tol_n}{q_{n,1}}\right)^{1/(n_1-1)}\right\}, \tag{4.6.5}$$

para algún $\nu = 2, 3, \ldots, n_1$. Esta última inecuación indica que no es probable que la convergencia sea alcanzada en n_1 iteraciones. En caso, que se detecte falta de convergencia tras ν iteraciones, entonces se reduce el tamaño de paso a través de la fórmula

$$h_{nuevo} = \delta' h_{viejo}, \quad \delta' = \text{máx}\left\{\theta_7\sqrt{\frac{s_n}{\tau_{n,\nu}}},\ \theta_8\right\}. \tag{4.6.6}$$

Esta reducción del tamaño de paso está basada en el hecho que para problemas lineales, la razón de convergencia satisface $\tau_{n,\nu} = \mathcal{O}(h_n)^2$ para las frecuencias bajas, las cuales asumimos que son dominantes.

(VI) La factorización LU es actualizada cuando o bien la matriz Jacobiana ha sido computada en el paso actual o bien cuando el tamaño de paso ha sido cambiado, véase (II)–(V).

Algoritmo. Gauss–2etapas $(t_0, t_f, y_0, y_0', atol, rtol, nmax, ier)$

1. *Comentario:* t_0 y $t_f > t_0$ denotan el punto inicial y el punto final de integración respectivamente. y_0 y y_0' son vectores donde se almacenan las condiciones iniciales (en entrada) y la solución numérica en el punto final (en salida). $atol$ y $rtol$ denotan las tolerancias absolutas y relativas respectivamente, y $nmax$ es el número máximo de pasos permitidos en la integración. ier es un código de error (error flag), $ier = 0$ indica que la integración fue exitosa.

   Se supone que la segunda derivada $f(t, y)$ y la matriz Jacobiana $f'(t, y) = \frac{\partial f}{\partial y}(t, y)$ están disponibles. Aunque muchas de las variables auxiliares indicadas más abajo están claras por el contexto y por los comentarios previos, especificaremos sin embargo algunas de ellas, las cuales reflejan mejor el flujo del algoritmo. $nstep$ es el número de pasos exitosos; $ntrial$ cuenta el número de pasos intentados; $ijac = 0, 1$ ($ijac = 1$ en los pasos cuando la matriz Jacobiana ha sido recientemente actualizada y $ijac = 0$ en otro caso); $h_{\text{mín}}$ es el tamaño de paso mínimo permitido en sentido relativo ($h_{\text{mín}} := 10 \cdot \theta_0$ podría ser adecuado). h_0 es el tamaño de paso previo aceptado; h es el nuevo tamaño de paso propuesto; y, h' almacena el tamaño de paso de la última factorización LU llevada a cabo. También hemos considerado algunos contadores locales (para el paso actual t) tal como $idiv$ para el número de veces que la divergencia es detectada en el esquema iterativo e $iest$ para el número de rechazos por el error del estimador local.

   El algoritmo dado aquí es para problemas no lineales en general, es decir, $ilin = 0$. Sólo algunas modificaciones menores son necesarias para el caso más simple $ilin = 1$. La primera versión (2006) de un código en FORTRAN 90 para este algoritmo puede ser bajada de [50], y la última versión (2008) de [47].

2. Para $nstep = 0;\ ntrial = 0;\ f_0 = f(t_0, y_0);\ J = f'(t_0, y_0);\ ijac = 1$ y $ier = 0$.

3. Computar $tol = atol + \| y_0 \| rtol$; $idiv = 0$ y $iest = 0$.

4. Si $[ntrail = 0]$ entonces [calcule h_0 de (4.4.3); $h = h_0$ y $h' = h$].

5. Asignar $ntrail = ntrail + 1$.

6. Si $[ntrail > nmax]$ entonces [$ier = -3$ y parar por integración no exitosa].

7. Si $[h < h_{\text{mín}} \cdot \text{máx}\{1, |t|\}]$ entonces [$ier = -2$ y parar por integración no exitosa].

8. Selecciona los predictores para las etapas internas como se indica en la sección 3.2.

9. Si $[ijac = 1$ ó $|h'/h - 1| > 0{,}08]$ entonces [hacer $LU = (\gamma h^2)^{-1} I - J$; Si$[((\gamma h^2)^{-1} I - J)$ es numéricamente singular] entonces [$h = h/2$ e ir a (5)]; tomar $h' = h$].

10. Hacer para $\nu = 1, 2, \ldots, n_1$ (para obtener convergencia de la iteración (4.2.4)–(4.2.5)–(4.2.7))

    a) Calcule $Y_n^{(\nu)}$ y $q_{n,\nu}$.

    b) Si $[\nu = 1]$ entonces $[r^* = \theta_4]$.

    c) Si $[q_{n,\nu} \leq \theta_5 tol$, véase (4.6.4)] entonces [calcule y_1, y_1' de (4.1.3); evalúe $f_1 = f(t + h, y_1)$; guarde Y_n^{ν}, ν y r^* e ir a (11)].

    d) Si $[\nu = n_3 + 1]$ entonces [tomar $r^* = r_n^*$ de acuerdo con (4.6.3)].

    e) Si $[\nu > 1$ y $\tau_{n,\nu} > s_n$, véase (V)] entonces [$h = \delta'\, h$, véase (4.6.6); $idiv = idiv + 1$;

    Si $[ijac = 0]$ entonces $[J = f'(t_0, y_0);\ ijac = 1]$;

    ir a (5)].

    f) If$[\nu = n_1]$ entonces [$idiv = idiv + 1$; $h = r^* h$;

    Si $[ijac = 0]$ entonces $[J = f'(t_0, y_0);\ ijac = 1]$;

    ir a (5)].

    g) Fin del hacer.

11. Tomar $r = h'/h$ y calcule $est = \| \epsilon_1(t, h, h') \|$, véase (4.3.11).

12. Si $[est > tol]$ entonces $[\delta = \text{máx}\{\theta_8, \theta_1 (tol/est)^{1/5}\}$; $iest = iest + 1$; $h = \delta h$;

    Si $[iest = 2$ y $ijac = 0]$ entonces $[J = f'(t_0, y_0);\ ijac = 1]$;

ir a (5)].

13. Actualizar, $t_0 = t_0 + h;\ nstep = nstep + 1;\ h_0 = h;\ y_0 = y_1;\ y_0' = y_1'$ y $f_0 = f_1$.

14. Si $[|t_f - t_0| \leq h_{\text{mín}} \cdot \text{máx}\{1, |t_f|\}]$ entonces [*pare* y *salga* (por integración exitosa)].

15. Si $[\nu > n_2]$ entonces $[J = f'(t_0, y_0)$ y $ijac = 1]$ en otro caso $[ijac = 0]$.

16. Calcule, $r = \text{mín}\{\theta_4, \theta_1(tol/(h_{\text{mín}} + est))^{1/5}\}$ y $r' = (t_f - t_0)/h_0$.

17. Si $[r' \leq 1{,}2r]$ entonces $[r = r';\ h = rh_0$ e ir a (3)].

18. Si $[idiv > 0$ ó $iest > 0]$ entonces $[r = \text{mín}\{1, r\}]$.

19. Si $[\nu > n_2]$ entonces $[r = \text{mín}\{r^*, r\};\ r = \text{máx}\{\theta_8, r\}]$.

20. Si $[(\theta_3 \leq r \leq \theta_2)$ y $(ijac = 0)]$ entonces $[h = h_0]$; en otro caso $[h = rh_0]$.

21. Ir a (3).

22. Fin.

En la primera versión del código mostrado en [50], hemos seleccionado como estimador del error local $est = \| \epsilon_2(t, h, h') \|$ dado en (4.3.12), pero las elecciones $est = \varepsilon_3$ (véase (4.3.13)) o $est = \| \epsilon_1(t, h, h') \|$ dado en (4.3.11) parecen más eficientes, tal como se sugiere en [47].

Aunque el algoritmo fue diseñado para emplear descomposiciones LU en la solución de los sistemas lineales asociados a las ecuaciones de etapas (4.2.5), sólo serían necesarias modificaciones menores cuando se transforma la matriz Jacobiana J a la forma de Hessemberg $S^{-1}JS = H$, siendo S una matriz triangular y H una matriz de Hessemberg, como se propone en [26, 83] y en [57, p. 122]. Estas transformaciones de Hessemberg pueden ser ventajosas cuando tratamos con problemas de altas dimensiones. Para problemas del tipo

$$My''(t) = f\big(t, y(t)\big), \quad t \in [t_0, t_{fin}],$$
$$y(t_0) = y_0, \quad y'(t_0) = y_0', \quad y, y', f \in \mathbb{R}^m, \quad M \in \mathbb{R}^{m,m},$$

donde M es una matriz masa constante, el algoritmo Gauss2 puede adaptarse fácilmente de modo que se evite el cálculo de la inversa de M.

4.7. Experimentos numéricos

Hemos seleccionado seis problemas oscilatorios, dos lineales y cuatro no lineales, de dimensiones baja y media, donde tres de ellos provienen de EDOs y los otros tres de EDPs. Estos problemas han sido considerados como problemas test en la literatura. Nuestro objetivo en esta sección es doble: primero, evaluar nuestro código comparándolo con ciertos integradores de carácter general que son bien conocidos, y en segundo lugar, tasar el comportamiento de nuestro código cuando se usa la salida continua. En este último caso representaremos en gráficas la evolución de los errores globales de las soluciones numéricas y de los invariantes cuando proceda (en el caso de sistemas Hamiltonianos). Para el primer objetivo consideraremos los siguientes códigos:

(1) ODEX2 (véase [53, 55]), es un código de reconocido prestigio, específico para problemas de segundo orden del tipo (4.1.1). Está basado en extrapolación local de la regla de Störmer (una fórmula explícita de segundo orden bien conocida). El código es especialmente apto para integraciones que requieren una precisión alta.

(2) VODE–BDF (véase [5, 6]). Este es un integrador para PVI de primer orden. Hemos elegido la opción basada en las fórmulas implícitas BDF (Backward Differentiation Formulae) de Gear, la cual es la opción prevista por el código para problemas de tipo Stiff. Este código es uno de los integradores más populares de problemas de tipo Stiff cuando se requieren precisiones bajas y medias.

(3) RADAU5 (véase [53, 55]), es un integrador muy robusto para problemas Stiff de primer orden, que está basado en la fórmula Runge–Kutta Radau IIA de tres etapas. Ya que este método implícito es de orden cinco, proporciona integraciones satisfactorias en precisiones medias y medio–altas.

Nuestro código ha sido implementado en dos versiones, los cuales sólo difieren en el estimador de error local considerado. GAUSS2–1 se refiere al caso en el cual se emplea el estimador local $\|\epsilon_1\|$ dado en (4.3.11), mientras que GAUSS2–3 denota el caso en que se implementa el estimador del error local ε_3 en (4.3.13). La versión de GAUSS2 en [47] permite elegir entre los tres distintos estimadores del error local mencionados anteriormente, $\|\epsilon_1\|$, $\|\epsilon_2\|$ y ε_3, así como la opción de salida densa.

Ya que nuestro código sólo estima el error local en la componente y, en los resultados que se muestran en las tablas abajo, hemos ejecutado los otros códigos omitiendo también el

control del error local en la componente y'. Cabe mencionar que los resultados no cambian significativamente para los códigos ODEX2, RADAU5 y VODE cuando se requieren precisiones locales similares en las componentes y e y'. Para el código VODE-BDF, con las opciones de orden máximo limitado a tres, cuatro o cinco (la última opción es la que suministra el código por defecto) para las fórmulas BDF, encontramos algunos problemas cuando integramos los sistemas de segundo orden provenientes de las EDP (Ecuaciones en Derivadas Parciales) propuestas. En este caso, para hacer el código VODE-BDF más robusto, hemos manejado dos opciones: la primera consiste en reducir el orden máximo de las fórmulas BDF a cuatro, para ganar estabilidad en las fórmulas (se denotará esta opción por VODE–BDF4); y en segundo lugar, reducimos drásticamente el orden máximo de las fórmulas BDF usadas a dos, de tal manera que las fórmulas sean A–estables. Esta opción se denota por VODE–BDF2.

Los códigos fueron ejecutados con los parámetros proporcionados por defecto por los códigos, excepto para RADAU5, donde fue seleccionado la opción que evita doblar la dimensión del problema original, véase [53]. También hemos aprovechado la estructura de banda de la matriz Jacobiana en el problema (4.7) descrito abajo, la cual permite reducir costos en la resolución de sistemas lineales y ahorrar en almacenamiento de datos. Esta opción está disponible para los códigos RADAU5 (aplicado a problemas de primer y segundo orden) y GAUSS2.

- Problema 4.1. [1, p. 932], [27, p. 17] Es un modelo lineal que surge en estudios de vibraciones en Ingeniería. El sistema viene dado con condiciones iniciales homogéneas $y(0) = y'(0) = 0$, y un término no homogéneo $f(t)$ que conlleva discontinuidades en la tercera derivada de la solución, en los puntos $t = 5$ y $t = 10$,

$$y'' + \begin{pmatrix} 301 & -100 & -200 \\ -100 & 10102 & -10000 \\ -200 & -10000 & 10102 \end{pmatrix} y = \begin{pmatrix} 0 \\ 0 \\ f(t) \end{pmatrix}, \; f(t) = \begin{cases} 2t, & t \in [0,5] \\ 20 - 2t, & t \in (5,10] \\ 0, & t \in (10,40]. \end{cases}$$

  Su solución combina tres componentes con frecuencias $\omega = 1{,}00,\ 21{,}2,\ 142.$, las cuales no sufren excitación por el término no homogéneo f. La amplitud asociada a cada frecuencia tiene tamaño $\alpha = \mathcal{O}(\omega^{-3/2})$. Por tanto, las frecuencias más altas son menos dominantes.

- Problema 4.2 [56, pp. 17–18]. Este es un problema hamiltoniano altamente oscilatorio, considerado por Fermi–Pasta–Ulam [30] (este problema está descrito en mayor detalle en Apéndice B, como problema B.30),

$$\begin{aligned}
y_1'' &= (y_2 - y_5 - y_1 - y_4)^3 - (y_1 - y_4)^3, && y_1(0) = 1, && y_1'(0) = 1\\
y_2'' &= -(y_2 - y_5 - y_1 - y_4)^3 + (y_3 - y_6 - y_2 - y_5)^3, && y_2(0) = 0, && y_2'(0) = 0\\
y_3'' &= -(y_3 - y_6 - y_2 - y_5)^3 - (y_3 + y_6)^3, && y_3(0) = 0, && y_3'(0) = 0\\
y_4'' &= (y_2 - y_5 - y_1 - y_4)^3 + (y_1 - y_4)^3 - \omega^2 y_4, && y_4(0) = \omega^{-1}, && y_4'(0) = 1\\
y_5'' &= (y_2 - y_5 - y_1 - y_4)^3 + (y_3 - y_6 - y_2 - y_5)^3 - \omega^2 y_5, && y_5(0) = 0, && y_5'(0) = 0\\
y_6'' &= (y_3 - y_6 - y_2 - y_5)^3 - (y_3 + y_6)^3 - \omega^2 y_6, && y_6(0) = 0, && y_6'(0) = 0.
\end{aligned}$$

  Hemos tomado $\omega = 50$ como recomiendan Hairer, Lubich y Wanner en [56, pp. 17–18]. La solución en $t = 100$, con 5 dígitos significativos para la componente y es,

$$\begin{gathered}
y_1 = -0{,}76557, \qquad y_2 = 0{,}22667, \qquad y_3 = -0{,}25092,\\
y_4 = 0{,}91662 \cdot 10^{-2}, \quad y_5 = -0{,}53880 \cdot 10^{-2}, \quad y_6 = -0{,}18555 \cdot 10^{-1}.
\end{gathered}$$

  Aunque este problema es no lineal, a efectos de tener una ligera idea de sus modos de frecuencia, hemos calculado los seis autovalores de la matriz Jacobiana evaluada en el punto inicial $t = 0$, obteniendo que,

$$\lambda_1 \simeq \lambda_2 \simeq \lambda_3 = -2{,}5 \cdot 10^3 < \lambda_4 \simeq -8{,}0 < \lambda_5 \simeq -1{,}1 < \lambda_6 = 0. \tag{4.7.1}$$

- Problema 4.3. Es una ecuación diferencial parcial que describe la vibración de una barra empotrada [62, 81]. Hemos tomado las condiciones iniciales y de frontera dadas en [62, pp. 426–427] (véase también Apéndice B, problema B.16),

$$\begin{cases}
y_{tt}(x,t) + 200 y_{xxxx}(x,t) = 0, \quad 0 < x < l = 22, \quad t > 0,\\
y(x,0) = f(x), \quad y_t(x,0) = 0,\\
y(0,t) = y_x(0,t) = 0,\\
y_{xx}(l,t) = y_{xxx}(l,t) = 0,
\end{cases} \tag{4.7.2}$$

  donde

$$\begin{gathered}
f(x) = 0{,}1\big(\cosh(\lambda x) - \cos(\lambda x) - K(\sinh(\lambda x) - \sin(\lambda x)\big),\\
K = \big(\sinh(\lambda l) + \sin(\lambda l)\big)^{-1}\big(\cosh(\lambda l) + \cos(\lambda l)\big) \quad \text{y} \quad \lambda = 0{,}08523200128726258.
\end{gathered}$$

La solución exacta está dada por

$$y(x,t) = f(x)\cos(\omega t),\ \text{con}\ \omega = 0{,}102735464\ldots.$$

Definiendo la red equidistante $x_i := i\Delta x$, $\Delta x = 22/N$, $(i = 1,\ldots,N)$ y usando diferencias centrales de segundo orden para la derivada cuarta,

$$\frac{\partial^4 y_i}{\partial x^4} \simeq \frac{y_{i+2} - 4y_{i+1} + 6y_i - 4y_{i-1} + y_{i-2}}{(\Delta x)^4}, \quad i = 1,\ldots,N, \tag{4.7.3}$$

obtenemos el sistema diferencial de segundo orden [62, p. 427],

$$\begin{pmatrix} y_1'' \\ y_2'' \\ y_3'' \\ \vdots \\ y_{N-2}'' \\ y_{N-1}'' \\ y_N'' \end{pmatrix} = -200(\Delta x)^{-4} \begin{pmatrix} 7 & -4 & 1 & & & & \\ -4 & 6 & -4 & 1 & & 0 & \\ 1 & -4 & 6 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 6 & -4 & 1 \\ & 0 & & 1 & -4 & 5 & -2 \\ & & & & 2 & -4 & 2 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{N-2} \\ y_{N-1} \\ y_N \end{pmatrix}. \tag{4.7.4}$$

Para $N = 90$ y $t \in [0, 10^3]$, cinco dígitos significativos se preservan en la solución del sistema diferencial (4.7.4) con relación a la solución de la PDE (4.7.2). Ya que en la práctica estaremos interesados en la solución de la PDE, entonces, la integración del problema (4.7.4) será sólo de interés para tolerancias no demasiado pequeñas. En este caso las frecuencias del sistema diferencial se encuentran dispersas en el intervalo [0,1; 947], alcanzando los extremos del intervalo.

- Problema 4.4. Es la ecuación diferencial parcial no lineal [13], dada también en el Apéndice B como problema B.17,

$$\begin{cases} y_{tt}(x,t) + \big(1 + y(x,t)^2\big) y_{xxxx}(x,t) = \eta^4 y(x,t)^3, \quad -\pi/\eta < x \le \pi/\eta,\ t > 0, \\ y(x,0) = \cos \eta x, \\ y_t(x,0) = 0, \quad \eta = 1/4, \end{cases} \tag{4.7.5}$$

con solución $2\pi/\eta$–periódica en el espacio, es decir,

$$y(x \pm 2\pi/\eta, t) = y(x,t), \quad \forall x, t \in \mathbb{R}.$$

La solución exacta está dada por $y(x,t) = \cos(\eta x)\cos(\eta^2 t)$. Discretizando en el espacio como en el problema 4.3 y considerando las condiciones de periodicidad $2\pi/\eta$,

$$y_0(t) = y_N(t), \quad y_{-1}(t) = y_{N-1}(t), \quad y_{N+1}(t) = y_1(t), \quad y_{N+2}(t) = y_2(t). \tag{4.7.6}$$

Obtenemos el problema de valor inicial de segundo orden,

$$\left.\begin{array}{c} y_i''(t) + (\Delta x)^{-4}(1+y_i^2)(y_{i-2} - 4y_{i-1} + 6y_i - 4y_{i+1} + y_{i+2}) = \eta^4 y_i^3 + r_i(t), \\ \Delta x = 2\pi\eta^{-1}N^{-1}, \quad y_i(0) = \cos(\eta x_i), \quad y_i'(0) = 0, \quad x_i = -\eta^{-1}\pi + i\Delta x, \\ i = 1, 2, \ldots, N. \end{array}\right\} \tag{4.7.7}$$

Los términos forzados

$$r_i(t) = \alpha(1+\varepsilon_i(t)^2)\varepsilon_i(t), \quad \varepsilon_i(t) = \cos(\eta x_i)\cos(\eta^2 t)$$

con

$$\alpha := 8^{-1}\sum_{k=1}^{\infty} \frac{(1-4^{-k-1})}{(-4)^k}\frac{(\Delta x)^{2k}}{(2k+4)!},$$

han sido añadidos al lado derecho de la EDO (4.7.7), de modo que la EDO y la PDE (4.7.5) compartan la misma solución sobre las correspondientes líneas x_i. Este es un problema no lineal de dimensión medio-alta donde las frecuencias más altas son menos significativas.

- Problema 4.5. Es el problema de los dos cuerpos, dado en el Apéndice B como problema B.20. Hemos considerado los casos de una excentricidad media $e = 0{,}5$ y una excentricidad alta $e = 0{,}97$. Este problema Hamiltoniano posee dos invariantes a lo largo de cada curva integral:
  1) `El Hamiltoniano:`

$$H(p,q) = \frac{1}{2}\big((p_1)^2 + (p_2)^2\big) - \big((q_1)^2 + (q_2)^2\big)^{-1/2}, \qquad p_j = \dot{q}_j, \ (j = 1, 2).$$

  2) `El Momento Lineal:`

$$L(p,q) = q_1 p_2 - q_2 p_1.$$

- Problema 4.6. Es una ecuación de onda no-lineal considerada por Cohen, Hairer y Lubich en [20]. Este problema está descrito en el Apéndice B como problema B.34. La segunda derivada en espacio se ha discretizando usando diferencias centrales de segundo orden en el modo usual. Hemos tomado $N = 128$ líneas en el intervalo de la

variable espacial $x \in [-\pi, \pi]$. Por tanto resulta un sistema de EDOs de dimensión media $y''(t) = Jy + g(y)$, donde $g(y)$ denotan términos cuadráticos y la matriz Jacobiana J es una matriz banda con ancho 1, pero con la excepción de tener dos elementos no nulos en cada extremo de la diagonal secundaria.

En las Tablas 4.7.1, 4.7.2, 4.7.3 y 4.7.4, TOL es la Tolerancia (hemos considerado el mismo valor para la tolerancia absoluta y relativa), NSS(NREJ) representan el número de pasos exitosos y rechazados respectivamente. Todos los pasos rechazados han sido incluidos en NREJ ya que muy pocas veces ha habido rechazos por la divergencia en las iteraciones. NLU, NJAC, NFUN y NLS representan el número de descomposiciones LU, el número de matrices Jacobianas calculadas, el número de evaluaciones de la segunda derivada y el número de soluciones de sistemas triangulares (forward–backward) respectivamente. EG_Y y EG_Y' almacenan los errores globales en el punto final en las componentes y e y'. EGest_Y y EGest_Y' denotan respectivamente el error global estimado en el punto final para las componentes y e y', usando la técnica de la *Tolerancia Proporcional* descrita anteriormente. CPU indica el tiempo en segundos tomado por las integraciones, en un ordenador personal (3.4 Ghz).

De los resultados mostrados en las cuatro Tablas 4.7.1-4.7.4, podemos decir que:

- El código GAUSS2 equipado con alguno de los dos estimadores de error local considerado puede ser un buen candidato para integrar un amplio rango de problemas oscilatorios, especialmente cuando se desean precisiones medias o bajas. El código es más adecuado en problemas oscilatorios donde las frecuencias bajas son las dominantes (problemas 4.1, 4.3 y 4.4). En estos casos, los tamaños de paso son controlados por la exactitud requerida en los modos de las bajas frecuencias, en virtud de que las amplitudes muy cortas correspondientes a modos de frecuencia altos no necesitan ser detectados y el integrador no ofrece problemas al integrar estos modos debido a las buenas propiedades de estabilidad del método usado. Este no es el caso de métodos que no sean P-estables (por ejemplo aquellos basados en fórmulas explícitas), donde los modos de alta frecuencia obligan a los códigos a tomar un elevado número de pasos en las integraciones, independientemente del tamaño de las amplitudes asociadas a tales modos de alta frecuencia, tal como puede verse en la Tabla 7.4.3 para el código ODEX2 (problema 4.3). Sin embargo los códigos basados en métodos P-estables como el GAUSS2 completan las integraciones de este tipo de problemas (problemas 4.3 y

Tabla 4.7.1: Estadísticas para el problema 4.1, integrado hasta $t_{end} = 40$. Los resultados obtenidos por los distintos códigos están separados por líneas, y escritos en el siguiente orden: GAUSS2-1, GAUSS2-3, ODEX2, RADAU5, VODE-BDF2 y VODE-BDF4.

TOL	NSS	NREJ	NLU	NJAC	NFUN+NLS	EG_Y	EGest_Y	EG_Y'	EGest_Y'	CPU
10^{-3}	34	1	5	1	503	$2{,}8 \cdot 10^{-2}$	$1{,}8 \cdot 10^{-2}$	$1{,}1 \cdot 10^{-2}$	$3{,}6 \cdot 10^{-2}$	0,00
10^{-4}	74	0	9	1	875	$9{,}8 \cdot 10^{-3}$	$1{,}8 \cdot 10^{-3}$	$1{,}3 \cdot 10^{-3}$	$5{,}2 \cdot 10^{-3}$	0,00
10^{-5}	205	6	20	1	2885	$3{,}9 \cdot 10^{-4}$	$1{,}2 \cdot 10^{-4}$	$7{,}7 \cdot 10^{-4}$	$3{,}2 \cdot 10^{-4}$	0,00
10^{-6}	392	2	17	1	5403	$9{,}8 \cdot 10^{-5}$	$3{,}5 \cdot 10^{-5}$	$5{,}4 \cdot 10^{-4}$	$6{,}0 \cdot 10^{-4}$	0,01
10^{-7}	720	12	50	1	10143	$8{,}2 \cdot 10^{-6}$	$1{,}9 \cdot 10^{-5}$	$2{,}5 \cdot 10^{-5}$	$4{,}3 \cdot 10^{-4}$	0,01
10^{-8}	2102	101	300	1	30551	$2{,}1 \cdot 10^{-6}$	$1{,}3 \cdot 10^{-6}$	$9{,}9 \cdot 10^{-5}$	$1{,}0 \cdot 10^{-5}$	0,02
10^{-9}	3616	191	414	1	52147	$6{,}8 \cdot 10^{-7}$	$6{,}3 \cdot 10^{-7}$	$6{,}4 \cdot 10^{-5}$	$9{,}7 \cdot 10^{-7}$	0,03
10^{-10}	6038	375	751	1	77667	$1{,}2 \cdot 10^{-7}$	$2{,}1 \cdot 10^{-7}$	$9{,}7 \cdot 10^{-6}$	$1{,}7 \cdot 10^{-5}$	0,04
10^{-3}	32	1	4	1	516	$3{,}2 \cdot 10^{-2}$	$5{,}0 \cdot 10^{-4}$	$1{,}5 \cdot 10^{-2}$	$5{,}3 \cdot 10^{-2}$	0,00
10^{-4}	61	2	11	1	874	$4{,}1 \cdot 10^{-3}$	$2{,}3 \cdot 10^{-3}$	$4{,}0 \cdot 10^{-3}$	$9{,}3 \cdot 10^{-4}$	0,00
10^{-5}	114	4	14	1	1675	$1{,}8 \cdot 10^{-4}$	$6{,}4 \cdot 10^{-5}$	$5{,}7 \cdot 10^{-4}$	$7{,}8 \cdot 10^{-4}$	0,00
10^{-6}	371	4	15	1	5562	$1{,}5 \cdot 10^{-4}$	$6{,}2 \cdot 10^{-5}$	$1{,}1 \cdot 10^{-3}$	$1{,}0 \cdot 10^{-3}$	0,01
10^{-7}	634	10	27	1	9479	$1{,}1 \cdot 10^{-5}$	$4{,}4 \cdot 10^{-5}$	$1{,}3 \cdot 10^{-4}$	$1{,}2 \cdot 10^{-4}$	0,01
10^{-8}	1672	120	306	1	26861	$4{,}1 \cdot 10^{-7}$	$1{,}2 \cdot 10^{-6}$	$7{,}6 \cdot 10^{-5}$	$3{,}1 \cdot 10^{-5}$	0,02
10^{-9}	3004	177	395	1	46960	$8{,}0 \cdot 10^{-7}$	$3{,}2 \cdot 10^{-7}$	$6{,}9 \cdot 10^{-5}$	$1{,}8 \cdot 10^{-5}$	0,03
10^{-10}	6132	393	787	1	86014	$1{,}6 \cdot 10^{-7}$	$7{,}4 \cdot 10^{-8}$	$1{,}2 \cdot 10^{-5}$	$2{,}2 \cdot 10^{-5}$	0,05
10^{-4}	1783	151	–	–	21127	$2{,}2 \cdot 10^{-2}$	—	$1{,}7 \cdot 10^{+1}$	—	0,02
10^{-5}	1803	197	–	–	17495	$6{,}2 \cdot 10^{-5}$	—	$1{,}1 \cdot 10^{-2}$	—	0,02
10^{-6}	1849	205	–	–	18915	$5{,}8 \cdot 10^{-6}$	—	$1{,}3 \cdot 10^{-3}$	—	0,02
10^{-7}	1741	180	–	–	18067	$4{,}4 \cdot 10^{-6}$	—	$4{,}1 \cdot 10^{-5}$	—	0,02
10^{-8}	1637	282	–	–	19664	$4{,}4 \cdot 10^{-6}$	—	$8{,}1 \cdot 10^{-6}$	—	0,02
10^{-9}	1773	244	–	–	21430	$1{,}4 \cdot 10^{-6}$	—	$2{,}2 \cdot 10^{-6}$	—	0,02
10^{-10}	1837	112	–	–	26136	$5{,}3 \cdot 10^{-7}$	—	$4{,}2 \cdot 10^{-7}$	—	0,02
10^{-3}	56	10	275	11	648	$5{,}3 \cdot 10^{-3}$	—	$3{,}6 \cdot 10^{-3}$	—	0,01
10^{-4}	79	16	370	17	943	$1{,}3 \cdot 10^{-3}$	—	$1{,}3 \cdot 10^{-3}$	—	0,01
10^{-5}	116	14	330	15	1316	$5{,}5 \cdot 10^{-4}$	—	$1{,}1 \cdot 10^{-3}$	—	0,01
10^{-6}	174	13	390	14	1926	$2{,}6 \cdot 10^{-4}$	—	$4{,}1 \cdot 10^{-4}$	—	0,02
10^{-7}	264	21	655	22	2994	$5{,}6 \cdot 10^{-5}$	—	$2{,}0 \cdot 10^{-4}$	—	0,02
10^{-8}	526	152	2930	149	7076	$1{,}9 \cdot 10^{-5}$	—	$8{,}4 \cdot 10^{-5}$	—	0,04
10^{-9}	840	238	4600	239	11520	$5{,}6 \cdot 10^{-6}$	—	$2{,}7 \cdot 10^{-5}$	—	0,02
10^{-10}	1239	216	5275	216	15872	$2{,}6 \cdot 10^{-6}$	—	$3{,}1 \cdot 10^{-6}$	—	0,03
10^{-4}	505	23	62	9	1229	$8{,}5 \cdot 10^{-2}$	—	$4{,}8 \cdot 10^{-2}$	—	0,00
10^{-5}	1086	24	93	19	2457	$1{,}8 \cdot 10^{-2}$	—	$1{,}2 \cdot 10^{-2}$	—	0,01
10^{-6}	2443	26	162	41	5333	$4{,}0 \cdot 10^{-3}$	—	$1{,}9 \cdot 10^{-3}$	—	0,02
10^{-7}	5774	21	312	97	12431	$5{,}4 \cdot 10^{-4}$	—	$4{,}1 \cdot 10^{-4}$	—	0,04
10^{-8}	11308	21	595	190	24245	$1{,}4 \cdot 10^{-4}$	—	$8{,}6 \cdot 10^{-4}$	—	0,08
10^{-9}	62412	1966	6120	1048	148226	$4{,}0 \cdot 10^{-6}$	—	$2{,}0 \cdot 10^{-5}$	—	0,23
10^{-10}	102618	1683	7713	1759	232922	$1{,}8 \cdot 10^{-6}$	—	$1{,}0 \cdot 10^{-5}$	—	0,38
10^{-5}	17780	1107	2391	306	45965	$7{,}0 \cdot 10^{-2}$	—	$2{,}5 \cdot 10^{+1}$	—	0,09
10^{-6}	22636	998	2465	387	56037	$2{,}9 \cdot 10^{-3}$	—	$4{,}2 \cdot 10^{+0}$	—	0,14
10^{-7}	23318	1198	2803	402	58623	$2{,}8 \cdot 10^{-3}$	—	$8{,}4 \cdot 10^{-2}$	—	0,14
10^{-8}	23063	962	2475	395	56747	$5{,}4 \cdot 10^{-5}$	—	$3{,}1 \cdot 10^{-3}$	—	0,14
10^{-9}	23834	1313	2970	410	60683	$2{,}2 \cdot 10^{-5}$	—	$3{,}1 \cdot 10^{-3}$	—	0,11
10^{-10}	24165	1359	3033	416	61599	$1{,}0 \cdot 10^{-6}$	—	$3{,}9 \cdot 10^{-4}$	—	0,12

Tabla 4.7.2: Estadísticas para el problema 4.2, integrado hasta $t_{end} = 100$. Los resultados obtenidos por los distintos códigos están separados por líneas, y escritos en el siguiente orden: GAUSS2-1, GAUSS2-3, ODEX2, RADAU5, VODE-BDF2 y VODE-BDF4.

TOL	NSS	NREJ	NLU	NJAC	NFUN+NLS	EG_Y	EGest_Y	EG_Y'	EGest_Y'	CPU
10^{-6}	6753	16	33	1	131857	$5{,}6 \cdot 10^{-3}$	$4{,}6 \cdot 10^{-3}$	$2{,}6 \cdot 10^{-1}$	$1{,}9 \cdot 10^{-1}$	0,14
10^{-7}	10366	18	38	1	187719	$6{,}9 \cdot 10^{-4}$	$9{,}1 \cdot 10^{-4}$	$2{,}9 \cdot 10^{-2}$	$5{,}0 \cdot 10^{-2}$	0,23
10^{-8}	16871	50	151	1	287833	$1{,}5 \cdot 10^{-4}$	$1{,}2 \cdot 10^{-4}$	$6{,}1 \cdot 10^{-3}$	$6{,}6 \cdot 10^{-3}$	0,31
10^{-9}	27075	35	131	1	396827	$2{,}4 \cdot 10^{-5}$	$2{,}1 \cdot 10^{-5}$	$1{,}1 \cdot 10^{-3}$	$9{,}4 \cdot 10^{-4}$	0,47
10^{-6}	6693	13	28	1	139849	$4{,}9 \cdot 10^{-3}$	$4{,}6 \cdot 10^{-3}$	$2{,}6 \cdot 10^{-1}$	$2{,}0 \cdot 10^{-1}$	0,15
10^{-7}	10372	20	42	1	198315	$4{,}9 \cdot 10^{-4}$	$1{,}2 \cdot 10^{-3}$	$2{,}8 \cdot 10^{-2}$	$5{,}1 \cdot 10^{-2}$	0,25
10^{-8}	17968	14	43	1	315173	$1{,}3 \cdot 10^{-4}$	$8{,}7 \cdot 10^{-5}$	$6{,}1 \cdot 10^{-3}$	$4{,}6 \cdot 10^{-3}$	0,34
10^{-9}	27032	41	165	1	423590	$2{,}3 \cdot 10^{-5}$	$2{,}1 \cdot 10^{-5}$	$8{,}9 \cdot 10^{-4}$	$1{,}0 \cdot 10^{-3}$	0,51
10^{-6}	2460	0	—	—	40147	$2{,}4 \cdot 10^{-2}$	—	$5{,}8 \cdot 10^{-2}$	—	0,05
10^{-7}	2195	0	—	—	48651	$3{,}6 \cdot 10^{-3}$	—	$8{,}6 \cdot 10^{-3}$	—	0,05
10^{-8}	2705	0	—	—	60091	$3{,}2 \cdot 10^{-4}$	—	$7{,}5 \cdot 10^{-4}$	—	0,05
10^{-9}	2376	0	—	—	69051	$4{,}7 \cdot 10^{-5}$	—	$1{,}1 \cdot 10^{-4}$	—	0,08
10^{-6}	7777	360	10055	361	102097	$6{,}4 \cdot 10^{-2}$	—	$1{,}6 \cdot 10^{-1}$	—	0,19
10^{-7}	11399	218	11355	219	154727	$1{,}6 \cdot 10^{-2}$	—	$3{,}6 \cdot 10^{-2}$	—	0,26
10^{-8}	16649	146	16340	147	220450	$1{,}1 \cdot 10^{-3}$	—	$2{,}5 \cdot 10^{-3}$	—	0,39
10^{-9}	24413	155	23705	156	330278	$1{,}8 \cdot 10^{-5}$	—	$1{,}1 \cdot 10^{-4}$	—	0,58
10^{-6}	65927	40	3369	1100	145167	$4{,}5 \cdot 10^{-2}$	—	$6{,}1 \cdot 10^{-1}$	—	0,51
10^{-7}	136968	39	6907	2284	301373	$2{,}8 \cdot 10^{-2}$	—	$6{,}8 \cdot 10^{-1}$	—	1,08
10^{-8}	308960	42	15505	5151	678341	$9{,}4 \cdot 10^{-3}$	—	$1{,}9 \cdot 10^{-1}$	—	2,39
10^{-9}	663438	43	33229	11059	1455387	$2{,}4 \cdot 10^{-3}$	—	$4{,}2 \cdot 10^{-2}$	—	5,12
10^{-6}	22332	5	1129	373	49169	$1{,}0 \cdot 10^{-1}$	—	$1{,}3 \cdot 10^{+0}$	—	0,20
10^{-7}	33456	4	1685	558	73639	$9{,}9 \cdot 10^{-2}$	—	$2{,}9 \cdot 10^{-1}$	—	0,33
10^{-8}	48077	11	2424	802	105843	$1{,}4 \cdot 10^{-2}$	—	$5{,}9 \cdot 10^{-2}$	—	0,73
10^{-9}	76341	7	3834	1273	167991	$1{,}3 \cdot 10^{-3}$	—	$8{,}5 \cdot 10^{-3}$	—	1,20

Tabla 4.7.3: Estadísticas para el problema 4.3, integrado hasta $t_{end} = 1000$. Los resultados obtenidos por los distintos códigos están separados por líneas, y escritos en el siguiente orden: GAUSS2-1, GAUSS2-3, ODEX2, RADAU5, VODE-BDF2 y VODE-BDF4.

TOL	NSS	NREJ	NLU	NJAC	NFUN+NLS	EG_Y	EGest_Y	EG_Y'	EGest_Y'	CPU
10^{-4}	93	6	19	1	1439	$5{,}0 \cdot 10^{-3}$	$6{,}0 \cdot 10^{-3}$	$3{,}5 \cdot 10^{-4}$	$3{,}2 \cdot 10^{-4}$	0,08
10^{-5}	148	6	19	1	2095	$8{,}9 \cdot 10^{-4}$	$8{,}1 \cdot 10^{-4}$	$6{,}5 \cdot 10^{-5}$	$5{,}7 \cdot 10^{-5}$	0,11
10^{-6}	223	11	30	1	2961	$1{,}4 \cdot 10^{-4}$	$1{,}6 \cdot 10^{-4}$	$1{,}1 \cdot 10^{-5}$	$1{,}2 \cdot 10^{-5}$	0,17
10^{-7}	360	23	82	1	4657	$2{,}5 \cdot 10^{-5}$	$2{,}5 \cdot 10^{-5}$	$1{,}9 \cdot 10^{-6}$	$1{,}8 \cdot 10^{-6}$	0,25
10^{-8}	573	0	11	1	6663	$3{,}2 \cdot 10^{-6}$	$3{,}7 \cdot 10^{-6}$	$2{,}4 \cdot 10^{-7}$	$2{,}7 \cdot 10^{-7}$	0,44
10^{-4}	89	7	21	1	1521	$5{,}5 \cdot 10^{-3}$	$7{,}3 \cdot 10^{-3}$	$3{,}8 \cdot 10^{-4}$	$3{,}7 \cdot 10^{-4}$	0,09
10^{-5}	145	10	27	1	2324	$9{,}3 \cdot 10^{-4}$	$8{,}7 \cdot 10^{-4}$	$6{,}8 \cdot 10^{-5}$	$6{,}1 \cdot 10^{-5}$	0,13
10^{-6}	223	9	26	1	3167	$1{,}6 \cdot 10^{-4}$	$1{,}6 \cdot 10^{-4}$	$1{,}2 \cdot 10^{-5}$	$1{,}2 \cdot 10^{-5}$	0,19
10^{-7}	362	30	101	1	5199	$2{,}8 \cdot 10^{-5}$	$2{,}2 \cdot 10^{-5}$	$1{,}9 \cdot 10^{-6}$	$1{,}7 \cdot 10^{-6}$	0,28
10^{-8}	561	9	35	1	7257	$3{,}4 \cdot 10^{-6}$	$4{,}1 \cdot 10^{-6}$	$2{,}6 \cdot 10^{-7}$	$3{,}1 \cdot 10^{-7}$	0,47
10^{-5}	237152	1607	—	—	2586634	$8{,}8 \cdot 10^{-5}$	—	$7{,}3 \cdot 10^{-2}$	—	324,6
10^{-6}	237395	1760	—	—	2591532	$5{,}8 \cdot 10^{-6}$	—	$5{,}8 \cdot 10^{-3}$	—	322,9
10^{-7}	237371	1737	—	—	2590342	$1{,}4 \cdot 10^{-6}$	—	$2{,}3 \cdot 10^{-4}$	—	323,7
10^{-8}	237307	1627	—	—	2589696	$3{,}6 \cdot 10^{-7}$	—	$2{,}7 \cdot 10^{-5}$	—	323,8
10^{-3}	98	43	680	29	1418	$2{,}9 \cdot 10^{-3}$	—	$2{,}6 \cdot 10^{-4}$	—	0,12
10^{-4}	134	32	660	33	1670	$3{,}3 \cdot 10^{-4}$	—	$3{,}5 \cdot 10^{-5}$	—	0,14
10^{-5}	197	32	815	33	2373	$5{,}3 \cdot 10^{-5}$	—	$6{,}0 \cdot 10^{-6}$	—	0,19
10^{-7}	274	31	870	32	3226	$1{,}1 \cdot 10^{-5}$	—	$1{,}2 \cdot 10^{-6}$	—	0,24
10^{-8}	395	33	1155	34	4659	$1{,}9 \cdot 10^{-6}$	—	$1{,}6 \cdot 10^{-7}$	—	0,34
10^{-3}	269	20	51	5	725	$3{,}8 \cdot 10^{-2}$	—	$1{,}0 \cdot 10^{-2}$	—	0,52
10^{-4}	617	27	67	11	1523	$9{,}3 \cdot 10^{-2}$	—	$3{,}5 \cdot 10^{-5}$	—	0,80
10^{-5}	1455	46	146	24	3429	$1{,}9 \cdot 10^{-2}$	—	$9{,}4 \cdot 10^{-3}$	—	1,75
10^{-6}	3139	57	249	52	7079	$4{,}1 \cdot 10^{-3}$	—	$2{,}7 \cdot 10^{-4}$	—	3,24
10^{-7}	6626	66	443	112	14597	$8{,}8 \cdot 10^{-4}$	—	$6{,}3 \cdot 10^{-5}$	—	6,17
10^{-2}	117	12	25	3	341	$2{,}9 \cdot 10^{-1}$	—	$6{,}8 \cdot 10^{-2}$	—	0,33
10^{-3}	344	27	58	6	919	$1{,}6 \cdot 10^{-1}$	—	$2{,}1 \cdot 10^{-2}$	—	0,64
10^{-4}	> 150000	***	***	***	***	***	—	***	—	$> 100.$

Tabla 4.7.4: Estadísticas para el problema 4.4, integrado hasta $t_{end} = 500$. Los resultados obtenidos por los distintos códigos están separados por líneas, y escritos en el siguiente orden: GAUSS2-1, GAUSS2-3, ODEX2, RADAU5, VODE-BDF2 y VODE-BDF4.

TOL	NSS	NREJ	NLU	NJAC	NFUN+NLS	EG_Y	EGest_Y	EG_Y'	EGest_Y'	CPU
10^{-4}	42	2	10	1	939	$4{,}7 \cdot 10^{-4}$	$7{,}2 \cdot 10^{-4}$	$1{,}6 \cdot 10^{-4}$	$2{,}3 \cdot 10^{-4}$	0,09
10^{-5}	67	6	24	1	1409	$1{,}0 \cdot 10^{-4}$	$9{,}9 \cdot 10^{-5}$	$3{,}5 \cdot 10^{-5}$	$3{,}5 \cdot 10^{-5}$	0,16
10^{-6}	101	9	33	1	2051	$2{,}0 \cdot 10^{-5}$	$1{,}8 \cdot 10^{-5}$	$7{,}1 \cdot 10^{-6}$	$6{,}3 \cdot 10^{-6}$	0,21
10^{-7}	451	9	36	15	8889	$1{,}5 \cdot 10^{-6}$	$3{,}5 \cdot 10^{-6}$	$4{,}6 \cdot 10^{-6}$	$2{,}0 \cdot 10^{-6}$	0,80
10^{-8}	2062	9	47	24	44353	$6{,}1 \cdot 10^{-7}$	$1{,}5 \cdot 10^{-6}$	$1{,}9 \cdot 10^{-6}$	$1{,}2 \cdot 10^{-6}$	3,20
10^{-4}	41	3	12	1	1007	$4{,}8 \cdot 10^{-4}$	$7{,}7 \cdot 10^{-4}$	$1{,}7 \cdot 10^{-4}$	$2{,}6 \cdot 10^{-4}$	0,10
10^{-5}	67	6	24	1	1490	$8{,}5 \cdot 10^{-5}$	$1{,}1 \cdot 10^{-4}$	$3{,}0 \cdot 10^{-5}$	$3{,}8 \cdot 10^{-5}$	0,16
10^{-6}	101	9	33	1	2161	$2{,}1 \cdot 10^{-5}$	$1{,}8 \cdot 10^{-5}$	$7{,}4 \cdot 10^{-6}$	$6{,}2 \cdot 10^{-6}$	0,22
10^{-7}	160	4	26	1	3147	$2{,}3 \cdot 10^{-6}$	$3{,}9 \cdot 10^{-6}$	$7{,}2 \cdot 10^{-7}$	$1{,}2 \cdot 10^{-6}$	0,30
10^{-8}	334	6	78	35	7452	$3{,}1 \cdot 10^{-7}$	$8{,}7 \cdot 10^{-7}$	$1{,}1 \cdot 10^{-7}$	$1{,}3 \cdot 10^{-7}$	0,72
10^{-8}	9350	12	–	–	101851	$1{,}8 \cdot 10^{-2}$	—	$5{,}2 \cdot 10^{-4}$	—	3,48
10^{-9}	9276	1	–	–	101616	$1{,}6 \cdot 10^{-4}$	—	$4{,}7 \cdot 10^{-6}$	—	3,45
10^{-10}	9326	22	–	–	102292	$1{,}2 \cdot 10^{-5}$	—	$3{,}6 \cdot 10^{-7}$	—	3,47
10^{-11}	9352	187	–	–	104039	$9{,}6 \cdot 10^{-8}$	—	$4{,}3 \cdot 10^{-9}$	—	3,55
10^{-5}	89	10	385	56	1690	$5{,}2 \cdot 10^{-4}$	—	$1{,}5 \cdot 10^{-5}$	—	0,48
10^{-6}	123	9	430	43	2067	$6{,}0 \cdot 10^{-4}$	—	$1{,}7 \cdot 10^{-5}$	—	0,58
10^{-7}	177	10	555	38	2929	$7{,}1 \cdot 10^{-5}$	—	$2{,}1 \cdot 10^{-6}$	—	0,77
10^{-8}	257	10	695	25	4273	$3{,}4 \cdot 10^{-4}$	—	$1{,}0 \cdot 10^{-5}$	—	1,02
10^{-9}	371	10	1110	25	6075	$1{,}4 \cdot 10^{-6}$	—	$4{,}2 \cdot 10^{-8}$	—	1,55
10^{-4}	388	18	51	7	937	$1{,}9 \cdot 10^{-2}$	—	$3{,}7 \cdot 10^{-3}$	—	0,69
10^{-5}	784	18	74	14	1819	$5{,}3 \cdot 10^{-2}$	—	$1{,}9 \cdot 10^{-3}$	—	1,08
10^{-6}	1719	25	126	30	3819	$9{,}2 \cdot 10^{-4}$	—	$2{,}1 \cdot 10^{-4}$	—	1,98
10^{-7}	3828	29	241	65	8273	$1{,}5 \cdot 10^{-4}$	—	$4{,}6 \cdot 10^{-5}$	—	3,97
10^{-8}	8181	31	461	137	17467	$3{,}2 \cdot 10^{-5}$	—	$1{,}1 \cdot 10^{-5}$	—	7,89
10^{-2}	59	5	14	2	173	$4{,}4 \cdot 10^{-1}$	—	$7{,}4 \cdot 10^{-2}$	—	0,25
10^{-3}	92	6	17	2	247	$1{,}5 \cdot 10^{-1}$	—	$2{,}0 \cdot 10^{-3}$	—	0,27
10^{-4}	> 60000	***	***	***	***	***	—	***	—	$> 45.$

4.4) en un número moderado de pasos, siempre que no se exijan tolerancias muy restrictivas que obliguen al integrador a detectar las pequeñas amplitudes asociadas a las frecuencias altas.

Se puede observar también en las cuatro tablas que independientemente del problema y la tolerancia, el código GAUSS2 toma muy pocas evaluaciones de Jacobianos y descomposiciones LU para completar las integraciones. Esto significa que prácticamente integra con tamaños de paso constante en subintervalos relativamente grandes dentro del intervalo de integración, y que el proceso iterativo usado para resolver las ecuaciones de etapas es convergente en pocas iteraciones, normalmente la convergencia se alcanza entre 3 y 5 iteraciones. Aunque no se muestra en las tablas, la elección del tamaño del paso inicial fue prácticamente siempre satisfactoria. Todos estos hechos han sido confirmados sobre otros muchos problemas integrados con el código. También puede observarse que para el GAUSS2 los errores globales en el punto final, sobre ambas componentes, son de magnitud similar a los errores globales estimados, independientemente del problema y del estimador de error local usado. Además, en los problemas 4.2, 4.3 y 4.4, los errores globales mantienen una *Proporcionalidad respecto a la Tolerancia* (Tolerance Proportionality), en el sentido que cuando las tolerancias se reducen en un factor de 10, los errores globales disminuyen en un factor aproximado de $10^{4/5} \simeq 6$. No es este el caso del problema 4.1, donde los cambios en el factor son más bruscos. Aún así y a pesar de las discontinuidades en la tercera derivada de la solución de este último problema, los errores globales estimados en el punto final están en concordancia con los errores globales verdaderos.

Aunque ambas estimaciones del error local sólo controlan el error local en la componente y, parece que estas estimaciones son suficientes para producir soluciones numéricas aceptables en ambas componentes. Algunas veces, los errores globales en la componente y' son ligeramente mayores que en la componente y, pero en este caso debemos tener en cuenta que si la integración de una solución no es satisfactoria para el usuario, entonces en virtud de la estimación del error global, una nueva integración con tolerancias más restrictivas, podría resolver el problema seguramente.

El uso de cualquiera de ambos estimadores $\|\epsilon_1\|$ o ε_3, presentan sus ventajas e inconvenientes. El estimador ε_3 es más exacto que el otro, en el caso de modos de frecuencia altos, a efectos de estimar los errores locales cuando la componente y es nula y la y' no

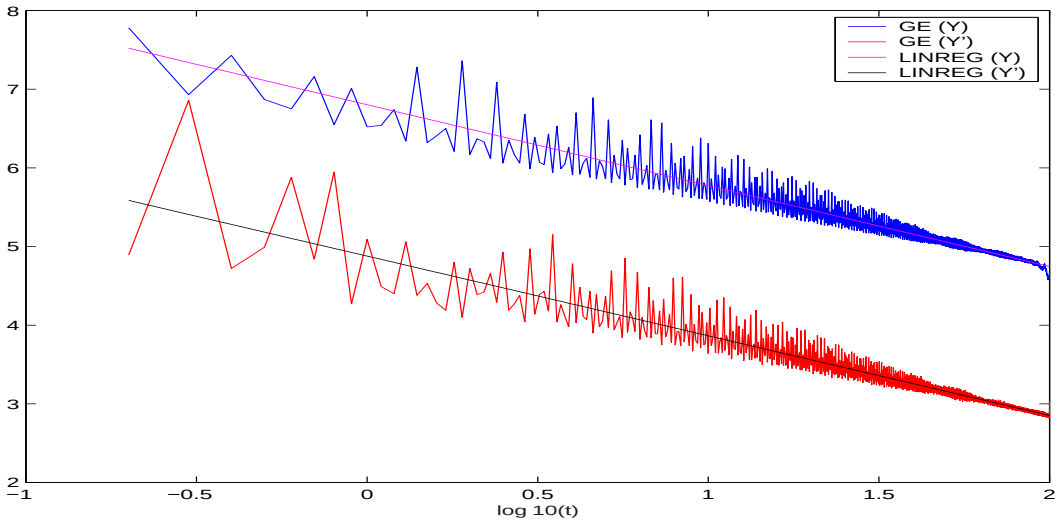
lo es, sin embargo tiene el inconveniente de que estima a la baja en el caso de frecuencias altas con componente y significativa. En este último caso es mejor el estimador de error local $\|\epsilon_1\|$. De acuerdo con los resultados de las 4 tablas podemos ver que el estimador $\|\epsilon_1\|$ hace el código ligeramente más eficiente para los 3 primeros problemas, mientras que el estimador ε_3 es más eficaz para el último problema (Problema 4.4). En nuestra opinión cualquiera de las dos opciones es válida, y si queremos que el código detecte mejor las amplitudes asociadas a los modos de frecuencia altos al nivel de la tolerancia prescrita, entonces quizás sea más conveniente usar el estimador $\|\epsilon_1\|$. También debemos reseñar que si lo que queremos son altos grados de precisión, entonces la opción de GAUSS2 no es la mejor, tal como puede verse en la Tabla 7.4.2 (problema 4.2).

- ODEX2 funciona satisfactoriamente cuando se requieren altos grados de precisión y algunas de las frecuencias del problema son significativas. Este es el caso del problema 4.2, en el que los tamaños de pasos son restringidos por la exactitud requerida antes que por razones de estabilidad. En la integración de los cuatro problemas (y en muchos otros), el código produce integraciones más satisfactorias cuando se usan tolerancias muy pequeñas. Sin embargo debe tomarse con cierta precaución los resultados dados por el código cuando se usan tolerancias bajas y medias y sólo se usa control de error local en la componente y. También puede observarse que el código requiere un número similar de tamaños de pasos y de evaluaciones de f, independientemente de la precisión requerida. Esto es una característica típica de métodos basados en la extrapolación local de una fórmula más simple, la cual se toma como fórmula base, en este caso la fórmula base es la regla de Störmer. El alto número de pasos requeridos por el código se debe a que las fórmulas explícitas de las que hace uso el código, poseen un intervalo acotado de estabilidad con una longitud del orden de la unidad. El código no es adecuado para el problema 4.3 (véase Tabla 7.4.3) ni otros similares, especialmente en el caso en que se considera un gran número de líneas en la discretización espacial, pues las frecuencias altas del problema obligan a tomar un número de pasos muy elevado, aun en el caso en que estas frecuencias no sean significativas. Con este código no se puede esperar *Proporcionalidad respecto a la Tolerancia* para los errores globales debido a que el código no usa la misma fórmula para todas las integraciones, ni para todos los pasos de integración. Este hecho se ve claramente reflejado en las tablas cuando se disminuye la tolerancia, pues los errores no decrecen en una proporción fija con las tolerancias.

- RADAU5 Funciona aceptablemente bien en todos los problemas, en particular cuando se emplean tolerancias medias y bajas. Este código parece ser fiable para problemas de tipo (4.1.1). Sin embargo es algo costoso en evaluaciones de Jacobianos y de factorizaciones LU cuando se emplea la opción estándar (por defecto) del código. En las tablas, se ha multiplicado por 5 el número de factorizaciones LU dadas por el código, ya que cada LU se refiere a una factorización LU en aritmética compleja más una LU en aritmética real. Esta circunstancia jugaría un papel importante en los tiempos de CPU tomados por el código cuando se integran problemas de dimensiones más altas. Debemos resaltar que RADAU5 fue concebido principalmente para integrar sistemas diferenciales de primer orden de tipo stiff, y por tanto necesita *una afinación* más conveniente cuando el código se desea usar en problemas de segundo orden. Se espera que RADAU5 exhiba *Proporcionalidad respecto a la Tolerancia* en los errores globales para los problemas de segundo orden. Esto se ve principalmente en los problemas 4.2 y 4.3, donde los errores globales disminuyen alrededor de un factor de 10 en ambas componentes cuando las tolerancias son disminuidas por el mismo factor. Sin embargo, una opción para estimar los errores globales no se ha incluido en ninguna de las versiones del código.

- Códigos VODE: VODE–BDF2 es una opción aceptable para los problemas del tipo PDE cuando se requiere una precisión baja, pero está en clara desventaja con respecto a los otros códigos. Esto se explica por el bajo orden de la fórmula BDF de 2 etapas. En los problemas 4.3 y 4.4, la opción VODE–BDF2 es más eficiente que la opción VODE–BDF4 debido a que el tamaño de paso es controlado por estabilidad, y las fórmulas de 3 y 4 pasos tienen intervalos de estabilidad muy cortos en entornos del origen del eje imaginario. En los problema provenientes de ecuaciones en ordinarias (problemas 4.1 y 4.2) el código VODE–BDF4 resulta ser superior al VODE–BDF2 ya que los tamaños de paso necesitan ser reducidos por exactitud antes que por estabilidad. La falta de estabilidad sobre el eje imaginario de las fórmulas BDF de k–pasos para $k \geq 3$, parece ser el mayor inconveniente para usar estas fórmulas en cierta clase de problemas de segundo orden, especialmente cuando éstos presentan un amplio rango de frecuencias dispersas. La versión del VODE–BDF2 parece exhibir *Proporcionalidad respecto a la Tolerancia*, tal cual como puede verse en las tablas correspondientes a los problemas 4.2, 4.3 y 4.4, donde los errores globales disminuyen por un factor aproximado de 4.5 cuando las tolerancias se dividen por 10.

Figura 4.7.1: *Gráficas de errores globales (en escala logarítmica) y de rectas de regresión de errores globales, usando la salida continua del GAUSS2-1 en 1000 puntos equidistantes en el intervalo* $[0, 100]$*, para el problema 4.2 con tol* $= 10^{-9}$.



Con respecto al segundo objetivo planteado en esta sección, el cual se refiere a la salida densa del código GAUSS2 y las estimaciones del error global en cualquier punto del intervalo de integración, y no sólo en el punto final, podemos decir lo siguiente.
En las Figuras 4.7.1, 4.7.2, 4.7.3, 4.7.4 y 4.7.5 se muestran respectivamente para los problemas 4.2, 4.3, 4.4, 4.5 y 4.6, las gráficas de los elementos indicados a continuación, (usando en todos los casos escalas logarítmicas en base 10 en ambos ejes):

1. $GE(Y) = -\log_{10}$ (Error Global en componente y).
2. $GE(Y') = -\log_{10}$ (Error Global en componente y').
3. LINREG(Y) y LINREG(Y'), denotan respectivamente las rectas de regresión lineal de los errores globales sobre cada componente.
4. GE(Y)EST y GE(Y')EST, denotan respectivamente los errores globales estimados en las componentes y e y'. Dichos errores estimados fueron calculados usando la fórmula (4.5.5) con $\tau = 5$, y la salida continua del GAUSS2 (con el estimador de error local $\|\epsilon_1\|$), denotando el código en este caso por GAUSS2-1.

Figura 4.7.2: *Gráficas (en escala logarítmica) de errores globales, errores estimados y rectas de regresión, usando la salida continua del GAUSS2-1 en 20 puntos equidistantes en el intervalo* $[0, 10^3]$*, para el problema 4.3 con tol* $= 10^{-5}$.

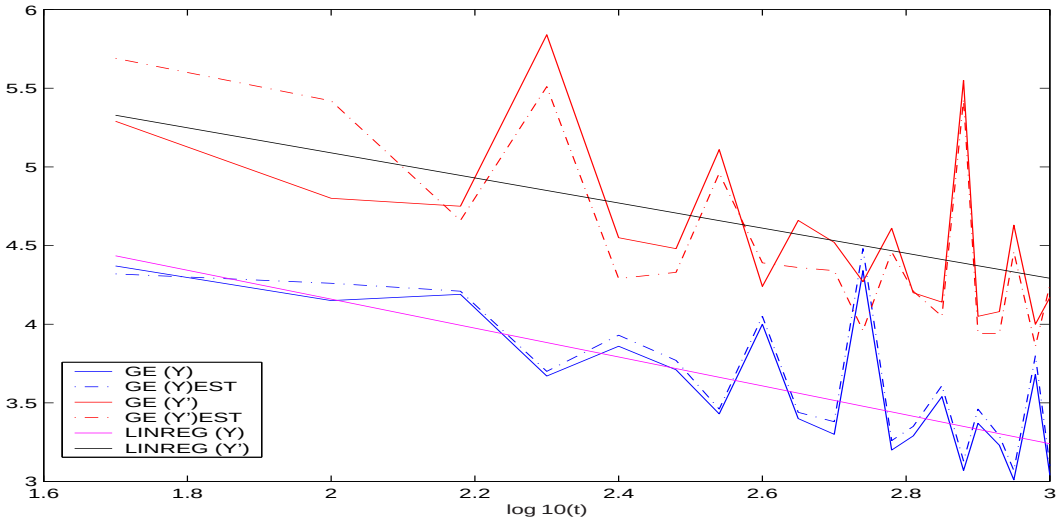


Figura 4.7.3: *Gráficas (en escala logarítmica) de errores globales, errores estimados y rectas de regresión, usando la salida continua del GAUSS2-1 en 20 puntos equidistantes en el intervalo* $[0, 500]$*, para el problema 4.4 con tol* $= 10^{-5}$.

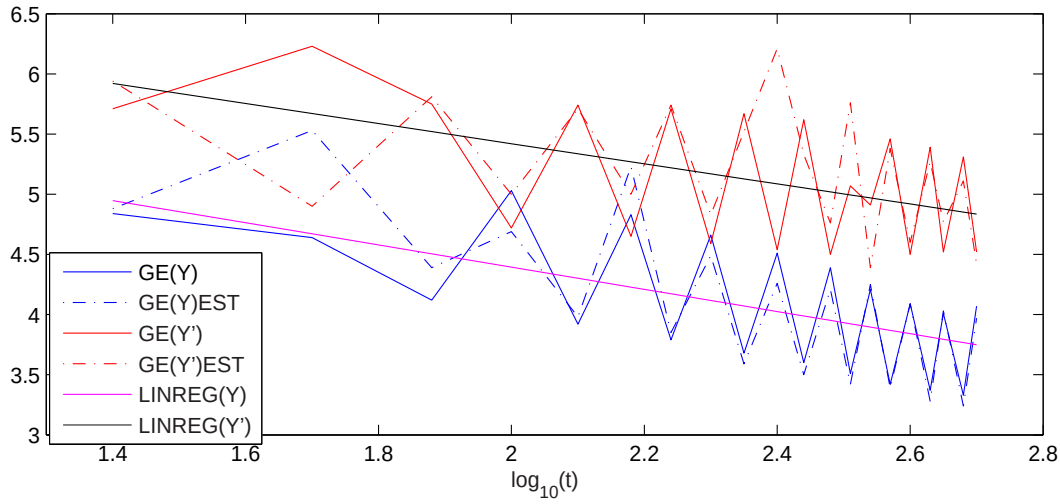


Figura 4.7.4: *Gráficas (en escala logarítmica) de errores globales y rectas de regresión, usando la salida continua del GAUSS2-1 en 64 puntos equidistantes en el intervalo* $[0, 32\pi]$*, para el problema 4.5 con excentricidad* $e = 0{,}97$ *y* $tol = 10^{-11}$.

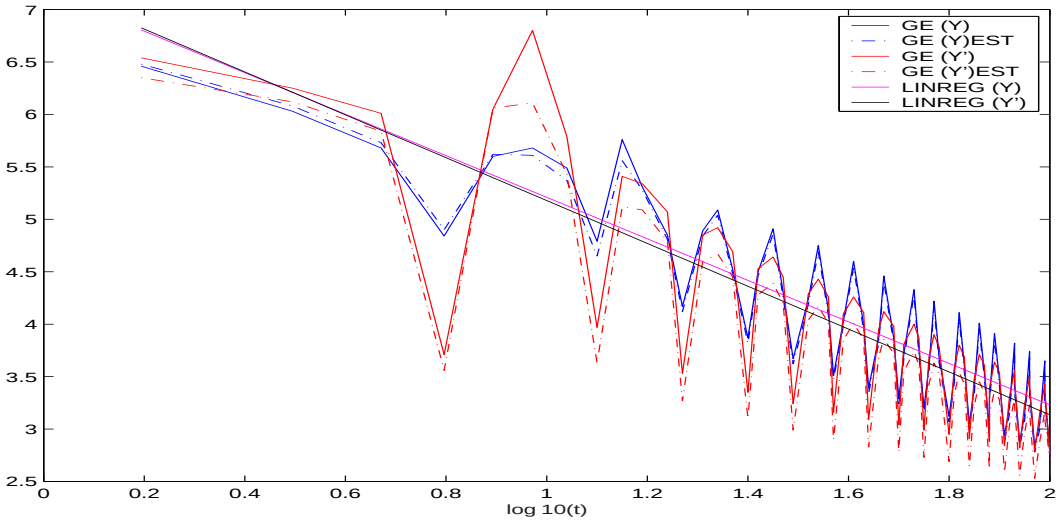


Figura 4.7.5: *Gráficas (en escala logarítmica) de errores globales y rectas de regresión, usando la salida continua del GAUSS2-1 en 10 puntos equidistantes en el intervalo* $[0, 550]$*, para el problema 4.6 con* $tol = 10^{-6}$.

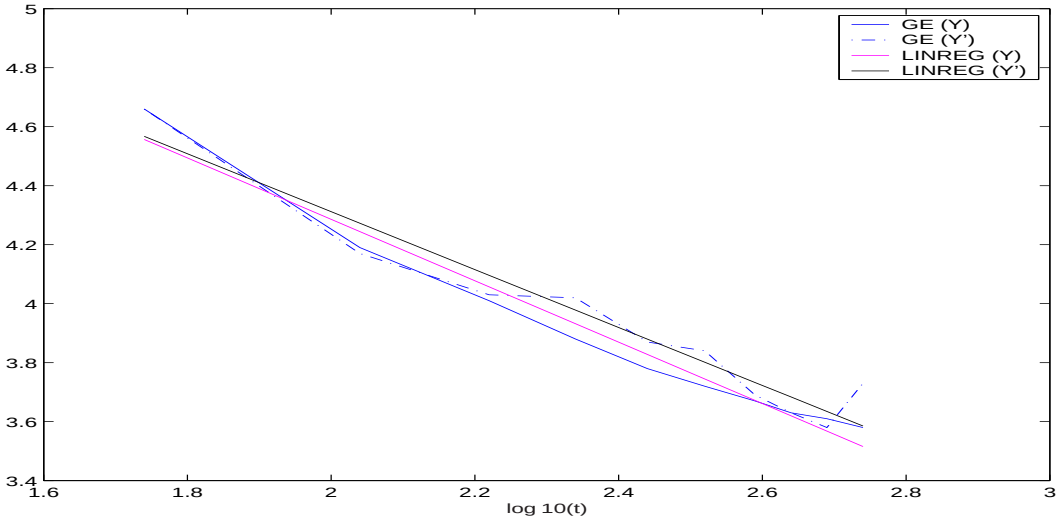


Figura 4.7.6: *Gráficas (en escala logarítmica) de errores globales del Hamiltoniano y el Momento Lineal y las rectas de regresión correspondientes, usando la salida continua del GAUSS2-1 en 64 puntos equidistantes en el intervalo* $[0, 32\pi]$*, para el problema 4.5 con excentricidad* $e = 0{,}97$ *y* $tol = 10^{-11}$.

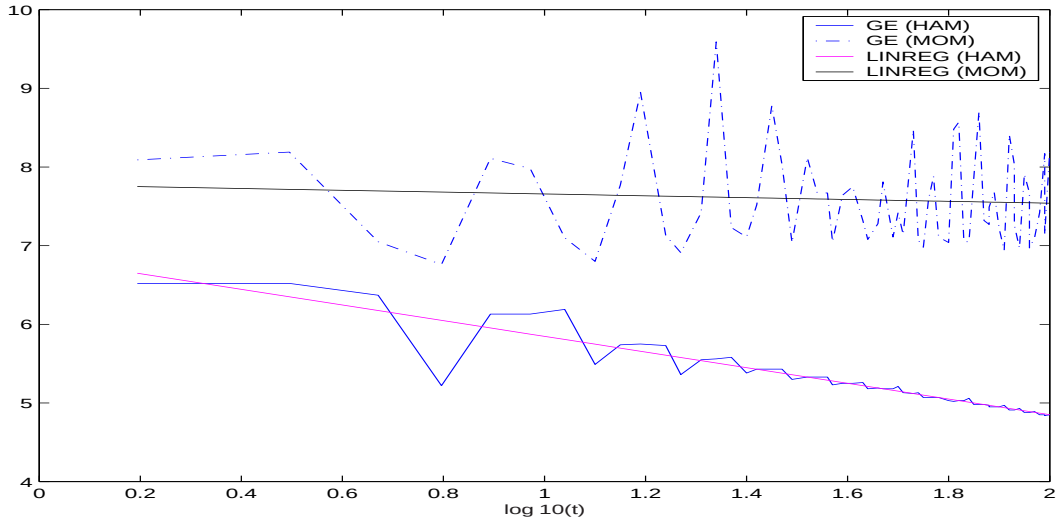


Figura 4.7.7: *Gráficas de las soluciones numéricas en las y-componentes para el problema 4.1, usando la salida continua del GAUSS2-1 con 1000 puntos equidistantes en el intervalo* $[0, 40]$ *y tol* $= 10^{-8}$.

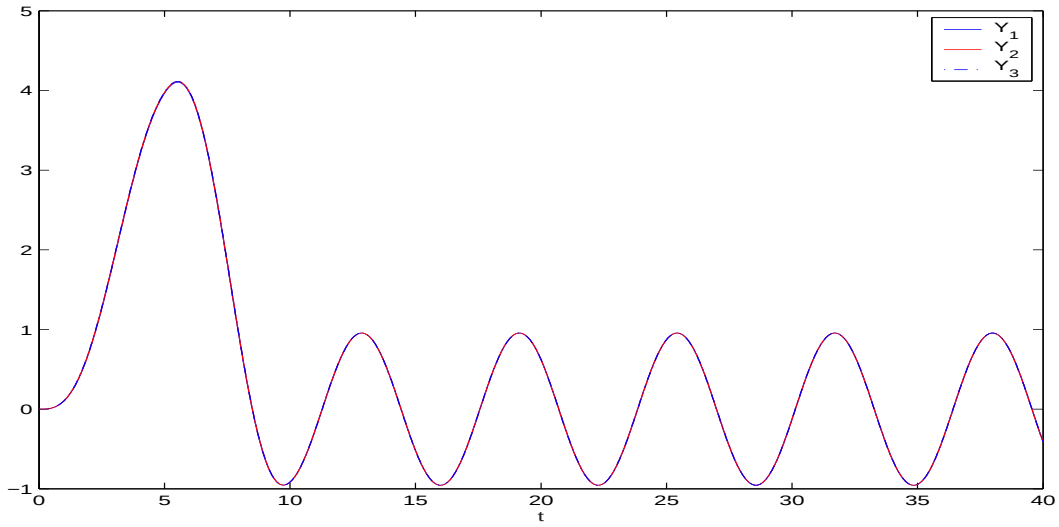


Figura 4.7.8: *Gráficas de las componentes* y_{32} *e* y_{64} *para el problema 4.6, usando la salida continua del GAUSS2-1 con 1100 puntos equidistantes en el intervalo* $[450, 550]$ *y tol* $= 10^{-7}$.

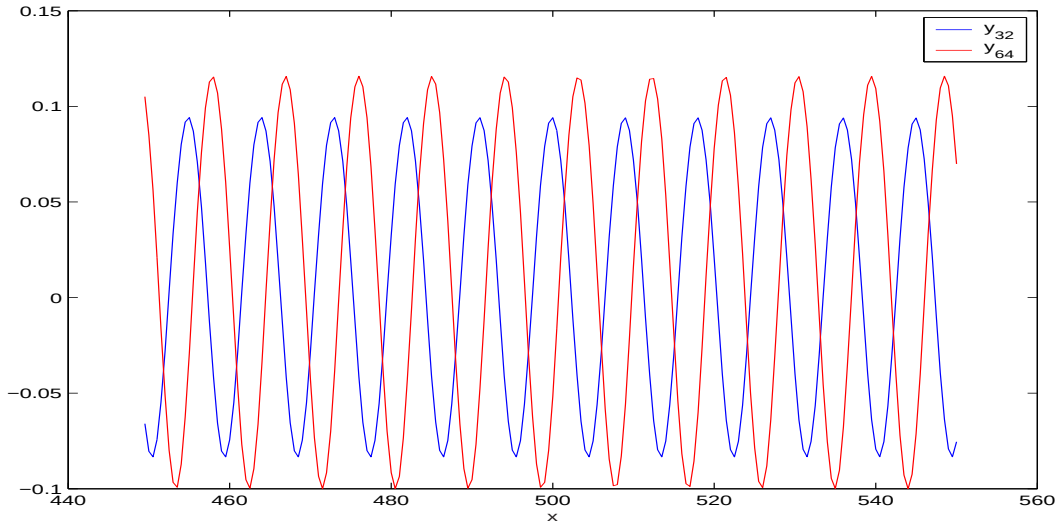


Figura 4.7.9: *Gráficas de errores globales en* y, y' *y en el Hamiltoniano para el problema 4.2, usando la salida continua del GAUSS2-1 en 1000 puntos equidistantes en el intervalo* $[0, 100]$ *(sin escala logarítmica), con tol* $= 10^{-9}$.

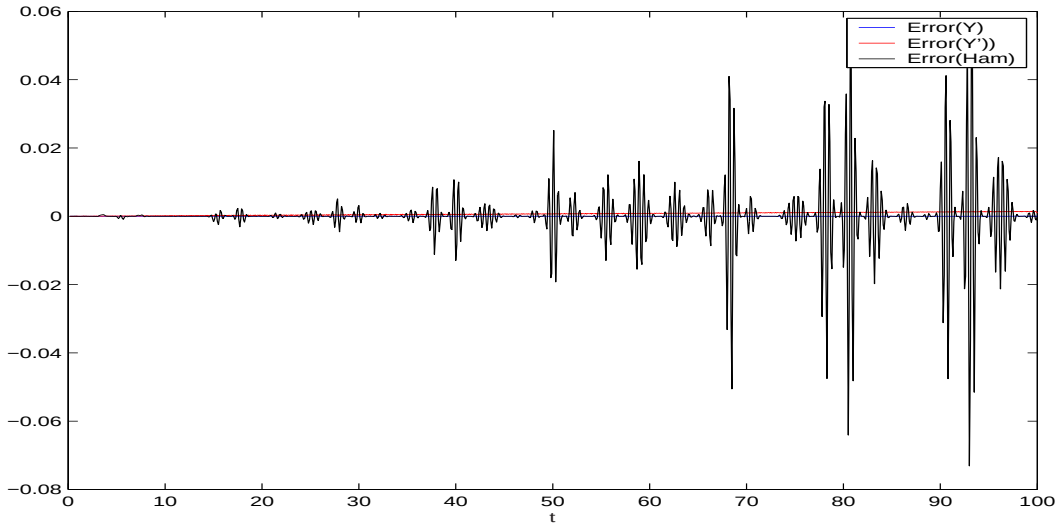
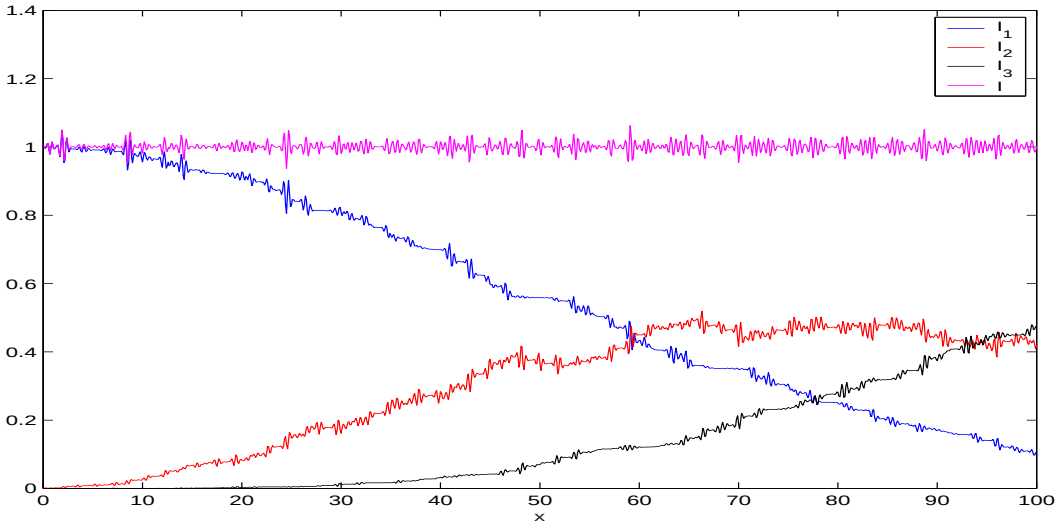


Figura 4.7.10: *Gráficas de las energías* $I_j = ((y'_{3+j}(t))^2 + \omega^2(y_{3+j}(t))^2)/2$, $(j = 1, 2, 3)$ *y de* $I = I_1 + I_2 + I_3$, *para el problema 4.2, usando la salida continua del GAUSS2-1 con 1000 puntos equidistantes en el intervalo* $[0, 100]$ *y tol* $= 10^{-9}$.



De las Figuras 4.7.2, 4.7.3 y 4.7.5, correspondientes a los problemas en EDPs, podemos decir que los errores globales coinciden en magnitud y en comportamiento cualitativo con los errores globales estimados obtenidos usando la salida densa. Además, teniendo en cuenta que la pendiente de las rectas de regresión es próxima a -1 en los problemas de EDPs, podemos inferir que en estos casos los errores globales crecen aproximadamente en proporción directa a la distancia entre el punto considerado y el punto inicial de integración. También se puede observar que las gráficas de los errores presentan ciertos *picos y valles*, pero en realidad estos *picos* resultan más acentuados por el carácter de la representación hecha en escala logarítmica, la cual produce pendientes más pronunciadas. En cualquier caso la diferencia entre los *picos y valles* próximos no suele pasar de una unidad, es decir esto supone una cifra significativa a lo más en la aproximación obtenida. Para el problema 4.6 (Figura 4.7) estos picos resultan bastante menos acentuados.

Debemos mencionar que los errores estimados se han computado usando la fórmula (4.5.5) para la componente y, pero para la estimación en la componente y' hemos reemplazado $\tau^{4/5}$ por $\tau^{3/5}$ en el lado derecho de la segunda ecuación de (4.5.5). Esto se ha hecho así, en virtud del resultado expresado en el Teorema 4.1. Hemos comprobado que esta estimación es mejor que cuando se usa $\tau^{4/5}$ para la componente y'. También se observa que para los 3 problemas de EDPs la correlación entre los errores estimados y los errores verdaderos es ligeramente mejor para la componente y, que para la componente y'. Este hecho suele ser así para cualquier tipo de problema oscilatorio. Aun así la estimación para la componente y' suele ser bastante fiable, en el sentido que está en el orden de magnitud del verdadero error.

Para los problemas Hamiltonianos provenientes de las ecuaciones diferenciales ordinarias consideradas, problemas 4.2 y 4.5, observamos en las Figuras 4.7.1 y 4.7.4 respectivamente, que mientras para el problema 4.2 (Fermi-Pasta-Ulam) los errores globales crecen en proporción directa a la distancia del punto considerado al punto inicial (pues las rectas de regresión de errores globales, en ambas componentes, tienen pendientes próximas a uno), para el problema de los dos cuerpos los errores globales van como el cuadrado de dicha distancia (las rectas de regresión de errores globales, en ambas componentes, tienen pendientes próximas a dos). Una explicación de este hecho proviene de que para el problema 4.2 el código realiza las integraciones prácticamente a paso fijo, con lo cual los errores globales deben crecer de forma lineal con la longitud del intervalo por ser el método de Gauss un integrador simpléctico. Sin embargo, para el problema 4.5 (de Kepler con excentricidad $e = 0{,}97$) el código toma pasos muy diferentes dependiendo de la posición del cuerpo que gira en torno al otro considerado

fijo (es decir si su posición es cercana al afelio o al perihelio). En este caso el integrador pierde la propiedad de que el error crezca de forma lineal con la longitud del intervalo al tomar tamaños de paso variables y el crecimiento del error global pasa de ser de tipo lineal a ser de tipo cuadrático.

Con respecto a la conservación de invariantes en los problemas de carácter Hamiltoniano, podemos decir lo siguiente:

1. Para el problema de los dos cuerpos con excentricidad $e = 0{,}97$, podemos ver en la Figura 4.7.6 (en escala logarítmica), la variación del error en el Hamiltoniano y su recta de regresión correspondiente. Esta recta posee una pendiente muy próxima a -1, lo cual nos dice que en este caso, el error en el Hamiltoniano crece de forma lineal con la longitud del intervalo (debe recordarse que los errores globales en las componentes crecían de forma cuadrática con la longitud del intervalo). Por otro lado, vemos que los errores en el Momento prácticamente se mantienen constantes (es decir, la pendiente de la recta de regresión es próxima a cero) a lo largo de toda la integración. Por tanto, el código GAUSS2 y su salida densa, parecen preservar este invariante, del mismo modo que el método de Gauss lo hace.

2. Para el problema 4.2 (Fermi-Pasta-Ulam), podemos observar en la Figura 4.7.9 (en escala normal), la variación del error global y del error en el Hamiltoniano. Ya habíamos visto en la Figura 4.7.1, que el error global en las componentes crecía de forma lineal con la distancia del punto al origen. Por otra parte, en la Figura 4.7.9, podemos apreciar que el error en el Hamiltoniano presenta un comportamiento bastante caótico con picos muy pronunciados en algunas zonas de anchura pequeña. A pesar de esto, se puede observar que en la mayor parte del intervalo de integración, el Hamiltoniano es quasi-preservado al usar la salida densa del código. De hecho, la recta de regresión de errores para el Hamiltoniano tiene pendiente muy próxima a cero, lo cual corrobora lo dicho anteriormente. Sin embargo, no tenemos una respuesta definitiva sobre porque los picos resultan mas pronunciados cuando la integración avanza. Debemos mencionar a este respecto que el Hamiltoniano es bastante atípico en el sentido de que posee un factor $\omega^2/2 = 1250$, que amplifica los errores en las componentes, véase la expresión del Hamiltoniano en (B.0.10), problema B.30 en Apéndice B. Además, en la salida densa el Hamiltoniano es evaluado en puntos que no son de red y esto propicia también la amplificación de errores. A este respecto, es conveniente reseñar que en el punto final

de integración (el cual es un punto de red) el error en el Hamiltoniano es pequeño, en concreto su valor es $1{,}6 \cdot 10^{-5}$, resultando ser ligeramente inferior al error en las componentes (es de tamaño $2{,}1 \cdot 10^{-5}$ para la y-componente y de tamaño $1{,}0 \cdot 10^{-3}$ para la y'-componente).

También hemos representado en la Figura 4.7.10, para este mismo problema la variación de las energías de los 3 muelles $I_j = (y'_{3+j}(t)^2 + \omega^2 y_{3+j}(t)^2)/2,\ (j = 1, 2, 3)$ y la suma de las mismas $I = I_1 + I_2 + I_3$, lo cual se dice ser *invariante adiabático* del sistema Hamiltoniano, aunque rigurosamente no se mantiene constante a lo largo del intervalo de integración, sino casi constante. Todo esto queda confirmado en la mencionada Figura, en la cual se observan las pequeñas fluctuaciones a lo largo de la integración, del mismo modo que pueden verse en las Figuras mostradas en la páginas 18-19 del texto Hairer et al. [56].

También hemos representado para los problemas 4.1 y 4.6 en las Figuras 4.7.7 y 4.7.8 respectivamente, las gráficas de las soluciones numéricas usando la salida densa del GAUSS2-1. En ellas se aprecia el carácter oscilatorio de ambos problemas, sin llegar a detectarse en las mismas los modos de altas frecuencias no-significativos (con pequeñas amplitudes), para los cuales tendríamos que usar resoluciones mucho más finas. Se da el hecho curioso que en el problema 4.1 las tres soluciones parecen idénticas, aunque en realidad no lo son, aunque sólo se diferencian en las amplitudes insignificantes asociadas a las dos frecuencias más altas.

4.8. Conclusiones

En el presente capítulo de esta memoria se desarrolla con cierta extensión los detalles del código GAUSS2 para problemas de valor inicial de segundo orden, de carácter especial. El código está basado en la fórmula de Gauss de dos etapas versión Runge–Kutta–Nyström, y está programado en FORTRAN 77-90 [47, 50] usando una política de paso variable. Se explica en detalle el esquema iterativo de tipo Newton propuesto junto con la elección de predictores para la resolución de las ecuaciones de etapas. Se estudian varios estimadores de error local, aunque sólo se recomiendan dos de ellos. La segunda versión del código [47] incorpora salida densa y un estimador del error global basado en la propiedad de Proporcionalidad del error global respecto a la tolerancia. El código es de propósito general y parece especialmente adecuado para la integración de sistemas de segundo orden de dimensiones medias y medio-altas, donde las matrices Jacobianas del sistema diferencial tengan estructuras

especiales tales como banda, circulantes, Toeplitz, etc, por ejemplo aquellas EDOs provenientes de Ecuaciones en Derivadas Parciales semidiscretizadas en las variables espaciales. En estos casos, sólo se requieren precisiones bajas o medias, debido a los errores introducidos en la discretización espacial de la EDP original. Para otros tipos de sistemas de grandes dimensiones, se debería incorporar al código un método adecuado para resolver los sistemas lineales resultantes, teniendo en cuenta la estructura especial de las matrices Jacobianas o bien usar técnicas de subespacios de Krylov. Nuestros experimentos confirman que ODEX2 es un buen código cuando se requieren precisiones altas y hay alguna frecuencia alta significativa. El código RADAU5 resulta adecuado para los problemas provenientes de las EDPs. Algunas mejoras en el código de modo que se evite hacer demasiados cambios de tamaño de paso redundaría notablemente en su eficacia. VODE no parece ser un código recomendable para esta clase de problemas.

Apéndice A

Conclusiones e Investigación Futura

En base a la investigación y experimentación numérica desarrolladas en esta memoria, podemos concluir lo siguiente:

1. Para problemas que comporten *Rigidez (Stiffness)*, la nueva iteración presentada en el Capítulo 2 de esta memoria, encaminada a resolver las ecuaciones de etapa de los métodos Runge–Kutta altamente implícitos, tales como aquellos de las familias Gauss, Radau IA, Radau IIA, Lobatto IIIA y Lobatto IIIC, parece ser eficiente y suele acelerar la velocidad de convergencia de las iteraciones de tipo Single-Newton. Además, la nueva iteración puede ser vista como una alternativa a la opción de la técnica de sobrerrelajación de Cooper y Butcher [22]. Para lograr una aceleración óptima se requeriría un conocimiento del espectro de la matriz Jacobiana del sistema de EDOS considerado. Un inconveniente de la nueva iteración es que no aumenta el orden en potencias de h del predictor (aproximación inicial), con las iteraciones. Por tanto una alternativa para evitar este hándicap en su aplicación general, sería usar en las primeras iteraciones, un esquema iterativo de tipo Single-Newton que aumente el orden de convergencia en potencias de h hasta lograr el orden del corrector, y luego usar el nuevo esquema iterativo para las siguientes iteraciones (dentro del mismo paso de integración) con el objetivo de acelerar la velocidad de convergencia, si ésta no se hubiera alcanzado con las iteraciones dadas con el esquema de tipo Single-Newton.

   El último aspecto requiere de una extensa experimentación que podría realizarse en el futuro con el objetivo de ver que tipo de esquemas podrían combinarse. También habría que probar la nueva iteración propuesta en códigos a paso variable, usando predictores convenientes, para ver como es su comportamiento en general.

2. La iteración de tipo Single-Newton presentada en el capítulo 3, para integrar problemas especiales de segundo orden, parece ser una alternativa más conveniente que la iteración Quasi-Newton, especialmente en problemas de dimensiones medias y altas. Esto se debe esencialmente a que la iteración de tipo Single-Newton presenta una buena velocidad de convergencia, que aunque es algo menor que la de iteración Quasi-Newton, evita la complejidad de las factorizaciones LU que conlleva la iteración Quasi-Newton. Las iteraciones de tipo Single-Newton desarrolladas han sido *optimizadas* para los métodos de Gauss de 2, 3 y 4 etapas. También se desarrollan predictores adecuados para iniciar las iteraciones del esquema iterativo desarrollado para estos métodos. Las iteraciones desarrolladas también parecen más eficientes que las propuestas en la literatura, por ejemplo la adaptación del esquema de Cooper y Butcher hecha para el método de Gauss de 2 etapas, por Gladwell y Thomas en [37].

   En el futuro se podría investigar la aceleración de la convergencia de los esquemas Single-Newton desarrollados es este capítulo 3. Especialmente, para el caso del los métodos de Gauss de 3 y 4 etapas, pues hay ciertas zonas (ciertos valores de $h\omega$ en la recta real) donde la razón de convergencia en el modelo lineal es algo lenta. La aceleración de la convergencia podría realizarse o bien mediante el nuevo esquema iterativo desarrollado en el Capítulo 2, o bien se usaría el esquema iterativo de tipo Single-Newton desarrollado en el capítulo 3, pero con la excepción de que si la convergencia no se ha alcanzado en una iteración fijada, por ejemplo la quinta iteración, entonces se realizaría una aceleración del esquema en este punto mediante una combinación adecuada de los valores de etapas calculados previamente y luego se continuaría con el esquema Single-Newton. Esta técnica se explica con mayor detalle en el capítulo 4 de esta memoria donde se ha aplicado con éxito al esquema iterativo Single-Newton para el Gauss de 2 etapas desarrollado en el capítulo 3.

3. En el capítulo 4 de esta memoria, se propone un código (GAUSS2) de propósito general para integrar Problemas de valor Inicial de segundo orden en precisión media. Decimos de propósito general, porque el código puede integrar cualquier tipo de problema de segundo orden dentro de una exactitud razonable, independientemente de la *Rigidez* del mismo. El código es especialmente adecuado para aquellos problemas de segundo orden que provienen de EDPs, pues en estos casos un par de cifras significativas en los resultados son a menudo suficientes, debido a que las soluciones numéricas ya se

encuentran contaminadas por los errores en las discretizaciones espaciales. El código está programado en FORTRAN 77-90 y puede descargarse de los portales [47, 50]. Además el código suministra, a petición del usuario, salida densa en puntos equiespaciados (aunque esto podría modificarse a gusto del usuario sin mas que modificar el driver de forma adecuada), la cual permite la representación gráfica de las trayectorias. También el código proporciona una estimación del error global cometido en la integración, el cual parece ser bastante fiable.

En el futuro se podrían desarrollar nuevos códigos de mayor precisión para integrar problemas oscilatorios. En concreto, creemos que el desarrollo de nuevos códigos basados en los métodos de Gauss de 3 etapas (orden 6) y de 4 etapas de orden (8) no es una tarea muy complicada, después de la extensa experiencia adquirida con el desarrollo del código GAUSS2. Debemos notar que ya los predictores y los esquemas iterativos de tipo Single-Newton han sido desarrollados en el capítulo 3 y en este aspecto sólo faltaría optimizarlos para acelerar la convergencia en alguna iteración prefijada. Por otra parte, un estimador de error local en la y-componente parece la labor más ardua a desarrollar para el caso de los métodos de Gauss de 3 y 4 etapas. Esto podría hacerse usando técnicas similares a las desarrolladas aquí en el capítulo 4 para el método de Gauss de 2 etapas. En cuanto a la estimación del error global, podemos decir que las técnicas basadas en la propiedad de que los errores globales son proporcionales a una potencia de la Tolerancia, siguen siendo de aplicación en estos casos.

Apéndice B

Anexo sobre Problemas especiales de segundo orden

En este anexo se recogen una buena colección de problemas de segundo orden de carácter oscilatorio o Hamiltoniano, tomados de la literatura y que han sido ampliamente usados como problemas test para probar los métodos numéricos. Se hace también referencia a la literatura de donde han sido tomados.

1. **Problema B.1** *Problema lineal simple con solución cosenoidal*

$$y''(t) = -\omega^2 y,$$

$$y(0) = \eta, \quad y'(0) = 0,$$

con solución exacta

$$y(t) = \eta \cos(\omega t).$$

2. **Problema B.2** *Problema lineal simple con solución senoidal*

$$y''(t) = -\omega^2 y,$$

$$y(0) = 0, \quad y'(0) = \eta\omega,$$

con solución exacta

$$y(t) = \eta \sin(\omega t).$$

3. **Problema B.3** *Problema lineal homogéneo ([32, Ejemplo 3, p. 86])*

$$y''(t) = \begin{pmatrix} \omega - 2 & 2\omega - 2 \\ 1 - \omega & 1 - 2\omega \end{pmatrix} y(t), \quad t \in [0, 10]$$

$$y(0) = \begin{pmatrix} 2 \\ -1 \end{pmatrix}, \quad y'(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

cuya solución exacta es

$$y(t) = \big(2\cos(t), -\cos(t)\big)^T, \quad \forall \omega > 0.$$

Valores recomendados $\omega = 25$ *y* $\omega = 1600$.

4. **Problema B.4** *Problema lineal no homogéneo ([32, Ejemplo 1, p. 84])*

$$y''(t) = -\omega^2 y(t) + 0{,}01\sin(t), \quad t \in [0, 20]$$

$$y(0) = 1, \quad y'(0) = \omega + \frac{0{,}01}{\omega^2 - 1}, \quad \omega \gg 1,$$

con solución exacta

$$y(t) = \cos(\omega t) + \sin(\omega t) + \frac{0{,}01}{\omega^2 - 1}\sin(t).$$

$\omega = 10$.

5. **Problema B.5** *Sistema lineal Hamiltoniano ([38, Problema 1, p. 140])*

$$y''(t) = -\frac{1}{2}\begin{pmatrix} \omega^2 + 1 & \omega^2 - 1 \\ \omega^2 - 1 & \omega^2 + 1 \end{pmatrix} y(t), \quad \omega \in \mathbb{R}, \quad t \in [0, 20]$$

$$y(0) = \begin{pmatrix} 1 + \lambda \\ 1 - \lambda \end{pmatrix}, \quad y'(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \lambda \in \mathbb{R}$$

con solución exacta

$$y(t) = \big(\cos(t) + \lambda\cos(100t), \cos(t) - \lambda\cos(100t)\big)^T.$$

$\omega = 10^2$, $\lambda = 1, 10^{-2}, 10^{-4}$.

6. **Problema B.6** *Ecuación de Duffing no amortiguada. Problema no lineal ([21, Problema 4, p. 68])*

$$y''(t) = -y(t) - y(t)^3 + 0{,}002\cos(1{,}01t), \quad t \in [0, 40\pi]$$

$$y(0) = 0{,}20042672806699998, \quad y'(0) = 0,$$

con solución exacta

$$\begin{aligned} y(t) &= 0{,}200179477536\cos(1{,}01t) + 0{,}246946143 \times 10^{-3}\cos(3{,}03t) \\ &\quad +0{,}304014 \times 10^{-6}\cos(5{,}05t) + 0{,}374 \times 10^{-9}\cos(7{,}07t). \end{aligned}$$

7. **Problema B.7** *Sistema no lineal ([38, Problema 2, p. 142])*

$$y''(t) = \begin{pmatrix} -1 & 0 \\ 0 & -\omega^2 \end{pmatrix} y(t) + 0{,}02 \begin{pmatrix} 1 \\ 1 \end{pmatrix} \big(\|y(t)\|_2^2 + \cos^2(t) - 1\big), \quad t \in [0, 10000]$$

$$y(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad y'(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

con solución exacta

$$y(t) = \big(\sin(t), 0\big)^T, \quad \forall \omega > 0.$$

$\omega = 10^3$.

8. **Problema B.8** *Sistema no lineal Hamiltoniano ([38, Problema 5. p. 150])*

$$y_1''(t) = -\frac{y_1(t)}{r^3(t)} - 2\omega^2\big(r^2(t) - 1\big)^9 y_1(t)$$

$$y_2''(t) = -\frac{y_2(t)}{r^3(t)} - 2\omega^2\big(r^2(t) - 1\big)^9 y_2(t),$$

$$y(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad y'(0) = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

$t \in [0, 10000]$ *y* $r(t) = \sqrt{y_1^2(t) + y_2^2(t)}$*, con solución exacta*

$$y(t) = \big(\cos(t), \sin(t)\big)^T.$$

$\omega = 10^4$.

9. **Problema B.9** *Problema no lineal ([38, Problema 5, p. 159])*

$$y_1''(t) = -4t^2 y_1(t) - \frac{2\sin(t^2)}{\sqrt{y_1^2(t) + \sin^2(t^2)}}$$
$$y_2''(t) = -\eta^2 y_2(t) + \omega\big(y_1^2(t) + \sin^2(t^2) - 1\big), \quad t \in [0, 50]$$
$$y(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad y'(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

con solución exacta

$$y(t) = \big(\cos(t^2), 0\big)^T, \quad \forall \eta \in \mathbb{R}.$$

$\eta = 10^4$ *y* $\omega = 0{,}1$.

10. **Problema B.10** *Ecuación de la órbita. Un situación idealizada (no lineal) ([62, Problema 5.5, p. 425])*

$$y_1''(t) = -4t^2 y_1(t) - \frac{2y_2(t)}{\sqrt{y_1^2(t) + y_1^2(t)}}$$
$$y_2''(t) = -4t^2 y_2(t) + \frac{2y_1(t)}{\sqrt{y_1^2(t) + y_1^2(t)}} \quad , \quad t \in [0, 2\sqrt{\pi/2}],$$
$$y(0) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad y'(0) = \begin{pmatrix} -\sqrt{2\pi} \\ 0 \end{pmatrix},$$

con solución exacta

$$y(t) = \big(\cos(t^2), \sin(t^2)\big)^T.$$

11. **Problema B.11** *Problema no lineal. Esta ecuación diferencial es un caso particular de la ecuación undamped Duffing's ([21, Problema 3, p. 68])*

$$y''(t) = -\big(1 + 0{,}01y^2(t)\big)y(t) + 0{,}01\cos^3(t), \quad t \in [0, 40\pi]$$
$$y(0) = 1, \quad y'(0) = 0,$$

con solución exacta

$$y(t) = \cos(t).$$

12. **Problema B.12** *Problema no lineal.*

$$y''(t) = -10^4(y - sen(t)) - sen(t), \quad t \in [0, 10]$$
$$y(0) = 0, \quad y'(0) = 1,$$

con solución exacta

$$y(t) = sen(t).$$

13. **Problema B.13** *Problema no lineal.*

$$y''(t) = -4t^2 y - 2sen(t^2), \quad t \in [0, 10]$$

$$y(0) = 1, \quad y'(0) = 0,$$

con solución exacta

$$y(t) = \cos(t^2).$$

14. **Problema B.14** *Problema Oscilatorio ([37, Problema 5.7, p. 201])*

$$y''(t) = -\sinh\big(y(t)\big), \quad t \in [0, 6]$$

$$y(0) = 1, \quad y'(0) = 0,$$

con solución numérica en $t = 6$

$$y(6) = 0{,}9954139409446862, \quad y'(6) = -0{,}1036661333139567.$$

15. **Problema B.15** *La ecuación que describe la vibración de una barra empotrada ([62, Problema 5.6., p. 426])*

$$\begin{cases} \dfrac{q}{g}\dfrac{\partial^2 u}{\partial t^2} + EI\dfrac{\partial^4 u}{\partial x^4} = 0, \quad 0 \le x \le l, \quad t \ge 0 \\ u(0,t) = u_x(0,t) = 0 \\ u(x,0) = f(x), \quad u_t(x,0) = 0 \\ u_{xx}(l,t) = u_{xxx}(l,t) = 0, \end{cases} \tag{B.0.1}$$

donde $f(x)$ *es dada por*

$$f(x) = A[\cosh(\omega x) - \cos(\omega x) - \frac{\cosh(\omega l) + \cos(\omega l)}{\sinh(\omega l) + \sin(\omega l)}\big(\sinh(\omega x) - \sin(\omega x)\big)],$$

$$A = 0{,}1, \quad l = 22, \quad q/g = 50, \quad EI = 10^4, \quad \omega = 0{,}08523200128726258.$$

Primeramente debemos semidiscretizar (B.0.1). Por lo cual definimos una red equidistante $x_j := j\Delta, \Delta = l/N, (j = 1, \ldots, N)$ *y usamos la aproximación simétrica de segundo orden*

$$\frac{\partial^4 u_i}{\partial x^4} \approx \frac{u_{i+2} - 4u_{i+1} + 6u_i - 4u_{i-1} + u_{i-2}}{(\Delta x)^4} \ .$$

La sustitución de las condiciones de frontera en la discretización, producen el sistema lineal

$$\begin{pmatrix} u_1'' \\ u_2'' \\ u_3'' \\ \vdots \\ u_{N-2}'' \\ u_{N-1}'' \\ u_N'' \end{pmatrix} = -\frac{1}{200\Delta^4} \begin{pmatrix} 7 & -4 & 1 & & & & \\ -4 & 6 & -4 & 1 & & 0 & \\ 1 & -4 & 6 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 6 & -4 & 1 \\ & 0 & & 1 & -4 & 5 & -2 \\ & & & & 2 & -4 & 2 \end{pmatrix} \cdot \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-2} \\ u_{N-1} \\ u_N \end{pmatrix}. \tag{B.0.2}$$

Se puede verificar que los autovalores de la matriz Jacobiana son reales independientemente del valor de N.

16. **Problema B.16** *Es similar al problema B.15 , es decir,*

$$y''(t) = -200(\Delta x)^{-4} J y(t) + r(t), \quad y(0) = y_0,\ y'(0) = y_0', \quad \Delta x = 22/N,\ N = 90,$$

donde la matriz de banda J $(N \times N)$ está dada en la ecuación (B.0.2), y $r(t) = \mathcal{O}((\Delta x)^2)$, y_0 y y_0' son elegidos para que la solución del sistema ODE coincida en cada línea x_j $(x_j = j\Delta x)$ con la solución exacta de la ecuación diferencial parcial

$$y_{tt}(x,t) + 200\, y_{xxxx}(x,t) = 0,\ t > 0,\ 0 < x < l = 22,$$

descrita en (B.0.1) (Problema 13). Donde, $r(t) = (r_i(t))_{i=1}^N$ está dado por

$$r_i(t) = \alpha(1 + \varepsilon_i(t)^2)\varepsilon_i(t), \quad \varepsilon_i(t) = \cos(x_i/4)\cos(t/16), \quad i = 1, 2, \ldots, N,$$
$$\alpha := 8^{-1} \textstyle\sum_{k=1}^{\infty} (-4)^{-k}(1 - 4^{-k-1})(\Delta x)^{2k}/(2k+4)!.$$

17. **Problema B.17** *. Es la ecuación diferencial parcial no lineal [13]*

$$\begin{cases} y_{tt}(x,t) + \big(1 + y(x,t)^2\big) y_{xxxx}(x,t) = \eta^4 y(x,t)^3, \quad -\pi/\eta < x \le \pi/\eta,\ t > 0, \\ y(x,0) = \cos \eta x, \\ y_t(x,0) = 0, \quad \eta = 1/4, \end{cases} \tag{B.0.3}$$

con solución $2\pi/\eta$–periódica en el espacio, es decir,

$$y(x \pm 2\pi/\eta, t) = y(x,t), \quad \forall x, t \in \mathbb{R}.$$

La solución exacta está dada por $y(x,t) = \cos(\eta x)\cos(\eta^2 t)$. Discretizando en el espacio como en el problema 3.3 y considerando las condiciones de periodicidad $2\pi/\eta$,

$$y_0(t) = y_N(t), \quad y_{-1}(t) = y_{N-1}(t), \quad y_{N+1}(t) = y_1(t), \quad y_{N+2}(t) = y_2(t). \tag{B.0.4}$$

Obtenemos el problema de valor inicial de segundo orden,

$$\left.\begin{array}{c} y_i''(t) + (\Delta x)^{-4}(1+y_i^2)(y_{i-2} - 4y_{i-1} + 6y_i - 4y_{i+1} + y_{i+2}) = \eta^4 y_i^3 + r_i(t), \\ \Delta x = 2\pi\eta^{-1}N^{-1}, \quad y_i(0) = \cos(\eta x_i), \quad y_i'(0) = 0, \quad x_i = -\eta^{-1}\pi + i\Delta x, \\ i = 1, 2, \ldots, N. \end{array}\right\} \tag{B.0.5}$$

Los términos forzados

$$r_i(t) = \alpha(1+\varepsilon_i(t)^2)\varepsilon_i(t), \quad \varepsilon_i(t) = \cos(\eta x_i)\cos(\eta^2 t)$$

con

$$\alpha := 8^{-1}\sum_{k=1}^{\infty} \frac{(1-4^{-k-1})}{(-4)^k}\frac{(\Delta x)^{2k}}{(2k+4)!},$$

han sido añadidos del lado derecho de la ODE (B.0.5) y el PDE (B.0.3) ya que tienen la misma solución en las correspondientes líneas x_i. Este es un problema no lineal de dimensión de media a alta donde las frecuencias más altas no son significativas.

18. **Problema B.18** *El modelo con rigidez en elastodinámica lineal definido por la EDP [38, Problema 3, p. 145],*

$$\begin{cases} \dfrac{\partial^2 u}{\partial t^2} + \dfrac{\partial^4 u}{\partial x^4} - x(1-x)\dfrac{\partial^2 u}{\partial x^2} - u = 0, \quad 0 < x < l, \quad t > 0 \\ u(0,t) = u(l,t) = 0 \\ u_{xx}(0,t) = u_{xx}(l,t) = -2\cos(t) \\ u(x,0) = x(1-x), \quad u_t(x,0) = 0, \end{cases} \tag{B.0.6}$$

con solución exacta $u(x,t) = x(1-x)\cos(t)$.
Primeramente debemos semidiscretizar (B.0.6). Por lo tanto definimos una red equidistante $x_j := j\Delta, \Delta = l/(N+1), (j = 1, \ldots, N)$ y usamos las aproximaciones de segundo orden

$$\frac{\partial^4 u_i}{\partial x^4} \approx \frac{u_{i+2} - 4u_{i+1} + 6u_i - 4u_{i-1} + u_{i-2}}{(\Delta x)^4},$$

$$\frac{\partial^2 u_i}{\partial x^2} \approx \frac{u_{i+1} - 2u_i + 4u_{i-1}}{(\Delta x)^2} \ .$$

En nuestro test usamos $l = 1$. Resultando un sistema lineal de EDOs no homogénea de dimensión N.

19. **Problema B.19** *Sistema moderadamente stiff de dimensión 2 ([37, Problema 5.8, p. 201])*

$$y_1''(t) = -\sinh\big(y_1(t) + y_2(t)\big), \quad t \in [0, 6]$$
$$y_2''(t) = -10^4 y_2(t),$$
$$y(0) = \begin{pmatrix} 1 \\ 10^{-8} \end{pmatrix}, \quad y'(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

con solución exacta en la segunda componente

$$y_2(t) = 10^{-8}\cos(100t),$$

y solución numérica en $t = 6$ para la primera componente

$$y_1(6) = 0{,}9954139409446862, \quad y_1'(6) = -0{,}1036661333139567.$$

20. **Problema B.20** *El problema de los dos cuerpos (no lineal) ([21, Problema 6, p. 71])*

$$y_1''(t) = -\frac{y_1(t)}{r^3(t)}$$

$$y_2''(t) = -\frac{y_2(t)}{r^3(t)} \ ,$$

$$y(0) = \begin{pmatrix} 1 - e \\ 0 \end{pmatrix}, \quad y'(0) = \begin{pmatrix} 0 \\ \sqrt{\frac{1+e}{1-e}} \end{pmatrix},$$

$t > 0$, $r(t) = \sqrt{y_1^2(t) + y_2^2(t)}$. *La solución exacta es*

$$y(t) = \Big(\cos(E) - e, \sqrt{1 - e^2}\sin(E)\Big)^T,$$

donde e es la excentricidad de la orbita, y E es la excentricidad anómala la cual es expresada como una función implícita de la variable independiente t, por medio de la ecuación de Kepler

$$t = E - e\sin(E).$$

21. **Problema B.21** *([21, Problema 5, p. 70]).*

La función $y(t) = \sqrt{t}J_0(10x)$*, donde* J_0 *es la función de Bessel de primera especie de orden cero, la cual satisface la ecuación diferencial*

$$y''(t) = -\Big(100 + \frac{1}{4(t+1)^2}\Big)y(t), \quad t \in [0, 9],$$

con valores iniciales,

$$y(0) = J_0(10), \quad y'(0) = \frac{1}{2}J_0(10) - 10J_1(10),$$

donde J_1 *es la función de Bessel de orden 1. Con solución numérica en* $t = 9$

$$y(9) = 0{,}063200807936514188, \quad y'(9) = 2{,}439550232600525649.$$

22. **Problema B.22** *El problema no lineal de tipo Klein–Gordon ([38, Problema 6, p. 54])*

$$\begin{cases} \dfrac{\partial^2 u}{\partial t^2} - \dfrac{\partial^2 u}{\partial x^2} + b(x)u + c(x)u^3 = 0, \quad 0 < x < l, \quad t > 0 \\ u(0,t) = u(l,t) = 0 \\ u(x,0) = 0, \quad u_t(x,0) = \omega x(x-1), \end{cases} \tag{B.0.7}$$

donde

$$b(x) = \omega^2 + k^2 + \frac{2}{x(x-1)} \ , \quad c(x) = -\frac{2k^2}{x^2(x-1)^2} \ ,$$

con solución exacta

$$u(x,t) = x(x-1)sn(\omega t, k/\omega), \quad 0 \le k/\omega < 1.$$

Primeramente debemos semidiscretizar (B.0.7). Por lo tanto definimos una red equidistante $x_j := j\Delta, \Delta = l/(N+1), (j = 1, \ldots, N)$ *y usamos la aproximación de segundo orden*

$$\frac{\partial^2 u_i}{\partial x^2} \approx \frac{u_{i+1} - 2u_i + 4u_{i-1}}{(\Delta x)^2} \ .$$

En nuestro test usamos $l = 1$*. Resultando un sistema de EDOs de dimensión* N*.*

23. **Problema B.23** *El problema del péndulo (no lineal) ([16])*

$$y''(t) = -\sin(y), \quad t \in [0, 2\pi]$$

$$y(0) = 0, \quad y'(0) = 1,$$

con solución numérica en $t = 2\pi$

$$y(2\pi) = 0{,}443944662290259, \quad y'(2\pi) = 0{,}897846803312944.$$

24. **Problema B.24** *Outer planet system. Sistema de ODEs no lineal de dimensión 18 ([56, pp. 10–12], [16]).*

Este sistema describe el movimiento de los cinco planetas mas externos (Plutón esta considerado aquí como planeta) y del sol mismo. Este sistema ha sido estudiado extensivamente por los astrónomos.

Denotaremos por $p_i = (p_{ik})_{k=1}^3$, $q_i = (q_{ik})_{k=1}^3$ *los vectores de momento y posición respectivamente, para cada uno de los seis cuerpos celestes involucrados* $(i = 1, \ldots, 6)$. *Las unidades elegidas son: la masa relativa al sol, de modo que el sol tiene masa 1. Hemos tomado siguiendo los estudios astronómicos, ver por ejemplo [56, pp. 10–12] y las referencias allí dadas,*

$$m_0 = 1{,}00000597682$$

para compensar la masa de los planetas internos. Las distancias son expresadas en unidades astronómicas (1[A.U.]=149597870 [km]), el tiempo en días en la tierra, y la constante de gravitación es

$$G = 2{,}95912208286 \cdot 10^{-4}.$$

Los valores iniciales para el sol son tomados como $q_6 = (0, 0, 0)^T$ *y* $\dot{q}_6(0) = (0, 0, 0)^T$. *Todos los otros datos (masa de los planetas, sus posiciones y velocidades iniciales) son dadas en la tabla B.0.1. Se obtiene así un sistema no lineal de dimensión 18 en las posiciones* q_{ik},

$$\ddot{q}_{i,k} = G \sum_{\substack{j=1 \\ j \neq i}}^{6} \frac{M_j (q_{3(j-1)+k} - q_{3(i-1)+k})}{d_{i,j}}, \quad 1 \leq k \leq 3, \quad 1 \leq i \leq 6$$

con

$$d_{i,j} = \left(\sqrt{\sum_{k=1}^{3} (q_{3(i-1)+k} - q_{3(j-1)+k})^2} \right)^3, \qquad 1 \leq i, j \leq 6$$

y $d_{j,i} = d_{i,j}$

Tabla B.0.1: Datos del problema del sistema solar

planeta	masa	posición inicial	velocidad inicial
Júpiter	$m_1 = 0{,}000954786104043$	$-3{,}5023653$ $-3{,}8169847$ $-1{,}5507963$	$0{,}00565429$ $-0{,}00412490$ $-0{,}00190589$
Saturno	$m_2 = 0{,}000285583733151$	$9{,}0755314$ $-3{,}0458353$ $-1{,}6483708$	0.00168318 0.00483525 0.00192462
Uranio	$m_3 = 0{,}0000437273164546$	$8{,}3101420$ $-16{,}2901086$ $-7{,}2521278$	$0{,}00354178$ $0{,}00137102$ $0{,}00055029$
Neptuno	$m_4 = 0{,}0000517759138449$	$11{,}4707666$ $-25{,}7294829$ $-10{,}8169456$	$0{,}00288930$ $0{,}00114527$ $0{,}00039677$
Plutón	$m_5 = 1/(1{,}3 \cdot 10^8)$	$-15{,}5387357$ $-25{,}2225594$ $-3{,}1902382$	$0{,}00276725$ $-0{,}00170702$ $-0{,}00136504$

25. **Problema B.25** *Problema no lineal ([62, Problema 5.4, p. 425])*

$$y''(t) = -y^3(t), \quad t \in [0, 315]$$

$$y(0) = 0, \quad y'(0) = 1,$$

con solución numérica en $t = 315$

$$y(315) = -0{,}6488047613698329, \quad y'(315) = -0{,}9999955699475079.$$

26. **Problema B.26** *Problema no lineal ([62, Problema 5.2, p. 424])*

$$y''(t) = -\ln(2+t)y(t), \quad t \in [0, 100]$$

$$y(0) = 0, \quad y'(0) = 1,$$

con solución numérica en $t = 100$

$$y(100) = 0{,}4310918570834484, \quad y'(100) = -0{,}1246299559189595.$$

27. **Problema B.27** *Problema no lineal ([61, Problema 3.2, p. 561])*

$$\begin{aligned} y_1''(t) &= -7y_1(t) + 3y_2(t) + \omega \sin^3\big(y_1(t) - y_2(t)\big) \\ y_2''(t) &= 2y_1(t) - 6y_2(t) + \omega \cos^3\big(y_1(t) - y_2(t)\big), \quad t \in [0, 10], \end{aligned}$$

$$y(0) = y'(0) = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

con solución exacta para $\omega = 0$

$$y_1(t) = \frac{3}{5}\big(\cos(2x) - \cos(3x)\big) + \frac{1}{10}\big(3\sin(2x) - 2\sin(3x)\big),$$

$$y_2(t) = \frac{1}{5}\big(3\cos(2x) + 2\cos(3x)\big) + \frac{1}{30}\big(9\sin(2x) + 4\sin(3x)\big),$$

y con soluciones numéricas en $t = 10$ *para los valores de* ω *siguientes vienen dadas por:*

$\omega = 0{,}1$

$$y_1(10) = 0{,}6713447018924612, \quad y_1'(10) = -0{,}2639980117628816,$$

$$y_2(10) = 0{,}4122418557493486, \quad y_2'(10) = 0{,}3901236575103554.$$

$\omega = 0{,}2$

$$y_1(10) = 0{,}714267793231979, \quad y_1'(10) = -0{,}2546361428881664,$$
$$y_2(10) = 0{,}3776901967300211, \quad y_2'(10) = 0{,}3735105857769548.$$

$\omega = \dfrac{1}{3}$

$$y_1(10) = 0{,}7637107577148519, \quad y_1'(10) = -0{,}2407272046501208,$$
$$y_2(10) = 0{,}3351470430544931, \quad y_2'(10) = 0{,}3392386831354331.$$

28. **Problema B.28** *La ecuación de la onda linealizada ([82, Problema 2. p. 947])*

$$\begin{cases} \dfrac{\partial^2 u}{\partial t^2} - 4\dfrac{\partial^2 u}{\partial x^2} - \sin(t)\cdot\cos\left(\dfrac{\pi x}{b}\right) = 0, \quad 0 \le x \le b = 100, \quad t \in [0, 40\pi] \\ u_x(0,t) = u_x(b,t) = 0 \\ u(x,0) = 0, \quad u_t(x,0) = \dfrac{b^2}{4\pi^2 - b^2}\cdot\cos\left(\dfrac{\pi x}{b}\right), \end{cases} \tag{B.0.8}$$

con solución teórica

$$u(x,t) = \frac{b^2}{4\pi^2 - b^2}\cdot\sin(t)\cdot\cos\left(\frac{\pi x}{b}\right).$$

Discretizando $\partial^2 u/\partial x^2$ usando diferencias simétricas de cuarto orden en los puntos del intervalo y en las fronteras, y usando la fórmula de quinto orden

$$\frac{\partial u_i}{\partial x} = \frac{-137u_{i-1} + 300u_{i-2} - 300u_{i-3} + 200u_{i-4} - 75u_{i-5} + 12u_{i-6}}{60\Delta x}, \tag{B.0.9}$$

resulta el sistema lineal de EDOs:

$$\begin{pmatrix} y_1'' \\ y_2'' \\ y_3'' \\ \vdots \\ y_{N-2}'' \\ y_{N-1}'' \\ y_N'' \end{pmatrix} = \frac{4}{(\Delta x)^2} \begin{pmatrix} -\frac{415}{72} & 8 & -3 & \frac{8}{9} & -\frac{1}{8} & & & \\ \frac{257}{144} & -\frac{10}{3} & \frac{7}{4} & -\frac{2}{9} & \frac{1}{48} & & 0 & \\ -\frac{1}{12} & \frac{4}{3} & -\frac{5}{2} & \frac{4}{3} & -\frac{1}{12} & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & -\frac{1}{12} & \frac{4}{3} & -\frac{5}{2} & \frac{4}{3} & -\frac{1}{12} \\ & & 0 & \frac{1}{48} & -\frac{2}{9} & \frac{7}{4} & -\frac{10}{3} & \frac{257}{144} \\ & & & -\frac{1}{8} & \frac{8}{9} & -3 & 8 & -\frac{415}{72} \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{N-2} \\ y_{N-1} \\ y_N \end{pmatrix}$$

$$+\sin(t)\cdot\begin{pmatrix}\cos\left(\frac{1\cdot\Delta x}{b}\cdot\pi\right)\\ \cos\left(\frac{2\cdot\Delta x}{b}\cdot\pi\right)\\ \cos\left(\frac{3\cdot\Delta x}{b}\cdot\pi\right)\\ \vdots\\ \cos\left(\frac{(N-2)\cdot\Delta x}{b}\cdot\pi\right)\\ \cos\left(\frac{(N-1)\cdot\Delta x}{b}\cdot\pi\right)\\ \cos\left(\frac{N\cdot\Delta x}{b}\cdot\pi\right)\end{pmatrix}.$$

Con $\Delta x = \dfrac{b}{N-1}$ *, obtenemos un sistema lineal con coeficientes constante de dimensión* N*.*

29. **Problema B.29** *Problema lineal no homogéneo ([82, Problema 1, p. 947])*

$$y''(t) = \begin{pmatrix}\frac{1}{100} & -\frac{1}{10}\\ -\frac{1}{10} & \frac{1}{100}\end{pmatrix} y(t) + \begin{pmatrix}0\\ \sin(t)\end{pmatrix}, \quad t\in[0,10\pi],$$

$$y(0) = \begin{pmatrix}1\\ 1\end{pmatrix}, \quad y'(0) = \begin{pmatrix}-\frac{1000}{10101}\\ \frac{10100}{10101}\end{pmatrix},$$

con solución teórica:

$$y(t) = \Big(\cos(\tfrac{3}{10}t) - \tfrac{1000}{10101}\sin(t), \cos(\tfrac{3}{10}t) - \tfrac{10100}{10101}\sin(t)\Big)^T.$$

30. **Problema B.30** *Problema no lineal ([56, p. 17]).*
El problema de Fermi–Pasta– Ulam (1955) es un modelo simple para simulaciones en estadística mecánica. Este problema está inspirado por la modelación de un muelle con $2m$ *puntos de masa, el cual tiene acoplados* $2m$ *resortes unidos linealmente en los puntos de masa. Se combinan resortes que tienen constantes de recuperación suaves con otros resortes más rígidos. Además el muelle conjunto tiene los extremos fijos. Las variables* $q_1,\ldots,q_{2m}$ $(q_0 = q_{2m+1} = 0)$ *representan los desplazamientos de los puntos de masa respecto de la posición de equilibrio, y* $p_i = \dot{q}_i$ *representan las velocidades respectivas. El movimiento queda descrito por el sistema Hamiltoniano con energía total dada por*

$$H(p,q) = \frac{1}{2}\sum_{i=1}^{m}\left(p_{2i-1}^2 + p_{2i}^2\right) + \frac{\omega^2}{4}\sum_{i=1}^{m}\left(q_{2i} - q_{2i-1}\right)^2 + \sum_{i=0}^{m}\left(q_{2i+1} - q_{2i}\right)^4,$$

donde ω *es una constante que toma valores tales como* $\omega = 20, 50, 100$*.*

Es natural introducir las nuevas variables

$$\begin{aligned} x_i &= (q_{2i} + q_{2i-1})/\sqrt{2}, \qquad & x_{m+i} &= (q_{2i} - q_{2i-1})/\sqrt{2}, \\ y_i &= (p_{2i} + p_{2i-1})/\sqrt{2}, \qquad & y_{m+i} &= (p_{2i} - p_{2i-1})/\sqrt{2}, \end{aligned}$$

donde $x_i (i = 1, \ldots, m)$ *representa el desplazamiento del i-ésimo resorte (escalado), y* y_i, y_{m+i} *sus velocidades (o momento). Con este cambio de coordenadas, el movimiento en las nuevas variables es nuevamente descrito por un sistema Hamiltoniano, con*

$$\begin{aligned} H(y,x) \;=\; & \frac{1}{2}\sum_{i=1}^{2m} y_i^2 + \frac{\omega^2}{2}\sum_{i=1}^{m} x_{m+i}^2 + \frac{1}{4}\Big((x_1 - x_{m+1})^4 \\ & + \sum_{i=1}^{m-1}\big(x_{i+1} - x_{m+i+1} - x_i - x_{m+i}\big)^4 + (x_m + x_{2m})^4\Big). \end{aligned} \tag{B.0.10}$$

Además de tratarse de un sistema Hamiltoniano, se tiene también que la energía total de intercambio entre los resortes, la cual viene dada por $I = I_1 + \cdots + I_m$ *permanece constante. Aquí la energía correspondiente al del* j*–ésimo resorte viene dada por*

$$I_j(x_{m+j}, y_{m+j}) = \frac{1}{2}\big(y_{m+j}^2 + \omega^2 x_{m+j}^2\big).$$

Además $I(x, y)$ *es un invariante adiabático del sistema Hamiltoniano. Siguiendo las referencias arriba mencionadas, elegimos* $m = 3$, $\omega = 50$, *y se obtiene el sistema Hamiltoniano siguiente cuyas condiciones iniciales son,*

$$x_1(0) = 1, \quad y_1(0) = 1, \quad x_4(0) = \omega^{-1}, \quad y_4(0) = 1,$$

y cero para los restantes valores iniciales.

$$\begin{aligned}
x_1'' &= (x_2 - x_5 - x_1 - x_4)^3 - (x_1 - x_4)^3, && x_1(0) = 1, && x_1'(0) = 1, \\
x_2'' &= -(x_2 - x_5 - x_1 - x_4)^3 + (x_3 - x_6 - x_2 - x_5)^3, && x_2(0) = 0, && x_2'(0) = 0, \\
x_3'' &= -(x_3 - x_6 - x_2 - x_5)^3 - (x_3 - x_6)^3, && x_3(0) = 0, && x_3'(0) = 0, \\
x_4'' &= (x_2 - x_5 - x_1 - x_4)^3 + (x_1 - x_4)^3 - \omega^2 x_4, && x_4(0) = \omega^{-1}, && x_4'(0) = 1, \\
x_5'' &= (x_2 - x_5 - x_1 - x_4)^3 + (x_3 - x_6 - x_2 - x_5)^3 - \omega^2 x_5, && x_5(0) = 0, && x_5'(0) = 0, \\
x_6'' &= (x_3 - x_6 - x_2 - x_5)^3 - (x_3 - x_6)^3 - \omega^2 x_6, && x_6(0) = 0, && x_6'(0) = 0,
\end{aligned}$$

31. **Problema B.31** *Problema no lineal ([46, Problema 2, p. 947])*

$$\begin{aligned} y_1''(t) &= -\sinh\big(y_1(t) + y_2(t)\big) \\ y_2''(t) &= -\frac{\omega}{1+t} y_2(t), \quad t \in [0, 4] \end{aligned}$$

$$y(0) = \begin{pmatrix} 1 \\ 10^{-8} \end{pmatrix}, \quad y'(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

En nuestro test elegimos $\omega = 10^{10}$.

32. **Problema B.32** *La ecuación de la onda no lineal([63, Problema 4.5, p. 610])*

$$\begin{cases} \dfrac{\partial^2 u}{\partial t^2} = gd(x)\dfrac{\partial^2 u}{\partial x^2} + \dfrac{1}{4}\omega^2(x,u)u, \quad 0 \le x \le b, \quad t \ge 0, \\ u_x(0,t) = u_x(b,t) = 0, \\ u(x,0) = \sin\left(\dfrac{\pi x}{b}\right), \quad u_t(x,0) = -\dfrac{\pi}{b}\sqrt{gd}\cos\left(\dfrac{\pi x}{b}\right). \end{cases} \tag{B.0.11}$$

Aquí, $d(x)$ *es la función de profundidad dada por* $d(x) = d_0[2 + \cos(2\pi x/b)]$, g *denota la aceleración de la gravedad y* $\omega(x,u)$ *es el coeficiente de fricción de fondo el cual está definido por* $\omega = g|u|/C^2 d$, *con coeficiente de Chezy* C.

Discretizando $\partial^2 u/\partial x^2$ *usando diferencias simétricas de cuarto orden en los puntos del intervalo y en las fronteras, y usando la fórmula (B.0.9) resulta un sistema no lineal de EDOs de dimensión* N.

33. **Problema B.33** *Problema del Péndulo Acoplado (CPEN) [55, p. 297]. Para este problema las energías Cinética y Potencial se expresan por medio de las ecuaciones:*

$$T = \frac{m_1 l_1^2 \dot{\varphi}_1^2}{2} + \frac{m_2 l_1^2 \dot{\varphi}_2^2}{2}$$
$$V = -m_1 l_1 \cos\varphi_1 - m_2 l_2 \cos\varphi_2 + \frac{c_0 r^2 (\sin\varphi_1 - \sin\varphi_2)^2}{2}.$$

La teoría de Lagrange conduce a las ecuaciones diferenciales:

$$\ddot{\varphi}_1 = -\frac{\sin\varphi_1}{l_1} - \frac{c_0 r^2}{m_1 l_1^2}(\sin\varphi_1 - \sin\varphi_2)\cos\varphi_1 + f(t)$$
$$\ddot{\varphi}_2 = -\frac{\sin\varphi_2}{l_2} - \frac{c_0 r^2}{m_2 l_1^2}(\sin\varphi_2 - \sin\varphi_1)\cos\varphi_2.$$

Elegimos los parámetros

$l_1 = l_2 = m_1 = 1$, $m_2 = 0{,}99$, $c_0 = 0{,}01$, $t_{end} = 496$, *y todos los valores y velocidades iniciales son iguales a cero para* $t = 0$. *El péndulo es puesto en movimiento por medio de una fuerza*

$$f(t) = \begin{cases} \sqrt{1-(1-t)^2} & si\ |t-1| \le 1; \\ 0 & en\ otro\ caso. \end{cases}$$

34. **Problema B.34** *Ecuación no lineal de onda [20, p.114;120],*

$$\begin{cases} \dfrac{\partial^2 u}{\partial t^2} = \dfrac{\partial^2 u}{\partial x^2} - 0{,}5u + u^2, \quad -\pi \leq x < \pi, \quad t \in [0, 550], \\ u(-\pi, t) = u(\pi, t), \\ u(x, 0) = 0{,}1(x/\pi - 1)^3(x/\pi + 1)^2, \quad u_t(x, 0) = 0{,}01x/\pi(x/\pi - 1)(x/\pi + 1)^2. \end{cases} \tag{B.0.12}$$

Discretizando mediante diferencias centrales de segundo orden la derivada espacial $\partial^2 u/\partial x^2$ *y usando las condiciones de periodicidad en la frontera se obtiene un sistema diferencial de dimensión* N*, cuya matriz Jacobiana resulta ser una matriz banda con ancho de banda igual a uno, pero con la particularidad de tener algunos elementos no nulos en los esquinas de la diagonal secundaria, debido precisamente a la periodicidad en las condiciones de frontera. Hemos tomado* $N = 128$ *como se hace en [20, p.114;120].*

Bibliografía

[1] C. A. Addison, *A comparison of several formulas on lightly damped, oscillatory problems*, SIAM J. Sci. Stat. Comput., **5** 4 (1984), 920–936.

[2] G. Bader & P. Deuflhard, *A semi–implicit mid–point rule for stiff systems of ordinary differential equations*, Numer. Math., **41** (1983), 377–398.

[3] T. A. Bickart, *An efficient solution process for implicit Runge–Kutta methods*, SIAM J. Numer. Anal., **14** (1977), 1022–1027.

[4] R. W. Brankin, I. Gladwell, J. R. Dormand, P. J. Prince & W. L. Seward, *Algorithm 670. A Runge–Kutta–Nyström code*, ACM Trans. Math. Softw., **15** (1989), 31–40.

[5] P. N. Brown, G. D. Byrne & A. C. Hindmarsh, *VODE: a variable coefficient ODE solver*, SIAM J. Sci. Stat. Comput., **10** (1989), 1038–1051.

[6] P. N. Brown, G. D. Byrne & A. C. Hindmarsh, htt://www.netlib.org/ode/vode.f

[7] K. Burrage, J. C. Butcher and F. H. Chipman, *An implementation of singly–implicit Runge–Kutta methods*, BIT, **20** (1980), 326–340.

[8] J. C. Butcher, *Implicit Runge–Kutta processes*, Math. Comput., **18** (1964), 50–64.

[9] J. C. Butcher, *On the implementation of implicit Runge–Kutta methods*, BIT, **16** (1976), 237–240.

[10] J. C. Butcher, *Some implementation schemes for implicit Runge–Kutta methods*, Proceedings of the Dundee conference o Numerical Analysis, Lecture Notes in Mathematics **733** (1979), 12–24.

[11] J. C. Butcher, *The numerical analysis of ordinary differential equations*, John Wiley & Sons, Chichester (1987), 237–240.

[12] J. C. Butcher, *Numerical Methods for ordinary differential equations*, John Wiley & Sons, Chichester (2003).

[13] M. Calvo, S. González–Pinto and J.I. Montijano, *Global error estimation based on the Tolerance Proportionality for some adaptive Runge-Kutta codes*, J. Comput. Appl. Math., 218 (2008), pp. 329-341.

[14] J. R. Cash, *High order, P–stable formulae for the numerical integration of periodic initial values problems*, Numer. Math., **37** (1981), 355–370.

[15] J. R. Cash, *Efficient P–stable methods for periodic initial values problems*, BIT, **24** (1984), 248–252.

[16] J. Cash, *Numerical solution of ODEs Boundary value problems an initial value problems. Geometric Integration Software*, http://www.ma.ic.ac.uk/ jcash/GI_software/problem.f, 2002.

[17] M. M. Chawla, *Unconditionally stable Numerov–type methods for second order differential equations*, BIT, **23** (1983), 541–542.

[18] M. M. Chawla, P. S. Rao and Benny Neta, *Two–step fourth order P–stable methods with phase–lag of order six for* $y'' = f(t, y)$, J. Comput. Appl. Math., **16** (1986), 233–236.

[19] T. H. Chipman, *The implementation of implicit Runge–Kutta processes*, BIT, **13** (1973), 391–393.

[20] D. Cohen, E. Hairer and C. Lubich, *Conservation of energy momentum and actions in numerical discretizations of nonlinear wave equations*, Numer. Math., **110** (2008), pp. 113–143.

[21] J. P. Coleman & S. C. Duxbury, *Mixed collocation methods for* $y'' = f(x, y)$, J. Comput. Appl. Math., **126** (2000), 47–75.

[22] G. J. Cooper and J. C. Butcher, *An iteration scheme for implicit Runge–Kutta methods*, IMA J. Numer. Anal., **3** (1983), 127–140.

[23] G. J. Cooper and R. Vignesvaran, *A scheme for the implementation of implicit Runge–Kutta methods*, Computing, **45** (1990), 321–332.

[24] K. Dekker and J. G. Verwer, *Stability of Runge-Kutta methods for stiff nonlinear differential equations*, North Holland, Amsterdam, 1984.

[25] B. L. Ehle, *On padé approximations to the exponential function and A–stable methods for the numerical solution of initial stiff problems*, Research Report CSRR **2010**, Univ. of Waterloo, Ontorio, Canada, (1969).

[26] W. H. Enright, *Improving the efficiency of matrix operations in the numerical solution of stiff ordinary differential equations*, ACM Trans. on Math. Software, **4** (1978), 127–136.

[27] W. H. Enright, *On the efficient time integration of systems of second order equations arising of Structural Dynamics*, inter. J. Num. Meth. Ingng., **16** (1980), 13–18.

[28] W. H. Enright and J. D. Pryce, *Two FORTRAN packages for assessing initial value methods*, ACM Trans. on Math. Software, **13** (1) (1987), 1–27.

[29] E. Fehlberg, S. Filippi and J. Gräf, *Ein Runge–Kutta–Nyström–Formelpaar der Ordnung 11(10) für Differentiagleichungen der Form* $y'' = f(x, y)$, ZAMM 66 (1986), **7**, 265–270.

[30] E. Fermi, J. Pasta and S. Ulam, *Los Alamos Report No. LA–1940 (1965)*, later published in Lect. Appli. Math., 15, 143 (1974).

[31] S. Filippi and J. Gräf, *New Runge–Kutta–Nyström formula–pair of order 8(7), 9(8), 10(9) and 11(10) for differential equations of the form* $y'' = f(x, y)$, J. Comput. Appl. Math., **14** (1986), 361–370.

[32] J. M. Franco, *An explicit hybrid method of Numerov type for second–order periodic initial–value problems*, J. Comput. Appl. Math., **59** (1995), 79–90.

[33] J. M. Franco, J. M. Gómez and L. Rández, *SDIRK methods for ODEs with oscillating solutions*, J. Comput. Appl. Math., **81** (1997), 197–209.

[34] R. Frank and C. W. Ueberhuber, *Iterated defect correction for the efficient solution of stiff systems of ordinary differential equations*, BIT, **17** (1977), 146–159.

[35] C. W. Gear, *Numerical initial value problems in ordinary differential equations*, Prentice Hall, Englewood Cliffs, New Jersey, (1971).

[36] I. Gladwell and R. M. Thomas, *Efficiency of methods for second–order problems*, Numerical Analysis Report **129**, Dept. of Mathematics, Univ. Manchester, (1987).

[37] I. Gladwell and R. M. Thomas, *Efficiency of methods for second–order problems*, IMA J. Numer. Anal., **10** (1990), 181–207.

[38] I. Gómez–Ibáñez, *Resolución numérica de problemas tipo stiff*, Tesis Doctoral, Universidad de Zaragoza, (2002).

[39] I. Gómez,I. Higueras and T. Roldán, *Starting algorithms for low stage order RKN methods*, J. Comput. Appl. Math., **140** (2002), 345–367.

[40] S. González–Pinto, C. González–Concepción & J. I. Montijano, *Iterative schemes for Gauss methods*, Computers Math. Applic., **27** (7) (1994), 67–81.

[41] S. González–Pinto, J. I. Montijano & L. Rández, *Iterative schemes for three–stage implicit Runge–Kutta methods*, Appl. Numer. Math., **17** (1995), 363–382.

[42] S. González–Pinto, S. Pérez–Rodríguez & J. I. Montijano, *On the numerical solution of stiff IVPs by Lobatto IIIA Runge–Kutta methods*, J. Comput. Appl. Math., **82** (1997), 129–148.

[43] S. González–Pinto, J. I. Montijano & S. Pérez–Rodríguez, *On the implementation of high order implicit Runge-Kutta methods*, Computers Math. Applic., **41** (7/8) (2001), 1009–1024.

[44] S. González–Pinto, J. I. Montijano & S. Peréz–Rodríguez, *Variable–order starting algorithms for implicit Runge–Kutta methods on stiff problems*, Appl. Numer. Math., **44** (2003), 77–94.

[45] S. González–Pinto & R. Rojas–Bello, *Speeding up Newton–type iterations for stiff problems*, J. Comput. Appl. Math., **181** (2005), 266–279.

[46] S. González–Pinto, S. Pérez–Rodríguez & R. Rojas–Bello, *Efficient iterations for Gauss methods on second order problems*, J. Comput. Appl. Math., **189** (2006), 80–97.

[47] S. González–Pinto & R. Rojas–Bello, *Gauss2, a Fortran 90 code for second order initial value problems,http://pcmap.unizar.es/numerico/*, (2008).

[48] S. González–Pinto, S. Pérez–Rodríguez & R. Rojas–Bello, CORRECCIÓN A *Efficient iterations for Gauss methods on second order problems:* [J. Comput. Appl. Math., **189** (2006), 80–97], **205** (2007), 583.

[49] S. González–Pinto, J. I. Montijano, S. Peréz–Rodríguez, L. Randez and R. Rojas–Bello, *A code based on Gauss Method for second defferential systems*, American Institute of Physics, AIP Procedings. No 936, pp.424-427, Ed. T.E. Simos, New York (2007).

[50] S. González–Pinto, S. Pérez–Rodríguez & R. Rojas–Bello, *Gauss2 a Fortran 90 code for second order initial value problems (2^{nd} versión), http://www.netlib.org/ode/*, (2008).

[51] S. González–Pinto & R. Rojas–Bello, *A code based on the two–stage Gauss method for second order initial value problems*, Submitted in fifth version to ACM Trans. Math. Soft. in (2008), with Status: *Minor Revision.*

[52] K. Gustafsson, *control–theoretic techniques for stepsize selection in implicit Runge–Kutta methods*, ACM Trans. Math. Soft., **20** (1994), 496–517.

[53] E. Hairer homepage, http://www.unige.ch/ hairer/software.html.

[54] E. Hairer, *Unconditionally stable methods for second order differential equations*, Numer. Math., **32** (1979), 373–379.

[55] E. Hairer, S. P. Nørsett and G. Wanner, *Solving ordinary differential equations I*, Springer Verlag, 2^{nd} ed., Berlin, (1993).

[56] E. Hairer, C. Lubich and G. Wanner, *Geometric Numerical Integration*, Springer–Verlag, 2^{nd} ed., Berlin, (2002).

[57] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II*, Springer–Verlag, 2^{nd} ed., Berlin, (1996).

[58] D. J. Higham, *The tolerance proportionality of adaptative ODE solvers*, J. Comput. Appl. Math. **45** (1993), 227–236.

[59] A. C. Hindmarsh, *LSODE and LSODI, two new initial value Ordinary Differential Equations solvers*, ACM–SIGNUM Newsletter **15** (1980), 10–11.

[60] van der Houwen and E. Messina, *Parallel linear system solvers for Runge–Kutta–Nyström methods*, J. Comput. Appl. Math., **82** (1997), 407–422.

[61] P. J. van der Houwen, E. Messina & B. P. Sommeijer, *Oscillatory störmer–Cowell methods Iterate Runge–Kutta methods on parallel computers*, J. Comput. Appl. Math., **115** (2000), 547–564.

[62] P. J. van der Houwen and B. P. Sommeijer, *Diagonally implicit Runge–Kutta–Nyström methods for oscillatory problems*, SIAM J. Numer. Anal., **26** (2) (1989), 414–429.

[63] P. J. van der Houwen and B. P. Sommeijer, *Explicit Runge–Kutta (–Nyström) methods with reduced phase errors for computing oscillating solutions*, SIAM J. Numer. Anal., **24** (3) (1987), 595–617.

[64] P.J. van der Houwen and B.P. Sommeijer, *Iterate Runge–Kutta methods on parallel computers*, SIAM J. Sci. Stat. Comput., **12** (5) (1991), 1000–1028.

[65] P. J. van der Houwen and J. J. B. de Swart, *Triangularly implicit iteration methods for ODE–PVI solvers*, SIAM J. Sci. Stat. Comput., **18** (1) (1997), 41–55.

[66] W. Hundsdorfer, J.G. Verwer (2003), *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*, Springer Series in Computational Mathematics, Vol. 33, Springer, Berlin.

[67] P. Laburta, *Starting algorithms for implicit Runge-Kutta Nyström methods*, appl. Numer. Math., **27 no.3** (1998), 233–251.

[68] J. D. Lambert and I. A. Watson, *Symmetric multistep methods for periodic initial value problems*, J. Inst. Math. Appl., **18** (1976), 189–202.

[69] W. M. Lioen, J. J. B. de Swart and W. A. van der Veen, *Test set for IVP solvers*, http://www.cwi.nl/cwi/projects/IVPtestset.html, Test set for IVP solvers, 1996.

[70] M. Marden, *Geometrie of polynomials*. American Mathematical Society, Providence, Rhode Island, 2^{nd} ed. (1966).

[71] Netlib, ver http://www.netlib.org/ode/

[72] S. P. Nørsett, *One step Hermite type methods for numerical integration of stiff system*, BIT, **14** (1974), 63–67.

[73] S. P. Nørsett and G. Wanner, *The real-pole sandwich for rational aproximations and oscillation equations*, BIT, **19** (1979), 79–94.

[74] S. Pérez–Rodríguez, *Integración de problemas stiff a través de métodos Runge–Kutta*, Tesis Doctoral, Universidad de La Laguna, (2000).

[75] R. Rojas–Bello & J. Bottino–Cedeño, *Métodos de líneas para el problema lineal de la onda*, CITEG, **1** (2007), 21–26.

[76] J. M. Sanz–Serna and M. P. Calvo, *Numerical Hamiltonian Problems*, Chapman & Hall, London (1994).

[77] L. F. Shampine and L. S. Baca, *Error estimators for stiff differential equations*, J. Comput. Appl. Math., **11** (1984), 197–207.

[78] P. W. Sharp, J. M. Fine & K. Burrage, *Two–stage and Three–stage Diagonally Implicit Runge–Kutta Nyström Methods of orders three and four*, IMA J. Numer. Anal., **10** (1990), 489–504.

[79] H. J. Stetter, *Tolerance proportionality in ODE codes*, Proc. Second conf. on Numerical Treatment of Ordinary Differential Equations; R März, ed. Humbold University Berlin (1980), 109–123.

[80] R. M. Thomas, *Phase Properties of hight order, almost P–stable formulae*, BIT **24** (1984), 225–238.

[81] S. Timoshenko, *Vibrations problems in Engineering*, 2^{nd} Ed., Van Nostrand, New York (1947).

[82] Ch. Tsitouras, *Explicit two–step methods for second–order linear IVPs*, Computers Math. Applic., **43** (2002), 943–949.

[83] J. M. Varah, *On the efficient implementation of implicit Runge–Kutta methods*, Math. Comput., **33** (1979), 557–561.