ESTUDIO NUMÉRICO PARA LA APROXIMACIÓN DE SISTEMAS RÍGIDOS

CLAUDIA PATRICIA ORDÓÑEZ RODRÍGUEZ

FACULTAD DE CIENCIAS EXACTAS Y NATURALES
DEPARTAMENTO DE MATEMÁTICAS Y ESTADÍSTICA
UNIVERSIDAD DE NARIÑO
SAN JUAN DE PASTO
2021

ESTUDIO NUMÉRICO PARA LA APROXIMACIÓN DE SISTEMAS RÍGIDOS

CLAUDIA PATRICIA ORDÓÑEZ RODRÍGUEZ

Trabajo presentado como requisito parcial para optar al título de Licenciada en Matemáticas

Asesoras
Catalina M. Rúa Alvarez
Doctora en Matemática Aplicada
Priscila Cardoso Calegari
Doctora en Matemática Aplicada

FACULTAD DE CIENCIAS EXACTAS Y NATURALES
DEPARTAMENTO DE MATEMÁTICAS Y ESTADÍSTICA
UNIVERSIDAD DE NARIÑO
SAN JUAN DE PASTO
2021

Nota de Responsabilidad

Todas las ideas y conclusiones aportadas en el siguiente trabajo son responsabilidad exclusiva de los autores.

Artículo 1^{ro} del Acuerdo No. 324 de octubre 11 de 1966 emanado por el Honorable Consejo Directivo de la Universidad de Nariño.

Nota de Aceptación

Catalina M. Rúa Alvarez
Directora de Tesis

Priscila Cardoso Calegari
Co-Directora de Tesis

Alexandre M. Roma
Jurado

Miller Cerón Gómez
Jurado

San Juan de Pasto, Enero 23 de 2021

Este trabajo está dedicado:
A mi madre, Marlen Rodríguez.
A mis hermanos, Johana y Ronald Ordóñez.
A mi ángel, mi padre, Ferney Ordóñez.
Claudia P.

Agradecimientos

Primero que todo quiero agradecer a Dios por permitirme terminar mi carrera con éxitos, por darme toda la sabiduría, paciencia y perseverancia para culminar este trabajo y por guiarme en todo este camino para hacer las cosas de la mejor manera.

Agradezco a mis padres quienes han sido mi motor, mi orgullo y mi bendición. A mi madre por siempre estar dándome su apoyo y moral para salir adelante y a mi padre que desde el cielo me regala toda su protección y bendición. A mis hermanos Johana y Ronald por todo su apoyo, unión y confianza. A mi familia por todo el apoyo incondicional que me han regalado, por ayudarme a hacer de este sueño una realidad. Gracias por confiar en mis capacidades y por estar siempre.

A mi asesora Catalina Rua, por guiarme en la elaboración de este trabajo, por toda su paciencia, tiempo, sugerencias y conocimientos que han sido esenciales en este proceso. Por haber hecho en mí una persona que mira siempre adelante, en alguien que debe soñar para cumplir lo que quiere, por enseñarme a confiar en todas mis capacidades y a ser fuerte en toda circunstancia. Gracias profe por cada consejo tanto académico como personal, gracias por crear en mí una estudiante que le encantó la investigación. Igualmente agradezco a mi co-asesora Priscila Cardoso, quien aceptó acompañarme en la realización de este proyecto, sus consejos y sugerencias hicieron que el trabajo tuviera una mejor organización. Infinitas gracias profes.

A la Universidad de Nariño por acogerme durante todos estos años, recibí muchos conocimientos para formarme como una profesional. Por brindarme las oportunidades y el respaldo necesario para profundizar en los conocimientos de este trabajo. A los profesores del Departamento de Matemáticas y Estadística, por su acompañamiento en mi formación académica.

Finalmente agradezco a mis amigos y compañeros, gracias por hacer de cada momento compartido, momentos de felicidad, porque con su presencia días llenos de angustia se convertían en momentos de alegría. Gracias por confiar siempre en mí.

Resumen

Los Sistemas de Ecuaciones Diferenciales Ordinarios (SEDO) permiten modelar matemáticamente varias aplicaciones de diferentes ciencias como la ingeniería, la medicina y la química, las cuales muchas veces no cuentan con una solución exacta y para obtenerla se debe aproximar mediante métodos numéricos. El desarrollo y avance de los métodos numéricos surgió de la necesidad de aproximar numéricamente problemas cada vez más complejos, con altas restricciones en la elección del tamaño de paso cuando se usan métodos explícitos para generar aproximaciones. Los SEDO con estas restricciones de solución numérica son denominados rígidos o tipo *"stiff"*, problemas que conllevan a la investigación de métodos con una mayor zona de estabilidad como lo son los métodos implícitos.

En este trabajo se realiza un estudio teórico y computacional de algunos métodos numéricos implícitos para aproximar soluciones de SEDO rígidos. Inicialmente se formulan y analizan algunas de las características que permiten distinguir los sistemas rígidos de los sistemas que no tienen esta propiedad. Luego se estudian métodos numéricos implícitos de paso único y paso múltiple como el método de Euler, los métodos de Runge-Kutta, los métodos Adams-Moulton y los métodos Predictor-Corrector. Además, se presentan las deducciones y se analizan características y propiedades importantes como la consistencia, convergencia y estabilidad. Adicionalmente, se destaca la necesidad de solución numérica de sistemas de ecuaciones no lineales, para los cuales se utiliza el método de Newton. Finalmente, se analiza un método adaptativo para variar el tamaño de paso y controlar la magnitud del error según el comportamiento de las aproximaciones.

Con base a la parte teórica estudiada sobre los métodos numéricos se realizaron implementaciones en Lenguaje C, para las cuales se validaron y verificaron propiedades teóricas con SEDO rígidos. Por último, para destacar una aplicación de los SEDO rígidos, con las implementaciones realizadas se soluciona el problema de Rober relacionado con reacciones químicas. Este problema no cuenta con una solución teórica por lo cual los resultados obtenidos son comparados con los presentes en la literatura, permitiendo concluir acerca de ellos.

Abstract

The Systems of Ordinary Differential Equation (SODEs) allow mathematical modeling many applications of different sciences such as engineering, medicine and chemistry which often do not have an exact solution and to obtain it, it must be approached by numerical methods. The development and advancement of numerical methods arose from the need to numerically approximate increasingly complex problems with great restrictions in the choice of step size when explicit methods are used to generate approximations. The SODEs with these numerical solution restrictions are called stiff, problems that lead to the investigation of methods with a greater area of stability as the implicit methods.

In this research, a theoretical and computational study of some implicit numerical methods is carried out to approximate solutions of stiff SODEs. Initially, some of the characteristics are formulated and analyzed that allow us to distinguish stiff systems from systems that do not have this property. Then, Implicit single step and multistep numerical methods such us Euler's method, Runge-Kutta's methods, Adams-Moulton's methods and Predictor-Corrector methods, are studied. In addition, deductions are presented and important characteristics and properties such as consistency, convergence, and stability are analyzed. Additionally, the need for a numerical solution of systems of non-linear equations for which Newton's method is used, is highlighted. Finally, an adaptive method is analyzed to vary the step size and control the magnitude of the error according to the behavior of the approximations.

Based on the theoretical part studied on numerical methods, their implementations were made in C Language, for which properties were validated and verified with stiff SODEs. Finally, to highlight an application of stiff SODEs, with the implementations carried out the Rober problem related to chemical reactions is solved. This problem does not have a theoretical solution, so the results obtained are compared with those found in the literature, allowing us to conclude about them.

Índice general

Índice de figuras

Índice de tablas

Notación

SEDO	Sistemas de Ecuaciones Diferenciales Ordinarias
MPU	Métodos de Paso Único
RK	Métodos Runge-Kutta explícitos
RKps	Métodos Runge-Kutta explícitos orden p con s estados
RK22	Runge-Kutta explícito orden dos con dos estado
RK33	Runge-Kutta explícito orden tres con tres estados
RK44	Runge-Kutta explícito orden cuatro con cuatro estados
RKI	Métodos Runge-Kutta implícitos
RKIps	Métodos Runge-Kutta implícitos orden p con s estados
RKI21	Runge-Kutta Implícito orden dos con un estado
RKI22	Runge-Kutta Implícito orden dos con dos estados
RKI32	Runge-Kutta Implícito orden tres con dos estado
RKI42	Runge-Kutta Implícito orden cuatro con dos estado
MPM	Métodos de Paso Múltiple
AB	Métodos Adams-Bashforth
AB2	Adams-Bastforth de dos pasos
AB3	Adams-Bastforth de tres pasos
AB4	Adams-Bastforth de cuatro pasos
AM	Métodos Adams-Moulton
AM2	Adams-Moulton de dos pasos
AM3	Adams-Moulton de tres pasos
AM4	Adams-Moulton de cuatro pasos
ABM	Adams-Bastforth-Moulton
ABM3	Adams-Bastforth-Moulton de tercera orden
ABM4	Adams-Bastforth-Moulton de cuarta orden
BDF	Backward Differentiation Formulae
$AbsTol$	Tolerancia Absoluta
$RelTol$	Tolerancia Relativa

Introducción

Los Sistemas de Ecuaciones Diferenciales Ordinarios (SEDO) se pueden relacionar con la modelación matemática de diferentes aplicaciones de varias ciencias como la ingeniería, la medicina y la química. En ocasiones estos son complicados de solucionar o incluso aún la solución analítica no se ha determinado, por tanto es conveniente el estudio del análisis numérico para encontrar aproximaciones a sus soluciones.

La aparición o formulación de SEDO que modelan diversas aplicaciones y tienen restricciones para encontrar sus soluciones numéricas, han permitido el desarrollo y el avance de los métodos numéricos. De estos problemas o aplicaciones que cuentan con estas restricciones han surgido los sistemas rígidos o tipo *"stiff"*, los cuales no son fáciles de solucionar numéricamente con métodos explícitos, por tanto se necesita el estudio de métodos numéricos con una mayor zona de estabilidad como lo son los métodos implícitos para aproximarlos. Ver más en [10, 16] y [22].

Este trabajo tiene un gran interés en el estudio teórico y computacional de algunos métodos numéricos implícitos para aproximar soluciones de sistemas de ecuaciones diferenciales rígidos. La solución numérica de los problemas rígidos conlleva a estudiar y formular algunas de las características que permiten distinguir los sistemas rígidos de los sistemas que no tienen esta propiedad. Estas particularidades permiten identificar el SEDO y abordarlo numéricamente con el método adecuado para no causar mayores costos computacionales, ver [2, 8, 18] y [30]. Para el estudio de las características de estos problemas se exigen conocimiento de álgebra lineal numérica y propiedades matriciales, temas que fueron estudiados respectivamente en el trabajo de grado de Cesar Fernando Bolaños en el 2016, como puede ser visto en [5].

Investigaciones sobre el estudio teórico y numérico para aproximar SEDO con métodos numéricos explícitos de paso único, como el método de Euler creado por Leonhard Euler en 1678 y los métodos Runge Kutta de diferentes estados que cuentan con una mayor precisión, desarrollados alrededor de 1900 por los matemáticos Carl T. Runge y Martin W. Kutta. Además de investigaciones de los métodos multipaso de Adams-Bastforth y Adams-Moulton. Fueron realizadas una por Christiam Fernando Pistala y otra por Rosa Janeth Alpala en el 2017, donde ambos resaltan como trabajos futuros la solución numérica de problemas rígidos, ver con detalle en [3] y [23]. Proyectos desarrollados en la Universidad de Nariño, que se complementan con la realización de este trabajo al estudiar métodos numéricos implícitos, lo cual se puede observar en el Capítulo 2.

Para proseguir con el estudio de los métodos numéricos se realiza un estado del arte de algunos métodos numéricos implícitos. Se hace una introducción a esta temática comenzando con la deduc-

ción del método de Euler implícito, método con orden de convergencia uno pero que cuenta con una excelente región de estabilidad. Se continúa con el estudio de los métodos de Runge-Kutta implícitos y se presentan las deducciones de algunos de ellos siguiendo las condiciones de orden simplificado, ya que existe una variedad de métodos dependiendo de características como la forma de obtener sus órdenes y deducciones. Es importante estudiar características y propiedades de estos métodos, como son la consistencia, convergencia y estabilidad, e incluso el orden de convergencia y análisis de errores, ya que esto permite analizar y concluir acerca de los resultados numéricos que se obtengan y de los métodos mismos. La mayoría de los métodos presentes cuentan con una característica importante en los métodos numéricos que es la A-estabilidad, propiedad esencial en los métodos para abordar los problemas rígidos. Se sugiere [8, 16, 18, 20] y [24].

Para la implementación computacional de los métodos numéricos implícitos al igual que el estudio teórico de los mismos, conlleva al estudio de métodos de solución de sistemas de ecuaciones no lineales como puede ser el método de Newton o sus modificaciones. Este estudio se mira con detalle en el trabajo de grado realizado por Juneth Andrea Terán en el 2018, ver [32].

Además de hacer el estudio de los métodos numéricos implícitos se investiga un método de paso adaptativo, ya que los métodos numéricos de paso único y paso múltiplo presentan errores de mayor magnitud en sus aproximaciones en las zonas donde los SEDO muestran su rigidez. Se aborda un método que permite controlar este error, ya que este consigue identificar el tamaño de h adecuado en cada paso para conseguir buenas aproximaciones. El método que se muestra es el método adaptativo de Euler implícito, ver con detalle en [15] y [28].

Es importante implementar y validar numéricamente propiedades teóricas de los métodos investigados, ya que esto permite observar dificultades, ventajas y desventajas que estos generan al encontrar aproximaciones a la solución de SEDO rígidos. Con la validación de estos se posibilita abordar aplicaciones con SEDO rígidos que pueden o no contar con soluciones teóricas, pero sus validaciones conllevan a confiar en sus aproximaciones. Al final del trabajo se soluciona numéricamente una aplicación relacionada con las reacciones químicas, ya que estas están asociadas a problemas con rigidez. Se trabaja el problema de ROBER, un SEDO rígido que según [4, 25, 26] y [33] es utilizado para la validación de los métodos numéricos para abordar este tipo de problemas. Los resultados de esta aplicación son comparados con los presentes en la literatura, lo cual permite concluir acerca de ellos.

De esta forma, este trabajo de grado complementa otros estudios que se desarrollaron en la licenciatura y los enfoca en el estudio de métodos numéricos más avanzados abriendo un nuevo camino para continuar futuras investigaciones. Por tal motivo se proponen los siguientes objetivos a cumplir:

- **Objetivo general:** Analizar teórica y computacionalmente algunos métodos numéricos implícitos para aproximar soluciones de sistemas de ecuaciones diferenciales rígidos.
- **Objetivos específicos:**
    1. Formular algunas de las características que permiten distinguir los sistemas rígidos de los sistemas que no tienen esta propiedad.
    2. Realizar un estado del arte de los métodos de Runge-Kutta implícitos.

3. Implementar y validar numéricamente propiedades teóricas de los métodos numéricos investigados.

Para alcanzar los objetivos propuestos, este trabajo se distribuye de la siguiente manera:

- En el Capítulo 1 se presenta una introducción sobre los métodos numéricos explícitos de paso único para la solución de SEDO, mostrando propiedades teóricas de los mismos. Además, se describen características y propiedades que ayudan a identificar un sistema rígido.
- El Capítulo 2 corresponde al estudio de métodos numéricos implícitos de paso único para la solución de SEDO, donde se muestran propiedades teóricas y deducciones. Además, se analizan algunos métodos de solución de sistemas de ecuaciones no lineales ya que los métodos implícitos necesitan de su aproximación.
- El Capítulo 3 presenta el estudio de los métodos numéricos de paso múltiple como los métodos de Adams Bastfoth y Moulton, ilustrando conceptos teóricos y posibles deducciones. Además, se introducen los métodos Predictor-Corrector.
- En el Capítulo 4 se muestran resultados numéricos, donde se validan las implementaciones realizadas en Lenguage C de los métodos numéricos estudiados, incluyendo resultados con un método de paso adaptativo. Adicionalmente, se soluciona numéricamente una aplicación relacionada con las reacciones químicas.
- Finalmente, en el Capítulo 5 se mencionan las conclusiones de esta investigación y algunos posibles trabajos futuros.

Capítulo 1

Preliminares

En muchas ocasiones es muy tedioso o no es posible encontrar la solución analítica de los SEDO que modelan matemáticamente situaciones, por tanto es necesario encontrar a través de los métodos numéricos aproximaciones a sus soluciones. En este capítulo se realiza una introducción sobre los métodos numéricos explícitos para la solución de SEDO, ya que estos son la base para el estudio de los métodos numéricos de nuestro interés. Además, se mencionan características que ayudan a identificar un SEDO rígido de los que no cuentan con esta propiedad, dado que los problemas a solucionar numéricamente son de este tipo. Esta teoría es seguida de [2, 3, 8, 12, 18, 24] y [30].

En la siguiente sección se introducen algunos métodos numéricos explícitos de paso único como el método de Euler y algunos métodos de Runge-Kutta de distintas órdenes, se estudian algunas características y se mencionan propiedades de los mismos que se pueden encontrar en [3, 8, 18] y [23].

1.1. Métodos de paso único

Los métodos numéricos estudiados son métodos iterativos que aproximan el *problema de Cauchy*, un problema de un SEDO sujeto a condiciones iniciales

$$\begin{cases} \boldsymbol{y'}(t) = \boldsymbol{f}(t, \boldsymbol{y}(t)), \\ \boldsymbol{y}(t_0) = \boldsymbol{y}_0, \end{cases} \tag{1.1.1}$$

donde $\boldsymbol{f} : [a, b] \times \mathbb{R}^n \to \mathbb{R}^n$ y $t \in [a, b]$.

A continuación, se resaltan algunos términos que serán útiles en el desarrollo e implementación de los métodos numéricos, ya que las aproximaciones no se obtienen de forma continua.

Se considera un intervalo $[a, b]$, el valor inicial para t es $t_0 = a$ y el punto donde se tiene la condición inicial es y_0; b es el punto donde se quiere encontrar la solución. El intervalo se divide en n subintervalos iguales, con n la *cantidad de pasos*, es así como se obtiene un conjunto $\{t_0, t_1, \ldots, t_n\}$ de *puntos de malla*, donde $t_{i+1} = t_i + h$, $i = 0, 1, \cdots, n-1$, y h es el *tamaño de paso* que se determina por $h = \frac{b-a}{n}$. Este proceso se conoce como *discretización*.

Es válido resaltar que cuando se utiliza n como exponente de un conjunto de número se refiere a las dimensiones de un vector, caso contrario n hace referencia a la cantidad de paso en el intervalo de integración.

Los métodos numéricos que utilizan el paso anterior $\boldsymbol{y}_i$, donde $\boldsymbol{y}_i$ es la aproximación de la solución exacta $\boldsymbol{y}(t_i)$, para encontrar la aproximación que se quiere $\boldsymbol{y}_{i+1}$ y que cumplan con una forma específica se denominan Métodos de Paso Único (MPU).

Definición 1.1. La forma general de un *método de paso único* aplicado al problema de Cauchy (1.1.1) es

$$\begin{cases} \boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \boldsymbol{\phi}\left(t_i, t_{i+1}, \boldsymbol{y}_i, \boldsymbol{y}_{i+1}, h\right), \\ \boldsymbol{y}(t_0) = \boldsymbol{y}_0, \end{cases} \tag{1.1.2}$$

donde $\boldsymbol{\phi}$ se denomina *función incremento* y h tamaño de paso. Si el término $\boldsymbol{y}_{i+1}$ aparece en los dos lados de la igualdad consideramos un *método implícito*, de lo contrario se obtiene un *método explícito*.

La forma general de *los MPU explícitos* es

$$\begin{cases} \boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \boldsymbol{\phi}\left(t_i, t_{i+1}, \boldsymbol{y}_i, h\right), \\ \boldsymbol{y}(t_0) = \boldsymbol{y}_0. \end{cases} \tag{1.1.3}$$

En el estudio del análisis numérico es importante conocer propiedades y características de los métodos numéricos como la consistencia, convergencia y estabilidad, ya que estas ayudan a concluir sobre las aproximaciones obtenidas y los métodos numéricos trabajados. Estos conceptos son necesarios en este trabajo ya que son válidos tanto para los métodos explícitos como para los métodos implícitos. Las definiciones y teoremas que se muestran a continuación se tomaron de [3, 8] y [23].

1.1.1. Consistencia, convergencia y estabilidad

Cuando se utilizan métodos numéricos para obtener aproximaciones a las soluciones de SEDO, es necesario verificar que tan cerca están las aproximaciones con respecto a las soluciones exactas. Para ello se presentan dos tipos de error, error local de discretización y error global de discretización.

Definición 1.2. Para el MPU (1.1.2) el *error local de discretización* en el instante t_i se define por

$$\boldsymbol{\tau}_i = \frac{\boldsymbol{y}(t_{i+1}) - \boldsymbol{y}(t_i)}{h} - \boldsymbol{\phi}(t_i, \boldsymbol{y}(t_i), h). \tag{1.1.4}$$

Definición 1.3. Para el MPU (1.1.2) el *error global de discretización* en el instante t_i se define por

$$\boldsymbol{e}_i = \|\boldsymbol{y}(t_i) - \boldsymbol{y}_i\|, \tag{1.1.5}$$

que representa el error acumulado entre la solución exacta $\boldsymbol{y}(t_i)$ y la solución aproximada $\boldsymbol{y}_i$.

Al realizar un analizar con respecto a los errores mencionados, se puede determinar qué tan precisa es la aproximación obtenida. En relación a esto, se presentan las siguientes definiciones.

Definición 1.4. Un MPU (1.1.2) es *consistente* con el problema de Cuachy (1.1.1) si y solo si la función incremento $\boldsymbol{\phi}(t, \boldsymbol{y}, h)$, satisface la siguiente relación

$$\boldsymbol{\phi}(t, \boldsymbol{y}, 0) = \boldsymbol{f}(t, \boldsymbol{y}).$$

Es decir, un MPU es consistente si y solo si

$$\lim_{h \to 0} \|\boldsymbol{\tau}_i\| = 0, \quad \forall t \in [a, b], \quad \boldsymbol{y} \in \mathbb{R}^n.$$

La definición de consistencia es importante tenerla en cuenta, porque permite notar que las aproximaciones obtenidas se acercan a la solución requerida y no se desvían a una otra solución. La consistencia controla las magnitudes de los errores locales en las aproximaciones, es decir hay una precisión local suficiente en la solución numérica.

Definición 1.5. Si existieran constantes positivas C, h_0 y p, independiente del paso de integración h y del subíndice i, con $0 < h < h_0$, $1 \leq i \leq n$ e $i \in \mathbb{N}$, tales que el error local de discretización satisface

$$\max_{1 \leq i \leq n} \|\boldsymbol{\tau}_i\| \leq Ch^p,$$

entonces se dice que el método numérico tiene *orden de consistencia* p y se denota $O(h^p)$.

Definición 1.6. Si existieran constantes positivas C, h_0 y p, independiente del paso de integración h y del subíndice i, con $0 < h < h_0$, $1 \leq i \leq n$ e $i \in \mathbb{N}$, tales que el error global de discretización satisface

$$\max_{1 \leq i \leq n} \|\boldsymbol{e}_i\| \leq Ch^p, \tag{1.1.6}$$

entonces se dice que el método numérico tiene *orden de convergencia* p y se denota $O(h^p)$.

Observación 1.1. El orden de consistencia indica que tan rápido el error local de discretización se acerca a cero cuando h disminuye y el orden de convergencia indica que tan rápido el error global de discretización disminuye. Como en este trabajo los dos tipos de errores coinciden, se dirá que un método es de orden p.

Definición 1.7. Un MPU (1.1.2) es *convergente* en un punto t_i si y solo si

$$\lim_{h\to 0} \|\boldsymbol{e}_i\| = 0.$$

El método numérico es convergente si fuera convergente para todo $t \in [a, b]$ y para cualquier problema de Cauchy. Esta definición hace referencia a la convergencia puntual de un método numérico.

A seguir se presenta un resultado que ofrece una forma alternativa para garantizar la convergencia de un MPU a partir de la consistencia. La demostración de este teorema se puede ver en [3] y [23].

Teorema 1.1. *Considere el MPU (1.1.2), donde la función incremento $\boldsymbol{\phi(t, y, h)}$ es Lipschiziana en $\boldsymbol{y}$ y continua en sus argumentos. Si el MPU es consistente es convergente.*

A seguir se estudian algunos MPU explícitos como el método de Euler y los métodos de Runge-Kutta.

1.1.2. Método de Euler

El método de Euler es uno de los métodos más antiguos creado por Leonhard Euler en 1678. La deducción de este método se puede realizar a través de la serie de Taylor, el método de integración o por medio de la definición de rectas tangentes que es la denominada deducción geométrica, como se ve en [3, 7] y [23].

El método de *Euler explícito* para solucionar el problema de Cauchy (1.1.1), está dado por

$$\begin{cases} \boldsymbol{y}_{i+1} = \boldsymbol{y}_i + h\boldsymbol{f}(t_i, \boldsymbol{y}_i), \\ \boldsymbol{y}(t_0) = \boldsymbol{y}_0, \end{cases} \tag{1.1.7}$$

donde $t_{i+1} = t_i + h$, $0 \le i \le n-1$, $h = \frac{b-a}{n}$, $\boldsymbol{f} : [a, b] \times \mathbb{R}^n \to \mathbb{R}^n$.

Al comparar (1.1.7) con (1.1.3), se concluye que el método de Euler es un MPU explícito con $\boldsymbol{\phi}(t_i, \boldsymbol{y}_i, h) = \boldsymbol{f}(t_i, \boldsymbol{y}_i)$. Se ignora el término $\boldsymbol{y}_{i+1}$ en (1.1.2).

Ejemplo 1.1. Solucionar numéricamente el siguiente SEDO con el método de Euler.

$$\begin{cases} m'(t) = 2m(t) - n(t), \\ n'(t) = m(t), \\ m(0) = 6, \quad n(0) = 2 \quad \text{y} \quad t = [0, 1]. \end{cases} \tag{1.1.8}$$

Con soluciones exactas $m(t) = 6e^t + 4te^t$ y $n(t) = 2e^t + 4te^t$, donde $m(1) = 27.18282$ y $n(1) = 16.30969$.

Solución. Utilizando el método de Euler (1.1.7) y un tamaño de paso $h = 1/16$ se tiene

$$\boldsymbol{y}_1 = \boldsymbol{y}(t_0) + h\boldsymbol{f}(t_0, \boldsymbol{y}_0) = \begin{bmatrix} m(t_0) \\ n(t_0) \end{bmatrix} + 1/16 \begin{bmatrix} 2m(t_0) - n(t_0) \\ m(t_0) \end{bmatrix}$$

$$= \begin{bmatrix} 6 \\ 2 \end{bmatrix} + 0.0625 \begin{bmatrix} 10 \\ 6 \end{bmatrix} = \begin{bmatrix} 6.625 \\ 2.375 \end{bmatrix} \approx \boldsymbol{y}(0.0625),$$

$$\boldsymbol{y}_2 = \boldsymbol{y}(t_1) + h\boldsymbol{f}(t_1, \boldsymbol{y}_1) = \begin{bmatrix} m(t_1) \\ n(t_1) \end{bmatrix} + 1/16 \begin{bmatrix} 2m(t_1) - n(t_1) \\ m(t_1) \end{bmatrix}$$

$$= \begin{bmatrix} 6,625 \\ 2.375 \end{bmatrix} + 0.0625 \begin{bmatrix} 10.875 \\ 6.625 \end{bmatrix} = \begin{bmatrix} 7.30469 \\ 2.78906 \end{bmatrix} \approx \boldsymbol{y}(0.125).$$

Para conseguir las aproximaciones de $m(1)$ y $n(1)$ se realiza el proceso anterior hasta alcanzar el extremo derecho del intervalo $t = [0, 1]$. En la Tabla 1.1 se muestran los resultados numéricos obtenidos con distintos valores de h, se observa que cuando h disminuye a la mitad el error también reduce a la mitad. Como el problema abordado es un SEDO en $\mathbb{R}^2$ se usa una norma vectorial para el cálculo del error global. Se utiliza la norma dos para obtener estos resultados. En el Apéndice A.1 se puede observar con detalle esta teoría.

Tamaño de paso h	1/16	1/32	1/64	1/128
Aproximación $m(1)$	25.75860	26.44542	26.80746	26.99343
Aproximación $n(1)$	15.20688	15.73746	16.01808	16.16247
Error global	1.80128	9.33387×10^{-1}	4.75323×10^{-1}	2.39876×10^{-1}

Tabla 1.1: Aproximaciones método de Euler para (1.1.8).

□

1.1.3. Métodos de Runge-Kutta

Los métodos de Runge-Kutta surgieron de la generalización del método de Euler, con el fin de mejorar el orden de convergencia, este trabajo generalmente se lo atribuyen a Runge 1895 y a Heun y Kutta que hicieron contribuciones adicionales alrededor de 1900. El interés del estudio de estos métodos ha avanzado enormemente aportando varias investigaciones que han contribuido con la teoría y desarrollo de los mismos. En un inicio, el principal interés fue el estudio de los métodos de Runge-Kutta explícitos (RK) que son los que se observan en este capítulo, ahora, existe un interés

por los métodos implícitos que son útiles para solucionar numéricamente SEDO con propiedades más restringidas, los cuales más adelante se estudian con detalle. Para complementar esta información se recomienda seguir [3, 8, 12, 18] y [23].

Definición 1.8. *Un Método de Runge-Kutta explícito de s estados* es un MPU, donde el número de estados s hace referencia a la cantidad de evaluaciones de la función $\boldsymbol{f}(t, \boldsymbol{y}(t))$ que se debe hacer en cada paso. Un método de este tipo se expresa de la forma

$$\begin{cases} \boldsymbol{y}_{i+1} = \boldsymbol{y}_i + h\boldsymbol{\phi}\left(t_i, \boldsymbol{y}_i, h\right), \\ \boldsymbol{y}(t_0) = \boldsymbol{y}_0, \end{cases} \tag{1.1.9}$$

donde

$$\boldsymbol{\phi}\left(t, \boldsymbol{y}, h\right) = \sum_{j=1}^{s} b_j \boldsymbol{k}_j,$$

con

$$\begin{aligned} \boldsymbol{k}_1(t, \boldsymbol{y}) &= \boldsymbol{f}(t, \boldsymbol{y}) \\ \boldsymbol{k}_2(t, \boldsymbol{y}) &= \boldsymbol{f}(t + hc_2, \boldsymbol{y} + ha_{21}\boldsymbol{k}_1) \\ \boldsymbol{k}_3(t, \boldsymbol{y}) &= \boldsymbol{f}(t + hc_3, \boldsymbol{y} + ha_{31}\boldsymbol{k}_1 + ha_{32}\boldsymbol{k}_2) \\ &\vdots \\ \boldsymbol{k}_s(t, \boldsymbol{y}) &= \boldsymbol{f}\left(t + hc_j, \boldsymbol{y} + h\sum_{k=1}^{s-1} a_{jk}\boldsymbol{k}_k\right), \quad \text{con} \quad j = 2, \ldots, s. \end{aligned} \tag{1.1.10}$$

Los coeficientes a_{jk}, b_j y c_j son los parámetros que determinan el método numérico y estos deben satisfacer que

$$\sum_{j=1}^{s} b_j = 1 \quad \text{y} \quad c_j = \sum_{k=1}^{s-1} a_{jk} \quad \text{con} \quad j = 2, \cdots, s.$$

Para la deducción de los métodos Runge-Kutta explícitos se debe tener en cuenta la Definición 1.8 y los coeficientes a_{jk}, b_j y c_j se determinan según el orden que se desee para cada método. Una forma de deducir un método es asignando un valor a s y determinar los coeficientes desarrollando las expansiones de Taylor para las funciones $\boldsymbol{k}_s$ en (1.1.10) y comparar este desarrollo con los coeficientes de la serie de Taylor. Las deducciones de estos métodos se pueden encontrar en [3] y [23].

Se sigue la notación de RKps para hacer referencia a un método de Runge-Kutta explícito con orden p y s estados. A continuación, se muestra la estructura de algunos métodos de Runge-Kutta de diferentes órdenes.

Runge-Kutta de orden dos con dos estados (RK22)

$$\begin{cases} \boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \frac{h}{2}\left(\boldsymbol{k}_1 + \boldsymbol{k}_2\right), \\ \boldsymbol{k}_1 = \boldsymbol{f}(t_i, \boldsymbol{y}_i), \\ \boldsymbol{k}_2 = \boldsymbol{f}\left(t_i + h, \boldsymbol{y}_i + h\boldsymbol{k}_1\right). \end{cases} \tag{1.1.11}$$

Runge-Kutta de tercera orden de tres estados (RK33)

$$\begin{cases} \boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \frac{h}{6}\left(\boldsymbol{k}_1 + 4\boldsymbol{k}_2 + \boldsymbol{k}_3\right), \\ \boldsymbol{k}_1 = \boldsymbol{f}(t_i, \boldsymbol{y}_i), \\ \boldsymbol{k}_2 = \boldsymbol{f}\left(t_i + \frac{1}{2}h, \boldsymbol{y}_i + \frac{1}{2}h\boldsymbol{k}_1\right), \\ \boldsymbol{k}_3 = \boldsymbol{f}(t_i + h, \boldsymbol{y}_i - h\boldsymbol{k}_1 + 2h\boldsymbol{k}_2). \end{cases} \tag{1.1.12}$$

Runge-Kutta de cuarta orden de cuatro estados (RK44)

$$\begin{cases} \boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \frac{h}{6}\left(\boldsymbol{k}_1 + 2\boldsymbol{k}_2 + 2\boldsymbol{k}_3 + \boldsymbol{k}_4\right), \\ \boldsymbol{k}_1 = \boldsymbol{f}(t_i, \boldsymbol{y}_i), \\ \boldsymbol{k}_2 = \boldsymbol{f}\left(t_i + \frac{1}{2}h, \boldsymbol{y}_i + \frac{1}{2}h\boldsymbol{k}_1\right), \\ \boldsymbol{k}_3 = \boldsymbol{f}\left(t_i + \frac{1}{2}h, \boldsymbol{y}_i + \frac{1}{2}h\boldsymbol{k}_2\right), \\ \boldsymbol{k}_4 = \boldsymbol{f}(t_i + h, \boldsymbol{y}_i + h\boldsymbol{k}_3). \end{cases} \tag{1.1.13}$$

Según los métodos de RK anteriores se podría suponer que si se toma un método de RK de s estados, este tendrá un orden equivalente al número de sus estados, pero la literatura muestra un resultado que no permite generalizar esta idea. Según [8] se tiene que un método de RK con s estados puede tener un orden del mismo valor a sus estados si $1 \leq s \leq 4$. Si $s > 4$ su orden puede variar con una o dos unidades menor al número de estados. En consecuencia, al utilizar un método de RK de orden mayor a 4 se estaría causando un gran costo computacional, dado que se realizan una gran cantidad de evaluaciones de la función $\boldsymbol{f}$, donde el orden de convergencia del método es menor al número de estados. Por tanto, en la práctica es conveniente utilizar el método de RK de orden 4.

Dado que los SEDO de interés a solucionar numéricamente en este trabajo son los sistemas rígidos, en la siguiente sección se mencionan algunas de las propiedades que los permiten identificar.

1.2. Sistemas rígidos

Muchos campos de aplicación, en particular la ingeniería química, cuentan con problemas de valores iniciales que involucran SEDO que exhiben un fenómeno conocido como *rigidez*, donde al intentar usar los métodos numéricos usuales como los métodos explícitos para resolverlos se presentan ciertas dificultades como el costo computacional. El problema de la rigidez se conoce desde hace algún

tiempo por Curtiss y Hirschfelder, pero en los últimos años ha llamado la atención por muchos analistas numéricos. Algunos conceptos teóricos de álgebra lineal y de SEDO que son útiles para la investigación de los sistemas rígidos se muestran en los apéndices A.1 y A.2. Además, la teoría que se estudia a continuación ha sido tomada de la literatura para la cual se recomienda seguir y profundizar en [2, 8, 18, 22, 24, 29] y [30].

El concepto de rigidez es difícil definir, dado que este puede depender de muchas características de los SEDO, como sus condiciones iniciales, parámetros, sus valores propios, el intervalo de integración y hasta el sistema mismo. Algunas de las posibles definiciones son las siguientes:

1. Se presenta un sistema rígido cuando las componentes de la solución varían unas más rápidos que otras, ver [2] y [22].

   Ejemplo 1.2. EDO lineal no homogénea, tomado de [2].

   $$\begin{cases} y' = -40y + 40t + 1 \\ y(0) = 4, \end{cases} \tag{1.2.1}$$

   con solución exacta $y(t) = t + 4e^{-40t}$.

   Solución. Analizando la solución anterior se observa que esta cuenta con dos componentes en su solución. La componente t que varía lentamente en comparación con la componente e^{-40t} que varía rápidamente, a medida que aumenta el valor de t la solución se convierte en $y \approx t$. Es así que al utilizar un método numérico explícito para aproximar su solución exacta se deben elegir tamaños de paso muy pequeños, lo cual causa un mayor trabajo computacional, por tanto se ve la necesidad de estudiar métodos con una adecuada zona de estabilidad. □

2. Un sistema de la forma $\boldsymbol{y'}(t) = A\boldsymbol{y} + \boldsymbol{b}$ es rígido cuando todos sus valores propios tienen parte real negativa y la relación entre las mismas es muy grande, seguir [22, 29] y [30].

   Ejemplo 1.3. SEDO lineal 2×2, tomado de [22].

   $$\begin{cases} x' = 0.01y, \\ y' = -100x - 100y + 2020, \\ x(0) = 0 \quad \text{y} \quad y(0) = 20. \end{cases} \tag{1.2.2}$$

   Solución. La matriz A del sistema (1.2.2) es

   $$A = \begin{bmatrix} 0 & 0.01 \\ -100 & -100 \end{bmatrix},$$

y sus valores propios son

$$\lambda_1 \approx -0.01 \quad y \quad \lambda_2 \approx -99.99.$$

Realizando el análisis acerca de los valores propios obtenidos, se mira que tienen parte real negativa y la diferencia entre ellos es muy grande, por tanto a este ejemplo se lo considera un problema rígido. □

3. Un sistema $\boldsymbol{y'}(t) = \boldsymbol{f}(t, \boldsymbol{y})$ es llamado rígido cuando el número de condicionamiento de la matriz asociada es muy grande, es decir

$$\kappa(A) = |||A||| \cdot \left|\left|\left|A^{-1}\right|\right|\right| >> 1,$$

con A la matriz asociada del sistema.

En particular cuando se trabaja con la norma dos de matrices se dice que un sistema es rígido, si todo valor propio λ_i de la matriz asociada posee una parte real negativa y si

$$q := \frac{\max\limits_{i=1,\cdots,n} |Re(\lambda_i)|}{\min\limits_{i=1,\cdots,n} |Re(\lambda_i)|} >> 1. \tag{1.2.3}$$

El sistema se llama débilmente rígido si $q = 10$ y rígido si $q > 10$. Ver [6, 20] y [29].

Ejemplo 1.4. Analizar los valores propios del Ejemplo 1.3 según esta definición.

Solución. Al verificar el cociente que presenta la definición anterior se tiene que

$$q = \frac{|-99.99|}{|-0.01|} = 9999.$$

Dado que q es estrictamente mayor que 1 y mayor que 10 el sistema del ejemplo analizado es un sistema rígido. □

Observación 1.2. Al trabajar con la norma dos en la definición del tercer ítem, el cociente (1.2.3) coincide con el análisis de los valores propios realizado en el segundo ítem. Para este caso las definiciones son equivalentes.

En las definiciones y ejemplos anteriores solo se ha trabajado con sistemas lineales, a continuación se da a conocer una definición para el caso de sistemas no lineales.

4. *SEDO rígido no lineal.* Un sistema $\boldsymbol{y'} = \boldsymbol{f}(t, \boldsymbol{y})$ no lineal es rígido si los valores propios del jacobiano $J = \frac{\partial f}{\partial y}$ se comportan de la misma manera que los valores propios de un sistema lineal. En el caso de los sistemas no lineales los valores propios dependen de la variable t, ver [2, 18].

Ejemplo 1.5. Sistema *stiff* no lineal 2×2, tomado de [1].

$$\begin{cases} x' = -1002x + 1000y^2, \\ y' = x - y - y^2, \\ x(0) = 1 \quad \text{y} \quad y(0) = 1. \end{cases} \tag{1.2.4}$$

Con soluciones exactas $x(t) = e^{-2t} \quad \text{y} \quad y(t) = e^{-t}$.

Solución. Para determinar si el sistema (1.2.4) es rígido, se calcula la matriz jacobiana y se encuentran los valores propios en varios instantes de tiempo t, además, se realiza el cálculo del número de condicionamiento de la la matriz obtenida en cada tiempo y se hace el análisis correspondiente. Según (1.2.4) la matriz jacobiana del sistema es

$$J = \begin{bmatrix} -1002 & 2000y(t) \\ 1 & -1 - 2y(t) \end{bmatrix},$$

donde sus valores propios y el número de condicionamiento en varios instantes de tiempo t son:

- En $t = 0$ la matriz jacobiana es

  $$J = \begin{bmatrix} -1002 & 2000 \\ 1 & -3 \end{bmatrix},$$

  sus valores propios son $\lambda_1 \approx -1003.9980$ y $\lambda_2 \approx -1.0020$ y el número de condicionamiento es $\kappa(J(0)) \approx 4974.1688$.

- En $t = 1$ la matriz jacobiana es

  $$J = \begin{bmatrix} -1002 & 735.7589 \\ 1 & -1.7358 \end{bmatrix},$$

  sus valores propios son $\lambda_1 \approx -1002.7350$ y $\lambda_2 \approx -1.0007$ y el número de condicionamiento es $\kappa(J(1)) \approx 1540.0006$.

- En $t = 5$ la matriz jacobiana es

  $$J = \begin{bmatrix} -1002 & 13.4759 \\ 1 & -1.0135 \end{bmatrix},$$

  sus valores propios son $\lambda_1 \approx -1002.0135$ y $\lambda_2 \approx -1.0000$ y el número de condicionamiento es $\kappa(J(5)) \approx 1002.1312$. □

De esta manera se mira que al calcular los valores propios del sistema (1.2.4) en varios valores de t, estos se comportan de forma similar al caso de los sistemas lineales. Los valores propios cuentan con una parte real negativa y la distancia entre ellas es muy grande, además, $\kappa(J(t))$ es un valor muy grande, por lo cual el sistema no lineal del Ejemplo 1.5 se considera un problema rígido.

En el siguiente capítulo se estudian métodos apropiados para la solución de sistemas rígidos como lo son los métodos numéricos implícitos.

Capítulo 2

Métodos de paso único implícitos

En este capítulo se presentan detalladamente los métodos numéricos implícitos de paso único para la solución de SEDO, el método de Euler y los métodos de Runge-Kutta. Se analizan algunos métodos de solución numérica de sistemas de ecuaciones no lineales, el método de punto fijo y el método de Newton, ya que estos son necesarios para el uso de los métodos implícitos, para profundizar se sugiere [7, 8, 12, 16, 18] y [32].

El desarrollo y avance de los métodos numéricos ha sido causado por la misma aparición de problemas más complicados de aproximar con los métodos usuales. Por ejemplo los SEDO que modelan problemas relacionados con la cinética química, la teoría de circuitos eléctricos y problemas de la guía de misiles, presentan cambios notorios en su comportamiento y cuentan con condiciones muy estrictas cuando se usan métodos explícitos para generar aproximaciones. Ver más en [10] y [22].

Al tratar de abordar un tipo de problemas más restringidos de solucionar numéricamente, los métodos numéricos explícitos tienen para garantizar la precisión, altas restricciones en el tamaño de paso utilizado. Por esta razón, es necesario el estudio de métodos numéricos con una mayor zona de estabilidad para lograr solucionarlos. Es así como aparecen los métodos numéricos implícitos que permiten encontrar la solución de sistemas de este tipo, como se ve en [8, 10] y [29].

En general, según [8] *un método es implícito* cuando conlleva la necesidad de resolver una ecuación o un sistema de ecuaciones no lineales de las que depende la solución en un nuevo valor de paso. En la Definición 1.1 un MPU de la forma (1.1.2) se lo considera implícito cuando el valor buscado $\boldsymbol{y}_{i+1}$ está en ambos lados de la ecuación, en este caso puede ser complicado despejar explícitamente el valor $\boldsymbol{y}_{i+1}$ de la función $\boldsymbol{f}(t, \boldsymbol{y}(t))$, por tanto se deben usar métodos numéricos para la solución de sistemas de ecuaciones no lineales como el método de punto fijo o el método de Newton.

Para la Definición 1.1 la forma de *los MPU implícitos* es

$$\begin{cases} \boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \boldsymbol{\phi}\left(t_{i+1}, \boldsymbol{y}_{i+1}, h\right), \\ \boldsymbol{y}(t_0) = \boldsymbol{y}_0. \end{cases} \tag{2.0.1}$$

A continuación, se introduce la deducción del método de Euler implícito. Siguiendo la idea de la deducción del método de Euler explícito por medio de la integración numérica vista en [3] y [23], esta puede ser generalizada a través de reglas de integración de un parámetro de la forma

$$\int_{t_i}^{t_{i+1}} \boldsymbol{f}(t, \boldsymbol{y}(t))dt \approx h[(1-\theta)\boldsymbol{f}(t_i, \boldsymbol{y}(t_i)) + \theta \boldsymbol{f}(t_{i+1}, \boldsymbol{y}(t_{i+1}))], \tag{2.0.2}$$

con $\theta \in [0, 1]$. Esta regla de integración introduce a la siguiente familia de métodos de un parámetro

$$\begin{cases} \boldsymbol{y}(t_{i+1}) = \boldsymbol{y}(t_i) + h[(1-\theta)\boldsymbol{f}(t_i, \boldsymbol{y}(t_i)) + \theta \boldsymbol{f}(t_{i+1}, \boldsymbol{y}(t_{i+1}))], \\ \boldsymbol{y}(t_0) = \boldsymbol{y}_0, \end{cases} \tag{2.0.3}$$

donde $t_{i+1} = t_i + h$, $0 \leq i \leq n-1$, $h = \frac{b-a}{n}$ y $\boldsymbol{f} : [a, b] \times \mathbb{R}^n \to \mathbb{R}^n$. Según (2.0.3) cuando $\theta = 0$ se obtiene el *método de Euler explícito* (1.1.7) y cuando $\theta = 1$ se tiene un método de la forma

$$\begin{cases} \boldsymbol{y}_{i+1} = \boldsymbol{y}_i + h\boldsymbol{f}(t_{i+1}, \boldsymbol{y}_{i+1}), \\ \boldsymbol{y}(t_0) = \boldsymbol{y}_0, \end{cases} \tag{2.0.4}$$

referido *al método de Euler implícito*, ya que a diferencia de Euler explícito requiere de la solución de ecuaciones implícitas para determinar $\boldsymbol{y}_{i+1}$.

Al comparar (2.0.4) con la forma general de un MPU en (1.1.2) se concluye que el método de Euler implícito es un MPU implícito con $\boldsymbol{\phi}\left(t_i, t_{i+1}, \boldsymbol{y}_i, \boldsymbol{y}_{i+1}, h\right) = \boldsymbol{f}(t_{i+1}, \boldsymbol{y}_{i+1})$. Además, se considera este método como un método implícito introductorio para solucionar numéricamente el problema de Cauchy (1.1.1).

Seguidamente, se estudian otros métodos numéricos implícitos como lo son los Métodos de Runge-Kutta.

2.1. Métodos de Runge-Kutta implícitos

Definición 2.1. *Un método de Runge-Kutta* puede ser escrito de la forma

$$\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + h\sum_{j=1}^{s} b_j \boldsymbol{k}_j, \tag{2.1.1}$$

donde

$$\boldsymbol{k}_j = \boldsymbol{f}(t_i + hc_j, \boldsymbol{y}_i + h\sum_{k=1}^{s} a_{jk}\boldsymbol{k}_k), \quad \text{con} \quad j = 1, \ldots, s \quad \text{y} \tag{2.1.2}$$

los coeficientes a_{jk}, b_j y c_j, deben cumplir que

$$\sum_{j=1}^{s} b_j = 1 \quad \text{y} \quad c_j = \sum_{k=1}^{s} a_{jk} \quad \text{con} \quad j = 1, \ldots, s. \tag{2.1.3}$$

Los valores de los coeficientes a_{jk}, b_j y c_j determinan el método numérico generalmente con la Tabla 2.1.

c_1	$\boldsymbol{a_{11}}$	$\boldsymbol{a_{12}}$	$\cdots$	$\boldsymbol{a_{1s}}$
c_2	a_{21}	$\boldsymbol{a_{22}}$	$\cdots$	$\boldsymbol{a_{2s}}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_s	a_{s1}	a_{s2}	$\cdots$	$\boldsymbol{a_{ss}}$
	b_1	b_2	$\cdots$	b_s

Tabla 2.1: Tabla de Butcher.

Los métodos de Runge-Kutta son implícitos (RKI), si la triangular superior y la diagonal de la Tabla 2.1 para los coeficientes a_{jk} tiene al menos un valor distinto de cero, en caso contrario los métodos son explícitos como los que se trabajan en el Capítulo 1. Para el caso de los métodos RKI a diferencia de Euler implícito son las etapas $\boldsymbol{k}_j$ las que aparecen de forma implícita. Estas no sólo dependen de las anteriores sino también de las no calculadas, por tanto para encontrar sus soluciones es necesario utilizar métodos numéricos para la solución de sistemas no lineales. Para complementar se recomienda [3, 8] y [12].

Para la deducción de los métodos RKI se puede proceder de varias maneras, una de ellas es considerando (2.1.1) donde los coeficientes se determinan según el orden que se desee para el método siguiendo expansiones Taylor, similarmente como ocurre para el caso de los métodos explícitos. Este proceso conduce a la solución de un sistema de ecuaciones no lineales no tan sencillo de resolver, por tanto, una herramienta útil para solucionar este problema son las *condiciones de orden simplificado*. Estas condiciones permiten obtener con mayor facilidad los valores de los coeficientes que concluyen el método, visto en [8, 16] y [20].

En este trabajo las deducciones que se presentan para los métodos RKI se realizan utilizando las condiciones de orden simplificado, condiciones que se ilustran en la siguiente definición. Además, para su deducción se recomienda seguir la teoría de la cuadratura de Gauss y los polinomios ortogonales que se mira en el Apéndice A.3, ya que a partir de esta se obtiene una clasificación de los mismos.

Definición 2.2. *Condiciones de orden simplificado*

$$B(p): \sum_{i=1}^{s} b_i c_i^{k-1} = \frac{1}{k}, \quad k = 1, \cdots, p,$$
$$C(l): \sum_{j=1}^{s} a_{ij} c_j^{k-1} = \frac{1}{k} c_i^k, \quad i = 1, \cdots, s, \quad k = 1, \cdots, l,$$
$$D(m): \sum_{i=1}^{s} b_i c_i^{k-1} a_{ij} = \frac{1}{k} b_j (1 - c_j^k), \quad j = 1, \cdots, s, \quad k = 1, \cdots, m,$$

con p el orden del método y s el número de etapas o estados del mismo. La deducción y el surgimiento de estas condiciones conllevan a la comprensión de conceptos teóricos más avanzados a los que se abordan en este trabajo, por tanto no se incluyen pero pueden ser consultadas en [8]. Sin embargo, se tiene presente que los coeficientes que se eligen satisfacen la cuadratura de Gauss, de tal manera que esta sea exacta para un polinomio de cierto grado. Estas condiciones permiten realizar las deducciones de los métodos de RKI de forma más práctica y sencilla, más adelante se realizan algunas de estas.

Antes de realizar las deducciones de algunos de los métodos de RKI, es válido aclarar que estos métodos varían y se clasifican de acuerdo al orden del método y a la forma en como este se obtiene, a las condiciones de orden que deben satisfacer y al polinomio de Legendre que se utiliza. Según lo mencionado es posible deducir varios métodos de RKI como los que se muestran en este trabajo y otros que se pueden encontrar en [8, 16] y [20]. Se usa la notación RKIps para hacer referencia a un RKI de orden p con s estados.

2.1.1. Método de Gauss-Legendre

El máximo orden que puede tener el método de RKI de s estados es de $p = 2s$, debe satisfacer $B(s), C(s)$ y $D(s)$ y los valores de c_j corresponden a las raíces del polinomio de Legendre $P_s^*(x)$. A seguir se presentan algunos ejemplos de RKI Gauss-Legendre.

RKI con $s = 1$ y orden $p = 2$ **(RKI21):** Según las condiciones de orden simplificado se tiene:

- $B(1): b_1 = 1.$
- $C(1): a_{11} = c_1.$
- $D(1): b_1 a_{11} = b_1(1 - c_1).$

El polinomio de Legendre es $P_1^* = 2x - 1$ y su raíz c_1 es $\frac{1}{2}$. Sustituyendo c_1 y resolviendo el sistema de ecuaciones anterior, los coeficientes del método son $a_{11} = 1$ y $b_1 = 1$. Consecuentemente se tiene

$$\begin{array}{c|c} 1/2 & 1/2 \\ \hline & 1 \end{array}.$$

Concluyendo el método

$$\begin{cases} \boldsymbol{y}_{i+1} = \boldsymbol{y}_i + h\boldsymbol{k}_1, \\ \boldsymbol{k}_1 = \boldsymbol{f}\left(t_i + \frac{h}{2}, \boldsymbol{y}_i + \frac{h}{2}\boldsymbol{k}_1\right). \end{cases} \tag{2.1.4}$$

RKI con $s = 2$ y orden $p = 4$ **(RKI42):** Según las condiciones de orden se debe satisfacer $B(2), C(2)$ y $D(2)$. Como los valores de c_j son las raíces del polinomio P_2^* es suficiente trabajar con $B(2)$ y $C(2)$. El sistema de ecuaciones que se obtiene es el siguiente:

- $B(2) : b_1 + b_2 = 1, \quad b_1c_1 + b_2c_2 = \frac{1}{2}$.
- $C(2) : a_{11} + a_{12} = c_1, \quad a_{11}c_1 + a_{12}c_2 = \frac{c_1^2}{2}, \quad a_{21} + a_{22} = c_2, \quad a_{21}c_1 + a_{22}c_2 = \frac{c_2^2}{2}$.

Calculando las raíces del polinomio $P_2^* = 6x^2 - 6x + 1$ se tiene que c_1 y c_2 son

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

Sustituyendo los valores de c_j y resolviendo los sistemas de ecuaciones, los coeficientes del método son

$$a_{11} = \frac{1}{4}, \quad a_{12} = \frac{\sqrt{3}}{6} + \frac{1}{4}, \quad a_{21} = \frac{-\sqrt{3}}{6} + \frac{1}{4}, \quad a_{22} = \frac{1}{4}, \quad b_1 = \frac{1}{2}, \quad b_2 = \frac{1}{2}.$$

Consecuentemente se tiene

$$\begin{array}{c|cc} \frac{1}{2} + \frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{\sqrt{3}}{6} + \frac{1}{4} \\ \frac{1}{2} - \frac{\sqrt{3}}{6} & \frac{-\sqrt{3}}{6} + \frac{1}{4} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}.$$

Concluyendo el método

$$\begin{cases} \boldsymbol{y}_{i+1} = \boldsymbol{y}_i + h\left(\frac{\boldsymbol{k}_1}{2} + \frac{\boldsymbol{k}_2}{2}\right), \\ \boldsymbol{k}_1 = \boldsymbol{f}\left(t_i + h\left(\frac{1}{2} + \frac{\sqrt{3}}{6}\right), \boldsymbol{y}_i + h\left(\frac{\boldsymbol{k}_1}{4} + \left(\frac{\sqrt{3}}{6} + \frac{1}{4}\right)\boldsymbol{k}_2\right)\right), \\ \boldsymbol{k}_2 = \boldsymbol{f}\left(t_i + h\left(\frac{1}{2} - \frac{\sqrt{3}}{6}\right), \boldsymbol{y}_i + h\left(\left(\frac{-\sqrt{3}}{6} + \frac{1}{4}\right)\boldsymbol{k}_1 + \frac{\boldsymbol{k}_2}{4}\right)\right). \end{cases} \tag{2.1.5}$$

Un producto de las condiciones que deben satisfacer los coeficientes de los métodos RKI Gauss-Legendre de orden $2s$, se muestra en el Corolario 2.1. Este resultado junto a su demostración se pueden encontrar en [8].

Corolario 2.1. *Un método de RKI tiene orden* $2s$ *si y solo si, sus coeficientes se eligen de la siguiente manera:*

1. *Elija* $c_1, c_2, \ldots, c_s$ *como los ceros de* P_s^*.
2. *Elija* $b_1, b_2, \ldots, b_s$ *para satisfacer la condición de* $B(s)$.
3. *Elija* $a_{jk}, \quad j, k = 1, 2, \ldots, s$ *para satisfacer la condición de* $C(s)$.

2.1.2. Método de Gauss-Radau

El máximo orden que puede alcanzar un método de RKI con s estados es $p = 2s - 1$ y similarmente a los métodos de Legendre satisface algunas de las condiciones de orden simplificado.

Algunas de las clasificaciones de los de métodos de Gauss-Radau son:

- **Radau I:** Este método debe satisfacer que $c_1 = 0$, los demás valores de c_j son las raíces del polinomio $P_s^* + P_{s-1}^*$, los valores de b_j satisfacen las condiciones de $B(s)$ y los a_{jk} satisfacen $C(s)$.
- **Radau II:** Este método debe satisfacer que $c_s = 1$, los demás valores de c_j son las raíces del polinomio $P_s^* - P_{s-1}^*$, los valores de b_j satisfacen las condiciones de $B(s)$ y los a_{jk} satisfacen $D(s)$.

Seguidamente, se presentan las deducciones de los métodos RKI Gauss-Radau con $s = 2$ y orden $p = 3$.

RKI32-Radau I: Según las condiciones de Radau I, los coeficientes de este método deben satisfacer $B(2)$ y $C(2)$, $c_1 = 0$ y $c_2 = \frac{2}{3}$ raíz del polinomio $P_2^* + P_1^* = 6x^2 - 4x = 0$. Desarrollando el sistema de ecuaciones que se genera y sustituyendo los valores que se obtienen, los coeficientes del método son

$$b_1 = \frac{1}{4}, \quad b_2 = \frac{3}{4}, \quad a_{11} = 0, \quad a_{12} = 0, \quad a_{21} = \frac{1}{3}, \quad a_{22} = \frac{1}{3}.$$

Consecuentemente se obtiene

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \frac{2}{3} & \frac{1}{3} & \frac{1}{3} \\ \hline & \frac{1}{4} & \frac{3}{4} \end{array}.$$

Concluyendo el método

$$\begin{cases} \boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \frac{h}{4}\left(\boldsymbol{k}_1 + 3\boldsymbol{k}_2\right), \\ \boldsymbol{k}_1 = \boldsymbol{f}\left(t_i, \boldsymbol{y}_i\right), \\ \boldsymbol{k}_2 = \boldsymbol{f}\left(t_i + \frac{2}{3}h, \boldsymbol{y}_i + \frac{h}{3}\left(\boldsymbol{k}_1 + \boldsymbol{k}_2\right)\right). \end{cases} \tag{2.1.6}$$

RKI32-Radau II: Según las condiciones de Radau II se tiene que:

- $B(2) : b_1 + b_2 = 1, \quad b_1c_1 + b_2c_2 = \frac{1}{2}$.
- $D(2) : b_1a_{11} + b_2a_{21} + b_2a_{21} = b_1(1 - c_1), \quad b_1c_1a_{11} + b_2c_2a_{21} = \frac{1}{2}b_1(1 - c_1^2), \quad b_1a_{12} + b_2a_{22} = b_2(1 - c_2), \quad b_1c_1a_{12} + b_2c_2a_{22} = \frac{1}{2}b_2(1 - c_2^2)$.

Calculando las raíces del polinomio $P_2^* - P_1^* = 6x^2 - 8x + 2 = 0$ se obtiene que $c_1 = \frac{1}{3}$ y $c_2 = 1$, luego sustituyendo los valores de c_j y solucionando el sistema de ecuaciones, los coeficientes del método son

$$b_1 = \frac{3}{4}, \quad b_2 = \frac{1}{4}, \quad a_{11} = \frac{1}{3}, \quad a_{12} = 0, \quad a_{21} = 1, \quad a_{22} = 0.$$

Consecuentemente se obtiene

$$\begin{array}{c|cc} \frac{1}{3} & \frac{1}{3} & 0 \\ 1 & 1 & 0 \\ \hline & \frac{3}{4} & \frac{1}{4} \end{array}.$$

Concluyendo el método

$$\begin{cases} \boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \frac{h}{4}\left(3\boldsymbol{k}_1 + \boldsymbol{k}_2\right), \\ \boldsymbol{k}_1 = \boldsymbol{f}\left(t_i + \frac{h}{3}, \boldsymbol{y}_i + \frac{h}{3}\boldsymbol{k}_1\right), \\ \boldsymbol{k}_2 = \boldsymbol{f}\left(t_i + h, \boldsymbol{y}_i + h\boldsymbol{k}_1\right). \end{cases} \tag{2.1.7}$$

2.1.3. Método de Gauss-Lobatto

Estos métodos de RKI se caracterizan porque el máximo orden que puede alcanzar un método de s estados es $p = 2s - 2$. Igualmente que los métodos anteriores sus coeficientes deben satisfacer algunas de las condiciones de orden simplificado, además Gauss-Lobatto satisface que $c_1 = 0$, $c_s = 1$ y los demás valores de c_j corresponden a las raíces del polinomio de Legendre $P_s^*(x) - P_{s-2}^*(x)$.

Algunas de las clasificaciones de los de métodos de Gauss-Lobatto son:

- **Lobatto I:** En este método los valores de b_j satisfacen las condiciones de $B(s)$ y los a_{jk} satisfacen $C(s)$.
- **Lobatto II:** En este método los valores de b_j satisfacen las condiciones de $B(s)$ y los a_{jk} satisfacen $D(s)$.

Posteriormente se presentan las deducciones de los métodos RKI Gauss-Lobatto con $s = 2$ y orden $p = 2$, para los cuales $c_1 = 0$ y $c_2 = 1$, valores que coinciden con las raíces del polinomio $P_2 - P_0 = 6x^2 - 6x + 1 - 1 = 0$.

RKI22-Lobatto I: Según las condiciones de orden los coeficientes de este método deben satisfacer $B(2)$ y $C(2)$, por tanto al desarrollar los sistemas de ecuaciones que se generan y al reemplazar los valores de c_j, los coeficientes que determinan el método son

$$a_{11} = 0, \quad a_{12} = 0, \quad a_{21} = \frac{1}{2}, \quad a_{22} = \frac{1}{2}, \quad b_1 = \frac{1}{2}, \quad b_2 = \frac{1}{2}.$$

Consecuentemente se obtiene

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

Concluyendo el método

$$\begin{cases} \boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \frac{h}{2}\left(\boldsymbol{k}_1 + \boldsymbol{k}_2\right), \\ \boldsymbol{k}_1 = \boldsymbol{f}\left(t_i, \boldsymbol{y}_i\right), \\ \boldsymbol{k}_2 = \boldsymbol{f}\left(t_i + h, \boldsymbol{y}_i + \frac{h}{2}\left(\boldsymbol{k}_1 + \boldsymbol{k}_2\right)\right). \end{cases} \tag{2.1.8}$$

RKI22-Lobatto II: Según las condiciones de orden los coeficientes de este método deben satisfacer $B(2)$ y $D(2)$, por tanto, al desarrollar los sistemas de ecuaciones que se generan y al reemplazar los valores de c_j se obtiene que

$$a_{11} = \frac{1}{2}, \quad a_{12} = 0, \quad a_{21} = \frac{1}{2}, \quad a_{22} = 0, \quad b_1 = \frac{1}{2}, \quad b_2 = \frac{1}{2}.$$

Consecuentemente se tiene

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

Concluyendo el método

$$\begin{cases} \boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \frac{h}{2}\left(\boldsymbol{k}_1 + \boldsymbol{k}_2\right), \\ \boldsymbol{k}_1 = \boldsymbol{f}\left(t_i, \boldsymbol{y}_i + \frac{h}{2}\boldsymbol{k}_1\right), \\ \boldsymbol{k}_2 = \boldsymbol{f}\left(t_i + h, \boldsymbol{y}_i + \frac{h}{2}\boldsymbol{k}_1\right). \end{cases} \tag{2.1.9}$$

Después de la deducción de cada uno de estos métodos, se analizan los coeficientes presentes y se tiene que para todos los métodos deducidos a excepción de RKI22-Lobatto II cumplen con las condiciones para ser un RKI, es decir los coeficientes satisfacen las condiciones de la Ecuación 2.1.3, por tanto RKI22-Lobatto II no se lo considera como un RKI, o en abuso de notación se lo asigna como un RKI22-Falso.

Como resultado del proceso anterior se muestra en la Tabla 2.2 la clasificación y variación de los métodos de RKI, se observa un resumen de las condiciones que necesita cada método para su deducción.

En comparación con los métodos de Runge-Kutta explícitos, los RKI son mucho más complicados en el momento de la implementación ya que requieren de mucho más tiempo y trabajo computacional. Sin embargo, la ventaja de estos es que son muy útiles para trabajar problemas de SEDO rígidos.

Dado que para la solución de SEDO con métodos implícitos se requiere de la solución de sistemas de ecuaciones no lineales, a continuación se introducen algunos de estos métodos iterativos.

Método	Orden	Condiciones	Polinomio	$\mathbf{c_j}$
Legendre	$2s$	$B(s), C(s), D(s)$	P_s^*	
Radau I	$2s-1$	$B(s), C(s)$	$P_s^* + P_{s-1}^*$	$c_1 = 0$
Radau II	$2s-1$	$B(s), D(s)$	$P_s^* - P_{s-1}^*$	$c_s = 1$
Lobatto I	$2s-2$	$B(s), C(s)$	$P_s^* - P_{s-2}^*$	$c_1 = 0$ y $c_s = 1$
Lobatto II	$2s-2$	$B(s), D(s)$	$P_s^* - P_{s-2}^*$	$c_1 = 0$ y $c_s = 1$

Tabla 2.2: Características métodos RKI.

2.2. Solución de ecuaciones no lineales

En ocasiones encontrar las raíces de ecuaciones no lineales no presentan un método analítico de solución, por tanto se necesita de métodos numéricos para poder encontrar sus aproximaciones. Entre estos métodos se destacan el método de punto fijo y el método de Newton, como se ve en [7, 31] y [32].

2.2.1. Método del punto fijo

Un *punto fijo* de una función g es un valor x para el cual $g(x) = x$. Estos problemas están asociados o relacionados con los problemas de la búsqueda de las raíces de una función $f(x) = 0$, dado que se puede definir g de distintas formas con respecto a la función f.

Se considera que el método de punto fijo es uno de los métodos más sencillos de analizar en la solución de problemas no lineales. Para utilizar este método es conveniente examinar propiedades y fundamentos teóricos como las que se enuncian a continuación, los cuales se estudian con profundidad en [7] y [32].

Para aproximar el punto fijo se escoge una aproximación inicial x_0 y se genera una sucesión $\{x_n\}$ a partir de la iteración

$$x_{n+1} = g(x_n), \tag{2.2.1}$$

denominada la *iteración de punto fijo* o *iteración funcional*.

Es importante verificar propiedades como la existencia y unicidad del punto fijo en un intervalo, por tanto el siguiente teorema presenta condiciones para su verificación además de garantizar su convergencia. La demostración de este teorema se puede ver en [7].

Teorema 2.1. *Teorema de punto fijo: Sea $g \in C[a,b]$ tal que $g(x) \in [a,b]$ para toda x en $[a,b]$. Además supongamos que existe que g' en (a,b) y una constante positiva $0 < k < 1$ tales que*

$$|g'(x)| \leq k, \quad \textit{para toda} \quad x \in (a,b),$$

entonces para cualquier número x_0 *en* $[a, b]$ *la sucesión definida por*

$$x_{n+1} = g(x_n), \quad n \geq 0,$$

converge al único punto fijo x^* *en* $[a, b]$.

A partir de las hipótesis del teorema del punto fijo, en [32] muestran resultados sobre las cotas de error para aproximar las raíces de una función, las cuales se relacionan con la convergencia del método.

Tomando como válidas las hipótesis del Teorema 2.1 y agregando la condición $g'(x^*) \neq 0$, se garantiza que la sucesión del punto fijo converge sólo linealmente. Para que el método de punto fijo puede lograr una convergencia cuadrática se debe confirmar que $g'(x^*) = 0$ y $g''(x^*) \neq 0$. De forma general para garantizar un orden de convergencia de orden p en el método de punto fijo, se debe cumplir que $g'(x^*) = g''(x^*) = \cdots = g^{p-1}(x^*) = 0$ y $g^p(x^*) \neq 0$, siendo $g \in C^p[a, b]$. Estos resultados y sus demostraciones se muestran con detalle en [32].

Según lo mencionado la búsqueda de los métodos de punto fijo cuadráticamente convergentes, necesariamente deben señalar las funciones cuyas derivadas se anulan en el punto fijo. La manera más fácil de plantear un problema de punto fijo relacionado con el de la búsqueda de raíces $f(x) = 0$ consiste en restar a x un múltiplo de $f(x)$, tomando la función g de la siguiente forma

$$g(x) = x - \phi(x)f(x), \tag{2.2.2}$$

donde $\phi(x)$ se conoce más adelante.

Para que el proceso iterativo de g sea cuadráticamente convergente se debe cumplir que $g(x^*) = 0$ cuando $f(x^*) = 0$, derivando (2.2.2) se tiene

$$g'(x) = 1 - \phi'(x)f(x) - f'(x)\phi(x)$$

y

$$g'(x^*) = 1 - \phi'(x^*)f(x^*) - f'(x^*)\phi(x^*),$$

de donde se llega a

$$g'(x^*) = 1 - f'(x^*)\phi(x^*),$$

para la cual $g'(x^*) = 0$ si y sólo si $\phi(x^*) = 1/f'(x^*)$, por tanto

$$x_{n+1} = g(x_n) = x_n - \frac{f(x_n)}{f'(x^*)}. \tag{2.2.3}$$

Del proceso realizado anteriormente se tiene que (2.2.3) es *la iteración del método de Newton*, en donde se hace la restricción que $f'(x^*) \neq 0$, con x^* raíz de $f(x) = 0$. A continuación se estudia detalladamente este método.

2.2.2. Método de Newton

Este es un método iterativo para aproximar las raíces de una ecuación no lineal, puede ser deducido gráficamente a partir de una de las convergencias más rápidas que ofrece el punto fijo o utilizando la serie de Taylor como se mira en [7] y [32].

Considerando la iteración del método de Newton, esta comienza con una aproximación inicial x_0 que genera una sucesión $\{x_n\}$, se tiene

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad \text{para} \quad n \geq 0,$$

la cual se cumple siempre que f sea diferenciable en $\mathbb{R}$ y $f'(x_n) \neq 0$ para todo $n \geq 0$.

Para este método es importante la elección de la condición inicial, dado que esta puede influir en la convergencia del mismo. A continuación, se enuncia el teorema de convergencia para el método de Newton y su demostración se muestra en [7].

Teorema 2.2. *Sea $f \in C^2[a,b]$. Si $x^* \in [a,b]$ es tal que $f(x^*) = 0$ y $f'(x^*) \neq 0$, entonces existe $\delta > 0$ tal que el método de Newton genera una sucesión $\{x_n\}$ que converge a x^* para cualquier aproximación inicial $x_0 \in [x^* - \delta, x^* + \delta]$.*

La convergencia del método de Newton se dice que es *local* y además esta se garantiza siempre y cuando x_0 sea lo suficiente cercana a x^*, ver más en [32].

Dado que las raíces de una ecuación pueden ser simples o tener una multiplicidad, en la literatura se enuncian definiciones, teoremas y demostraciones que permite identificar cuando un cero de una función tiene estas características. Algunos de estos resultados se relacionan con la convergencia cuadrática del método de Newton en sus raíces. Estos argumentos teóricos como otros que permiten estudiar con más profundidad todo acerca de estos métodos iterativos se muestran en [7, 31] y [32].

Así como se mencionó el método de Newton iterativo para encontrar la aproximación de la solución de una ecuación no lineal, este se puede generalizar para encontrar las soluciones numéricas de un sistema de n ecuaciones no lineales. Se introduce el método de Newton en $\mathbb{R}^n$ como se ve en [32].

Sea $F : U \subseteq \mathbb{R}^n \to \mathbb{R}^n$ una función definida y diferenciable en el conjunto abierto U tal que $F(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x}))^t$, donde $f_i : \mathbb{R}^n \to \mathbb{R}$, con $i = 1, 2, \ldots, n$, es una función diferenciable, el vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ y t la transpuesta del vector. Luego la solución de un sistema de ecuaciones no lineales viene dado por

$$F(x_1, x_2, \ldots, x_n) = \begin{bmatrix} f_1(x_1, x_2, \ldots, x_n) \\ f_2(x_1, x_2, \ldots, x_n) \\ \vdots \\ f_n(x_1, x_2, \ldots, x_n) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

donde para una aproximación inicial $\boldsymbol{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)})$ la iteración del método de Newton para el sistema es un vector de la forma

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \left[JF(\boldsymbol{x}^k)\right]^{-1} F(\boldsymbol{x}^k), \tag{2.2.4}$$

con $\boldsymbol{x}^k = (x_1^k, x_2^k, \ldots, x_n^k)$ y $[JF(\boldsymbol{x})]^{-1}$ la inversa de la matriz jacobiana

$$JF(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\boldsymbol{x}) & \frac{\partial f_1}{\partial x_2}(\boldsymbol{x}) & \ldots & \frac{\partial f_1}{\partial x_n}(\boldsymbol{x}) \\ \frac{\partial f_2}{\partial x_1}(\boldsymbol{x}) & \frac{\partial f_2}{\partial x_2}(\boldsymbol{x}) & \ldots & \frac{\partial f_2}{\partial x_n}(\boldsymbol{x}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1}(\boldsymbol{x}) & \frac{\partial f_n}{\partial x_2}(\boldsymbol{x}) & \ldots & \frac{\partial f_n}{\partial x_n}(\boldsymbol{x}) \end{bmatrix}.$$

Después de hacer la introducción a estos dos métodos se muestran las funciones iterativas de los métodos punto fijo y Newton asociadas a la aplicación de los métodos numéricos implícitos para la solución de SEDO. Para el caso del método de punto fijo su función de iteración es $\boldsymbol{g}(\boldsymbol{x})$ de la expresión (2.2.1). Para el caso de Newton la función iterativa es (2.2.4) aplicada a la función no lineal F, donde F es la expresión que cuenta con el término implícito igualada a cero. Se muestran algunos ejemplos presentando las formas de las funciones de iteración y algunas propiedades necesarias a tener en cuenta en el momento de su aplicación.

Ejemplo 2.1. Iteraciones de punto fijo y Newton en Euler implícito y RKI21.

Solución. Las funciones de iteración para los métodos de solución de ecuaciones no lineales son:

1. Euler implícito: $\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + h\boldsymbol{f}(t_{i+1}, \boldsymbol{y}_{i+1})$.

   - **Punto fijo:** $\boldsymbol{y}_{i+1}^{k+1} = \boldsymbol{g}(\boldsymbol{y}_{i+1}^k)$, donde

     $$\boldsymbol{g}(t_{i+1}, \boldsymbol{y}_i, \boldsymbol{y}_{i+1}, h) = \boldsymbol{y}_i + h\boldsymbol{f}(t_{i+1}, \boldsymbol{y}_{i+1}).$$

   - **Newton:** $\boldsymbol{y}_{i+1}^{k+1} = \boldsymbol{y}_{i+1}^k - \left[JF(\boldsymbol{y}_{i+1}^k)\right]^{-1} F(\boldsymbol{y}_{i+1}^k)$, donde

     $$F(t_{i+1}, \boldsymbol{y}_i, \boldsymbol{y}_{i+1}, h) = \boldsymbol{y}_{i+1} - \boldsymbol{y}_i - h\boldsymbol{f}(t_{i+1}, \boldsymbol{y}_{i+1}).$$

   La condición inicial para la primera iteración de estos es $\boldsymbol{y}_{i+1}^0 = \boldsymbol{y}_i$.

2. RKI21: $\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + h\boldsymbol{k}_1$, con $\boldsymbol{k}_1 = \boldsymbol{f}\left(t_i + \frac{h}{2}, \boldsymbol{y}_i + \frac{h}{2}\boldsymbol{k}_1\right)$. En este caso las iteraciones de los métodos de solución de ecuaciones no lineales se realizan con respectos a $\boldsymbol{k}_1$, ya que este es el término implícito.

- **Punto fijo:** $\boldsymbol{k}_1^{k+1} = \boldsymbol{g}(\boldsymbol{k}_1^k)$, donde

$$\boldsymbol{g}(t_i, \boldsymbol{y}_i, \boldsymbol{k}_1, h) = \boldsymbol{f}\left(t_i + \frac{h}{2}, \boldsymbol{y}_i + \frac{h}{2}\boldsymbol{k}_1\right).$$

- **Newton:** $\boldsymbol{k}_1^{k+1} = \boldsymbol{k}_1^k - \left[JF(\boldsymbol{k}_1^k)\right]^{-1} F(\boldsymbol{k}_1^k)$, donde

$$F(t_i, \boldsymbol{y}_i, \boldsymbol{k}_1, h) = \boldsymbol{k}_1 - \boldsymbol{f}\left(t_i + \frac{h}{2}, \boldsymbol{y}_i + \frac{h}{2}\boldsymbol{k}_1\right).$$

La condición inicial para cada método puede ser $\boldsymbol{k}_1^0 = \boldsymbol{f}\left(t_i + \frac{h}{2}, \boldsymbol{y}_i + \frac{h}{2}\boldsymbol{f}\left(t_i, \boldsymbol{y}_i\right)\right)$ o $\boldsymbol{k}_1^0 = \boldsymbol{f}(t_i, \boldsymbol{y}_i)$.

□

Con respecto a las condiciones iniciales para los términos implícitos en el caso de RKI, se hicieron ensayos y se observó que no se obtienen cambios notorios al utilizar una u otra condición inicial. Dado que los resultados fueron muy similares, para estos métodos se elige como condición inicial $\boldsymbol{k}_i^0 = \boldsymbol{f}(t_i, \boldsymbol{y}_i)$ ya que es menos costosa computacionalmente.

Ahora se presenta un método especial, un método que no sólo cuenta con un término implícito sino que tiene dos términos implícitos generando dos funciones no lineales, este es el caso del método RKI42. En este método los términos implícitos son $\boldsymbol{k}_1$ y $\boldsymbol{k}_2$ y las funciones no lineales a las cuales se aplica la iteración del método de Newton en el caso de $\mathbb{R}$ son

$$F_1(t_i, y_i, k_1, k_2, h) = k_1 - f\left(t_i + h\left(\frac{1}{2} + \frac{\sqrt{3}}{6}\right), y_i + h\left(\frac{k_1}{4} + \left(\frac{\sqrt{3}}{6} + \frac{1}{4}\right)k_2\right)\right).$$

$$F_2(t_i, y_i, k_1, k_2, h) = k_2 - f\left(t_i + h\left(\frac{1}{2} - \frac{\sqrt{3}}{6}\right), y_i + h\left(\left(\frac{-\sqrt{3}}{6} + \frac{1}{4}\right)k_1 + \frac{k_2}{4}\right)\right).$$

La iteración de Newton para RKI42 es

$$\boldsymbol{k}^{k+1} = \boldsymbol{k}^k - \left[\boldsymbol{JF}(\boldsymbol{k}^k)\right]^{-1} \boldsymbol{F}(\boldsymbol{k}^k), \tag{2.2.5}$$

con $\boldsymbol{k} = (k_1, k_2)^T$, $\boldsymbol{F} = (F_1, F_2)^T$, condición inicial $\boldsymbol{k}^0 = f\left(t_i, y_i\right)$ y

$$\boldsymbol{JF} = \begin{bmatrix} 1 - \frac{\partial f}{\partial k_1} & -\frac{\partial f}{\partial k_2} \\ -\frac{\partial f}{\partial k_1} & 1 - \frac{\partial f}{\partial k_2} \end{bmatrix}.$$

Para este método se obtiene una matriz de tamaño 2×2 en el caso de los $\mathbb{R}$, ya que k_1 y k_2 son los dos términos implícitos, por tanto al trabajar en $\mathbb{R}^n$ el tamaño de la matriz jacobiana es mucho más

grande. Por ejemplo para $\mathbb{R}^2$ se tiene una matriz de tamaño 4×4 ya que cuenta con dos funciones vectoriales no lineales

$$\boldsymbol{F_1} = \begin{bmatrix} k_1[0] - f_1\left(t_i + hc_1, \boldsymbol{y}_i + h\left(a_{11}\boldsymbol{k}_1 + a_{12}\boldsymbol{k}_2\right)\right) \\ k_1[1] - f_2\left(t_i + hc_1, \boldsymbol{y}_i + h\left(a_{11}\boldsymbol{k}_1 + a_{12}\boldsymbol{k}_2\right)\right) \end{bmatrix},$$

$$\boldsymbol{F_2} = \begin{bmatrix} k_2[0] - f_1\left(t_i + hc_2, \boldsymbol{y}_i + h\left(a_{21}\boldsymbol{k}_1 + a_{22}\boldsymbol{k}_2\right)\right) \\ k_2[1] - f_2\left(t_i + hc_2, \boldsymbol{y}_i + h\left(a_{21}\boldsymbol{k}_1 + a_{22}\boldsymbol{k}_2\right)\right) \end{bmatrix}.$$

Se considera la iteración de Newton (2.2.5) que se aplica en las funciones anteriores, con condiciones iniciales $\boldsymbol{k}_1^0 = \boldsymbol{f}\left(t_i, y_i\right)$, $\boldsymbol{k}_2^0 = \boldsymbol{f}\left(t_i, y_i\right)$ y con la matriz jacobiana de la forma

$$\boldsymbol{JF} = \begin{bmatrix} 1 - \frac{\partial f_1}{\partial k_1[0]} & -\frac{\partial f_1}{\partial k_1[1]} & -\frac{\partial f_1}{\partial k_2[0]} & -\frac{\partial f_1}{\partial k_2[1]} \\ -\frac{\partial f_2}{\partial k_1[0]} & 1 - \frac{\partial f_2}{\partial k_1[1]} & -\frac{\partial f_2}{\partial k_2[0]} & -\frac{\partial f_2}{\partial k_2[1]} \\ -\frac{\partial f_1}{\partial k_1[0]} & -\frac{\partial f_1}{\partial k_1[1]} & 1 - \frac{\partial f_1}{\partial k_2[0]} & -\frac{\partial f_1}{\partial k_2[1]} \\ -\frac{\partial f_2}{\partial k_1[0]} & -\frac{\partial f_2}{\partial k_1[1]} & -\frac{\partial f_2}{\partial k_2[0]} & 1 - \frac{\partial f_2}{\partial k_2[1]} \end{bmatrix}.$$

Es válido mencionar que para encontrar las soluciones de los sistemas no lineales no se realiza el cálculo de la inversa, en la práctica lo que se hace es solucionar el sistema $A\boldsymbol{x} = \boldsymbol{b}$.

Según el procedimiento que se debe realizar para la aplicación de newton en RKI42 para $\mathbb{R}^n$, se tiene que uno de los cálculos que se llevan a cabo es el cálculo de matrices inversas. Proceso que resulta ser muy costoso computacionalmente cuando se trabaja con matrices de grandes dimensiones, por tanto es conveniente realizar ciertos cambios y estudiar este método con algunas mejorías o modificaciones con el fin de disminuir este problema.

2.2.3. Modificaciones método de Newton

En esta sección se presenta una modificación del método de Newton realizando una aproximación de la inversa de la matriz jacobiana, ya que este es el procedimiento que causa un gran trabajo computacionalmente.

Método de Jacobi

El método de Jacobi es un método iterativo, es decir es un método que a partir de una serie no exacta de pasos aproximan la solución de un sistema lineal

$$A\boldsymbol{x} = \boldsymbol{b}, \tag{2.2.6}$$

donde $A \in \mathbb{R}^{n\times n}$ es una matriz invertible, $\boldsymbol{x} \in \mathbb{R}^n$ es el vector solución y $\boldsymbol{b} \in \mathbb{R}^n$ el vector resultante del producto entre A y el vector $\boldsymbol{x}$. Este método resulta de descomponer la matriz A de la forma

$$A = M - N, \tag{2.2.7}$$

donde M es una matriz arbitraria no singular y N resulta de la diferencia $M - A$.

En este método se considera que $M = D$ y $N = L + U$, donde D es la matriz diagonal de A, L una matriz triangular inferior y U una matriz triangular superior, según esto se tiene que

$$A = M - N = D - (L + U),$$

luego al sustituir A en (2.2.6) se llega a

$$[D - (L + U)]\,\boldsymbol{x} = \boldsymbol{b}.$$

Al multiplicar por D^{-1} y despejar $\boldsymbol{x}$ en la igualdad anterior se obtiene

$$\boldsymbol{x} = D^{-1}(L + U)\,\boldsymbol{x} + D^{-1}\boldsymbol{b},$$

donde su forma iterativa es

$$\boldsymbol{x}^{k+1} = D^{-1}(L + U)\,\boldsymbol{x}^k + D^{-1}\boldsymbol{b}. \tag{2.2.8}$$

Como $L + U = D - A$, reemplazando en (2.2.8) la forma iterativa se abrevia a

$$\boldsymbol{x}^{k+1} = \left[I - D^{-1}A\right]\boldsymbol{x}^k + D^{-1}\boldsymbol{b}, \tag{2.2.9}$$

la cual es la *iteración del método de Jacobi* e I es la matriz identidad. Considerando la iteración (2.2.4) del método de Newton y al multiplicar esta igualdad por $JF(\boldsymbol{x}^k)$ se tiene

$$JF\left(\boldsymbol{x}^k\right)\boldsymbol{x}^{k+1} = JF\left(\boldsymbol{x}^k\right)\boldsymbol{x}^k - \boldsymbol{F}\left(\boldsymbol{x}^k\right). \tag{2.2.10}$$

Al relacionar (2.2.10) con el problema $A\boldsymbol{x} = \boldsymbol{b}$, se mira que $A = JF\left(\boldsymbol{x}^k\right)$ y $\boldsymbol{b} = JF\left(\boldsymbol{x}^k\right)\boldsymbol{x}^k - \boldsymbol{F}\left(\boldsymbol{x}^k\right)$. Además la matriz diagonal $D = DJF\left(\boldsymbol{x}^k\right)$ está formada por la diagonal principal de la matriz jacobiana $JF\left(\boldsymbol{x}^k\right)$, luego aplicando la iteración del método de Jacobi para el sistema (2.2.10) se obtiene

$$\boldsymbol{x}^{k+1} = \left[I - \left[DJF\left(\boldsymbol{x}^k\right)\right]^{-1} JF\left(\boldsymbol{x}^k\right)\right]\boldsymbol{x}^k + \left[DJF\left(\boldsymbol{x}^k\right)\right]^{-1}\left[JF\left(\boldsymbol{x}^k\right)\boldsymbol{x}^k - \boldsymbol{F}\left(\boldsymbol{x}^k\right)\right],$$

de donde

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \left[DJF\left(\boldsymbol{x}^k\right)\right]^{-1}\boldsymbol{F}\left(\boldsymbol{x}^k\right), \tag{2.2.11}$$

que hace referencia a la *iteración de Newton modificado con método de Jacobi*. Al comparar la iteración de Newton (2.2.4) con la iteración de Jacobi (2.2.11), lo que se realiza es un cambio en el cálculo de las matrices en el momento de hacer cada iteración, es decir se sustituye la matriz jacobiana $JF\left(\boldsymbol{x}^k\right)$ por la diagonal de esta matriz $DJF\left(\boldsymbol{x}^k\right)$.

A continuación, se presenta la definición de una matriz diagonalmente dominante que es necesaria para la comprensión del teorema que garantiza la convergencia del método de Jacobi. Este teorema y su demostración se muestran en [5].

Definición 2.3. Sea $A \in \mathbb{R}^{m\times n}$. Una matriz A es *diagonalmente dominante* por filas si

$$a_{ii} \geq \sum_{k=1}^{n} |a_{ik}| \quad \forall i = 1, 2, \ldots, m.$$

Cuando se cumpla la desigualdad anterior, la matriz A es *estrictamente diagonal dominante por filas*. Similarmente una matriz A es *diagonalmente dominante por columnas* si y sólo si

$$a_{ii} \geq \sum_{k=1}^{m} |a_{ki}| \quad \forall i = 1, 2, \ldots, n$$

y es *estrictamente diagonalmente dominante* por columnas si se cumple la desigualdad anterior.

Teorema 2.3. *Si A es una matriz estrictamente diagonalmente dominante, entonces la iteración asociada al método de Jacobi converge para cualquier valor inicial.*

A continuación, se realiza un análisis cualitativo de la estabilidad de los métodos numéricos, ya que es una propiedad esencial para la solución numérica de los problemas rígidos.

2.3. Estabilidad absoluta

Para obtener la solución numérica en el problema de Cauchy (1.1.1), requiere en la práctica la elección del tamaño de paso de integración $h > 0$ además de un método numérico. Pues al utilizar un valor de h muy pequeño conlleva a realizar una gran cantidad de cálculos causando un costo computacional, como también por la precisión finita del computador podría causar errores mayores de redondeo. Es por esto que se debe utilizar un tamaño de paso lo más grande posible, teniendo en cuenta que la solución numérica se comporte de manera similar a la solución teórica. Esta teoría ha sido tomada de [8, 16, 23, 24] y [30].

La elección de un tamaño de paso de integración h fijo está relacionado con el concepto de *estabilidad absoluta*. Para su estudio se debe realizar un análisis cualitativo para el conjunto de puntos en el plano complejo, en el que la solución numérica tenga el mismo comportamiento de la solución analítica, este conjunto de puntos se conoce como la *región de estabilidad*.

Para los problemas rígidos es necesario encontrar métodos numéricos que no impongan una limitación en el tamaño de paso como los métodos explícitos que cuentan con una región de estabilidad absoluta pequeña. Debido a la complejidad de este análisis se restringe este estudio a problemas puramente lineales con coeficientes constantes y se realiza para el caso de SEDO.

Se considera un sistema de la forma

$$\begin{cases} \boldsymbol{y}'(t) = A\boldsymbol{y}(t), \\ \boldsymbol{y}(0) = \boldsymbol{y}_0, \end{cases} \tag{2.3.1}$$

con $A \in \mathbb{R}^{n\times n}$, $\boldsymbol{y}_0 \in \mathbb{R}^n$ y solución teórica $\boldsymbol{y}(t_{i+1}) = \boldsymbol{y}(t_i)e^{Ah}$ en $t_{i+1} = t_i + h$.

Haciendo un cambio de base

$$\boldsymbol{y}(t) = Q\boldsymbol{w}(t) \Rightarrow \boldsymbol{y}'(t) = Q\boldsymbol{w}'(t), \tag{2.3.2}$$

con Q una matriz constante no singular y reemplazando (2.3.2) en (2.3.4) se tiene

$$Q\boldsymbol{w}'(t) = AQ\boldsymbol{w}(t), \tag{2.3.3}$$

al multiplicar (2.3.3) por la matriz inversa de Q se llega a

$$\boldsymbol{w}'(t) = Q^{-1}AQ\boldsymbol{w}(t)$$

consiguiendo el sistema

$$\boldsymbol{w}'(t) = \hat{A}\boldsymbol{w}(t), \tag{2.3.4}$$

donde $\hat{A} = Q^{-1}AQ$ y su solución teórica es $\boldsymbol{w}(t_{i+1}) = \boldsymbol{w}(t_i)e^{\hat{A}h}$.

Si la matriz $\hat{A}$ se elige de la forma canónica de Jordan de A, el Sistema 2.3.4 y la aproximación numérica llegan a ser en cierta medida desacoplados. Esto significa que para cada valor propio distinto λ_i, una de las ecuaciones en el Sistema 2.3.4 tiene la forma

$$y'(t) = \lambda y(t) \tag{2.3.5}$$

y otras componentes que corresponden al mismo bloque de Jordan dependerán de esta solución pero no contribuirá en su comportamiento, ver más en [8]. Conceptos básicos de la forma canónica de Jordan se muestran en el Apéndice A.1 y se estudian con detalle en [17]. La solución teórica de (2.3.5) es $y_{i+1} = e^z y_i$ en $t_{i+1} = t_i + h$, donde se hace $z = \lambda h$.

A continuación, se presentan algunas definiciones y propiedades útiles para el estudio de la estabilidad de los métodos numéricos.

Definición 2.4. Sea un MPU aplicado al problema (2.3.5) que conduzca a $y_{i+1} = R(z)y_i$, la expresión $R(z)$ se denomina *factor de amplificación* y el conjunto

$$S := \{z \in \mathbb{C} : |R(z)| < 1\}$$

se conoce como *región de estabilidad absoluta* o *dominio de estabilidad*. La intersección de la región S con la recta real determina el *intervalo de estabilidad absoluta* del MPU.

Algunos métodos numéricos se caracterizan porque son estables en todo el semiplano izquierdo $\mathbb{C}^-$, este es precisamente el conjunto de valores propios donde la solución exacta de (2.3.5) también es estable. Una característica deseable para un método numérico es que conserve esta propiedad de estabilidad, ver [16].

Definición 2.5. Un método es *A-estable*, si su dominio de estabilidad satisface

$$\mathbb{C}^- \subset S = \{z \in \mathbb{C} : \mathrm{Re}(z) < 0\},$$

donde $\mathrm{Re}(z)$ es la parte real de z.

Hay un gran interés especialmente en métodos numéricos para los cuales la región de estabilidad incluye todo el semiplano izquierdo, este es el caso de los métodos implícitos, métodos con una propiedad ideal para ser aplicados a los problemas rígidos, ver [8].

Definición 2.6. Un método es llamado *L-estable*, si este es A-estable y además

$$\lim_{z\to\infty} R(z) = 0.$$

Después de mencionar algunos de los conceptos teóricos importantes en el estudio de la estabilidad de los métodos numéricos, iniciamos un análisis de la estabilidad para algunos métodos explícitos en el caso de los $\mathbb{R}$ con el fin de introducir esta temática. El estudio de la estabilidad absoluta para estos métodos se puede ver con detalle en [3] y [23].

Para el caso del método Euler explícito $y_{i+1} = y_i + hf(t_i, y_i)$, aplicando la expresión (2.3.5) se tiene

$$y_{i+1} = y_i + h\lambda y_i,$$

de lo cual se obtiene que

$$y_{i+1} = (1 + h\lambda)y_i.$$

Haciendo $z = h\lambda$ se llega a que el factor de amplificación es

$$R(z) = 1 + z. \tag{2.3.6}$$

Analizando la expresión (2.3.6) según la Definición 2.4, se tiene

$$|1 + z| < 1,$$

luego como $z \in \mathbb{C}$ y es de la forma $z = a + bi$, reemplazando en la expresión anterior y calculando la norma se llega a

$$(a + 1)^2 + b^2 < 1, \tag{2.3.7}$$

siendo (2.3.7) el dominio de estabilidad para Euler explícito, referente a la parte interna de una circunferencia de centro $(-1, 0)$ y radio 1.

Para el caso de los métodos de Runge-Kutta explícitos se procede de forma similar al caso anterior. Se analiza la región de estabilidad para RK22 aplicando (2.3.5) al esquema de estos métodos (1.1.10) y reemplazando en (1.1.9).

De lo anterior se tiene que

$$k_1 = f(t_i, y_i) = \lambda y_i \quad \text{y} \quad k_2 = f(t_i + h, y_i + hk_1) = \lambda y_i + h\lambda^2 y_i. \tag{2.3.8}$$

Reemplazando (2.3.8) en la expresión genera del método $y_{i+1} = y_i + h/2(k_1 + k_2)$ se llega a

$$y_{i+1} = \left(1 + h\lambda + \frac{h^2\lambda^2}{2}\right) y_i.$$

Haciendo $z = h\lambda$, el factor de amplificación es

$$R(z) = \left(1 + z + \frac{z^2}{2}\right). \tag{2.3.9}$$

Analizando la expresión (2.3.9) según la Definición 2.4 se tiene que

$$\left|1 + z + \frac{z^2}{2}\right| < 1,$$

siendo esta la expresión que define la región de estabilidad del método RK22.

En la Figura 2.1 se muestran las regiones de estabilidad para Euler y RK de varias órdenes. Se observa que para los métodos presentes en esta ilustración, entre mayor es su orden mayor es su región de estabilidad.

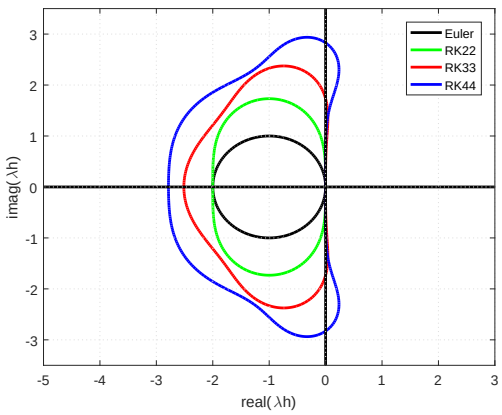


Figura 2.1: Regiones de estabilidad absoluta métodos numéricos explícitos. Tomada de [3].

A continuación, se analizan propiedades de estabilidad para algunos métodos implícitos de paso único como el método de Euler y los métodos de Runge-Kutta.

2.3.1. Estabilidad del método Euler implícito

Se analiza la región de estabilidad del método de Euler implícito en $\mathbb{R}$. La forma general del método de Euler implícito para una EDO es $y_{i+1} = y_i + hf(t_{i+1}, y_{i+1})$, donde aplicando (2.3.5) se tiene

$$y_{i+1} = y_i + h\lambda y_{i+1},$$

de lo cual se llega a

$$y_{i+1} = \frac{1}{(1-h\lambda)} y_i,$$

luego haciendo $z = h\lambda$ se sigue

$$\begin{aligned} y_{i+1} &= \frac{1}{(1-z)} y_i \\ y_{i+1} &= R(z)\, y_i. \end{aligned} \tag{2.3.10}$$

Analizando $R(z)$ visto en (2.3.10) y según la Definición 2.4, se tiene

$$\left| \frac{1}{1-z} \right| < 1,$$

donde $z \in \mathbb{C}$ y es de la forma $z = a + bi$, luego reemplazando se obtiene que

$$|1 - a - bi| > 1,$$

aplicando la norma se llega a

$$(a-1)^2 + b^2 > 1. \tag{2.3.11}$$

Según (2.3.11) el dominio de estabilidad para el método de Euler implícito es el complemento de la parte interna de la circunferencia de centro $(1, 0)$ y radio 1. Este cubre todo el semiplano negativo y gran parte del semiplano positivo, por tanto el método Euler implícito es un método A-estable. Este método además de ser A-estable cumple la condición de la Definición 2.6 para ser un método L-estable. La Figura 2.2 ilustra la región de estabilidad que representa este método.

Observación 2.1. Para elaborar la Figura 2.2 se utilizó el software Matlab. Esta ilustración se realiza al considerar el conjunto de puntos z en el plano complejo, tales que cumplan la condición $|R(z)| < 1$. Es decir, se representó gráficamente los puntos que están en la región de estabilidad del método Euler implícito. Las regiones de estabilidad que se muestran más adelante se trabajan de forma similar.

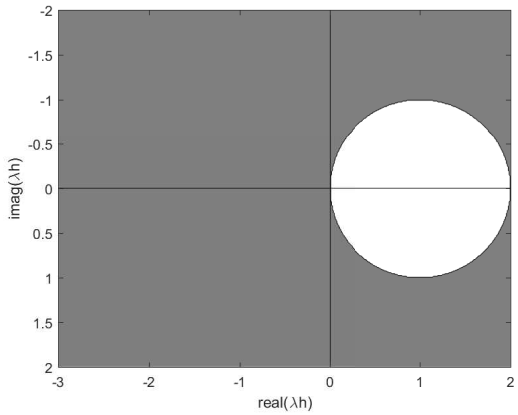


Figura 2.2: Región de estabilidad Euler implícito.

A continuación, se trabaja la zona de estabilidad de este método para el caso de SEDO. La forma general del método es $\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + h\boldsymbol{f}(t_{i+1}, \boldsymbol{y}_{i+1})$, luego aplicando (2.3.4) se tiene

$$\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + h\hat{A}\boldsymbol{y}_{i+1},$$

de donde

$$\left(I - h\hat{A}\right)\boldsymbol{y}_{i+1} = \boldsymbol{y}_i,$$

de lo cual se obtiene que

$$\boldsymbol{y}_{i+1} = \left(I - h\hat{A}\right)^{-1}\boldsymbol{y}_i, \tag{2.3.12}$$

luego, haciendo $Z = h\hat{A}$ se tiene

$$\boldsymbol{y}_{i+1} = (I - Z)^{-1}\boldsymbol{y}_i,$$
$$\boldsymbol{y}_{i+1} = R(Z)\boldsymbol{y}_i.$$

Analizando $R(Z)$ visto en (2.3.12) y según la Definición 2.4, se tiene

$$\left|\left|\left|\left(I - h\hat{A}\right)^{-1}\right|\right|\right| < 1, \tag{2.3.13}$$

lo cual corresponde a una desigualdad entre una norma matricial y la unidad. En el Apéndice A.1 se muestran los conceptos teóricos sobre las normas matriciales.

Teniendo en cuenta que $\hat{A}$ es una matriz canónica de Jordan, con B_i el bloque de Jordan asociado a λ_i, (2.3.13) se puede escribir de la forma

$$\left|\left|\left|\begin{pmatrix} I_1 - hB_1 & 0 & \cdots & 0 \\ 0 & I_2 - hB_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & I_r - hB_r \end{pmatrix}^{-1}\right|\right|\right| < 1,$$

calculando la inversa se llega a

$$\left|\left|\left|\begin{pmatrix} (I_1 - hB_1)^{-1} & 0 & \cdots & 0 \\ 0 & (I_2 - hB_2)^{-1} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & (I_r - hB_r)^{-1} \end{pmatrix}\right|\right| < 1,$$

lo cual es equivalente analizar cada uno de las componentes de la matriz, es decir

$$|||(I_1 - hB_1)^{-1}||| < 1, |||(I_2 - hB_2)^{-1}||| < 1, \ldots, |||(I_r - hB_r)^{-1}||| < 1, \tag{2.3.14}$$

que son las normas asociadas a cada uno de los bloques de Jordan que dependen de un valor propio λ_i. Como las funciones que dependen del mismo valor propio no afectan su comportamiento, entonces el análisis que se realiza para cada norma en (2.3.14) es el mismo que en el caso de los $\mathbb{R}$. Este análisis es realizado para cada uno de los λ_i, de esta forma se consigue la región para cada uno de estos valores, dado que algunas regiones son mucho más estrictas que otras, entonces se realiza su intersección para obtener la región de estabilidad absoluta final del método.

2.3.2. Estabilidad métodos Runge-Kutta implícitos

Para un método RKI de s estados definido por la tabla

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array},$$

similarmente como se mencionó en el método de Euler implícito, el estudio de sus regiones de estabilidad para el caso de sistemas es el mismo que en los $\mathbb{R}$, ya que al realizar el análisis de la matriz canónica de Jordan en (2.3.4) cada valor propio distinto se restringe al análisis del problema (2.3.5) $y'(t) = \lambda y(t)$ y sus otras componentes que corresponden al mismo bloque de Jordan dependerán de esta solución pero no contribuirá en su comportamiento, ver [8]. Es así como para un método de RK las regiones de estabilidad se trabajan para el caso de los reales, pudiéndose extender esta idea para sistemas.

Las expresiones (2.1.1) y (2.1.2) que determinan un método de RKI, en el caso de los $\mathbb{R}$ se pueden escribir como

$$y_{i+1} = y_i + hb^T \boldsymbol{k}, \tag{2.3.15}$$

con

$$\boldsymbol{k} = f(t_i \mathbf{1} + hC, y_i \mathbf{1} + hA\boldsymbol{k}), \tag{2.3.16}$$

donde $b^T = (b_1, b_2, \ldots, b_s)$, $\boldsymbol{k} = (k_1, k_2, \ldots, k_s)^T$, $C = (c_1, c_2, \ldots, c_s)^T$, $A = (a_{jk})_{j,k=1}^s$ y $\mathbf{1} = (1_1, 1_2, \ldots, 1_s)^T$ el vector unitario de dimensión s.

Luego, aplicado el problema (2.3.5) a la expresión (2.3.16) se tiene

$$\boldsymbol{k} = \lambda(y_i\mathbf{1} + hA\boldsymbol{k}),$$

de lo cual se sigue

$$(I - \lambda hA)\boldsymbol{k} = \lambda y_i\mathbf{1},$$

de donde se obtiene que

$$\boldsymbol{k} = (I - zA)^{-1}\lambda y_i\mathbf{1}, \quad \text{con} \quad z = \lambda h. \tag{2.3.17}$$

Reemplazando (2.3.17) en (2.3.15) se llega a

$$y_{i+1} = (1 + zb^T(I - zA)^{-1}\mathbf{1})y_i,$$

donde el factor de amplificación para un método de RKI de s estados es

$$R(z) = 1 + zb^T(I - zA)^{-1}\mathbf{1}. \tag{2.3.18}$$

A seguir se realiza el procedimiento en el cual se obtiene la función de estabilidad para el método RKI21, este proceso se realiza utilizando la expresión (2.3.18).

Según RKI21 la tabla de Butcher para este método es

$$\begin{array}{c|c} 1/2 & 1/2 \\ \hline & 1 \end{array}.$$

Reemplazando los valores de los coeficientes del método en (2.3.18) se obtiene

$$R(z) = 1 + z(1)(I - z(1/2))^{-1}\mathbf{1}, \tag{2.3.19}$$

de lo cual se concluye que

$$R(z) = \frac{2 + z}{2 - z}. \tag{2.3.20}$$

Es válido mencionar que la matriz identidad I y el vector $\mathbf{1}$ de la expresión (2.3.19) para este método son reales, es decir $I = 1 = \mathbf{1}$, ya que el número de estados de RKI21 es $s = 1$. Caso contrario, las dimensiones de estos varían según el número de estados.

Analizando el factor de amplificación (2.3.20) según la Definición 2.4 se tiene

$$\left|\frac{2 + z}{2 - z}\right| < 1, \tag{2.3.21}$$

donde su dominio de estabilidad está formado por aquellos $z \in \mathbb{C}$, tales que $|2 + z| < |2 - z|$. Tomando $z = a + bi$ un número complejo y reemplazando en (2.3.21) se llega a

$$\left|\frac{2 + a + bi}{2 - a - bi}\right| < 1,$$

de donde

$$\sqrt{(a+2)^2+b^2} < \sqrt{(2-a)^2+b^2},$$

de lo cual se obtiene que

$$a < 0.$$

Es decir

$$Re(z) < 0.$$

Esto es justamente el conjunto de puntos en el plano complejo con parte real negativa, por lo tanto el método RKI21 es un método A-estable. La Figura 2.3 ilustra claramente esta región de estabilidad.

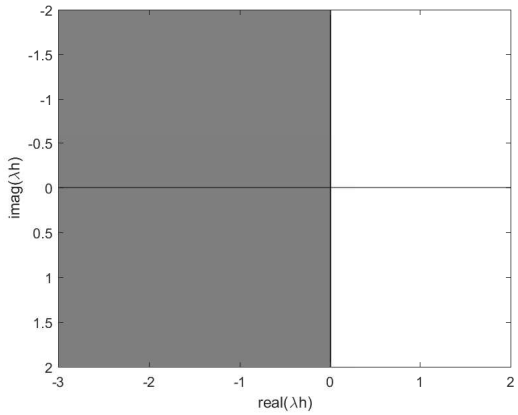


Figura 2.3: Región de estabilidad RKI21.

A seguir se mencionan algunas proposiciones relacionadas con la estabilidad de los métodos de RKI vistas en [8] y [16].

Proposición 2.1. *(A, b, c) denotan un método de RKI, luego su función de estabilidad está dada por*

$$R(z) = \frac{det\left(I - zA + z\mathbf{1}b^T\right)}{det(I - zA)}.$$

Proposición 2.2. *Si un método RKI de s estados con A no singular satisface una de las siguientes condiciones*

$$a_{sk} = b_k, \quad k = 1, \ldots, s, \tag{2.3.22}$$

$$a_{j1} = b_1, \quad j = 1, \ldots, s, \tag{2.3.23}$$

entonces $R(\infty) = 0$. Esto hace que un método A-estable sea L-estable.

Demostración: Según (2.3.18) se tiene que

$$R(\infty) = \lim_{z\to\infty} 1 + zb^T(I - zA)^{-1}\mathbf{1} = 1 - b^T(A)^{-1}\mathbf{1}. \tag{2.3.24}$$

Ahora, la expresión (2.3.22) es posible escribirla de la forma

$$A^T\boldsymbol{e}_s = b, \tag{2.3.25}$$

donde A^T es la transpuesta de la matriz $A = (a_{jk})_{j,k=1}^{s}$, $\boldsymbol{e}_s = (0,\ldots,0,1)^T$ el vector con 1 en la posición s y ceros en las demás componentes y $b = (b_1, b_2, \ldots, b_s)^T$.

Reemplazando (2.3.25) en (2.3.24) se tiene

$$R(\infty) = 1 - e_s^T A(A)^{-1}\mathbf{1},$$

de lo cual se llega a

$$R(\infty) = 1 - \boldsymbol{e}_s\mathbf{1} = 0. \tag{2.3.26}$$

Similarmente, la condición (2.3.23) se puede escribir de la forma

$$A\boldsymbol{e}_1 = \mathbf{1}b_1,$$

de lo cual se tiene que

$$A\boldsymbol{e}_1 b_1^{-1} = \mathbf{1}. \tag{2.3.27}$$

Luego sustituyendo (2.3.27) en (2.3.24) se llega a

$$R(\infty) = 1 - b^T(A)^{-1}(A\boldsymbol{e}_1 b_1^{-1}) = 0. \tag{2.3.28}$$

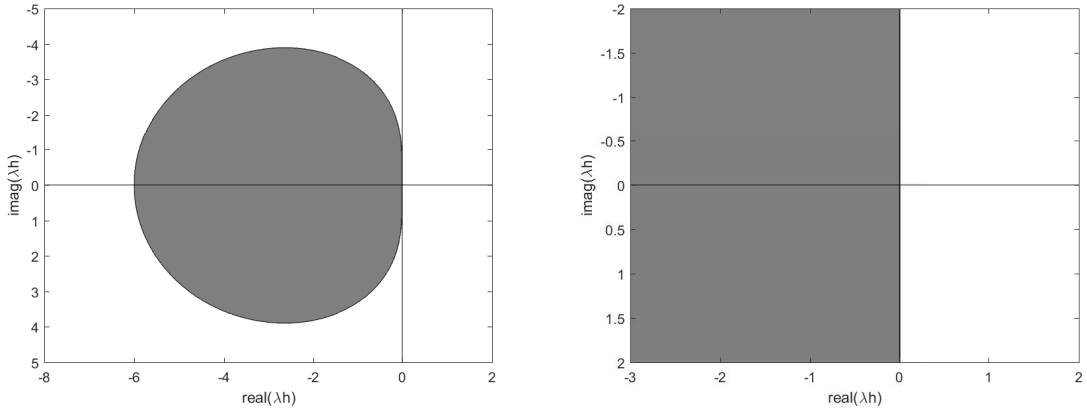
Por tanto, de (2.3.26) y (2.3.28) se concluye que las condiciones (2.3.22) y (2.3.23) conllevan a que $R(\infty) = 0$. Es decir, si una de ellas se cumple el método A-estable es L-estable. □

Algunos resultados de estabilidad absoluta para los métodos de RKI se resumen en la Tabla 2.3 y en la Figura 2.4. Se muestran las regiones de estabilidad para algunos métodos de RKI de orden dos, tres y cuatro.

Para obtener R(z) en el caso de los métodos RKI se debe solucionar la Ecuación 2.3.18 que cuenta con operaciones como el cálculo de matrices inversas o la multiplicación de matriz por vector. Operaciones que pueden ser tediosas y complejas de trabajar manualmente dependiendo de las componentes y dimensiones matriciales. De estos resultados es válido mencionar que entre mayor es el orden del método es más tedioso determinar su factor de amplificación, dado que al aumentar el número de estados el tamaño de su matriz A también crece, además sus componentes a_{jk} no son tan sencillos de trabajar, por lo cual es necesario utilizar un software de cálculo simbólico como

Método	Factor de Amplificación	Intervalo	A-estable	L-estable
Euler implícito	$1/(1-z)$	$(-\infty, 0) \cup (2, \infty)$	x	x
RKI21 Legendre RKI22 Lobatto I RK22 - Falso	$(2+z)/(2-z)$	$(-\infty, 0)$	x	
RKI32 Radau I RKI32 Radau II	$-(z^2+4z+6)/(2z-6)$	$(-6, 0)$		
RKI42 Legendre	$(z^2+6z+12)/(z^2-6z+12)$	$(-\infty, 0)$	x	

Tabla 2.3: Estabilidad absoluta para métodos numéricos implícitos.



(a) RKI orden 3.

(b) RKI orden 2 y orden 4.

Figura 2.4: Regiones de estabilidad para métodos numéricos implícitos.

Sage Math para conseguir estos resultados. Este software permite realizar de forma más rápida el cálculo de las operaciones matriciales y vectoriales presentes en la expresión que determina R(z).

Según la Proposición 2.2 los métodos RKI mencionados en la Tabla 2.3 no cumplen la condición de la Definición 2.6 para ser L-estables, pero si son A-estables excepto RKI32.

A partir de las regiones de estabilidad se deduce que los intervalos de estabilidad para los métodos explícitos son más restringidos que los que presentan los métodos implícitos. Por tanto al solucionar numéricamente los problemas rígidos con métodos explícitos se necesita de un tamaño de paso muy pequeño para lograr que estos converjan.

Capítulo 3

Métodos de paso múltiple

Los métodos numéricos que se han considerado en los capítulos anteriores son MPU, donde para obtener la aproximación de la solución al problema de Cauchy

$$\begin{cases} \boldsymbol{y'}(t) = \boldsymbol{f}(t, \boldsymbol{y}(t)), \\ \boldsymbol{y}(t_0) = \boldsymbol{y}_0, \end{cases}$$

se utiliza el paso anterior $\boldsymbol{y}_i$ para encontrar la aproximación de $\boldsymbol{y}_{i+1}$. En el caso de los Métodos de Paso Múltiple (MPM), para solucionar este mismo problema se necesitan conocer valores previos a $\boldsymbol{y}_{i+1}$ en el proceso de la discretización, como $\boldsymbol{y}_i, \boldsymbol{y}_{i-1}, \boldsymbol{y}_{i-2}, \ldots, \boldsymbol{y}_{i-k}$. Estos métodos aprovechan la información obtenida en etapas anteriores para conseguir el valor de $\boldsymbol{y}_{i+1}$. Los MPU se consideran como un caso particular de los MPM.

La idea de extender y mejorar el método de Euler dio paso al estudio de los MPM propuesta por Bashforth y Adams en (1883), conocidos como métodos de Adams-Bashforth. Otros tipos de métodos de multipasos lineales fueron introducidos por Nystrom en (1925) y Milne en (1926, 1953). Además, la teoría moderna de los MPM lineales fue desarrollada en grandes pasos por Dahlquist hasta (1956) y conocida a través de exposiciones presentadas por Henrici en (1962, 1963). Se recomienda seguir [8] y [23].

La forma general de los MPM lineales de N pasos es

$$\sum_{j=0}^{N} \alpha_j \boldsymbol{y}_{i+j} = h \sum_{j=0}^{N} \beta_j \boldsymbol{f}_{i+j}, \tag{3.0.1}$$

donde α_j y β_j son reales, $\alpha_N \neq 0$, $\boldsymbol{f} : [a, b] \times \mathbb{R}^n \longrightarrow \mathbb{R}^n$, $h = (b-a)/n$, $t_{i+j} = t_i + jh$ y $\boldsymbol{f}_{i+j} = \boldsymbol{f}\left(t_{i+j}, \boldsymbol{y}_{i+j}\right)$ para $0 \leq j \leq N$.

El método definido por (3.0.1) es implícito cuando $\beta_N \neq 0$ y explícito cuando $\beta_N = 0$, también se asume que $\alpha_N = 1$ ya que si no es así se podría despejar. Es importante mencionar que para aplicar uno de los MPM lineales los valores iniciales respectivos deben ser conocidos.

Existen diferentes formas para realizar la deducción de los MPM, una de ellas es a través del método de la integración numérica la cual se presenta en este trabajo. Para estudiar detalladamente otros métodos se recomienda seguir [8] y [23].

3.1. Métodos de Adams

En esta sección se analizan dos tipos de métodos de paso múltiple lineales: *explícitos e implícitos*. Los métodos de paso múltiple explícitos se denominan métodos de *Adams-Bashforth* (AB) y los métodos de paso múltiple implícitos métodos de *Adams-Moulton* (AM). A continuación se dan a conocer algunos de estos, tal y como se mira en [3, 8] y [23].

La deducción de MPM a través de la integración numérica se realiza integrando el problema de Cauchy entre t_i y t_{i+j} con $j \geq 1$. Clasificando varios MPM dependiendo del valor de j. En los métodos deducidos por integración numérica se destacan los métodos de Adams, para los cuales se toma $j = 1$, obteniendo la siguiente expresión

$$\boldsymbol{y}_{i+1} - \boldsymbol{y}_i = \int_{t_i}^{t_{i+1}} \boldsymbol{P}(t)\, dt, \tag{3.1.1}$$

donde $\boldsymbol{P}(t)$ es un polinomio interpolante a $\boldsymbol{f}(t, \boldsymbol{y}(t))$. Se usa un polinomio interpolante de grado $N-1$ para el caso de un método explícito y N para un método implícito, siendo N el número de pasos del método.

Observación 3.1. En el caso de los métodos de AB el orden de convergencia coincide con el número de pasos que se utilice, en cambio en los métodos de AM el número de pasos es menor en una unidad a su orden de convergencia. Esto se debe a que en los métodos implícitos siempre se tiene un coeficiente β_j adicional, ver [8] y [24].

Para la deducción del método explícito de Adams-Bashforth de dos pasos es necesario aproximar $\boldsymbol{f}(t, \boldsymbol{y}(t))$ por el polinomio de Lagrange de grado 1 que pasa por los puntos $(t_{i-1}, \boldsymbol{y}_{i-1})$ y $(t_i, \boldsymbol{y}_i)$, es decir el polinomio

$$\boldsymbol{P}_1(t) = \frac{(t - t_i)}{(t_{i-1} - t_i)} \boldsymbol{f}_{i-1} + \frac{(t - t_{i-1})}{(t_i - t_{i-1})} \boldsymbol{f}_i,$$

tomando

$$u = \frac{t - t_{i-1}}{h} \quad \text{y} \quad dt = h du,$$

se tiene que $t - t_{i-1} = hu$ y $t - t_i = t - t_{i-1} - h = h(u-1)$. Sustituyendo $\boldsymbol{P}_1(t)$ y los valores anteriores en (3.1.1) se obtiene

$$\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \int_{t_i}^{t_{i+1}} \boldsymbol{P}_1(t)dt,$$

donde

$$\begin{aligned}\boldsymbol{y}_{i+1} &= \boldsymbol{y}_i + \int_1^2 \left[u\boldsymbol{f}_i + (1-u)\boldsymbol{f}_{i-1}\right] h du \\ &= \boldsymbol{y}_i + h\left[\left(\frac{u^2}{2}\right)\boldsymbol{f}_i + \left(u - \frac{u^2}{2}\right)\boldsymbol{f}_{i-1}\right]_1^2 \\ &= \boldsymbol{y}_i + h\left(\frac{3}{2}\boldsymbol{f}_i - \frac{1}{2}\boldsymbol{f}_{i-1}\right).\end{aligned}$$

Concluyendo el **método de Adams-Bashforth de dos pasos (AB2)**

$$\begin{cases}\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \frac{h}{2}\left[3\boldsymbol{f}_i - \boldsymbol{f}_{i-1}\right], \text{con} \\ \boldsymbol{y}_0 = \boldsymbol{y}(t_0) \quad \text{y} \quad \boldsymbol{y}_1 \quad \text{conocido,}\end{cases} \tag{3.1.2}$$

donde el valor de $\boldsymbol{y}_1$ se obtiene con un MPU explícito del mismo orden al método AB2. En general los valores conocidos en los métodos de Adams son obtenidos con MPU explícitos de órdenes equivalentes al MPM.

Se mencionan otros métodos de Adams-Bashforth de orden superior, métodos con tres y cuatro pasos. Estos métodos se obtienen con la interpolación de polinomios de grados 2 y 3 y sus deducciones son equivalentes.

Método de Adams-Bashforth de tres pasos (AB3)

$$\begin{cases}\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \frac{h}{12}\left[23\boldsymbol{f}_i - 16\boldsymbol{f}_{i-1} + 5\boldsymbol{f}_{i-2}\right], \text{con} \\ \boldsymbol{y}_0 = \boldsymbol{y}(t_0) \quad \text{y} \quad \boldsymbol{y}_p \quad \text{conocido para} \quad 1 \leq p \leq 2.\end{cases} \tag{3.1.3}$$

Método de Adams-Bashforth de cuatro pasos (AB4)

$$\begin{cases}\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \frac{h}{24}\left[55\boldsymbol{f}_i - 59\boldsymbol{f}_{i-1} + 37\boldsymbol{f}_{i-2} - 9\boldsymbol{f}_{i-3}\right], \text{con} \\ \boldsymbol{y}_0 = \boldsymbol{y}(t_0) \quad \text{y} \quad \boldsymbol{y}_p \quad \text{conocido para} \quad 1 \leq p \leq 3.\end{cases} \tag{3.1.4}$$

En el caso de los MPM implícitos como los son los métodos de Adams-Moulton, la deducción de estos se realiza similarmente por medio del método de integración. Para el método Adams-Moulton de dos pasos se aproxima la función $\boldsymbol{f}(t, \boldsymbol{y}(t))$ por el polinomio de Lagrange de grado 2, que pasa por los puntos $(t_{i-1}, \boldsymbol{f}_{i-1}), (t_i, \boldsymbol{f}_i)$ y $(t_{i+1}, \boldsymbol{f}_{i+1})$, es decir el polinomio de la forma

$$\boldsymbol{P}_2(t) = \frac{(t-t_i)(t-t_{i+1})}{(t_{i-1}-t_i)(t_{i-1}-t_{i+1})}\boldsymbol{f}_{i-1} + \frac{(t-t_{i-1})(t-t_{i+1})}{(t_i-t_{i-1})(t_i-t_{i+1})}\boldsymbol{f}_i + \frac{(t-t_i)(t-t_i)}{(t_{i+1}-t_{i-1})(t_{i+1}-t_i)}\boldsymbol{f}_{i+1}.$$

Tomando

$$u = \frac{t - t_{i-1}}{h} \quad \text{y} \quad dt = hdu,$$

se sigue que $t - t_{i-1} = hu$, $t - t_i = t - t_{i-1} - h = h(u-1)$ y $t - t_{i+1} = t - (t_i + h) = t - t_i - h = h(u-1) - h = h(u-2)$, sustituyendo en $\boldsymbol{P}_2(t)$ se tiene

$$\begin{aligned}\boldsymbol{P}_2(t) &= \frac{h(u-1)h(u-2)}{2h^2}\boldsymbol{f}_{i-1} + \frac{(hu)(h(u-2))}{h(-h)}\boldsymbol{f}_i + \frac{(hu)(h(u-1))}{(2hh)}\boldsymbol{f}_{i+1} \\ &= \frac{(u-1)(u-2)}{-2}\boldsymbol{f}_{i-1} - u(u-2)\boldsymbol{f}_i + \frac{u(u-1)}{2}\boldsymbol{f}_{i+1}.\end{aligned}$$

Reemplazando la anterior expresión de $\boldsymbol{P}_2(t)$ en (3.1.1), se obtiene

$$\begin{aligned}\boldsymbol{y}_{i+1} &= \boldsymbol{y}_i + \int_1^2 \left[\frac{u(u-1)}{2}\boldsymbol{f}_{i+1} - u(u-2)\boldsymbol{f}_i + \frac{(u-1)(u-2)}{2}\boldsymbol{f}_{i-1}\right] hdu \\ &= \boldsymbol{y}_i + h\left[\left(\frac{u^3}{6} - \frac{u^2}{4}\right)\boldsymbol{f}_{i+1} - \left(\frac{u^3}{3} - u^2\right)\boldsymbol{f}_i + \left(\frac{u^3}{6} - \frac{3u^2}{4} + u\right)\boldsymbol{f}_{i-1}\right]_1^2 \\ &= \boldsymbol{y}_i + h\left[\frac{5}{12}\boldsymbol{f}_{i+1} + \frac{2}{3}\boldsymbol{f}_i - \frac{1}{12}\boldsymbol{f}_{i-1}\right] \\ &= \boldsymbol{y}_i + \frac{h}{12}\left[5\boldsymbol{f}_{i+1} + 8\boldsymbol{f}_i - \boldsymbol{f}_{i-1}\right].\end{aligned}$$

Concluyendo el **método de Adams-Moulton de dos pasos (AM2)**

$$\begin{cases}\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \frac{h}{12}\left[5\boldsymbol{f}_{i+1} + 8\boldsymbol{f}_i - \boldsymbol{f}_{i-1}\right], \text{con} \\ \boldsymbol{y}_0 = \boldsymbol{y}(t_0) \quad \text{y} \quad \boldsymbol{y}_1 \quad \text{conocido.}\end{cases} \tag{3.1.5}$$

Se mencionan otros métodos de Adams-Moulton con orden superior de tres y cuatro pasos. Estos métodos se obtienen con la interpolación de polinomios de grados 3 y 4 y sus deducciones se realizan de forma similar.

Método de Adams-Moulton de tres pasos (AM3)

$$\begin{cases}\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \frac{h}{24}\left[9\boldsymbol{f}_{i+1} + 19\boldsymbol{f}_i - 5\boldsymbol{f}_{i-1} + \boldsymbol{f}_{i-2}\right], \text{con} \\ \boldsymbol{y}_0 = \boldsymbol{y}(t_0) \quad \text{y} \quad \boldsymbol{y}_p \quad \text{conocido para} \quad 1 \leq p \leq 2.\end{cases} \tag{3.1.6}$$

Método de Adams-Moulton de cuatro pasos (AM4)

$$\begin{cases}\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \frac{h}{720}\left[251\boldsymbol{f}_{i+1} + 646\boldsymbol{f}_i - 264\boldsymbol{f}_{i-1} + 106\boldsymbol{f}_{i-2} - 19\boldsymbol{f}_{i-3}\right], \text{con} \\ \boldsymbol{y}_0 = \boldsymbol{y}(t_0) \quad \text{y} \quad \boldsymbol{y}_p \quad \text{conocido para} \quad 1 \leq p \leq 3.\end{cases} \tag{3.1.7}$$

A continuación, se estudian propiedades y características de los MPM como consistencia, estabilidad y convergencia.

3.2. Consistencia, convergencia y estabilidad

En esta sección se presentan las definiciones de consistencia, convergencia y estabilidad para los MPM, adicionando algunas propiedades que se cumplen para estos métodos. Esta teoría ha sido tomada de [8, 16, 23] y [24].

Definición 3.1. Dado un MPM lineal de la forma (3.0.1), el *error local de discretización* es definido por

$$\boldsymbol{\tau}_i = \frac{\sum_{j=0}^{N} \alpha_j \boldsymbol{y}(t_i + jh)}{h} - \sum_{j=0}^{N} \beta_j \boldsymbol{f}\left(t_i + jh, \boldsymbol{y}(t_i + jh)\right). \tag{3.2.1}$$

La ecuación (3.2.1) se puede escribir de la forma

$$h\boldsymbol{\tau}_i = \sum_{j=0}^{N} \alpha_j \boldsymbol{y}(t_i + jh) - h\sum_{j=0}^{N} \beta_j \boldsymbol{f}\left(t_i + jh, \boldsymbol{y}(t_i + jh)\right), \tag{3.2.2}$$

donde su expansión en series de Taylor sobre el centro $t = t_i$ es

$$h\boldsymbol{\tau}_i = \sum_{j=0}^{N} \alpha_j \left[\boldsymbol{y}(t_i) + jh\boldsymbol{y'}(t_i) + \frac{j^2h^2}{2!}\boldsymbol{y''}(t_i) + \cdots\right] - \sum_{j=0}^{N} \beta_j \left[h\boldsymbol{y'}(t_i) + jh^2\boldsymbol{y''}(t_i) + \frac{j^2h^3}{2!}\boldsymbol{y^{(3)}}(t_i) + \cdots\right],$$

de lo cual se llega a

$$h\boldsymbol{\tau}_i = C_0\boldsymbol{y}(t_i) + C_1h\boldsymbol{y'}(t_i) + C_2h^2\boldsymbol{y''}(t_i) + \cdots + C_ph^p\boldsymbol{y^{(p)}}(t_i) + \cdots, \tag{3.2.3}$$

con coeficientes C_i

$$\begin{aligned}
C_0 &= \sum_{j=0}^{N} \alpha_j, \\
C_1 &= \sum_{j=0}^{N} j\alpha_j - \sum_{j=0}^{N} \beta_j \\
C_2 &= \sum_{j=0}^{N} \frac{j^2\alpha_j}{2!} - \sum_{j=0}^{N} j\beta_j \\
&\vdots \\
C_p &= \sum_{j=0}^{N} \frac{j^p\alpha_j}{p!} - \sum_{j=0}^{N} \frac{j^{(p-1)}\beta_j}{(p-1)!}.
\end{aligned}$$

Definición 3.2. Un MPM es *consistente* si y solo si

$$\lim_{h\to 0} \|\boldsymbol{\tau}_i\| = 0,$$

es decir si y solo si las constantes $C_0 = 0$ y $C_1 = 0$.

Definición 3.3. Un MPM lineal tiene *orden de consistencia* p ($O(h^p)$) si y solo si existen constantes positivas C, h_0 y p, independiente del paso de integración h y del subíndice i, con $0 < h < h_0$, $1 \leq i \leq n$ e $i \in \mathbb{N}$, tales que el error local de discretización satisface

$$\max_{1\leq i\leq n} \|\boldsymbol{\tau}_i\| \leq Ch^p,$$

es decir si y solo si

$$C_0 = C_1 = \cdots = C_p = 0 \quad \text{y} \quad C_{p+1} \neq 0.$$

De esta forma, para asegurar la consistencia de un MPM lineal se debe verificar que este tenga un orden $p \geq 1$, en particular para $p = 1$ se verifica que $C_0 = 0 = C_1$.

Para realizar el estudio adecuado de la convergencia de los MPM lineales, es necesario el análisis de la teoría de las ecuaciones de diferencias lineales, por tanto, para este análisis se recomienda mirar [23] y [24] que es donde se muestra detalladamente lo requerido.

Definición 3.4. Un MPM lineal (3.0.1) de N pasos es *convergente*, si para todo problema de Cauchy (1.1.1) se tiene que

$$\lim_{h\to 0} \boldsymbol{y}_i = \boldsymbol{y}(t_i),$$

para todo $t_i \in [a, b]$ y para toda solución $\boldsymbol{y}_{i+N}$ de la ecuación de diferencias (3.0.1) que satisfacen las condiciones iniciales $\boldsymbol{y}_i = \boldsymbol{\eta}_i(h)$, donde $\lim_{h\to 0} \boldsymbol{\eta}_i(h) = \boldsymbol{y}_0$ para $i = 0, 1, \ldots N - 1$.

Observación 3.2. La consistencia es una condición necesaria para la convergencia, es decir, un MPM lineal consistente puede o no ser convergente. Sin embargo, si el MPM lineal no es consistente, entonces no es convergente.

A seguir, se describen algunas definiciones que son necesarias para el análisis de la convergencia, consistencia y estabilidad de los MPM lineales.

Definición 3.5. Dado un MPM lineal (3.0.1), el *primer y segundo polinomios característicos* asociados al método están definidos respectivamente por

$$\rho(r) = \sum_{j=0}^{N} \alpha_j r^j \quad \text{y} \quad \sigma(r) = \sum_{j=0}^{N} \beta_j r^j,$$

con $\alpha_N = 1$.

En relación a los polinomios característicos mencionados en la Definición 3.5, se enuncian algunos resultados con relación a la consistencia y estabilidad de los MPM lineales.

Teorema 3.1. *Un MPM lineal* (3.0.1) *es consistente si y solo si*

$$\rho(1) = 0 \quad y \quad \rho'(1) - \sigma(1) = 0.$$

La demostración de este teorema consiste en probar que C_0 y C_1 sean iguales a cero. Por un lado al calcular $\rho(1)$ este resulta ser igual a la expresión correspondiente de C_0, por lo que $C_0 = 0$. Por otro lado al calcular $\rho'(1)$ y $\sigma(1)$ y al reemplazar en la igualdad $\rho'(1) - \sigma(1) = 0$, esta diferencia es equivalente a C_1, por lo que se tiene que $C_1 = 0$ y se concluye que un MPM lineal es consistente. Esta demostración se puede ver con detalle en [23] y [24].

Observación 3.3. En un MPM lineal, el primer polinomio característico $\rho(r)$ siempre tiene una raíz igual a 1. Esta raíz es denominada *raíz principal* y generalmente se denota por r_1. Así, la consistencia de un MPM lineal depende sólo de la raíz principal $r_1 = 1$.

Definición 3.6. En el caso de raíces reales y distintas de $\rho(r)$, un MPM lineal (3.0.1) es *convergente* si

$$|r_j| \leq 1, \quad j = 1, \ldots, N,$$

siendo r_j las raíces del primer polinomio característico. En el caso de las raíces con multiplicidad mayor que 1, un MPM lineal (3.0.1) es *convergente* si esas raíces tienen módulo menor que 1.

Definición 3.7. Se dice que un MPM lineal (3.0.1) es *estable* si ninguna raíz del primer polinomio característico

$$\rho(r) = \sum_{j=0}^{N} \alpha_j r^j,$$

tiene módulo mayor que 1 y toda raíz con módulo 1 tiene multiplicidad 1.

El resultado que se muestra a seguir relaciona los conceptos de consistencia, estabilidad y convergencia. Se presenta y se demuestra en [8].

Teorema 3.2. *Las condiciones necesarias y suficientes para que un MPM lineal* (3.0.1) *sea convergente son que este sea consistente y estable.*

De esta forma es posible comparar algunas de las propiedades de los MPU y MPM, ya que para que un MPU sea convergente sólo basta con ser consistente, en cambio para que un MPM lo sea se debe garantizar su consistencia y estabilidad.

3.3. Estabilidad absoluta

La estabilidad absoluta para los MPM lineales se estudia de forma similar a la estabilidad de los MPU analizando la expresión

$$y'(t) = \lambda y(t).$$

La estabilidad consiste en escoger un tamaño de paso h adecuado de manera que la solución numérica se comporte de forma similar a la solución teórica. Se sugiere ver con más detalle en [8, 16, 23] y [24].

Para observar el comportamiento del error se analiza la expresión

$$\sum_{j=0}^{N} (\alpha_j - z\beta_j)\, e_{i+j} = \psi, \tag{3.3.1}$$

donde $z = \lambda h$, e_{i+j} el error global de discretización en el instante t_{i+j} y $\psi = h\boldsymbol{\tau}_i$ supuesto como una constante. Según la teoría estudiada en [23] y [24], la expresión (3.3.1) hace referencia a una ecuación de diferencias cuya solución general es una sucesión $\{e_n\}$, donde cada término es de la forma

$$e_i = \sum_{k=0}^{N} A\boldsymbol{r}_k^i - \frac{\gamma}{z\sum_{j=0}^{N}\beta_j}, \tag{3.3.2}$$

donde el primer término es la solución de la ecuación de diferencias lineal homogénea, representando la solución general para el caso de las raíces r_k del polinomio $\sum_{j=0}^{N}(\alpha_j - z\beta_j)\, r^j$ con una cierta multiplicidad. El segundo término es su solución particular, con γ constante. Según (3.3.2) el error es dependiente del primer sumando de la expresión, dado que si alguna raíz r_k tiene módulo mayor que 1, entonces el error crece cuando N crece.

A seguir se mencionan algunas definiciones útiles para el estudio de la estabilidad absoluta de estos métodos.

Definición 3.8. Se denomina *polinomio de estabilidad absoluta* de un MPM lineal al polinomio

$$\phi(r, z) = \rho(r) - (z)\sigma(r) = \sum_{j=0}^{N} (\alpha_j - z\beta_j) r^j, \tag{3.3.3}$$

donde $z = \lambda h$ y $\sigma(r)$ con $\rho(r)$ el primer y segundo polinomios característicos.

Definición 3.9. Se dice que un MPM lineal es *absolutamente estable* para un valor de $z = \lambda h$ dado, si para este z todas las raíces r_i del polinomio de estabilidad absoluta (3.3.3) satisfacen que

$$|r_i| < 1,$$

para $i = 1, \ldots, N$.

Definición 3.10. El conjunto

$$S = \{z \in \mathbb{C} / \, |r_i| < 1 \ \text{ para toda raíz } \ r_i \ \text{ de } \ \phi(r, z)\},$$

se conoce como *región de estabilidad absoluta* y la intersección de esta región con el eje real se conoce como *intervalo de estabilidad absoluta.*

Para encontrar las regiones de estabilidad de estos métodos se soluciona el polinomio de estabilidad (3.3.3), trabajando con la expresión

$$z = \frac{\rho(r)}{\sigma(r)} \tag{3.3.4}$$

obtenida de (3.3.3). Para que los métodos cumplan con la condición de absolutamente estables, todas las raíces r_i del polinomio deben tener módulos menores que uno, es decir se consideran los puntos en el círculo unitario y para esto se reemplaza r en (3.3.4) por $e^{i\theta}$ con $\theta \in [0, 2\pi]$, obteniendo los puntos de la región de estabilidad de cada método.

Los polinomios de estabilidad y las expresiones de z para los métodos AB2 y AM2 que se trabajan para obtener sus regiones de estabilidad son:

- **AB2:**
$$\phi(r, z) = r^2 - \left(1 + \frac{3}{2}z\right) r + \frac{1}{2}z \quad \text{y} \quad z = \frac{2e^{i\theta}(e^{i\theta} - 1)}{3e^{i\theta} - 1}.$$
- **AM2:**
$$\phi(r, z) = \left(1 - \frac{5}{12}z\right) r^2 - \frac{5}{3}r + \frac{1}{12}z \quad \text{y} \quad z = \frac{12e^{i\theta}(e^{i\theta} - 1)}{5e^{2i\theta} + 8e^{i\theta} - 1}.$$

En las figuras 3.1 y 3.2 y en las tablas 3.1 y 3.2, se muestran las regiones e intervalos de estabilidad para algunos MPM lineales Adams-Bashforth y Adams-Moulton. Se observa que las regiones de estabilidad para los métodos implícitos son de mayor amplitud que la de los métodos explícitos, pero ninguno de estos cuenta con una región de estabilidad con todo el semiplano izquierdo complejo, es decir su región es mucha más restringida. Se tiene que un MPM lineal es *A-estable* si $\mathbb{C}^- \subset S$, por tanto estos métodos no se consideran A-estables. Se concluye que los métodos de Adams no son del todo apropiados para solucionar numéricamente problemas rígidos, ya que según [8] la propiedad de ser A-estables está cerca de ser esencial para abordar problemas de este tipo.

Observación 3.4. Para elaborar las figuras 3.1 y 3.2 se utilizó el software Matlab. Esta ilustraciones se realizan al considerar el conjunto de puntos z en el plano complejo que satisfacen la expresión (3.3.4), teniendo en cuenta que las raíces de los polinomios característicos deben tener un módulo menor que uno. De esta forma, se representa gráficamente el contorno de las regiones de estabilidad de los MPM.

Nº de pasos N	2	3	4
orden p	2	3	4
Intervalo de estabilidad absoluta	$(-1, 0)$	$(-6/11, 0)$	$(-3/10, 0)$

Tabla 3.1: Características de los métodos de Adams-Bashforth.

Nº de pasos N	2	3	4
orden p	3	4	5
Intervalo de estabilidad absoluta	$(-6, 0)$	$(-3, 0)$	$(-90/49, 0)$

Tabla 3.2: Características de los métodos de Adams-Moulton.

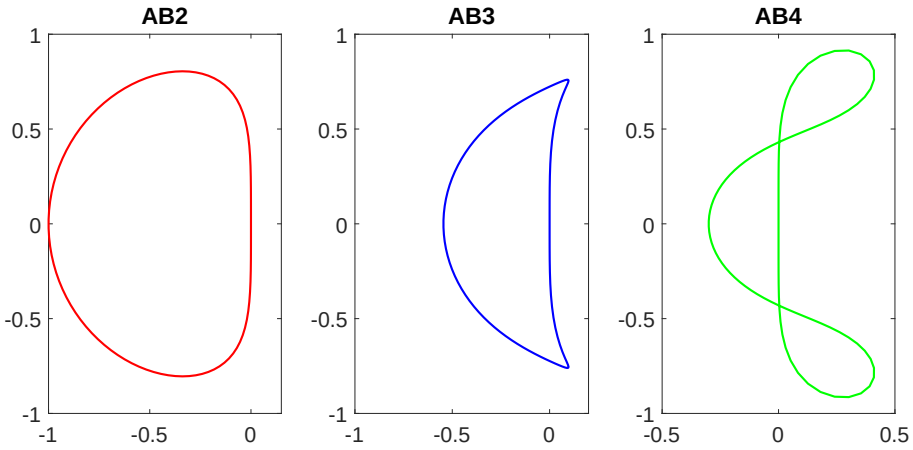


Figura 3.1: Regiones de estabilidad MPM lineales explícitos, Adams-Bashforth.

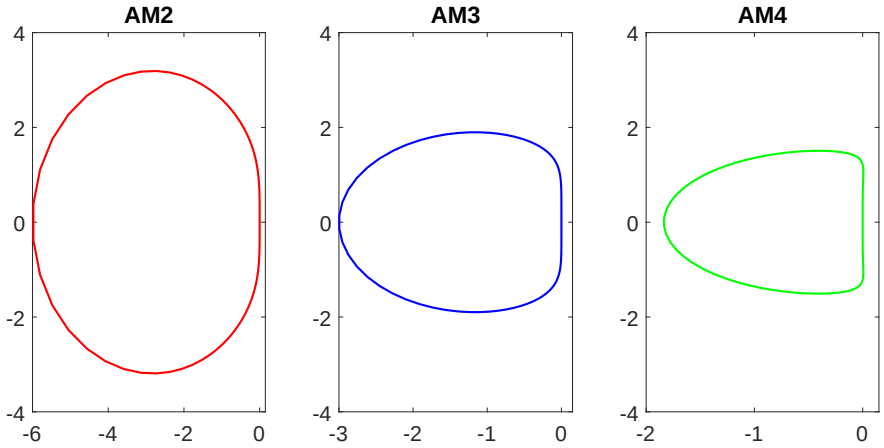


Figura 3.2: Regiones de estabilidad MPM lineales implícitos, Adams-Moulton.

3.4. Métodos Predictor-Corrector

Los métodos de Adams generalmente se implementan en forma de métodos *Predictor-Corrector*, estos se obtienen al combinar métodos de paso múltiple tanto explícitos como implícitos. En primer lugar se realiza un cálculo preliminar a la aproximación en el punto utilizando la fórmula de Bashforth y luego se encuentra una aproximación mejorada en el punto con la fórmula de Moulton, se sugiere ver [8, 12] y [18]. Según [3] y [11] los métodos de este tipo más utilizados son denominados *Adams-Bashfoth-Moulton* (ABM). Para la aplicación de ABM se usan conjuntamente dos métodos, un explícito que se llama *predictor* y un implícito que se llama *corrector*. El método explícito presenta una cantidad de pasos mayor en una unidad a la cantidad de pasos del método implícito, pero los dos métodos cuentan con el mismo orden de convergencia.

La forma general de un método de ABM es

$$\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + h\beta_0 \boldsymbol{f}(t_{i+1}, \boldsymbol{y}^*_{i+1}) + h\sum_{j=0}^{k-1} \beta_j \boldsymbol{f}(t_{i-j}, \boldsymbol{y}_{i-j}), \tag{3.4.1}$$

donde

$$\boldsymbol{y}^*_{i+1} = \boldsymbol{y}_i + h\sum_{j=0}^{k^*-1} \beta^*_j \boldsymbol{f}(t_{i-j}, \boldsymbol{y}_{i-j}). \tag{3.4.2}$$

La expresión (3.4.2) es el cálculo del método de AB que predice la aproximación de $\boldsymbol{y}(t_{i+1})$ con $\boldsymbol{y}^*_{i+1}$ y (3.4.1) corrige este valor al encontrar $\boldsymbol{y}_{i+1}$. Además, los valores de k^* y k son el número de pasos de los métodos AB y AM, con $k^* = k+1$ y $p = k+1$, donde p es el orden del método Predictor-Corrector ABM.

A continuación, se describe la forma en cómo se trabajan los métodos ABM para un orden determinado.

Método Predictor-Corrector ABM de tercera orden (ABM3)

Según la forma general presentada anteriormente para los métodos de ABM, para este método es necesario utilizar AB3 como un método predictor y AM2 como el método corrector. Para la implementación de este método se siguen los siguientes pasos:

1. Calcular los valores de partida $\boldsymbol{y}_0, \boldsymbol{y}_1$ y $\boldsymbol{y}_2$ con un método de paso único de tercera orden, generalmente RK33.

2. Calcular la predicción de $\boldsymbol{y}_{i+1}$ con AB3

$$\boldsymbol{y}^*_{i+1} = \boldsymbol{y}_i + \frac{h}{12}\left[23\boldsymbol{f}_i - 16\boldsymbol{f}_{i-1} + 5\boldsymbol{f}_{i-2}\right].$$

3. Calcular la corrección de $\boldsymbol{y}_{i+1}$ con AM2

$$\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \frac{h}{12}\left[5\boldsymbol{f}^*_{i+1} + 8\boldsymbol{f}_i - \boldsymbol{f}_{i-1}\right],$$

donde $\boldsymbol{f}^*_{i+1} = (t_{i+1}, \boldsymbol{y}^*_{i+1})$.

De esta forma se aproxima el valor de $\boldsymbol{y}(t_{i+1})$ para el problema de Cauchy con el método Predictor-Corrector.

A continuación, se muestra una observación sobre el estudio de la estabilidad absoluta para los métodos Predictor-Correcto. En este trabajo no se profundiza sobre este tema, pero puede ser estudiado con detalle en [24] y [30].

Observación 3.5. Para el estudio de la estabilidad absoluta de los métodos Predicto-Corrector, similarmente a los métodos de AB y AM consiste en solucionar un polinomio de estabilidad $\phi(r, z)$ que es escrito por medio del primer y segundo polinomios característicos, es decir

$$\phi(r, z) = \rho(r) - z\sigma(r) + z\beta_N[\rho^*(r) - z\sigma^*(r)], \tag{3.4.3}$$

donde

$$\rho^*(r) = \sum_{j=0}^{N} \alpha^*_j r^j, \quad \rho(r) = \sum_{j=0}^{N} \alpha_j r^j, \quad \sigma^*(r) = \sum_{j=0}^{N} \beta^*_j r^j \quad \text{y} \quad \sigma(r) = \sum_{j=0}^{N} \beta_j r^j,$$

siendo $\rho^*(r)$ y $\sigma^*(r)$ los polinomios de AB y $\rho(r)$ y $\sigma(r)$ los polinomios de AM. Cuando las raíces r_i del polinomio (3.4.3) satisfacen la condición

$$|r_i| < 1, \quad i = 1, 2, \ldots, N,$$

el método Predictor-Corrector será absolutamente estable y el intervalo de estabilidad absoluta del método será (α, β), con α, $\beta \in \mathbb{R}$ tal que $z = \lambda h \in (\alpha, \beta)$. Para profundizar se recomienda seguir [24] y [30].

Capítulo 4

Resultados numéricos

En este capítulo se dan a conocer diferentes resultados numéricos, con el fin de verificar la teoría sobre los métodos numéricos para solucionar SEDO rígidos presentada en los capítulos anteriores. Se realizan comparaciones de las aproximaciones obtenidas con los MPU y los MPM con las soluciones teóricas de diferentes problemas de EDO y SEDO. Además, se incluyen resultados numéricos para una aplicación sobre reacciones químicas.

4.1. Validación de implementaciones

Se han realizado las implementaciones en Lenguaje C de los métodos numéricos estudiados como los métodos de Euler, los métodos de Runge-Kutta, los métodos de Adams y los métodos Predictor-Corrector. A seguir se trabajan algunos SEDO rígidos, los cuales permiten hacer la validación de las implementaciones realizadas y verificar las propiedades teóricas de cada unos de los métodos.

Para cada uno de los problemas solucionado numéricamente se calcula el error global de discretización comparando las aproximaciones obtenidas con las soluciones teóricas, utilizando la norma dos para el caso escalar y vectorial. Además, para el cálculo del orden de convergencia de cada método se calcula la razón $\left(\boldsymbol{e}_{(i,h)}\right) / \left(\boldsymbol{e}_{(i,h/2)}\right)$ con h el tamaño de paso, donde al calcular log_2 de la razón coincide este valor con el orden del método. Es decir, de (1.1.6) se tiene que

$$\left\|\boldsymbol{e}_{(i,h)}\right\| \approx Ch^p \quad y \quad \left\|\boldsymbol{e}_{(i,h/2)}\right\| \approx C(h/2)^p$$

para errores globales con tamaños de paso h y $h/2$, luego calculando la razón entre estas expresiones se llega a

$$\frac{\left\|\boldsymbol{e}_{(i,h)}\right\|}{\left\|\boldsymbol{e}_{(i,h/2)}\right\|} \approx 2^p \tag{4.1.1}$$

y aplicando log_2 en (4.1.1) se obtiene que

$$log_2\left(\frac{\left\|\boldsymbol{e}_{(i,h)}\right\|}{\left\|\boldsymbol{e}_{(i,h/2)}\right\|}\right) \approx p,$$

expresión que permite calcular el orden de convergencia del método.

Como se ha mencionado en los capítulos anteriores los métodos implícitos necesitan de la solución de sistemas de ecuaciones no lineales como punto fijo, Newton o Newton modificado. En el caso del método de Newton se realiza el cálculo de la matriz inversa el cual es costoso computacionalmente, por tanto para matrices de tamaño 2×2 y 3×3 se realiza este cálculo de forma exacta. Para matrices de mayor dimensión se utiliza el método de Newton modificado. Métodos descritos en la Sección 2.2.

Para los criterios de parada en los métodos de solución de ecuaciones no lineales, se fijaron valores de una cantidad de iteraciones máxima de 100 y una tolerancia de 10^{-10} para compararla con la diferencia absoluta entre dos aproximaciones consecutivas. Estos datos se utilizan para todos los métodos. El criterio de parada viene dado por la comparación del error con respecto a la tolerancia y la cantidad de iteraciones con las iteraciones máximas. Se realizaron ensayos al disminuir el valor de la tolerancia y se observó que para tolerancias menores a 10^{-12}, los métodos no convergen o fracasan y para valores entre 10^{-11} y 10^{-12} se obtienen resultados muy similares a los que genera una tolerancia de 10^{-10}. Una diferencia que se mira al disminuir el valor de la tolerancia es que aumenta la cantidad de iteraciones en los métodos de solución de ecuaciones no lineales, caso que ocurre cuando se trabaja el cálculo de la matriz inversa de forma aproximada para matrices de dimensiones mayores a 3.

Los MPM necesitan de cálculos previos para obtener la aproximación requerida $\boldsymbol{y}_{i+1}$, en este caso para conseguir estos valores se utilizan MPU explícitos con el mismo orden de convergencia al MPM utilizado.

4.1.1. Convergencia

En este apartado se presentan y analizan resultados numéricos obtenidos con MPU y MPM lineales, como los métodos de Euler, Runge-Kutta, Adams-Bashforth, Adams-Moulton y Predictor-Corrector. Para validar sus implementaciones y verificar sus propiedades teóricas se han solucionado algunas EDO y SEDO.

Problema 4.1. Usar los MPU y los MPM tanto explícitos como implícitos y los métodos Predictor-Corrector para aproximar el problema de Cauchy

$$\begin{cases} y' = 2ty, \\ y(0) = 1, \quad \text{en} \quad t = [1, 1.5]. \end{cases}$$

Con solución exacta $y(t) = e^{t^2-1}$ y $y(1.5) = 3.4903429575$.

Inicialmente se soluciona de forma numérica el Problema 4.1, una EDO no rígida, con el fin de verificar el orden de convergencia de cada uno de los método trabajados.

Las tablas 4.1 y 4.2 corresponden a los errores y razones obtenidas numéricamente en el Problema 4.1, con MPU, MPM y métodos Predictor-Corrector. En la aplicación de los métodos implícitos se utilizó el método de Newton para la solución de los sistemas de ecuaciones no lineales. Se observa que a medida que se reduce el tamaño de h a la mitad el error también disminuye haciendo que los métodos converjan. En la última columna se mira que las razones de los errores para cada uno de los métodos están siendo aproximadas al valor correcto según su respectivo orden. Estos resultados muestran que los métodos de mayor orden presentan mejores aproximaciones para un tamaño de h más grande en comparación a los de menor orden. Esto ocurre en todos los métodos presentes. Sin embargo, se observa que algunos métodos implícitos cuentan con una mayor precisión que los explícitos pero con una diferencia no muy grande. Por tanto, para este problema no rígido no fue conveniente utilizar un método implícito ya que sólo está causando un mayor costo computacional.

A seguir se soluciona numéricamente una EDO rígida con los métodos anteriormente mencionados.

Problema 4.2. Aproximar con los MPU y los MPM tanto explícitos como implícitos y los métodos Predictor-Corrector el problema de Cauchy

$$\begin{cases} y' = -40y + 40t + 1, \\ y(0) = 4, \quad \text{en} \quad t = [0, 20]. \end{cases}$$

Con solución exacta $y(t) = t + 4e^{-40t}$ y $y(20) = 20$.

Según la solución teórica del Problema 4.2 se mira que una de las componentes de su solución varía mucho más rápido que la otra, por tanto se considera una EDO rígida.

El Problema 4.2 fue solucionado numéricamente con métodos explícitos y se observó que para un tamaño de paso $h \geq 0.1$ todos los métodos divergen. Para $h = 5 \times 10^{-2}$ Euler y RK22 presentan un error de 4 y RK33 y RK44 consiguen converger llegando a un error de orden 10^{-15}. Por tanto, para que Euler y RK22 logren la convergencia se usa un $h = 2.5 \times 10^{-2}$, para el cual los errores de los métodos alcanzan el cero computacional.

Las aproximaciones del proceso de discretización que se obtuvieron con los métodos explícitos se ilustran en las figuras 4.1 y 4.2. En la Figura 4.1a se mira que para un tamaño de paso $h = 5 \times 10^{-2}$ las aproximaciones en Euler se están comportando de forma oscilatoria y en RK22 están alejadas de la solución teórica. En la Figura 4.1b se ve que los métodos de orden mayor para el mismo

Método	n	Error	Razón
Euler Explícito	20	1.6935×10^{-1}	1.9316
	40	8.7673×10^{-2}	1.9646
	80	4.4627×10^{-2}	1.9820
	160	2.2517×10^{-2}	1.9909
	320	1.1310×10^{-2}	
Euler Implícito	20	1.9624×10^{-1}	2.0796
	40	9.4367×10^{-2}	2.0382
	80	4.6299×10^{-2}	2.0187
	160	2.2934×10^{-2}	2.0093
	320	1.1414×10^{-2}	
RK22	20	2.8254×10^{-3}	3.9115
	40	7.2233×10^{-4}	3.9558
	80	1.8260×10^{-4}	3.9780
	160	4.5903×10^{-5}	3.9891
	320	1.1507×10^{-5}	
RK33	20	4.1485×10^{-5}	7.8209
	40	5.3044×10^{-6}	7.9103
	80	6.7057×10^{-7}	7.9551
	160	8.4294×10^{-8}	7.9775
	320	1.0566×10^{-8}	
RK44	20	5.9984×10^{-7}	15.6156
	40	3.8413×10^{-8}	15.8076
	80	2.4300×10^{-9}	15.9038
	160	1.5280×10^{-10}	15.9526
	320	9.5781×10^{-12}	

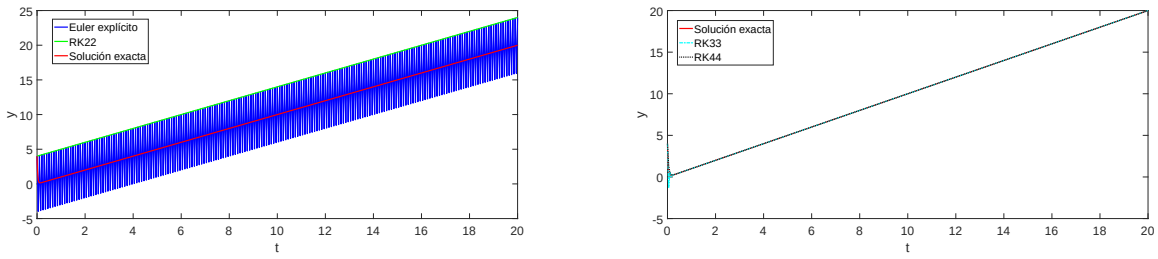
Método	n	Error	Razón
RKI21	20	1.4781×10^{-3}	4.0023
	40	3.6933×10^{-4}	4.0006
	80	9.2319×10^{-5}	4.0001
	160	2.3079×10^{-5}	4.0000
	320	5.7697×10^{-6}	
RKI22	20	2.8442×10^{-3}	4.0040
	40	7.1035×10^{-4}	4.0010
	80	1.7754×10^{-4}	4.0002
	160	4.4383×10^{-5}	3.9999
	320	1.1096×10^{-5}	
RKI32 Radau I	20	2.3650×10^{-5}	7.9318
	40	2.9817×10^{-6}	7.9655
	80	3.7432×10^{-7}	7.9829
	160	4.6891×10^{-8}	7.9924
	320	5.8669×10^{-9}	
RKI32 Radau II	20	4.8590×10^{-6}	7.9876
	40	6.0831×10^{-7}	7.9947
	80	7.6090×10^{-8}	7.9982
	160	9.5134×10^{-9}	8.0019
	320	1.1889×10^{-9}	
RKI42	20	5.7578×10^{-8}	15.9957
	40	3.5996×10^{-9}	15.9989
	80	2.2499×10^{-10}	16.0001
	160	1.4061×10^{-11}	16.0642
	320	8.7530×10^{-13}	

Tabla 4.1: Convergencia MPU explícitos e implícitos, Problema 4.1.

Método	n	Error	Razón	Método	h	Error	Razón
AB3	10	7.2459×10^{-3}	6.5445	AB4	10	1.2408×10^{-3}	11.7629
	20	1.1071×10^{-3}	7.2839		20	1.0548×10^{-4}	13.8869
	40	1.5201×10^{-4}	7.6431		40	7.5959×10^{-6}	14.9395
	80	1.9888×10^{-5}	7.8215		80	5.0844×10^{-7}	15.4682
	160	2.5427×10^{-6}			160	3.2870×10^{-8}	
AM2	10	1.0238×10^{-3}	7.4336	AM3	10	1.2293×10^{-4}	13.5888
	20	1.3772×10^{-4}	7.7170		20	9.0465×10^{-6}	14.8068
	40	1.7847×10^{-5}	7.8587		40	6.1097×10^{-7}	15.4055
	80	2.2709×10^{-6}	7.9295		80	3.9659×10^{-8}	15.7031
	160	2.8639×10^{-7}			160	2.5256×10^{-9}	
ABM3	10	5.0938×10^{-4}	5.0926	ABM4	10	4.6473×10^{-5}	7.7919
	20	1.0002×10^{-4}	6.5382		20	5.9643×10^{-6}	11.8763
	40	1.5298×10^{-5}	7.2665		40	5.0220×10^{-7}	13.9302
	80	2.1053×10^{-6}	7.6325		80	3.6051×10^{-8}	14.9626
	160	2.7584×10^{-7}			160	2.4094×10^{-9}	

Tabla 4.2: Convergencia MPM, Problema 4.1.

tamaño de h convergen presentando pequeñas oscilaciones de error en el inicio de las iteraciones. Al disminuir $h = 2.5 \times 10^{-2}$ se mira en la Figura 4.2 los errores que alcanza cada método para este tamaño de paso. Se tiene que los métodos de menor orden consiguen converger y en general todos los métodos continúan con las perturbaciones en la región inicial.

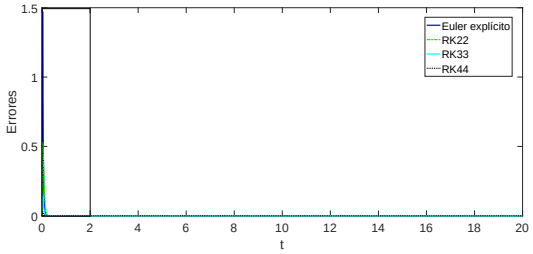


(a) Euler y RK22.

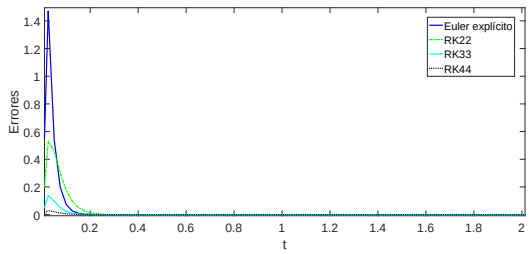
(b) RK33 y RK44.

Figura 4.1: Convergencia MPU explícitos con $h = 5 \times 10^{-2}$, Problema 4.2.

En la Tabla 4.3 se muestran resultados numéricos obtenidos con el método de Euler implícito para el Problema 4.2, donde las ecuaciones no lineales se han solucionado numéricamente mediante el método de punto fijo. Según la Sección 2.2 solucionar una ecuación no lineal en este problema hace referencia a encontrar un punto fijo de la función $y_{i+1} = y_i + hf(t_{i+1}, y_{i+1})$, donde la función de iteración es $y_i + hf(t_{i+1}, y_{i+1})$. Se observa que este método diverge para un tamaño de paso $h \geq 2.5 \times 10^{-2}$, para el cual Euler explícito converge, pero para un valor de $h \leq 1.2500 \times 10^{-2}$ el método converge presentando muy buenas aproximaciones. De esta forma al utilizar Euler implíci-



(a) Errores con $h = 2.5 \times 10^{-2}$.



(b) Zoom de c).

Figura 4.2: Errores numéricos MPU explícitos, Problema 4.2.

to con punto fijo, es necesario realizar muchas iteraciones para lograr que el método converja en comparación con los métodos explícitos. Es decir, este método es mucho más costoso computacionalmente, ya que debe realizar una mayor cantidad de cálculos para conseguir buenos resultados. Adicionalmente, en la tabla se muestra la cantidad de iteraciones i que realiza el método de punto fijo.

n	**h**	**Aproximación**	**i**	**Error**
800	2.5000×10^{-2}	Diverge	–	–
1600	1.2500×10^{-2}	20.000000000069	29	6.9850×10^{-11}
3200	6.2500×10^{-3}	20.000000000029	15	2.9104×10^{-11}
6400	3.1250×10^{-3}	19.999999999973	10	2.6183×10^{-11}

Tabla 4.3: Convergencia método Euler implícito - punto fijo, Problema 4.2.

Debido a los resultados obtenidos en la Tabla 4.3 se mira la necesidad de utilizar un método numérico de solución de ecuaciones no lineales con mayor orden de convergencia, para observar cuál es su comportamiento al solucionar numéricamente SEDO rígidos con métodos implícitos.

Se ha solucionado numéricamente el Problema 4.2 con los MPU implícitos en los cuales se ha utilizado el método de Newton para solucionar las ecuaciones no lineales. Las funciones a solucionar con Newton son las funciones que cuentan con el término implícito igualadas a cero, se sugiere ver la Sección 2.2.

En la Tabla 4.4 se muestran los tamaños de paso, las iteraciones en Newton y los errores globales que se generan en el Problema 4.2. Se mira que la mayoría de los métodos convergen para un tamaño de paso mucho mayor al utilizado en los métodos explícitos y en el método Euler implícito con punto fijo. Para estos métodos se necesita de muy pocas iteraciones para lograr obtener buenas aproximaciones a excepción de RKI22-Falso y RKI32. En la tabla no se mencionan los resultados para

el método RKI22 y no se hace distinción entre los Radau de RKI32, ya que los errores generados por estos son similares a los métodos que cuentan con el mismo orden, esto ocurre porque están realizando las mismas cuentas.

Método	**h**	**i**	**Error**
Euler Implícito	10	2	6.2189×10^{-6}
	5	2	6.1265×10^{-10}
	2.5	2	0.0000
RKI21	6.2500×10^{-1}	2	2.3643×10^{-2}
	3.1250×10^{-1}	2	4.2719×10^{-9}
	1.5625×10^{-1}	1	0.0000
RKI22-Falso	3.1250×10^{-1}	2	9.7656×10^{-1}
	1.5625×10^{-1}	1	2.4414×10^{-1}
	7.8125×10^{-2}	1	6.1035×10^{-2}
RKI32	7.8125×10^{-2}	1	0.0000
RKI42	1.25	42	8.5976×10^{-2}
	6.2500×10^{-1}	22	8.5466×10^{-7}
	3.1250×10^{-1}	1	2.3270×10^{-12}

Tabla 4.4: Convergencia MPU implícitos - Newton, Problema 4.2.

Para valores de h mayores a los que se muestran en la Tabla 4.4 se tiene que Euler Implícito alcanza un error de orden 10^{-3} y los demás métodos divergen. Se mira que en los métodos RKI32 es necesario disminuir h a un valor de 7.8125×10^{-2} para lograr que los métodos converjan, alcanzando el cero computacional, pero RKI22-Falso precisa continuar con este proceso para mejorar su precisión.

Según el análisis realizado con respecto a los resultados numéricos obtenidos para los problemas anteriores, se concluye que los MPU explícitos y los implícitos con punto fijo no son los más convenientes de utilizar para solucionar SEDO rígidos, ya que estos precisan de tamaños de paso muy reducidos para conseguir la convergencia, causando un gran costo computacional.

En la Tabla 4.5 se muestra la convergencia de los MPM lineales para el Problema 4.2, donde las ecuaciones no lineales de AM son solucionadas con Newton. Se mira que los métodos de AM convergen para un tamaño de paso h mayor al que se utiliza en los métodos AB y ABM, es decir los métodos Adams-Moulton presentan mejores aproximaciones que los demás métodos. Además, se observa que al comparar los métodos ABM y AB, los métodos Predictor-Corrector convergen mucho más rápido que los métodos explícitos. Es válido mencionar que para los ABM y AB no se necesita resolver un sistema de ecuaciones. En general todos los métodos en el momento que dejan de diverger consiguen una muy buena precisión, lo cual hace que los errores sean muy pequeños y no se logre obtener el orden de cada método, es decir la razón entre los errores no caen en el valor

esperado. Después de que los métodos consiguen converger no es conveniente disminuir mucho el tamaño de paso, ya que esto puede causar errores de máquina.

Método	**h**	**Error**	**Método**	**h**	**Error**
AB3	2.5000×10^{-2}	Diverge	AB4	1.2500×10^{-2}	Diverge
	1.2500×10^{-2}	0.0000		6.2500×10^{-3}	3.5527×10^{-15}
AM2	2.0000×10^{-1}	Diverge	AM3	1.0000×10^{-1}	Diverge
	1.0000×10^{-1}	0.0000		5.0000×10^{-2}	0.0000
ABM3	5.0000×10^{-2}	Diverge	ABM4	5.0000×10^{-2}	Diverge
	2.5000×10^{-2}	0.0000		2.5000×10^{-2}	0.0000

Tabla 4.5: Convergencia MPM, Problema 4.2.

Al comparar los resultados obtenidos con los MPU y MPM implícitos, se observó que los MPU consiguen converger con tamaños de pasos mayores a los que necesita los MPM. Esto se debe a que la región de estabilidad de los MPU es mucho mayor a la de los MPM. Es decir, la zona de estabilidad de los MPM es muy restringida, en cambio los MPU son A-estables o L-estables a excepción de RKI32.

Dado que los problemas trabajados anteriormente cuya solución está en el espacio de los números reales, es decir han sido EDO, ahora se trabaja un SEDO lineal rígido que cuenta con soluciones teóricas.

Problema 4.3. Solucionar numéricamente con los MPU y MPM implícitos y los métodos Predictor-Corrector el problema de Cauchy

$$\begin{cases} x' = -80.6x + 119.4y, \\ y' = 79.60x - 120.4y, \\ x(0) = 1 \quad \text{y} \quad y(0) = 4, \quad \text{en} \quad t = [0, 1]. \end{cases}$$

Con soluciones exactas $x(t) = 3e^{-t} - 2e^{-200t}$ y $y(t) = 2e^{-t} + 2e^{-200t}$, donde $x(1) = 1.1036383235$ y $y(1) = 0.7357588823$.

Se considera este SEDO lineal un problema rígido porque los valores propios de su matriz asociada tienen parte real negativa y además la relación entre ellos es muy grande, causando que las componentes de sus soluciones teóricas varíen unas más rápido que otras.

Al solucionar el Problema 4.3 con los MPU explícitos, se observó que estos divergen para tamaños de paso $h > 7.8125 \times 10^{-3}$. Para un $h = 7.8125 \times 10^{-3}$ los métodos convergen, Euler con un error de orden 10^{-3}, RK22 con orden 10^{-5}, RK33 con orden 10^{-8} y RK44 con orden 10^{-11}. Según estos

resultados todos los métodos consiguen aproximaciones para el mismo tamaño de h pero con una precisión diferente.

La Tabla 4.6 corresponde a las iteraciones en Newton, errores y razones calculadas en las aproximaciones del Problema 4.3 mediante los MPU implícitos. Se observa que al disminuir el tamaño de h a la mitad el error también disminuye, además las razones de los errores caen en el valor esperado con relación al orden de cada método. Para RKI42 se altera este valor porque cuenta con una precisión de magnitud 10^{-13}, aproximándose al cero computacional y causando errores de máquina. Para este problema los métodos de igual orden presentan los mismos resultados numéricos o valores similares.

Método	**h**	**i**	**Error**	**Razón**	**M No Lineal**
Euler Implícito	3.1250×10^{-2}	2	2.0459×10^{-2}	1.9872	Newton
	1.5625×10^{-2}	2	1.0296×10^{-2}	1.9935	
	7.8125×10^{-3}	2	5.1645×10^{-3}	1.9968	
	3.9063×10^{-3}	2	2.5864×10^{-3}		
RKI21	3.1250×10^{-2}	2	1.0796×10^{-4}	4.0003	Newton
	1.5625×10^{-2}	2	2.6987×10^{-5}	4.0001	
	7.8125×10^{-3}	2	6.7465×10^{-6}	4.0000	
	3.9063×10^{-3}	2	1.6866×10^{-6}		
RKI32	3.1250×10^{-2}	2	3.7877×10^{1}	–	Newton
	1.5625×10^{-2}	2	7.0570×10^{-8}	8.0168	
	7.8125×10^{-3}	2	8.8028×10^{-9}	8.0084	
	3.9063×10^{-3}	2	1.0992×10^{-9}		
RKI42	3.1250×10^{-2}	45	1.7592×10^{-9}	15.9856	Newton Modificado
	1.5625×10^{-2}	22	1.1005×10^{-10}	15.4617	
	7.8125×10^{-3}	14	7.1175×10^{-12}	28.5279	
	3.9063×10^{-3}	9	2.4949×10^{-13}		

Tabla 4.6: Convergencia MPU implícitos - Newton, Problema 4.3.

Para un $h > 3.1250 \times 10^{-2}$ se tiene que Euler implícito converge con errores de orden 10^{-1} y 10^{-2}, RKI de orden 2 convergen para $3.1250 \times 10^{-2} < h < 0.25$ con errores de orden 10^{-1} y 10^{-2}, para un h mayor divergen, RKI32 divergen y RKI42 fracasa en el método de Newton alcanzando las iteraciones máximas. Para RKI42 al modificar el criterio de parada que consiste en la verificación si las aproximaciones en Newton son las raíces de las funciones igualmente fracasa, ya que las aproximaciones de las raíces no cuentan con una buena precisión. Sin embargo, RKI42 para un $h \leq 3.1250 \times 10^{-2}$ presenta aproximaciones similares a las que se muestran en la tabla, lo único que varía es el aumento de iteraciones en Newton.

Al comparar los resultados de los MPU explícitos e implícitos para un $h = 7.8125 \times 10^{-3}$, se concluye

que los métodos logran alcanzar un error del mismo orden, pero los métodos implícitos permiten conseguir aproximaciones para tamaños de paso mayores, en cambio los métodos explícitos divergen. Es decir para que los métodos explícitos no diverjan necesitan disminuir mucho más el tamaño de paso.

Después de haber realizado un respectivo análisis de los resultados numéricos obtenidos con los MPU, se analizan las aproximaciones obtenidas con los MPM. Para el caso de los AB, se tiene que para el primer tamaño de paso que AB3 no diverge $h = 1.9531 \times 10^{-3}$ alcanza un error de orden 10^{-9} y AB4 diverge. Pero para el siguiente valor de $h = 9.7656 \times 10^{-4}$ el orden de error de AB3 es 10^{-10} y de AB4 10^{-13}.

En la Tabla 4.7 se presentan las aproximaciones para el Problemas 4.3 con los MPM. Se observa que los métodos con mejores aproximaciones son AM y ABM de cuarta orden, pero los AM consiguen obtener aproximaciones para un tamaño de paso mayor. Para valores de h mayores a los que se ilustran en la tabla todos los métodos divergen. Se muestra que las razones de los errores consiguen caer o son próximas a los valores correspondientes a su orden, en el caso de ABM4 estos valores se alteran ya que los errores son cercanos al cero computacional y pueden causar errores numéricos. Además, se menciona que la cantidad de iteraciones finales en Newton para los métodos AM tienen un comportamiento constante de $i = 2$.

El error como función del tamaño de paso para los MPU y MPM implícitos aplicados al Problema 4.3 se muestra en la Figura 4.3. La escala es logarítmica en sus dos ejes con un tamaño de paso inicial $h = 3.1250 \times 10^{-2}$ hasta $h = 6.1035 \times 10^{-5}$. Dado que el error es proporcional al tamaño de paso, se relaciona la pendiente de las rectas como el orden de la convergencia, entre mayor sea la inclinación mayor es el orden del método. Se observa que RKI32, AM2 y AM3 muestran un cambio brusco al inicio de las iteraciones, ya que para los tamaños de paso iniciales aún no rompen la zona de estabilidad. Para RKI42 y AM3 se mira que son los métodos que para un determinado h consiguen una muy buena precisión, sin embargo no es recomendable usar un tamaño de paso muy pequeño, pues la precisión finita del computador hace que se generen errores de máquina que afectan el error global de discretización. Como para los MPU de igual orden se obtuvieron los mismos resultados, se tiene que las rectas también coinciden para este problema.

En la Figura 4.4 se muestra la cantidad de iteraciones que realiza Newton modificado en cada paso del método RKI42 con $h = 3.1250 \times 10^{-2}$ en el Problema 4.3. Los cuadros de color azul hacen referencia a las iteraciones con Newton modificado con el primer criterio de parada que es la comparación de normas e iteraciones. Las equis de color verde muestran las iteraciones con el segundo criterio de parada que es la verificación de las raíces de sus funciones. Para los métodos implícitos diferentes a RKI42, Newton genera una cantidad de iteraciones constante igual a 2, en cambio la

Método	h	Error	Razón
AM2	3.1250×10^{-2}	6.4603×10^{-1}	–
	1.5625×10^{-2}	2.0873×10^{-7}	7.9599
	7.8125×10^{-3}	2.6223×10^{-8}	7.9801
	3.9062×10^{-3}	3.2860×10^{-9}	7.9901
	1.9531×10^{-3}	4.1126×10^{-10}	
AM3	1.5625×10^{-2}	3.4197×10^{-1}	–
	7.8125×10^{-3}	1.0746×10^{-10}	15.9245
	3.9062×10^{-3}	6.7482×10^{-12}	15.9456
	1.9531×10^{-3}	4.2322×10^{-13}	15.6713
	9.7656×10^{-4}	2.6867×10^{-14}	
ABM3	7.8125×10^{-3}	1.7394×10^{-5}	–
	3.9063×10^{-3}	3.3266×10^{-9}	8.0391
	1.9531×10^{-3}	4.1380×10^{-10}	8.0194
	9.7656×10^{-4}	5.1600×10^{-11}	8.0111
	4.8828×10^{-4}	6.4411×10^{-12}	
ABM4	3.9063×10^{-3}	8.2472×10^{-12}	16.0381
	1.9531×10^{-3}	5.1423×10^{-13}	15.8743
	9.7656×10^{-4}	3.2394×10^{-14}	12.5098
	4.8828×10^{-4}	2.5895×10^{-15}	

Tabla 4.7: Convergencia MPM, Problema 4.3.

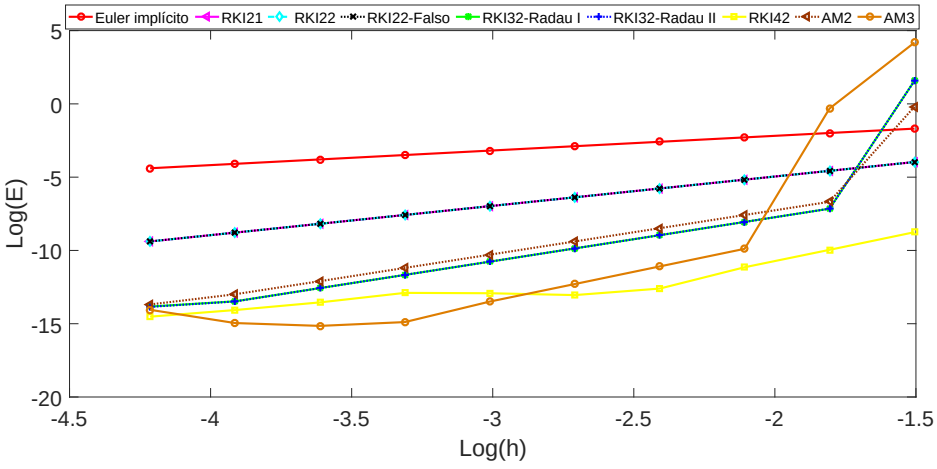


Figura 4.3: Convergencia MPU implícitos, Problema 4.3.

modificación de Newton en RKI42 genera una cantidad de iteraciones mucho mayor con los dos criterios de parada. Sin embargo, cuando se disminuye el tamaño de paso las iteraciones para este método también disminuyen como se mira en Tabla 4.6.

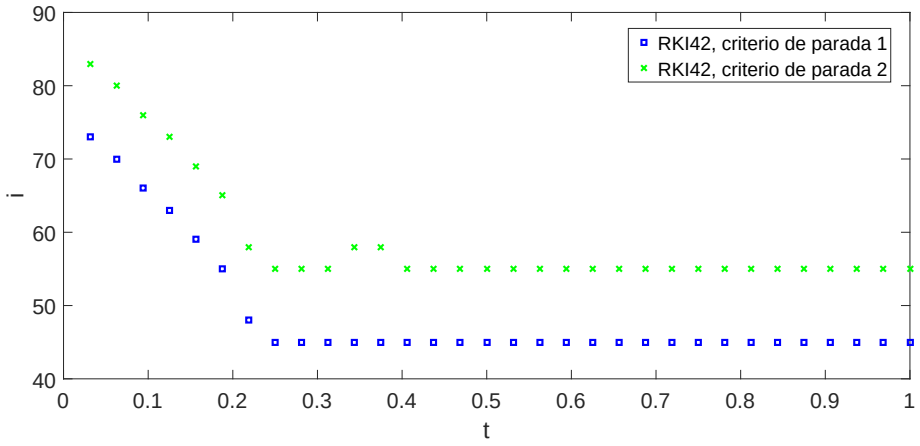


Figura 4.4: Iteraciones Newton modificado, $h = 3.1250 \times 10^{-2}$, Problema 4.3.

Se puede observar en la Figura 4.4 que Newton modificado al inicio del intervalo de integración presenta un mayor número de iteraciones, pero luego este comienza a disminuir quedando constante en un determinado valor. El comportamiento de estos puntos se puede relacionar con la rigidez del problema, ya que las soluciones de este SEDO presentan cambios bruscos en esa parte del intervalo, causando una mayor restricción de solución.

En la Figura 4.5 se ilustra la relación obtenida entre el tiempo de cómputo en segundos y los errores producidos en las aproximaciones con los MPU y MPM implícitos para el Problema 4.3. El tamaño de paso inicial es $h = 3.1250 \times 10^{-2}$ y se disminuye este valor hasta $h = 2.4414 \times 10^{-4}$. Los puntos que se miran en la figura hacen referencia a los resultados en cada tamaño de h, se disminuye el tamaño de paso en cada proceso. En la mayoría de los métodos se ilustran ocho puntos a excepción del AM3 y RKI42. Para AM3 se ha eliminado el primer punto porque el método diverge para el tamaño de paso inicial. Para RKI42 se ha eliminado el último punto, ya que el tiempo gastado es de 0.0225 segundos y es un tiempo mucho mayor al compararlo con los demás métodos, causando que no se mire con claridad la escala de la gráfica.

Se observa que el método RKI42 es un método que presenta una buena precisión para el primer tamaño de h, aunque es el método que mayor tiempo de cómputo gasta, ya que cuenta con dos términos implícitos consiguiendo más evaluaciones en sus funciones lo que justifica un tiempo mayor de cálculo. El error para este método después del quinto h se acerca al cero de máquina y comienza a

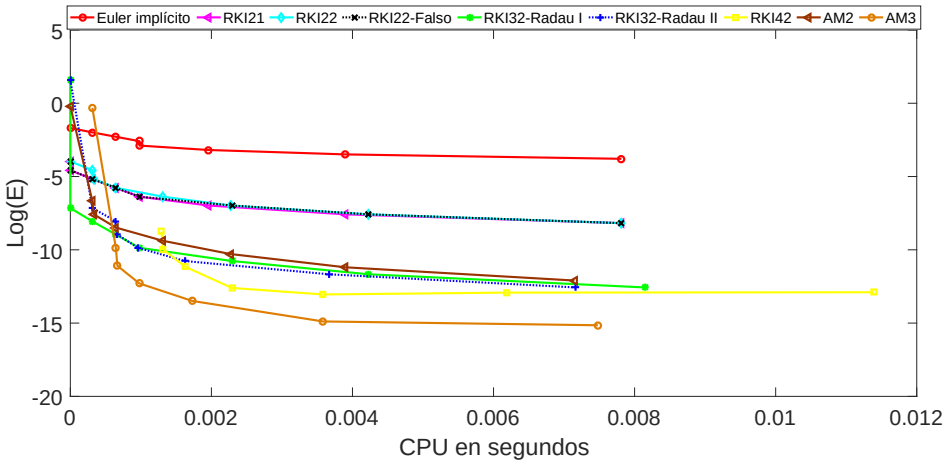


Figura 4.5: Prueba de tiempos métodos implícitos, Problema 4.3.

conseguir resultados no del todo confiables. Para RKI42 al igual que AM3 no se recomienda usar valores de h muy pequeños porque pueden llevar al cálculo de errores, es decir se debe tener cuidado al escoger el valor de h.

En el caso de AM3, se mira que después de un determinado h el método presenta una buena precisión mejorando el tiempo que muestra RKI42. Para los otros métodos los tiempos no varían mucho entre sí, pero si varían sus precisiones de acuerdo a su orden de convergencia.

Al realizar una comparación relativa de los tiempos finales con respecto al método más costoso que es RKI42, se tiene que todos los métodos gastan entre el 32 % y el 35 % del tiempo que gasta RKI42.

4.1.2. Estabilidad

Para verificar la estabilidad absoluta de los MPU y los MPM, se consideran los problemas 4.2 y 4.3 para realizar el análisis respectivo.

Para el Problema 4.2,

$$\begin{cases} y' = -40y + 40t + 1, \\ y(0) = 4, \quad \text{en} \quad t = [0, 20], \end{cases}$$

se tiene que $\lambda = -40$, obtenido de la solución homogénea de la EDO, relacionado con el problema $y' = \lambda y$. Para que λh esté dentro de la región de estabilidad de cada método el valor de h debe estar dentro del intervalo de estabilidad. La Tabla 4.8 presenta los intervalos de estabilidad de h para los MPU en este problema.

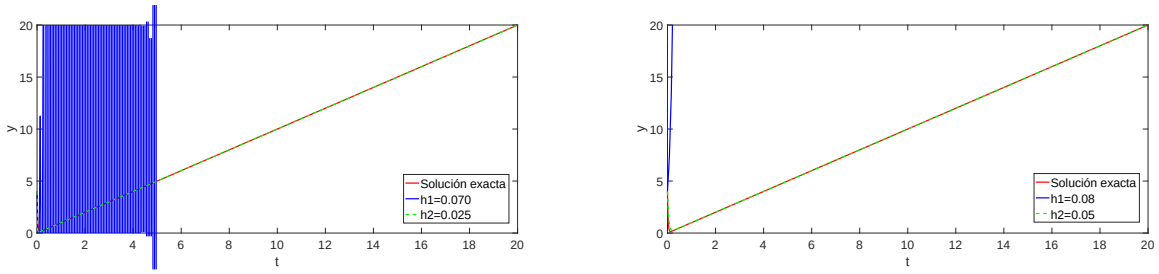
Método	Intervalo
Euler explícito, RK22	$(0, 0.05)$
RK33	$(0, 0.0628)$
RK44	$(0, 0.0695)$

Método	Intervalo
Euler implícito	$(-\infty, -0.05) \quad \vee \quad (0, \infty)$
RKI21, RKI22, RKI22-Falso, RKI42	$(0, \infty)$
RKI32 Radau I y II	$(0, 0.15)$

Tabla 4.8: Intervalos de estabilidad MPU explícitos e implícitos, Problema 4.2.

En la Tabla 4.8 se puede observar el tamaño de los intervalos de estabilidad de los métodos explícitos e implícitos y notar la diferencia que hay entre ellos. Se mira que en los métodos implícitos, RKI32 cuenta con una mayor restricción en su zona de estabilidad, sin embargo esta es mucho mayor que la de los métodos explícitos. Esto permite justificar el por qué los métodos explícitos deben reducir mucho más el tamaño de h para lograr que sus soluciones se estabilicen.

Para observar el comportamiento de las soluciones con los métodos explícitos según su estabilidad, en la Figura 4.6 se soluciona numéricamente el Problema 4.2 con algunos de estos métodos, Euler y RK44. Se han tomado tamaños de paso dentro y fuera del intervalo de estabilidad. Se observa que para un h_1 fuera de los intervalos los métodos divergen y para un h_2 dentro de los intervalos los métodos son estables alcanzando un error global equivalente al cero computacional. Además, se ve que Euler explícito necesita de un h_2 menor al de RK44 para que sus soluciones sean estables, ya que según los intervalos de estabilidad de la Tabla 4.8 este es más restringido.



(a) Soluciones Euler explícito.

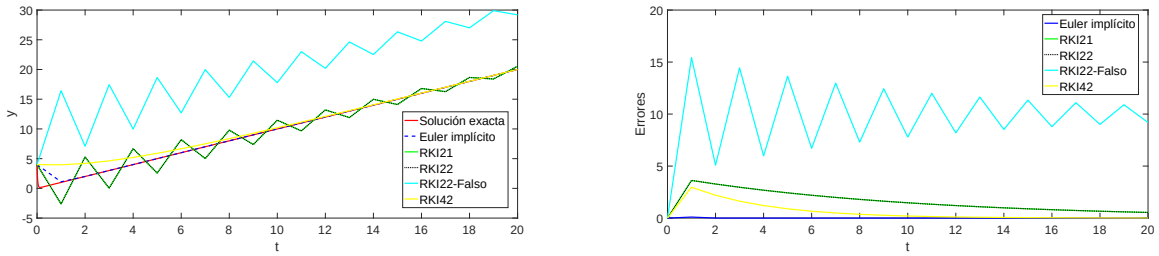
(b) Soluciones RK44.

Figura 4.6: Estabilidad absoluta métodos explícitos, Problema 4.2.

Así mismo, para el Problema 4.2 se realizaron varias simulaciones numéricas con diferentes tamaños de paso en todos los métodos implícitos. Para los métodos Euler implícito, RKI21, RKI22, RKI22-Falso y RKI42 los valores de h siempre están dentro del intervalo, ya que son métodos A-estables. Para el caso de RKI32 que no es un método A-estable se toman valores de h dentro y fuera del intervalo de estabilidad. Además, es válido aclarar que para el caso de Euler implícito no se considera el intervalo negativo, ya que los tamaños de paso de interés son $h > 0$.

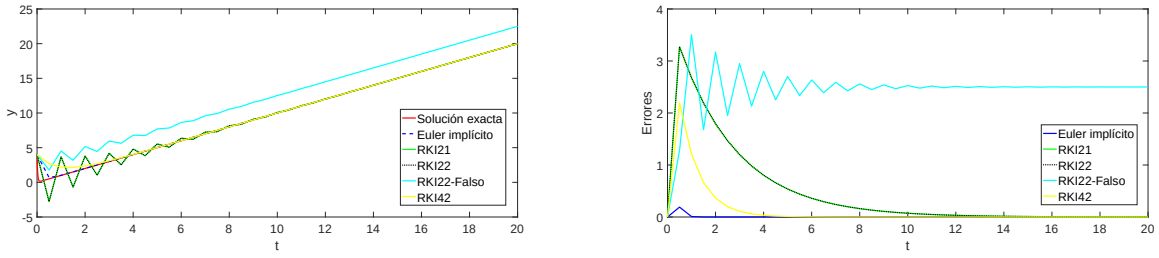
En las figuras 4.7, 4.8 y 4.9 se presentan las aproximaciones obtenidas con los métodos implícitos de orden 1, 2 y 4 para el Problema 4.2. Se observa el comportamiento de las soluciones numéricas junto con sus errores para tamaños de paso iguales a 1, 0.5 y 0.25, aunque para tamaños de paso mayores como $h = 10$ Euler implícito sea estable. Se mira que para $h = 1$ los métodos que consiguen estabilizarse son Euler implícito y RKI42, los otros métodos RKI21, RKI22 y RKI22-Falso son inestables generando oscilaciones en sus aproximaciones. Para $h = 1$ Euler alcanza el cero computacional en el error global, RKI de orden 2 un error de orden 10^{-1} y RKI42 un error de orden 10^{-3}. Al disminuir h los métodos se estabilizan aunque al inicio de las iteraciones sus soluciones se comporten de forma oscilatoria. Sin embargo, los métodos son convergentes consiguiendo errores globales menores o iguales a 10^{-9}. El método RKI22-Falso es inestable para estos tamaños de paso, es decir necesita disminuir mucho más el valor de h para lograr su estabilidad.

Se menciona que para RKI22-Falso el método de Newton sólo está realizando 2 iteraciones, debido a que las aproximaciones en cada una de las iteraciones son muy cercanas. El error entre ellas es muy pequeño y menor que la tolerancia que se utiliza, lo cual causa que las aproximaciones de las raíces no sean las mejores.



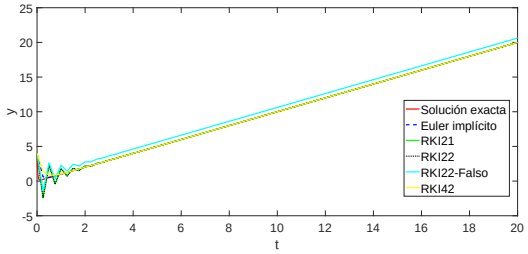
(a) Soluciones numéricas. (b) Errores.

Figura 4.7: Estabilidad absoluta métodos implícitos $h = 1$, Problema 4.2.

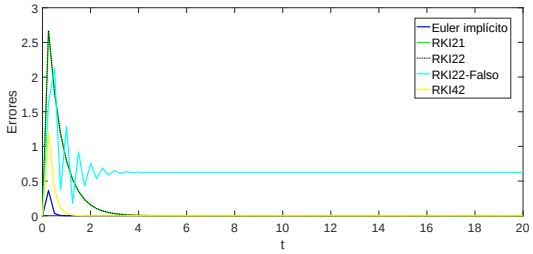


(a) Soluciones numéricas. (b) Errores.

Figura 4.8: Estabilidad absoluta métodos implícitos $h = 0.5$, Problema 4.2.



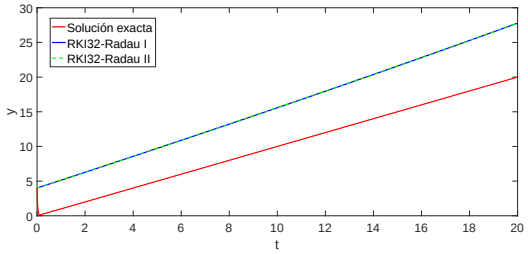
(a) Soluciones numéricas.



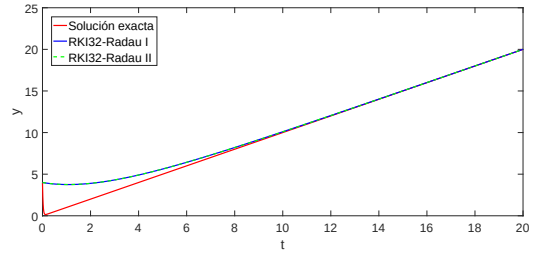
(b) Errores.

Figura 4.9: Estabilidad absoluta métodos implícitos $h = 0.25$, Problema 4.2.

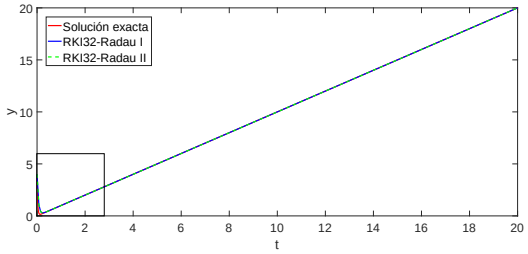
Para los métodos RKI32 se muestran los resultados numéricos en la Figura 4.10 para el Problema 4.2. Se han escogido tamaños de paso dentro y fuera del intervalo de estabilidad. Para el valor de $h = 1.5037 \times 10^{-1}$ fuera del intervalo los métodos divergen, este h ha sido escogido cercano al extremo derecho del intervalo, dado que si este es muy alejado el método fracasa. Para tamaños de h menores en el intervalo, se mira que las aproximaciones cuentan con una mayor precisión presentando estabilidad en el método. Con $h = 1.4598 \times 10^{-1}$ presenta un error global de orden 10^{-3} y con $h = 7.5188 \times 10^{-2}$ el error alcanza el cero computacional.



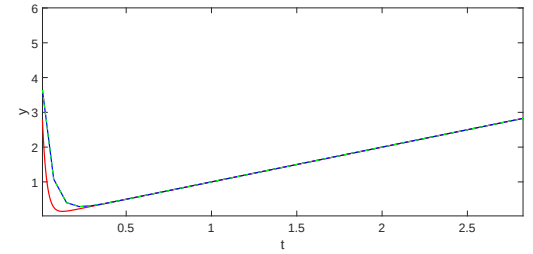
(a) Soluciones $h = 1.5037 \times 10^{-1}$.



(b) Soluciones $h = 1.4598 \times 10^{-1}$.



(c) Soluciones $h = 7.5188 \times 10^{-2}$.



(d) Zoom de c).

Figura 4.10: Estabilidad absoluta RKI32, Problema 4.2.

En la Figura 4.11 se ilustran con mayor claridad los errores generados con RKI32. Se mira que con un h fuera del intervalo el error es muy grade en comparación al error que se genera al utilizar h dentro del intervalo. Se verifica que para conseguir buenas aproximaciones se precisa trabajar con

tamaños de paso dentro del intervalo de estabilidad.

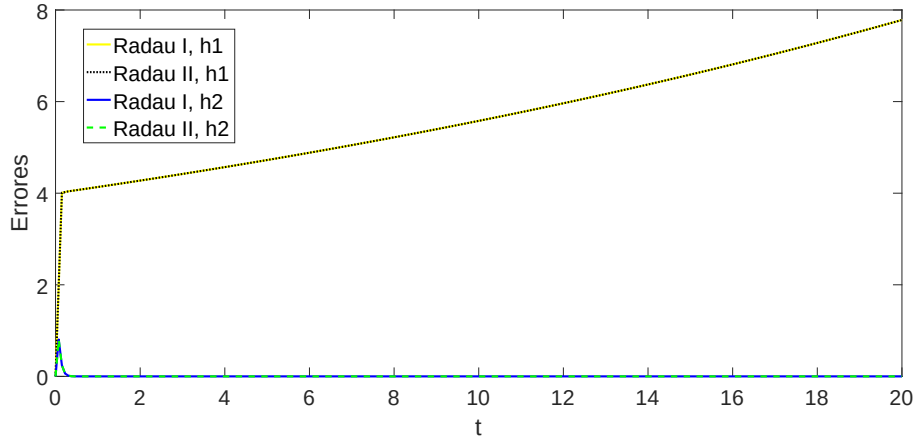


Figura 4.11: Errores método RKI32 con $h_1 = 1.5037 \times 10^{-1}$ y $h_2 = 7.5188 \times 10^{-2}$, Problema 4.2.

Se ha solucionado numéricamente el Problema 4.2 con el método de Euler implícito con punto fijo, en las figura 4.12 y 4.13 se muestran estos resultados. En la Figura 4.12 se mira que para un tamaño de paso $h = 1.9011 \times 10^{-2}$ dentro del intervalo de estabilidad el método es estable. Sin embargo, el tamaño de h utilizado es mucho más pequeño que los utilizados en los métodos anteriores, puesto que si se utiliza un h mayor el método de punto fijo fracasa. En la Figura 4.13 se presentan los errores cometidos con $h_1 = 1.9011 \times 10^{-2}$ y $h_2 = 1.2500 \times 10^{-2}$, para estos tamaños de paso el método genera un error global de orden 10^{-11}. Se ve que sin importar el tamaño de h utilizado se han causado errores al inicio de las iteraciones.

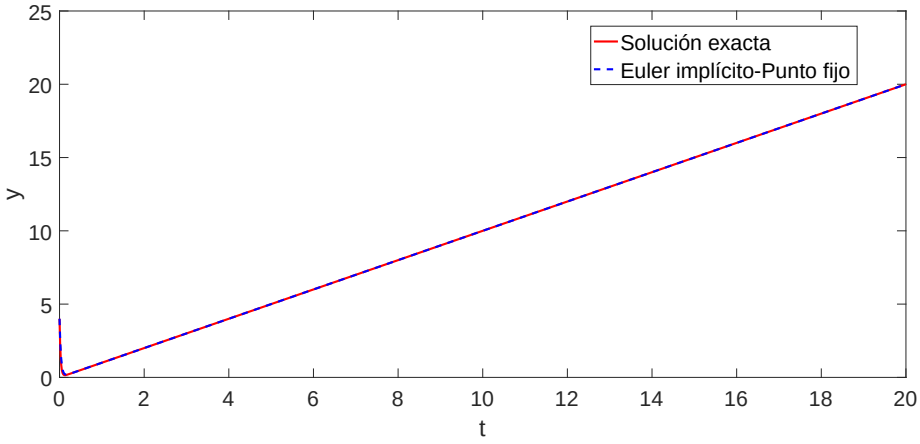
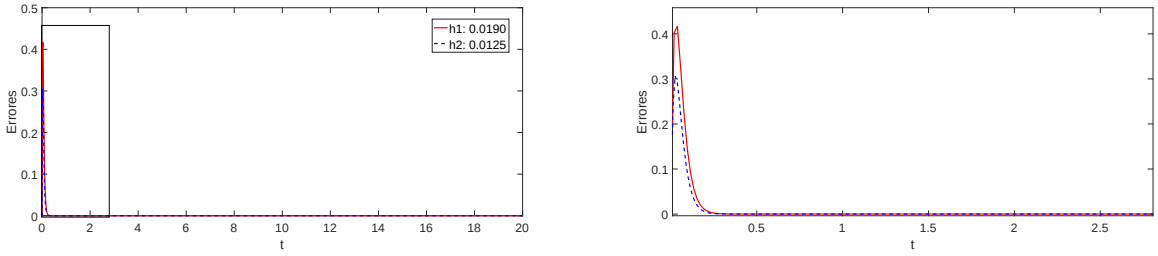


Figura 4.12: Estabilidad absoluta Euler implícito-punto fijo $h = 1.9011 \times 10^{-2}$, Problema 4.2.



(a) Errores h_1 y h_2. (b) Zoom de a).

Figura 4.13: Errores método Euler implícito-punto fijo $h_1 = 1.9011 \times 10^{-2}$ y $h_2 = 1.2500 \times 10^{-2}$, Problema 4.2.

Después de haber analizado la estabilidad para los MPU continuamos con los MPM. En la Tabla 4.9 se muestran los intervalos de estabilidad para estos métodos según el Problema 4.2, los cuales son útiles para analizar resultados numéricos con tamaños de paso fuera y dentro del intervalo.

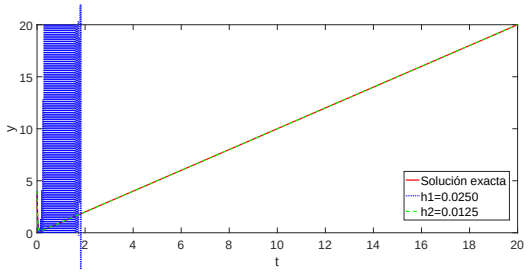
Método	**Intervalo**
AB3	$(0, 0.0136)$
AB4	$(0, 0.0075)$
AM2	$(0, 0.15)$
AM3	$(0, 0.075)$

Tabla 4.9: Intervalos de estabilidad MPM, Problema 4.2.

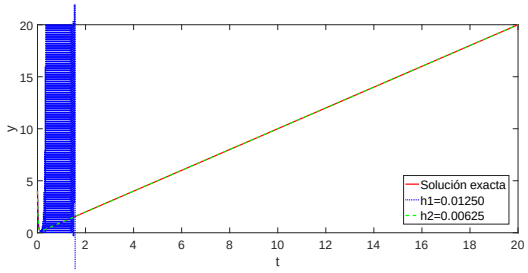
Se observa según los intervalos de los MPM que los métodos AM cuentan con una región de estabilidad mayor que los métodos AB, pero los MPM de orden 3 tienen un intervalo de estabilidad mayor a los de orden 4. Al comparar estos intervalos con los intervalos de los MPU de la Tabla 4.8, se tiene que los MPM cuentan con intervalos mayores a los MPU explícitos, pero menores a los MPU implícitos. El intervalo de AM2 coincide con el de RKI32.

En la Figura 4.14 se observa el comportamiento de las soluciones numéricas obtenidas para el Problema 4.2 con los métodos de Adams-Bashfoth de tercera y cuarta orden. Como cada uno de los métodos cuenta con un intervalo de estabilidad diferente, el análisis de estabilidad se ha realizado para valores distintos de h en cada uno de los métodos. Un h_1 fuera de los intervalos para los cuales los métodos divergen y un h_2 dentro de los intervalos donde los métodos consiguen estabilizar sus aproximaciones alcanzando un error del cero computacional. Por tanto, como las regiones de estabilidad de estos métodos son restringidas, se deben utilizar tamaños de pasos adecuados para lograr que los métodos sean estables.

Los métodos de AB a diferencia de los MPU explícitos estudiados, es que sus intervalos de estabili-



(a) Soluciones AB3.

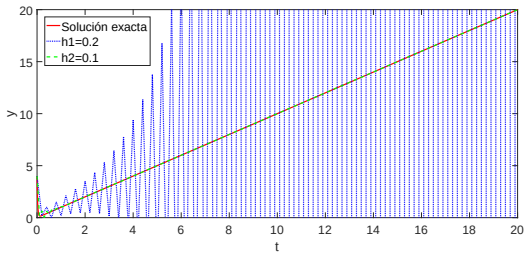


(b) Soluciones AB4.

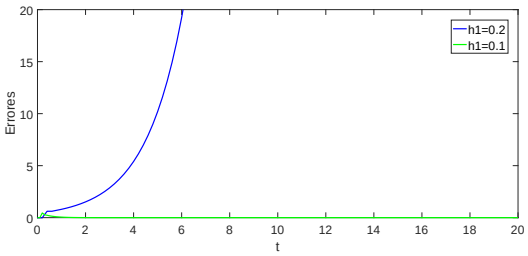
Figura 4.14: Estabilidad absoluta métodos Adams-Bashfoth, Problema 4.2.

dad no se relacionan como los MPU explícitos que entre mayor es el orden mayor es su intervalo. En este caso, el método AB4 presenta un intervalo más restringido que AB3, caso que también ocurre para los métodos AM.

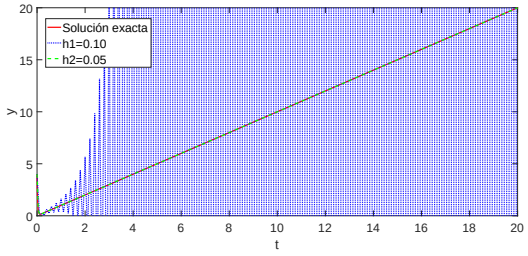
Similarmente, en la Figura 4.15 se ve el comportamiento de las soluciones numéricas y los errores causados en las aproximaciones durante el proceso de discretización con los métodos AM para el Problema 4.2. Se mira que para los valores de h_1 fuera de los intervalos los métodos divergen y por ende los errores crecen demasiado, los pequeños errores en el inicio de las iteraciones están afectando los resultados finales. En cambio para los valores de h_2 dentro de los intervalos, los métodos son estables y sus errores globales son muy cercanos a cero aunque presenten pequeñas oscilaciones al inicio de la integración.



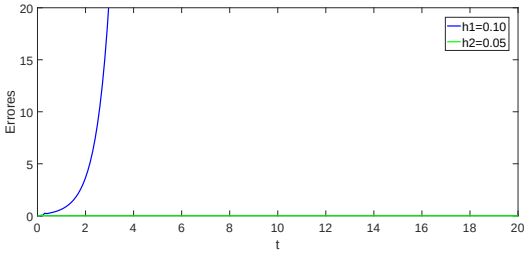
(a) Soluciones AM2.



(b) Errores AM2.



(c) Soluciones AM3.



(d) Errores AM3.

Figura 4.15: Estabilidad absoluta métodos Adams-Moulton, Problema 4.2.

Estos resultados han permitido validar la información de los intervalos de estabilidad que se ven en las tablas 4.8 y 4.9, como también los resultados de convergencia que se mostraron en la sección anterior para el Problema 4.2.

A continuación, se muestran resultados de la estabilidad numérica de los MPU y MPM para el Problema 4.3

$$\begin{cases} x' = -80.6x + 119.4y, \\ y' = 79.60x - 120.4y, \\ x(0) = 1 \quad \text{y} \quad y(0) = 4, \quad \text{en} \quad t = [0, 1]. \end{cases}$$

En la Tabla 4.10 se muestran los intervalos de estabilidad para este sistema con sus valores propios $\lambda_1 = -1$ y $\lambda_2 = -200$.

Método	**Intervalo**
Euler explícito, RK22	$\lambda_1 \to (0, 2)$ $\lambda_2 \to (0, 0.01)$
RK33	$\lambda_1 \to (0, 2.51)$ $\lambda_2 \to (0, 0.0126)$
RK44	$\lambda_1 \to (0, 2.78)$ $\lambda_2 \to (0, 0.0139)$

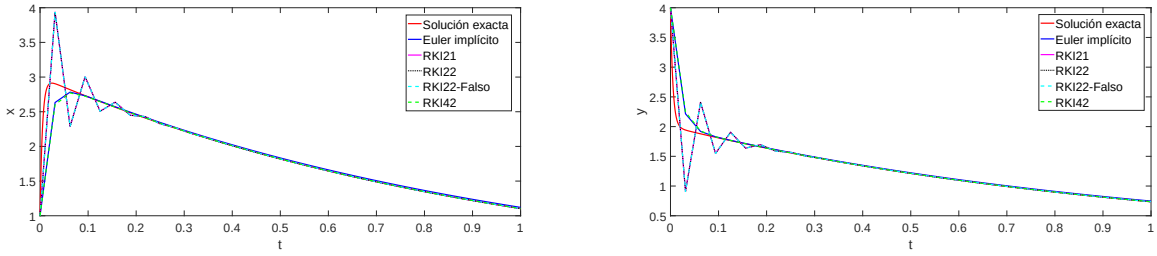
Método	**Intervalo**
Euler implícito	$\lambda_1 \to (-\infty, -2) \quad \vee \quad (0, \infty)$ $\lambda_2 \to (-\infty, -0.01) \quad \vee \quad (0, \infty)$
RKI21, RKI22, RK22 - Falso, RKI42	$\lambda_1 \to (0, \infty)$ $\lambda_2 \to (0, \infty)$
RKI32 Radau I y II	$\lambda_1 \to (0, 6)$ $\lambda_2 \to (0, 0.03)$

Tabla 4.10: Intervalos de estabilidad MPU explícitos e implícitos, Problema 4.3.

Según los intervalos que se muestran en la Tabla 4.10, el valor propio $\lambda_2 = -200$ es el que determina el intervalo de estabilidad, ya que es el que presenta mayor restricción según las condiciones de cada método. Al comparar el tamaño de los intervalos, se mira que los métodos explícitos cuentan con intervalos mucho más restringidos que los métodos implícitos. Para el caso de RKI32 que presenta el menor intervalo de los métodos implícitos, este continúa siendo mayor al de los explícitos.

En la Figuras 4.16 se muestran aproximaciones obtenidas con los métodos implícitos para el Problema 4.3. Se observa que para un tamaño de paso $h = 3.1250 \times 10^{-2}$ todos los métodos implícitos de orden 1, 2 y 4 son estables, presentando errores globales desde 10^{-2} a 10^{-9}. Estos métodos al inicio de las iteraciones presenten una mayor magnitud de error, para el caso de los métodos RKI de orden 2 con aproximaciones oscilatorias y para los demás métodos con un comportamiento similar a la solución exacta pero sin precisión. Se menciona que para tamaños de pasos mayores a este los métodos divergen, fracasan o consiguen errores globales de órdenes 10^{-1} y 10^{-2}.

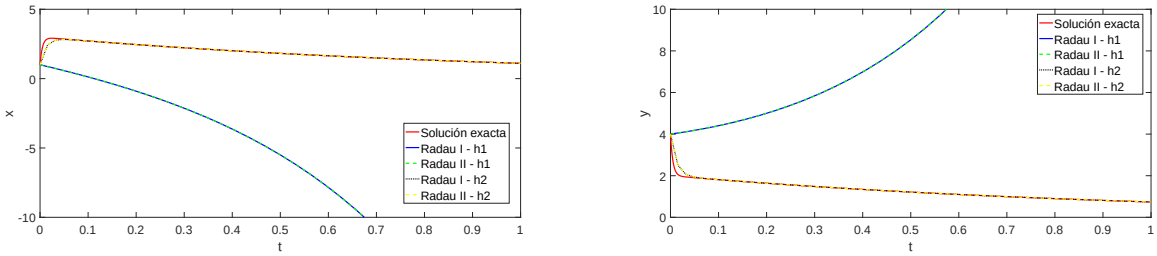
Para el caso de los métodos de RKI32 que cuentan con un intervalo de estabilidad con mayor restricción, en la Figura 4.17 se presentan resultados numéricos con valores de h dentro y fuera del



(a) Soluciones x. (b) Soluciones y.

Figura 4.16: Estabilidad absoluta métodos implícitos, $h = 3.1250 \times 10^{-2}$, Problema 4.3.

intervalo de estabilidad. Para $h_1 = 3.1250 \times 10^{-2}$ fuera del intervalo los métodos divergen, en cambio para $h_2 = 1.5625 \times 10^{-2}$ dentro del intervalo cuentan con estabilidad generando un error global de orden 10^{-8}, pero presentando un pequeño error al inicio de las iteraciones.



(a) Soluciones x. (b) Soluciones y.

Figura 4.17: Estabilidad absoluta RKI32, $h_1 = 3.1250 \times 10^{-2}$ y $h_2 = 1.5625 \times 10^{-2}$, Problema 4.3.

Seguidamente se verifica la estabilidad absoluta de los MPM para el mismo sistema lineal del Problema 4.3. Sus intervalos de estabilidad asociados a $\lambda_1 = -1$ y $\lambda_2 = -200$ se muestran en la Tabla 4.11. Se observa que los intervalos que toman mayor restricción son los asociados al valor propio λ_2, por tanto para garantizar la estabilidad absoluta de los MPM se debe elegir un h que esté dentro del intervalo más restringido.

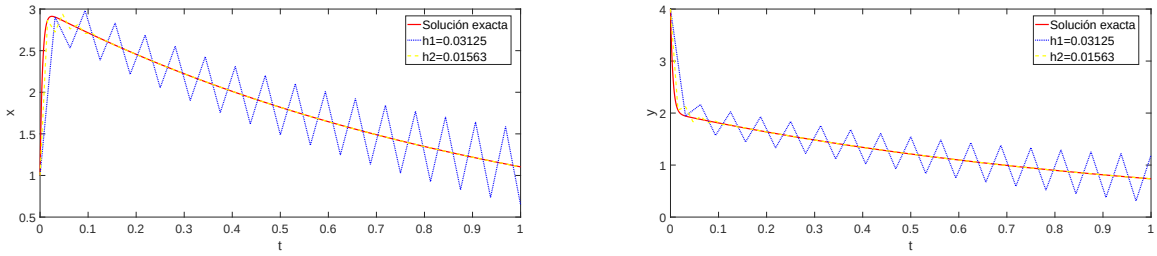
Método	**Intervalo**
AB3	$\lambda_1 \to (0, 0.5455), \quad \lambda_2 \to (0, 0.0027)$
AB4	$\lambda_1 \to (0, 0.3), \quad \lambda_2 \to (0, 0.0015)$
AM2	$\lambda_1 \to (0, 6), \quad \lambda_2 \to (0, 0.03)$
AM3	$\lambda_1 \to (0, 0.3), \quad \lambda_2 \to (0, 0.015)$

Tabla 4.11: Intervalos de estabilidad MPM lineales, Problema 4.3.

Al comparar los intervalos de estabilidad de la Tabla 4.10 y la Tabla 4.11, se mira que la mayoría de

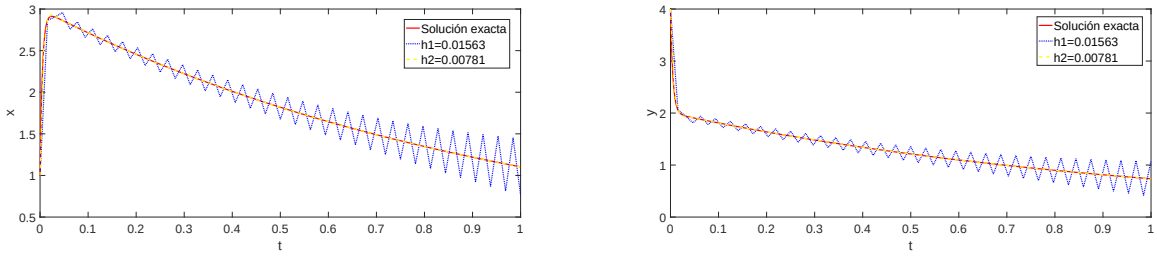
los MPM son los que cuentan con mayor restricción en la estabilidad numérica para este problema. Para el caso de AM2 su intervalo coincide con el de RKI32, siendo este menor que el de los MPU implícitos pero mayor al de los MPU explícitos. Sin embargo, este método es el que muestra un intervalo de estabilidad mayor en los MPM.

En las figuras 4.18 y 4.19 se observan los resultados numéricos para el Problema 4.3 obtenidos con los MPM. Se han tomado valores de h dentro y fuera de los intervalos de estabilidad dependiendo de cada método. Se observa que para los valores de h_1 fuera de los intervalos las aproximaciones en AM son inestables y alcanzan un error global de orden 10^{-1}. Al tomar un h_2 dentro de los intervalos las soluciones numéricas son estables con errores globales desde 10^{-7} a 10^{-13}.



(a) Soluciones x. (b) Soluciones y.

Figura 4.18: Estabilidad absoluta AM2, Problema 4.3.



(a) Soluciones x. (b) Soluciones y.

Figura 4.19: Estabilidad absoluta AM3, Problema 4.3.

Estos resultados al igual que el problema anterior permiten validar la información de las tablas 4.10 y 4.11, como también los resultados de convergencia que se presentaron en la sección anterior para el Problema 4.3.

Según los resultados anteriores se tiene que estos problemas cuentan con subintervalos de integración con una mayor restricción para ser solucionado numéricamente, es por esto que los métodos están generando una magnitud de error en el inicio de las iteraciones que es donde hay cambios bruscos

de las soluciones. Por esta razón, es necesario estudiar métodos numéricos que permitan corregir estos errores como lo son los métodos adaptativos, los cuales se estudian y analizan más adelante.

4.2. Métodos de tamaño de paso adaptativo

Por la existencia de SEDO difíciles de solucionar numéricamente, en un método numérico usar un tamaño de paso de integración constante no puede ser muy conveniente. En este caso lo recomendable es usar un tamaño de paso variable adaptable al problema, es decir algoritmos que se adaptan a la trayectoria de la solución. Cuando la solución varíe suavemente, convendrá adoptar un paso grande, mientras que cuando la misma varíe rápidamente, será conveniente adoptar un paso menor. Estos métodos se consideran métodos de paso adaptativo y para su implementación requieren de una estimación del error local en cada paso para determinar el valor adecuado para el paso siguiente. Los métodos adaptativos permiten observar y controlar el error local con el fin de mejorar la precisión de las aproximaciones y el tiempo de cómputo. Se recomienda seguir [9, 28] y [34].

La estimación de error es necesaria porque ayuda a garantizar que los tamaños de paso escogidos son los suficientemente pequeños para generar una precisión requerida en los resultados. Permite asegurar que los tamaños de paso sean lo suficientemente grandes para evitar trabajo computacional innecesario, ya que los valores de h demasiado pequeños hacen que se genere un gran costo computacional. Ver [16] y [28]

Algunos métodos alternativos para la estimación de error son la extrapolación de Richardson y el método de Melene que consiste en la combinación de las fórmulas integradas de Runge-Kutta. Estos métodos controlan el paso de integración. Un ejemplo de estos es la combinación de los métodos de RK como RK45 y RK56, conocida como el método de RK-Fehlberg que controla automáticamente el tamaño de paso. Estos métodos son presentados en [15, 23] y [24]. Por otro lado, el software de MATLAB ofrece un conjunto de funciones de pasos adaptativos para resolver SEDO. Algunas de estas funciones son ode45 recomendada para solucionar SEDO no rígidas y ode15$_s$ para solucionar problemas rígidos basada en las fórmulas de integración numérica. En [9] se implementa un algoritmo de las Backward Differentiation Formulae (BDF) de orden dos de tamaño de paso adaptativo utilizando la misma estrategia implementada en ode15$_s$, para complementar se recomienda seguir [9] y [34].

4.2.1. Control automático del tamaño de paso

En esta sección se introduce una estrategia adaptativa para la selección de un tamaño de paso, la cual se sigue del procedimiento descrito en [15] y [28].

Para implementar algoritmos de paso variable es necesario calcular en cada paso de integración un tamaño de h adecuado en base a la estimación del error. A continuación, se muestra la idea de un algoritmo que ajusta automáticamente el tamaño de paso, para lograr una tolerancia prescrita del error local.

Algoritmo 1: Ajuste automático de tamaño de paso.

1. Escoger de forma adecuada un valor de h inicial. Cada vez que se elige un tamaño de paso inicial el programa calcula dos aproximaciones a la solución $\boldsymbol{y}_i$ y $\hat{\boldsymbol{y}}_i$.

2. Calcular la aproximación para $\boldsymbol{y}(t_i)$ con dos paso consecutivos de tamaño h, asignándola a $\boldsymbol{y}_i$.

3. Calcular la aproximación para $\boldsymbol{y}(t_i)$ con un tamaño de paso $2h$, asignándola a $\hat{\boldsymbol{y}}_i$.

4. Calcular la estimación de error mediante la expresión

$$E = \sqrt{\frac{1}{d}\sum_{j=1}^{d}\left(\frac{\boldsymbol{y}_i^j - \hat{\boldsymbol{y}}_i^j}{Sc_i}\right)^2},$$

   donde d son las dimensiones del SEDO que se trabaja y

$$Sc_i = AbsTol + RelTol \cdot \text{máx}\left(\left\|\boldsymbol{y}_{i-1}^j\right\|, \left\|\boldsymbol{y}_i^j\right\|\right),$$

   donde $AbsTol$ hace referencia a la tolerancia absoluta y $RelTol$ la tolerancia relativa asignadas por el usuario.

5. Si $E \geq 1$ y $h > 10^{-15}$ el paso h se rechaza y se retorna al paso 2 con un nuevo tamaño de paso h_{nuevo} de la forma

$$h_{nuevo} = \text{máx}\left(10^{-15}, h \cdot \text{mín}\left(fac_{max}, \text{máx}\left(fac_{min}, fac\left(\frac{1}{E}\right)^{\frac{1}{p+1}}\right)\right)\right), \tag{4.2.1}$$

   con $fac_{max} = 1$, $fac_{min} = 0.1$, $fac = 0.25$ y p el orden del método. Sino, se aceptan los valores de h y $\boldsymbol{y}_i$ en t_i, pasando a $t_i + h_{nuevo}$ con h_{nuevo} de la expresión (4.2.1), pero con $fac_{max} = 5$, $fac_{min} = 0.25$ y $fac = 0.8$.

6. Realizar este proceso mientras $t_i <= b$, de lo contrario parar.

7. Si $t_i = b$ el proceso termina, sino se calcula la última aproximación en t_f con $h = t_f - t_i$.

Según [15] el aumento máximo del tamaño de paso fac_{max} se elige entre 1.5 y 5, esto evita que el código tenga pasos demasiado grandes, aumenta y contribuye su seguridad. De igual forma, se recomienda asignar al factor máximo $fac_{max} = 1$ en los pasos justo después de un rechazo de pasos.

Es importante recalcar que las combinaciones de los métodos adaptativos pueden ser diferentes. Por ejemplo, método adaptativo con Euler implícito y métodos adaptativos con RKI. Es decir, varían de acuerdo al método que se use para obtener la aproximación en $\boldsymbol{y}(t_i)$. Sin embargo, en este trabajo se implementó el método adaptativo con Euler implícito porque es el método que tiene menor orden y consiguió mejorar la precisión en las partes más restringidas de un problema rígido.

4.2.2. Tamaño de paso inicial

Inicialmente la elección del primer tamaño de paso para realizar la implementación de un método adaptativo estaba a cargo del usuario. El usuario era el encargado de asignar una idea aproximada de un buen tamaño de paso. Esta elección se realizaba a partir de la experiencia o ideas previas en la computación, pero una mala elección de h podría causar una pérdida de tiempo computacional. Por tanto, según la literatura varios personajes desarrollaron ideas para que la computadora elija el valor de este h. Siguiendo [15], se presenta la estructura del algoritmo de la elección de un buen tamaño de paso inicial.

Algoritmo 2: Tamaño de paso inicial.

1. Hacer una evaluación de la función en el punto inicial $\boldsymbol{f}(t_0, \boldsymbol{y}_0)$.
2. Hacer $d_0 = \|\boldsymbol{y}_0\|$ y $d_1 = \|\boldsymbol{f}(t_0, \boldsymbol{y}_0)\|$.
3. Si d_0 o $d_1 < 10^{-15}$, hacer $h_0 = 10^{-6}$, sino hacer $h_0 = 0.01(d_0/d_1)$.
4. Calcular $\boldsymbol{y}_1$ con Euler explícito, $\boldsymbol{y}_1 = \boldsymbol{y}_0 + h_0\boldsymbol{f}(t_0, \boldsymbol{y}_0)$.
5. Calcular $\boldsymbol{f}(t_0 + h_0, \boldsymbol{y}_1)$.
6. Hacer $d_2 = \|\boldsymbol{f}(t_0 + h_0, \boldsymbol{y}_1) - \boldsymbol{f}(t_0, \boldsymbol{y}_0)\| / h_0$.
7. Si $\max(d_1, d_2) \leq 10^{-15}$ hacer $h_1 = \max\left(10^{-6}, 10^{-3}h_0\right)$, sino calcular
$$h_1 = \sqrt[p+1]{\frac{0.01}{\max(d_1, d_2)}}.$$
8. Finalmente, el tamaño de paso inicial es
$$h = \min(100h_0, h_1).$$

En el algoritmo de tamaño de paso inicial, la manera de obtener el valor de $\boldsymbol{y}_1$ puede variar según su forma de ser calculado. Se puede realizar mediante el método de Euler explícito el cual se usa en este trabajo seguido de [15] o puede ser mediante un cálculo de derivadas como se ve en [28].

4.2.3. Soluciones numéricas

En esta sección se muestran algunas simulaciones obtenidas con el método de Euler implícito y el método Euler implícito adaptativo, con el fin de realizar una comparación entre estos al solucionar numéricamente un problema rígido.

Se ha solucionado numéricamente el siguiente problema, una EDO rígida, que cuenta con componentes en su solución que varían unas más rápido que otras.

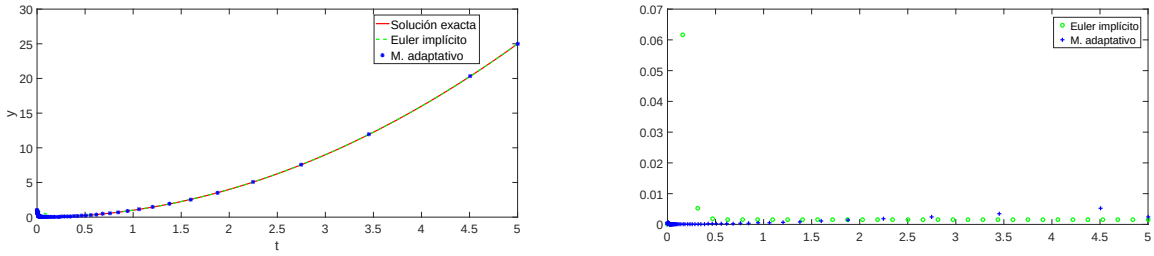
Problema 4.4. Usar los métodos Euler implícito y Euler implícito adaptativo para aproximar el problema de Cauchy

$$\begin{cases} y' = 2t - 100(y - t^2), \\ y(0) = 1, \quad \text{en} \quad t = [0, 5]. \end{cases}$$

Con solución exacta $y(t) = t^2 + e^{-100t}$, donde $y(5) = 25$.

Para aplicar el método adaptativo se han asignado tolerancias de $AbsTol = 10^{-4}$ y $RelTol = 10^{-3}$, las cuales han generado 103 pasos en la integración del intervalo.

En la Figura 4.20 se observan resultados con los dos métodos numéricos, para Euler implícito se han usado 32 pasos en el intervalo de integración y para el método adaptativo 103 que fueron los generaron según la tolerancia asignada. Se ha utilizado una cantidad de pasos distinta porque según Euler implícito para un $n = 32$ presenta un error de orden 10^{-3}, por tanto se eligieron tolerancias cercanas a este valor en el método adaptativo que generó 103 pasos. El menor tamaño de paso que utiliza el método adaptativo es $h = 1.9998 \times 10^{-4}$, para el cual Euler implícito realiza 25002 pasos generando un error global de orden 10^{-6} y el adaptativo un error de orden 10^{-5} en ese paso.



(a) Soluciones. (b) Errores.

Figura 4.20: Resultados numéricos Euler implícito y método adaptativo, Problema 4.4.

En la Figura 4.20a se presenta la comparación de la solución teórica del problema contra las aproximaciones obtenidas. Para Euler implícito el tamaño de paso es un valor de h fijo en cada iteración y para el método adaptativo se usa un valor de h variado. Este problema cuenta con subintervalos

de tiempo donde hay mayor restricción para ser solucionados numéricamente, por tanto el método adaptativo usa tamaños de pasos mucho más pequeños a los que usan donde la función es más suave. Por esta razón se mira que al inicio de la integración las iteraciones son mucho más consecutivas que al final de ellas.

El comportamientos de los errores producidos con estos métodos se ven en la Figura 4.20b. Se mira que Euler implícito está causando una magnitud de error mayor en el inicio de la integración que es donde se presenta el cambio brusco del problema rígido. En cambio el método adaptativo consigue controlar este error aunque este crezca en pequeñas magnitudes al final de la integración, ya que el tamaño de paso escogido en las iteraciones finales es mucho más grande. En el proceso de integración el método adaptativo consigue errores de órdenes 10^{-5}, 10^{-4} y 10^{-3} al final del intervalo.

Después de presentar estos resultados con una cantidad de pasos distinta para cada uno de los métodos, en la Figura 4.21 se muestra el comportamiento de los errores con los dos métodos pero con una misma cantidad de iteraciones. Se han utilizado 103 pasos en el intervalo de integración. Se puede observar que al haber aumentado la cantidad de pasos, Euler implícito continúa con una magnitud de error en el subintervalo donde el problema es mucho más estricto de solucionar numéricamente. Es decir, este método necesita de un tamaño de paso muy pequeño para lograr mejorar la precisión en esta zona.

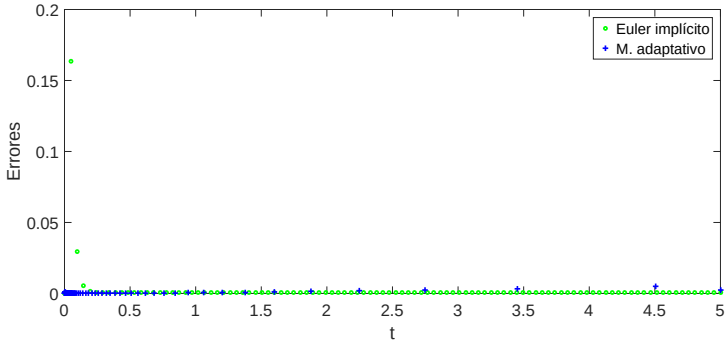


Figura 4.21: Errores numéricos métodos Euler implícito y adaptativo, Problema 4.4.

Más claramente, en la Figura 4.22 se ilustra el comportamiento de los errores obtenidos con Euler implícito para 32 y 103 pasos. Se observa que en los dos casos el error sigue presente en el inicio del intervalo.

Según los resultados obtenidos anteriormente, se puede concluir que el método adaptativo es un método que permite corregir los errores que se están obteniendo con el MPU. Este método logra tomar el h adecuado en cada paso para conseguir buenas aproximaciones. En la Figura 4.23 se

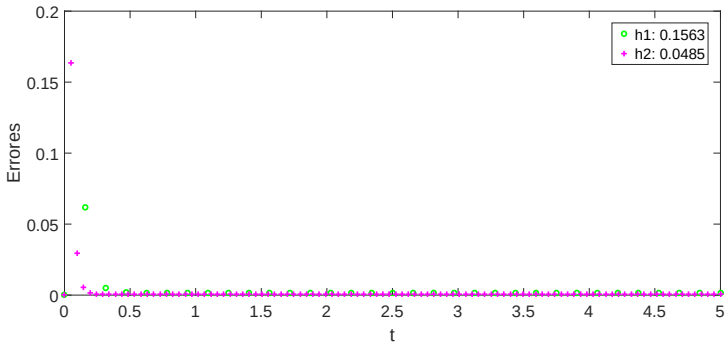


Figura 4.22: Errores con Euler implícito, Problema 4.4.

observa la variación del tamaño de paso en escala logarítmica durante el intervalo de integración. Se mira que el último valor de h disminuye con respecto a los anteriores, ya que el último tamaño de paso sugerido por el método hace que el valor de t se pase de t_f, por tanto el valor de h final se calcula realizando la diferencia entre t_f y t_{f-1}.

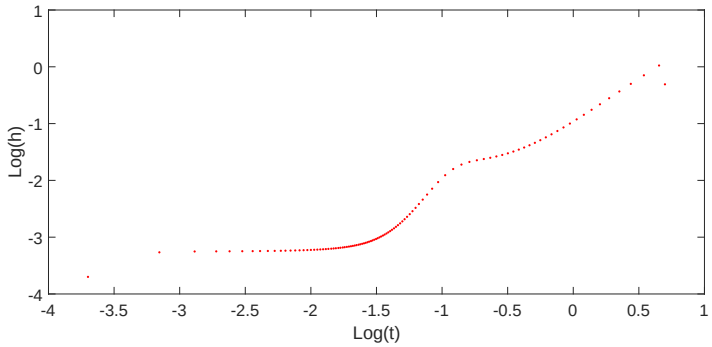


Figura 4.23: Variación del tamaño de paso Problema 4.4, método adaptativo.

En la Tabla 4.12 se muestran las pruebas de tiempos en segundos para el Problema 4.4 en el intervalo de $t = [0, 500]$, ya que para el intervalo inicial el método adaptativo no demanda tiempos considerables. Se observa que el tiempo varía según los valores de las tolerancias asignadas, es decir cuando se quiere que las aproximaciones sean mucho más precisas las tolerancias son más pequeñas y por ende el método genera una mayor cantidad de paso y un mayor tiempo computacional.

A continuación, se soluciona numéricamente con el método Euler implícito y el método adaptativo

AbsTol	**RelTol**	**N° de pasos**	**Tiempo**
10^{-4}	10^{-3}	109	0.0030
10^{-6}	10^{-4}	586	0.0081
10^{-9}	10^{-7}	27608	0.3246

Tabla 4.12: Prueba de tiempo en segundos - Método adaptativo, Problema 4.4.

el sistema lineal rígido del Problema 4.3, dado por

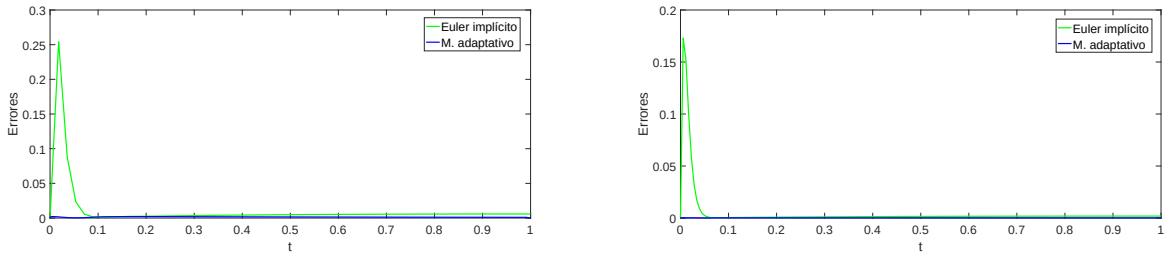
$$\begin{cases} x' = -80.6x + 119.4y, \\ y' = 79.60x - 120.4y, \\ x(0) = 1 \quad \text{y} \quad y(0) = 4, \quad \text{en} \quad t = [0, 1]. \end{cases}$$

En la Tabla 4.13 se muestran los datos asignados en los métodos de Euler, los cuales han generado una cantidad de pasos y un determinado error al final de la integración.

Método	**RelTol** y **AbsTol**	**n**	**Error**
Euler Adaptativo	10^{-3} y 10^{-4}	56	1.6910×10^{-2}
	10^{-4} y 10^{-6}	177	5.1304×10^{-3}
Euler Implícito	–	56	1.1756×10^{-2}
	–	177	3.7381×10^{-3}

Tabla 4.13: Resultados numéricos métodos de Euler, Problema 4.3.

Según los datos de la tabla anterior, se observa que los dos métodos consiguen un error del mismo orden, pero en la Figura 4.24 se ilustra los errores cometidos en el proceso de discretización. Se mira que al aumentar la cantidad de pasos, Euler implícito reduce su magnitud de error pero continúa con el comportamiento de su error al inicio de la integración, en cambio el método adaptativo presenta una mejor precisión.



(a) Errores con 56 pasos.

(b) Errores con 177 pasos.

Figura 4.24: Errores numéricos, Problema 4.3.

Aplicando el método de Euler implícito adaptativo, en la Figura 4.25 se muestra la variación de h para este problema usando distintos valores en las tolerancias. Los puntos de color rojo hacen

referencia a los h con 56 pasos y los de color azul con 177 pasos. Se observa que al inicio del intervalo de integración se consideran tamaños de paso más pequeños por la restricción que hay para su solución.

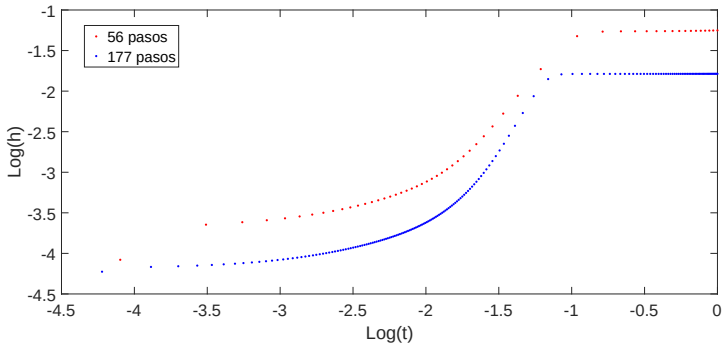


Figura 4.25: Variación del tamaño de paso Problema 4.3, método adaptativo.

Según los problemas abordados, se observó que el método adaptativo consigue mejorar los errores que se generan al solucionar problemas rígidos con los métodos MPU y MPM. Este método debido a su variación de h logra identificar cual es tamaño de paso apropiado para conseguir buenas aproximaciones.

A continuación, se trabaja una aplicación relacionada con las reacciones químicas, ya que estas están asociadas con los problemas rígidos.

4.3. Aplicación - Problema de ROBER

En esta sección se realiza una introducción sobre las reacciones químicas, se presenta el problema asociado a estas que es el problema de ROBER y se soluciona el mismo con los métodos numéricos estudiados en las anteriores secciones. Los resultados numéricos obtenidos se comparan con resultados presentes en la literatura, lo cual permite comparar y concluir acerca de ellos.

4.3.1. Reacciones químicas

La información general de la teoría de las reacciones químicas que se presenta en este trabajo se estudia con profundidad en [13] y [27].

El término *especie química* se refiere a cualquier elemento químico con una identidad dada, esta identidad la puede determinar el tipo, el número y la configuración de los átomos de esa especie. Aunque dos compuestos químicos tengan la misma cantidad de átomos en cada elemento, podrían

ser diferentes especies por tener diferentes configuraciones.

Se dice que ha ocurrido una *reacción química* cuando un número detectable de moléculas de una o más especies han perdido su identidad y han asumido una nueva forma, por un cambio en el tipo o número de los átomos del compuesto o por un cambio de estructura o configuración de dichos átomos.

Una especie puede perder su identidad química de tres maneras: por descomposición, por combinación o por isomerización. La *descomposición* hace referencia a la partición de una molécula en moléculas más pequeñas, átomos o fragmentos de átomos. La *combinación* ocurre cuando una molécula de alguna sustancia se combina con otra molécula para formar otra sustancia. La *isomerización* consta en un cambio de configuración de átomos.

Definición 4.1. Una *reacción química* es aquel proceso químico en el cual dos sustancias o más, denominados reactivos, por la acción de un factor energético, se convierten en otras sustancias designadas como productos.

En una reacción química se debe cumplir la *ley de conservación de la masa*, la cual consiste en que la suma de las masas de los reactivos es igual a la suma de las masas de los productos. Esto ocurre porque durante la reacción los átomos ni aparecen ni desaparecen, solo se ordenan de una forma distinta para formar nuevos enlaces. Cuando una reacción química cumple con esta ley, la reacción está *equilibrada*.

Ejemplo 4.1.

$$\underbrace{HCL + NaOH}_{Reactivos} \longrightarrow \underbrace{NaCl + H_2O}_{Productos}. \tag{4.3.1}$$

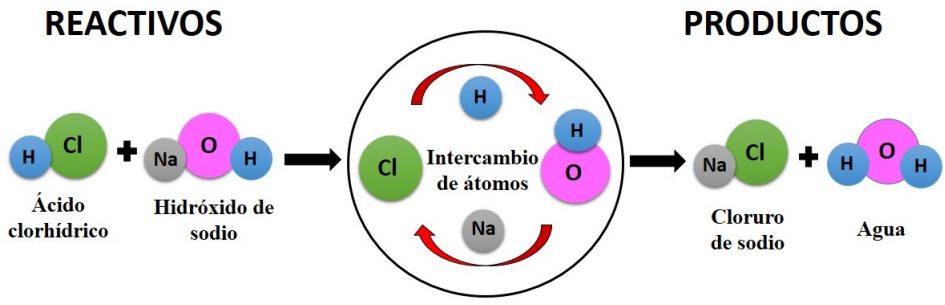


Figura 4.26: Reacción química, reordenamiento de átomos.

Solución. Según (4.3.1) y como se ve en la Figura 4.26, se observar que los reactivos están tomando un reordenamiento de átomos con el fin de formar los productos, es decir se une un átomo de cloro

(Cl) y un átomo de sodio (Na) para formar una molécula cloruro de sodio ($NaCl$) y un átomo de oxígeno (O) con dos átomos de hidróxido (H_2) para formar una molécula de agua (H_2O). □

Una reacción química se escribe esquemáticamente de la forma

$$aA + bB + \cdots = pP + qQ + \cdots , \tag{4.3.2}$$

donde $A, B, \ldots$ son especies químicas reactivas y $P, Q, \ldots$ son los productos de la reacción y $a, b, \ldots$ son números que en la ecuación significan moléculas. Es decir, a moléculas de A reaccionan con b moléculas de B, etc para dar p moléculas de P, q moléculas de Q, etc. Una reacción química representada por (4.3.2) se le denomina *ecuación estequimétrica* y los valores de $a, b, \ldots, p, q, \ldots$, se denominan *números estequiométricos*.

Agrupando todas las sustancias de la expresión (4.3.2) a un lado de la ecuación se tiene

$$-aA - bB - \cdots + pP + qQ + \cdots = 0.$$

En los problemas de la cinética química hay un interés sobre el estudio de lo que sucede en el interior de una reacción química, para lo cual se necesita del planteamiento de un modelo para realizar este proceso. Para dar una introducción sobre esto se supone un modelo que es la *ley de acción de masas*, la cual consiste en que si la temperatura se mantiene constante, la velocidad de una reacción química es proporcional al producto de las concentraciones de las sustancias que toman parte en la reacción. Ver más en [13] y [27].

De la definición de la ley de acción de masas se deriva la *velocidad de una reacción*, que es la variación de la concentración (en moles/unidad de volumen, supuesto el volumen constante) y dividido por el correspondiente número estequiométrico. Es la rapidez con la que se forman los productos o se consumen los reactivos. Además, esta cantidad es la misma para cada sustancia presente en la reacción.

Para representar las concentraciones de las sustancias A, B, P, Q, ... de la ecuación estequiométrica (4.3.2), se escribe $[A]$, $[B]$, $[P]$, $[Q]$, ..., donde la variación de cada una de estas es $\frac{d[A]}{dt}$, $\frac{d[B]}{dt}$, etc. Por lo tanto, la velocidad de la reacción para (4.3.2) está dada por

$$v = -\frac{1}{a}\frac{d[A]}{dt} = -\frac{1}{b}\frac{d[B]}{dt} = \frac{1}{p}\frac{d[P]}{dt} = \frac{1}{q}\frac{d[Q]}{dt} = \cdots \tag{4.3.3}$$

Para exponer la ecuación que representa lo que sucede en una reacción química, se debe decidir cuál es la función incógnita. En este caso, la función incógnita debe ser la concentración de cada sustancia en el instante t. En particular para una reacción de la forma $A + B = P + Q$, la ecuación diferencial que describe una reacción química para el compuesto A aplicando la ley de masas es

$$v = -\frac{d[A]}{dt} = k[A][B]. \tag{4.3.4}$$

Luego, llamando $x_A(t)$ y $x_B(t)$ a las concentración (en moles/volumen) de $[A]$ y $[B]$ que han reaccionado hasta el instante t y las concentraciones iniciales de estas $[A]_0 = \alpha$ y $[B]_0 = \beta$, resulta que $[A] = \frac{(\alpha - x_A(t))}{V}$ y $[B] = \frac{(\beta - x_B(t))}{V}$, con V el volumen donde se lleva a cabo la reacción. Dado que en la reacción considerada por cada mol de A actúa uno de B se tiene que $x_A(t) = x_B(t) = x(t)$. En consecuencia, reemplazando en (4.3.4) se tiene

$$v = -\frac{d\frac{(\alpha - x(t))}{V}}{dt} = k\frac{(\alpha - x(t))}{V}\frac{(\beta - x(t))}{V},$$

donde k es la *constante de velocidad de la reacción*. Haciendo $K = \frac{k}{V}$ se llega a

$$v = -\frac{dx}{dt} = K(\alpha - x)(\beta - x). \tag{4.3.5}$$

Lo mencionado anteriormente hace referencia a la teoría de las reacciones químicas para una EDO, lo cual se puede llevar al caso de SEDO. La ecuación (4.3.4) es la velocidad de la reacción para una sustancia química relacionada a una EDO. Para el caso de los SEDO, si la misma sustancia química está presente en dos o más reacciones que aparecen consecutivamente en un reactor, entonces la velocidad de reacción correspondiente a dicha sustancia es la suma de las velocidades en cada reacción.

Para una mayor comprensión, se presenta las siguientes reacciones químicas que se producen consecutivamente en un reactor como sigue

$$\begin{aligned} A + B &\xrightarrow{k_1} C, \\ C + B &\xrightarrow{k_2} D. \end{aligned} \tag{4.3.6}$$

En (4.3.6) k_1 y k_2 son las constantes de velocidad y A, B, C y D las sustancias involucradas. Se denota por $[A], [B], [C]$ y $[D]$ a las concentraciones molares de cada sustancia y según lo visto anteriormente, se tiene que la velocidad de cada una de las sustancia viene dada por

$$\begin{aligned} \frac{d[A]}{dt} &= -k_1[A][B], \\ \frac{d[B]}{dt} &= -k_1[A][B] - k_2[C][B], \\ \frac{d[C]}{dt} &= k_1[A][B] - k_2[C][B], \\ \frac{d[D]}{dt} &= k_2[C][B]. \end{aligned} \tag{4.3.7}$$

Según (4.3.7), se observa que ya no es una EDO sino un SEDO 4×4.

4.3.2. Formulación del modelo

El problema de ROBER describe la cinética de una reacción autocatalítica dada por Robertson en 1996. El nombre de ROBER fue dado por Hairer y Wanner. Este problema es muy popular en el

estudio numérico y a menudo se usa como un problema de prueba en el caso de los sistemas rígidos, además es presentado y simulado en [4, 8, 16, 25, 26] y [33].

El problema consiste en un sistema rígido no lineal de 3 ecuaciones diferenciales, es considerado rígido porque las magnitudes de las contantes de velocidad de la reacción presentan una gran diferencia, causando que el sistema tenga variaciones muy rápidas y muy suaves en sus componentes. La estructura de sus reacciones está dada por

$$\begin{aligned} A &\xrightarrow{k_1} B, \\ B + B &\xrightarrow{k_2} C + D, \\ B + C &\xrightarrow{k_3} A + C, \end{aligned} \tag{4.3.8}$$

donde k_1, k_2 y k_3 son las contantes de velocidad y A, B y C son las especies químicas involucradas. De [4, 16] y [26] se tomaron los valores numéricos $k_1 = 0.04$, $k_2 = 3 \times 10^7$ y $k_3 = 10^4$ y las concentraciones iniciales de las especies $y_1 = 1$, $y_2 = 0$ y $y_3 = 0$. Siguiendo la teoría general de las reacciones químicas con la suposición de la ley de acción de masas, se configura el modelo para el problema de ROBER

$$\begin{cases} y_1' = -0.04y_1 + 10^4 y_2 y_3, \\ y_2' = 0.04y_1 - 10^4 y_2 y_3 - 3 \times 10^7 y_2^2, \\ y_3' = 3 \times 10^7 y_2^2, \\ y_1(0) = 1, \quad y_2(0) = 0, \quad \text{y} \quad y_3(0) = 0. \end{cases} \tag{4.3.9}$$

Según la literatura, originalmente el Sistema 4.3.9 se planteó en el intervalo de tiempo $t = [0, 40]$, pero algunos trabajos como [26] y [33] lo abordan numéricamente en intervalos de integración mayores o iguales al intervalo inicial.

A continuación, se presentan aproximaciones para el problema de ROBER con algunos métodos numéricos trabajados anteriormente.

4.3.3. Soluciones numéricas

La idea de esta sección es abordar numéricamente el problema de ROBER en un intervalo de tiempo $t = [0, 40]$ y para esto se han utilizado métodos que fueron validados en la Sección 4.1.

El sistema (4.3.9) modela el problema propuesto y no cuenta con una solución teórica. Este problema se ha solucionado numéricamente con los métodos Euler implícito, RKI42, AM3, ABM4 y Euler adaptativo, ya que según las validaciones de las implementaciones estos han sido los más apropiados para solucionar problemas rígidos. Además, [26] muestra resultados para este problema

con un método de cuarta orden y la mayoría de estos son de la misma orden, por tanto se permite hacer una comparación entre ellos.

En [26] y [33] se presentan resultados numéricos para este problema en distintos intervalos de tiempo, en la Tabla 4.14 se muestran las aproximaciones para el intervalo que se aborda en este trabajo que es $t = [0, 40]$. En [26] se utiliza un método híbrido de paso múltiple que según la literatura es creado con el fin de utilizar una menor cantidad de pasos sin reducir su orden en comparación a los MPU y MPM que se han mencionado. Este artículo usa un método híbrido de orden cuatro con un tamaño de paso fijo $h = 0.001$ para conseguir las aproximaciones de la tabla. En [33] se usa un algoritmo denominado EPISODE, un paquete de FORTRAN, implementa un algoritmo de control automático de pasos basado en el método de BDF eligiendo un tamaño de h inicial de 10^{-8} y terminando con un $h = 1.3$. Asigna una tolerancia para el control del error de 10^{-9} la cual permite conseguir resultados con buenas precisiones, ya que al utilizar una tolerancia menor como 10^{-6}, el método refleja inestabilidad o una alta sensibilidad del problema, donde el método diverge si continúa con la integración.

Las aproximaciones que se ven en la Tabla 4.14 son las que se muestran en la literatura y se toman como referencia para verificar que las aproximaciones obtenidas no se están desviando a resultados erróneos. Se tomaron como soluciones exactas las aproximaciones de los artículos con cinco dígitos iguales en cada componente. Se miran los tamaños de paso que se usan en cada método con el fin de comparar este valor con los tamaños de paso que se usan en los métodos trabajados para obtener estos mismos resultados.

Ref	**h**	$\mathbf{y_1}$	$\mathbf{y_2}$	$\mathbf{y_3}$
[26]	0.001	0.715827068718994	$0.918553476456752 \times 10^{-5}$	0.284163745746361
[33]	1.3	0.715827	0.918552×10^{-5}	0.284164

Tabla 4.14: Soluciones numéricas para (4.3.9) en $t = 40$.

En la Tabla 4.15 se observan los resultados numéricos para el problema de ROBER obtenidos con Euler implícito, RKI42, AM3 y ABM4 en $t = 40$. La idea de esta tabla es comparar los resultados que se obtienen con estos métodos y el método presente en [26], ya que este artículo usa un método con un h fijo. Para adquirir estos resultados se utiliza en todos los métodos, a excepción de RKI42, el método de Newton para la solución de las ecuaciones no lineales y el cálculo de la inversa de la matriz jacobina se realiza de forma exacta. Para el caso de RKI42 se usa Newton con la modificación de Jacobi donde la matriz inversa se obtiene de forma aproximada, ya que es una matriz de dimensión 6×6. Además, la tolerancia asignada en Newton y su modificación ha sido $TOL = 10^{-6}$.

Método	Euler implícito	RKI42	AM3	ABM4
h	1.6×10^{-2}	6.4×10^{-2}	5.0×10^{-4}	2.5×10^{-4}
Error	8.1701×10^{-5}	1.1509×10^{-5}	8.3876×10^{-6}	7.9998×10^{-6}
Tiempo	0.0093	0.0039	0.3120	0.0394

Tabla 4.15: Resultados numéricos MPU y MPM.

En la Tabla 4.15 se muestran los resultados obtenidos para los cuales los métodos anteriores consiguen la misma precisión que en la literatura tomando cinco cifras iguales en las tres componentes, se muestran los tamaños de paso, los errores y los tiempos de cómputo. Según estos resultados el método RKI42 es el método que utiliza un tamaño de paso mayor y el métodos que precisan de un h más pequeño es ABM4 usando un $h = 2.5 \times 10^{-4}$. Así mismo, el método que gasta mayor tiempo de cómputo es AM3 y el de menor tiempo es RKI42. Comparando el tamaño de h utilizado en [26] con los de esta tabla, Euler implícito y RKI42 usan un tamaño de paso mayor al método híbrido del artículo.

Es importante resaltar que aproximaciones para tamaños de paso mayores al que se muestra en la tabla, Euler implícito converge con errores de órdenes 10^{-3} y 10^{-4}, RKI42 fracasa, AM3 diverge o fracasa y ABM4 diverge.

Para los métodos numéricos donde Newton o su modificación fracasan, se realizó el cambio del criterio de parada al criterio de la verificación de las raíces, pero no se consiguió que los métodos dejen de fracasar.

Los resultados numéricos para el problema de ROBER obtenidos con Euler implícito adaptativo se observan en la Tabla 4.16. Se soluciona el intervalo de integración con diferentes valores de tolerancias y con un tamaño de paso inicial $h = 9.1287 \times 10^{-4}$. Se comparan estas aproximaciones con los resultados que muestra [33] en la Tabla 4.14 calculados también con un método de control automático de pasos. Al comparar el tamaño de paso final se observa que los valores obtenidos no son similares al que muestra [33], este varía según las tolerancias asignadas al igual que las precisiones de sus aproximaciones. Se mira que para conseguir una precisión de orden 10^{-5} se necesita disminuir las tolerancias a 10^{-7} y 10^{-9}, ya que para tolerancias mayores se consiguen errores de órdenes mayores. En consecuencia, al disminuir estos valores el método tiene una mayor restricción en las precisiones, aumentando la cantidad de pasos y el tiempo de cómputo.

En la Figura 4.27 se ilustra el comportamiento del tamaño de paso que Euler implícito adaptativo varía según las tolerancias asignadas en la Tabla 4.16. Para tolerancias de 10^{-3} y 10^{-4} el mínimo y máximo tamaños de pasos usados son 9.1287×10^{-4} y 4.6230. Para tolerancias de 10^{-4} y 10^{-6} son

h final	AbsTol	RelTol	n	Error	Tiempo
4.6236	10^{-4}	10^{-3}	33	3.9617×10^{-3}	0.0006
6.5836×10^{-1}	10^{-6}	10^{-4}	142	9.8493×10^{-4}	0.0019
7.6157×10^{-3}	10^{-9}	10^{-7}	4390	3.4877×10^{-5}	0.0534

Tabla 4.16: Resultados numéricos Euler adaptativo, .

2.8577×10^{-4} y 1.4181 y para tolerancias 10^{-7} y 10^{-9} son 7.1705×10^{-6} y 4.7159×10^{-2}. Se puede notar que entre mayor sea la restricción en las tolerancias, mayor es la restricción en los tamaños de h.

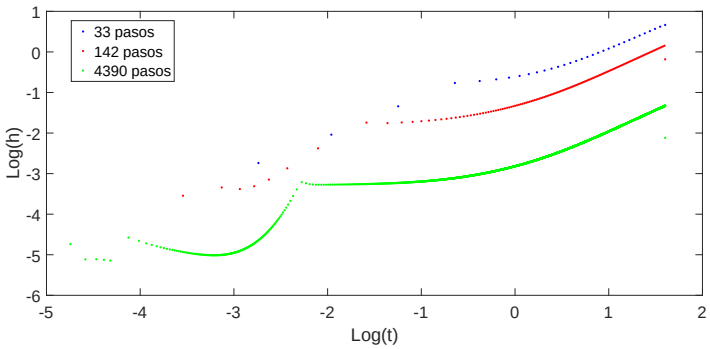
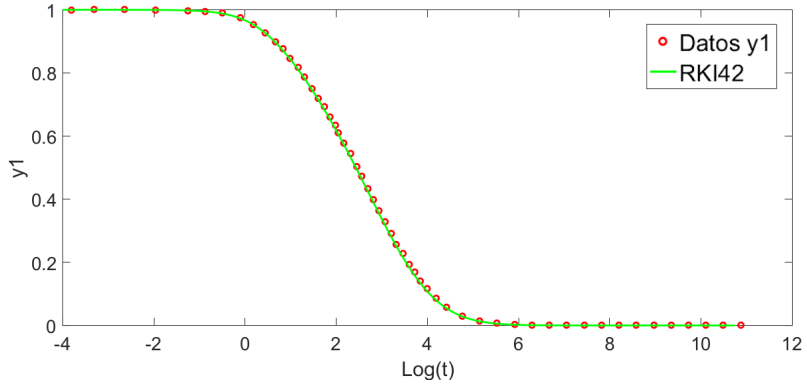


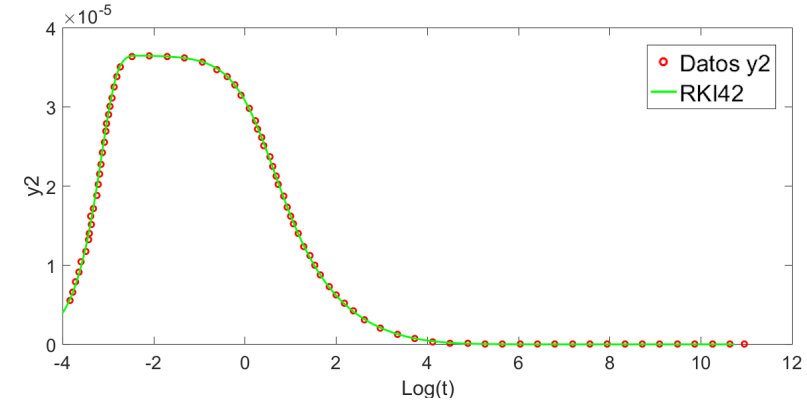
Figura 4.27: Variación del tamaño de paso, método adaptativo, problema de ROBER.

Se menciona que se usaron Euler explícito y RK44 con el fin de notar el comportamiento de las soluciones numéricas de este problema al abordarlo con métodos explícitos. Se observó que los dos métodos explícitos divergen para tamaños de paso $h < 5 \times 10^{-4}$. Para que los métodos consigan los 5 dígitos de igualdad en las soluciones deben disminuir más el tamaño de paso. El primer tamaño de paso para el cual los métodos no divergen es $h = 5 \times 10^{-4}$, para el cual Euler presenta un error de 7.9998×10^{-6} con un tiempo de cómputo de 0.0125 segundos y RK44 un error de 7.6480×10^{-6} en un tiempo de 0.0149 segundos. Al comparar estos resultados con los obtenidos con los métodos implícitos, se mira que la diferencia de estos es que los métodos implícitos permiten conseguir aproximaciones con tamaños de h mayores, en cambio los explícitos tiene una mayor restricción en el tamaño de paso para que estos no diverjan.

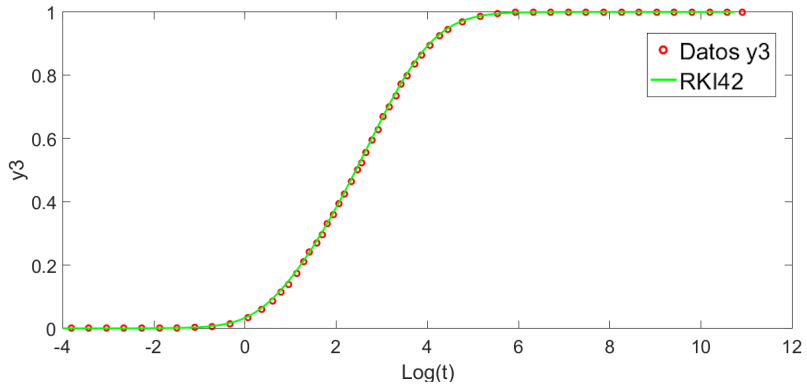
Finalmente, en la Figura 4.28 se ilustran gráficamente los comportamientos de las soluciones numéricas para el problema de ROBER en un intervalo de tiempo $t = [0, 10^{11}]$. Los datos para la simulación de las siguientes figuras se tomaron de [4] y las soluciones numéricas se obtuvieron con RKI42. Al realizar una comparación entre estas soluciones se observa que el método RKI42 es estable ya que sus soluciones numéricas se asemejan a los datos de la literatura.



(a) Solución numérica y_1.



(b) Solución numérica y_2.



(c) Solución numérica y_3.

Figura 4.28: Soluciones numéricas con RKI42, problema de ROBER.

Capítulo 5

Conclusiones y trabajos futuros

5.1. Conclusiones

Del estudio teórico y computacional realizado en este trabajo, se mencionan las siguientes conclusiones:

- Los conceptos teóricos e implementaciones estudiadas se profundizan especialmente en los métodos numéricos implícitos para la solución de SEDO rígidos. Como los SEDO se relacionan con la modelación matemática de distintas aplicaciones de la vida cotidiana, este estudio conlleva a la aplicación de estos métodos en la solución de problemas prácticos.

- El estudio de propiedades y características que permiten identificar los sistemas rígidos de los no rígidos se consideran sumamente importantes, ya que estas conllevan a la elección de los métodos numéricos adecuados en el momento de su solución con el fin de no realizar un mayor trabajo computacional.

- Se han validado las implementaciones de los métodos en el software Lenguaje C, solucionando numéricamente SEDO rígidos que cuentan con una solución teórica. Estos problemas ayudan verificar propiedades de los métodos numéricos como es su convergencia, consistencia y estabilidad, para utilizarlos con confiabilidad en la solución numérica de los problemas prácticos que no cuentan con una solución teórica. Las simulaciones numéricas permiten concluir características de las soluciones en los modelos matemáticos como se muestra en la Sección 4.3 al abordar el problema de ROBER.

- Es importante tener en cuenta las sugerencias mencionadas en el Capítulo 4, las cuales han surgido de la experiencia en las validaciones de los métodos numéricos. Estas conllevan al lector a no causar costos computacionales en el momento de aplicar un método numérico.

- Para la aplicación de los métodos numéricos implícitos se necesita de algunos métodos auxiliares para la solución de los sistemas de ecuaciones no lineales.

- Los métodos punto fijo, Newton y algunas de sus modificaciones como el método de Jacobi, han sido utilizados para las soluciones de los sistemas de ecuaciones no lineales en los métodos implícitos. Según el método utilizado para este proceso, se pueden causar ventajas o desventajas en el costo computacional. La aplicación del método punto fijo no fue conveniente en la solución de los sistemas no lineales, llevando al estudio del método de Newton que cuenta con un orden mayor de convergencia, una convergencia cuadrática. Como en la aplicación de Newton se precisa del cálculo de la matriz inversa, lo cual es costoso computacionalmente y además tediosos cuando se trabaja con matrices de grandes dimensiones, es conveniente utilizar este método con algunas de sus modificaciones como Jacobi para reducir este problema.

- Para la solución de sistemas rígidos se recomienda usar los métodos que cuenten con una zona de estabilidad adecuada como lo es la A-estabilidad o L-estabilidad, se considera que esta es una característica esencial de los métodos numéricos para abordar este tipo de problemas. El estudio de propiedades de los métodos numéricos para la solución de SEDO es sumamente importante, por ejemplo una buena elección de los tamaños de paso conlleva a mejorar las precisiones, costos computacionales y errores de redondeo que se pueden generar, ya que los computadores cuentan con una precisión finita y pueden causar este tipo de errores ya sea por los tamaños de paso escogidos o porque los métodos alcanzan el cero computacional.

- En las validaciones de los métodos numéricos utilizados en la solución de SEDO rígidos, los métodos con una mayor zona de estabilidad como los métodos implícitos son los más recomendables para abordar estos problemas. Aunque los métodos implícitos necesiten de la solución de sistemas de ecuaciones no lineales, los métodos explícitos precisan reducir en gran cantidad su tamaño de paso para no diverger. Al comparar estos métodos, los métodos implícitos presentan buenas precisiones para tamaños de paso mayores con una gran diferencia a los explícitos, es decir, el trabajo que realizan los métodos implícitos no logra compensar todas las cuentas que hacen los métodos explícitos.

- El proceso de la validación de las implementaciones de los métodos numéricos se considera una especie de laboratorio, donde se han escogido los métodos que presentaron mejores resultados para solucionar numéricamente un problema relacionado con las reacciones químicas. Estos métodos fueron MPU y MPM de orden cuatro implícitos y el método de Euler implícito, los cuales ofrecen una óptima relación computacional y de precisión.

- Como los SEDO rígidos cuentan con subintervalos donde su solución es mucho más estricta de solucionar numéricamente y los métodos estudiados presentan errores de mayor magnitud en esas zonas, fue necesario el estudio de los métodos adaptativo como Euler implícito adaptativo

para disminuir ese error. Estos métodos analizan el tamaño de h en cada paso consiguiendo una buena precisión.

5.2. Trabajos futuros

A seguir se presentan algunos temas futuros de investigación que se pueden continuar a partir de la realización de este trabajo:

- Métodos implícitos con la propiedad de L-estabilidad.
- Métodos de Rosenbrock.
- Métodos BDF implícitos.
- Métodos de paso múltiple no lineales.
- Profundización en los métodos adaptativos.
- Solución numérica de otras aplicaciones asociadas a SEDO rígidos.
- Profundización en el estudio de las condiciones de orden simplificado.
- Estudio de otras formas de deducción de los métodos RKI.
- Estudio y profundización de los temas en otras áreas, como la solución de sistemas de ecuaciones no lineales y la aproximación de la matriz inversa.

Apéndice A

Apéndice

A.1. Fundamentos del álgebra lineal

Para el estudio del análisis numérico con respecto a la aproximación de SEDO, especialmente para identificar los sistemas rígidos, es necesario tener en cuenta algunos conceptos fundamentales de álgebra lineal como la teoría de sistemas lineales, valores y vectores propios, determinante de una matriz, inversa de una matriz, normas matriciales y vectoriales, además del número de condicionamiento de una matriz, para los cuales se sugiere complementar en [5] y [17].

A.1.1. Valores y vectores propios

La teoría que se presenta en este apartado es de mucha utilidad para caracterizar cuando un SEDO es rígido, dado que el cálculo de estos valores está relacionado con una de las definiciones de estos problemas. Los conceptos y teoremas se tomaron de [5] y [17].

Definición A.1.1. Sea A una matriz de $n \times n$. El número real λ es un *valor propio* si existe un vector $\boldsymbol{x} \neq 0 \in \mathbb{R}^n$ tal que

$$A\boldsymbol{x} = \lambda\boldsymbol{x}. \tag{A.1.1}$$

Todo vector $\boldsymbol{x} \neq 0$ que satisfaga (A.1.1), se denomina un *vector propio* de A asociado al valor propio λ. Los valores propios también son llamados valores característicos, autovalores, valores latentes o eigenvalores. De forma similar para los vectores propios.

Observación A.1.1. λ puede ser real o complejo, y el vector $\boldsymbol{x}$ puede tener componentes reales o complejos.

Definición A.1.2. Sea $A = [a_{ij}]$ una matrix de $n \times n$, el determinante $f(\lambda) = det(A - \lambda I_n)$ es el *polinomio característico* de A, y la ecuación $f(\lambda) = det(A - \lambda I_n) = 0$ es la *ecuación característica de A*.

Teorema A.1.1. *Los valores propios de A son las raíces del polinomio característico de A.*

A.1.2. Inversa de una matriz

Definición A.1.3. Una matriz A de $n \times n$ es *invertible* (no singular) si existe una matriz B de $n \times n$ tal que

$$AB = BA = I_n.$$

La matriz B es llamada *inversa* de la matriz A. Si no existe esta matriz se dirá que A no es invertible.

Observación A.1.2. La inversa de una matriz A, si existe se denotará como A^{-1}.

Teorema A.1.2. *Si una matriz tiene inversa, la inversa es única.*

A continuación, se mencionan algunas propiedades de la inversa de una matriz:

- Se A es una matriz invertible, entonces A^{-1} es invertible y

$$(A^{-1})^{-1} = A.$$

- Si A y B son matrices invertibles, entonces AB es invertible y

$$(AB)^{-1} = B^{-1}A^{-1}.$$

- Si A es una matriz invertible, entonces

$$(A^T)^{-1} = (A^{-1})^T,$$

  donde A^T es la matriz transpuesta de la matriz A.

Corolario A.1.1. *Si $A_1, A_2, \ldots, A_r$ son matrices invertibles de $n \times n$, entonces $A_1A_2 \cdots A_r$, es invertible y*

$$(A_1A_2 \cdots A_r)^{-1} = A_r^{-1}A_{r-1}^{-1} \cdots A_1^{-1}.$$

A.1.3. Normas de matrices y vectores

Definición A.1.4. *Una norma vectorial* $\|\cdot\|$ para vectores reales es una función

$$\|\cdot\| : \mathbb{R}^n \longrightarrow \mathbb{R},$$

que cumple ciertas condiciones como:

1. *Definida positiva.* Para cualquier $\boldsymbol{x} \neq 0 \in \mathbb{R}^n$ se tiene que

$$\|\boldsymbol{x}\| > 0,$$

   y para $\boldsymbol{x} = 0$

$$\|\boldsymbol{x}\| = 0.$$

2. *Homogeneidad.* Para cualquier $\boldsymbol{x} \in \mathbb{R}^n$ y $\alpha \in \mathbb{R}$, se cumple que

$$\|\alpha \boldsymbol{x}\| = |\alpha| \, \|\boldsymbol{x}\| .$$

3. *Desigualdad triangular.* Para cualquier $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, se satisface que

$$\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\| .$$

Existen diferentes normas vectoriales denotadas por $\|\boldsymbol{x}\|_1$, $\|\boldsymbol{x}\|_2$ y $\|\boldsymbol{x}\|_\infty$. Mirar más en [5] y [17].

Similarmente hay normas matriciales inducidas que permiten estudiar y calcular propiedades en los sistemas lineales.

Definición A.1.5. *Una norma matricial* $|||\cdot|||$ para matrices reales cuadradas es una función

$$|||\cdot||| : \mathbb{R}^{n \times n} \longrightarrow \mathbb{R},$$

que cumple las siguientes condiciones:

1. *Definida positiva.* Para cualquier matriz $A \neq 0 \in \mathbb{R}^{n \times n}$, se satisface que

$$|||A||| > 0,$$

y para $A = 0$, entonces

$$|||A||| = 0.$$

2. *Homogeneidad.* Para cualquier matriz $A \in \mathbb{R}^{n \times n}$ y $\alpha \in \mathbb{R}^n$, se cumple que

$$|||\alpha A||| = |\alpha| \, |||A||| .$$

3. *Desigualdad triangular.* Para cualquier matriz A y $B \in \mathbb{R}^{n \times n}$, se tiene que

$$|||A + B||| \leq |||A||| + |||B||| .$$

4. *Consistencia.* Para cualquier matriz A y $B \in \mathbb{R}^{n \times n}$

$$|||AB||| \leq |||A||| \, |||B||| .$$

Después de la definición de norma, se concluyen algunas normas matriciales inducidas como la norma uno, la norma infinito y la norma dos para cualquier matriz $A \in \mathbb{R}^{n \times n}$. Para estas es posible demostrar que cumplen cada una de las propiedades de la definición de la norma matricial. Las siguientes igualdades son normas matriciales inducidas:

1. *Norma uno*: $|||A|||_1 = \text{máx}_{i=1,\cdots,n} \|Row_i A\|$, donde Row_i A representa la i-ésima fila de A.
2. *Norma infinito*: $|||A|||_\infty = \text{máx}_{i=1,\cdots,n} \|Col_i A\|$, donde Col_i A representa la i-ésima columna de A.
3. *Norma dos*: $|||A|||_2 = \sqrt{\rho(A^T A)}$, donde ρ es el radio espectral de A, es decir el mayor valor propio de la matriz $A^T A$.

A continuación, se menciona la definición de número de condicionamiento de una matriz, dado que es importante en el estudio numérico para aproximar sistemas lineales además de permitir realizar un análisis de los sistemas rígidos, ver [16].

A.1.4. Número de condicionamiento.

En muchas ocasiones al solucionar sistemas lineales aparecen sistemas de grandes dimensiones muy tediosos de solucionar, llevando a la utilización de herramientas computacionales para encontrar aproximaciones a sus soluciones. Debido a los procesos realizados computacionalmente y a la precisión finita del computador, se generan pequeñas perturbaciones que pueden afectar a las soluciones numéricas desviándolas de los resultados exactos, es por esto que es importante analizar qué tan cerca se encuentra la aproximación de la solución exacta.

Para un sistema lineal $A\boldsymbol{x} = \boldsymbol{b}$ se define el vector error $\boldsymbol{e} = \boldsymbol{x} - \boldsymbol{x^*}$, donde $\boldsymbol{x}$ es la solución exacta y $\boldsymbol{x}^*$ es la solución aproximada. Para analizar el error se usan las normas vectoriales.

Algunas de las perturbaciones en las componentes del sistema pueden ser $A(\boldsymbol{x} + \Delta\boldsymbol{x}) = \boldsymbol{b}$, $(A + \Delta A)(\boldsymbol{x} + \Delta\boldsymbol{x}) = \boldsymbol{b}$ o $A\boldsymbol{x} = (\boldsymbol{b} + \Delta\boldsymbol{b})$, siendo $\Delta A, \Delta\boldsymbol{b}$ y $\Delta\boldsymbol{x}$ pequeñas perturbaciones al sistema. Estas perturbaciones pueden afectar o no a la solución numérica del sistema, por tanto, a seguir se presenta una definición que permite saber cuándo un sistema puede ser afectado por las perturbaciones ocasionas en los procesos numéricos.

Definición A.1.6. El *número de condicionamiento* de una matriz A, con respecto a la norma inducida $|||\cdot|||$, es definido como

$$\kappa(A) = |||A||| \left|||A^{-1}|||\right.\,.$$

Para $\kappa(A)$ se puede encontrar una cota independiente de la norma inducida empleada, dada por

$$1 = |||I||| = \left|||AA^{-1}|||\right. \leq |||A||| \left|||A^{-1}|||\right. = \kappa(A).$$

Se dice que la matriz A está *bien condicionada* si el número de condicionamiento es cercano a 1 y *mal condicionada* si es mucho mayor que 1.

Ejemplo A.1.1. Solucionar los sistemas $H\boldsymbol{x} = \boldsymbol{b}$ y $H\boldsymbol{x} = (\boldsymbol{b} + \boldsymbol{\Delta b})$, para la matriz de Hilbert

$$H = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad y \quad \boldsymbol{\Delta b} = \begin{bmatrix} 0 \\ 0.1 \\ 0 \end{bmatrix}.$$

Solución. La solución exacta del sistema original es

$$\boldsymbol{x} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

y la solución numérica del sistema perturbado es

$$\boldsymbol{x}^* = \begin{bmatrix} -3.6 \\ 19.2 \\ -18 \end{bmatrix}.$$

□

Era esperado que las aproximaciones encontradas fueran cercanas, dado que solo se hace una pequeña perturbación al sistema lineal $H\boldsymbol{x} = \boldsymbol{b}$, pero $\boldsymbol{x}$ no es próximo de $\boldsymbol{x}^*$.

Al calcular el número de condicionamiento de la matriz H utilizando la norma 2, se tiene

$$\kappa_2(H) = |||H|||_2 \left|||H^{-1}|||_2 \approx 524.05678.\right.$$

Como el valor de $\kappa_2(H)$ es muy lejano a 1 se concluye que la matriz de Hilbert es mal condicionada y por ende los sistemas formados con esta también lo son.

En este ejemplo pequeñas perturbaciones afectan los resultados numéricos, pero algunas veces las matrices del sistema son matrices bien condicionadas, lo que implica que los resultados numéricos van a ser muy próximos a los exactos, es decir las perturbaciones en los procesos computacionales no afectan los resultados.

El número de condicionamiento de una matriz es importante en la solución de SEDO, ya que permite identificar si la matriz asociada a un SEDO está bien condicionada o no, y por ende caracterizar si el sistema es rígido o no.

A continuación, se presentan algunos teoremas y definiciones importantes para el estudio de la estabilidad absoluta en los métodos numéricos trabajados, esta teoría se presenta en [14].

A.1.5. Matrices semejantes

Definición A.1.7. Se dice que dos matrices A y B de $n \times n$ son *semejantes* si existe una matriz invertible C de $n \times n$ tal que

$$B = C^{-1}AC.$$

Teorema A.1.3. *Si A y B son matrices semejantes de $n \times n$, entonces A y B tienen el mismo polinomio característico y, por consiguiente, tienen los mismos valores característicos.*

Definición A.1.8. Una matriz A de $n \times n$ es *diagonizable* si existe una matriz diagonal D tal que A es semejante D.

Teorema A.1.4. *Una matriz A de $n \times n$ es diagonizable si y sólo si tiene n vectores característicos linealmente independientes. En tal caso, la matriz diagonal D semajante a A está dada por*

$$D = \begin{pmatrix} \lambda_1 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_n \end{pmatrix},$$

donde $\lambda_1, \lambda_2, \ldots, \lambda_n$ son los valores característicos de A. Si C es una matriz cuyas columnas son vectores característicos linealmente independientes de A, entonces

$$D = C^{-1}AC.$$

Corolario A.1.2. *Si la matriz A de $n \times n$ tiene n valores propios diferentes, entonces A es diagonizable.*

A.1.6. Forma canónica de Jordan

Según los teoremas y definiciones anteriormente mencionadas, una matriz A de $n \times n$ con n vectores característicos linealmente independientes se puede expresar de forma más sencilla ya que todos sus valores propios son distintos, es decir A es diagonizable y por tanto A es semejante a D. Sin embargo, en la práctica surgen matrices que no son diagonizables, pero es posible encontrar una matriz que sea semejante a esta aunque ya no sea diagonal.

Para analizar las matrices que no son diagonizables de define la matriz N_k de $k \times k$

$$N_k = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix},$$

donde N_k es una matriz con unos arriba de la diagonal principal y ceros en sus otras entradas. Para un escalar dado λ se define la *matriz de bloques de Jordan* $B(\lambda)$ por

$$B(\lambda) = \lambda I + N_k = \begin{pmatrix} \lambda & 1 & 0 & \cdots & 0 & 0 \\ 0 & \lambda & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda & 1 \\ 0 & 0 & 0 & \cdots & 0 & \lambda \end{pmatrix},$$

es decir, $B(\lambda)$ es la matriz de $k \times k$ con el valor de λ en la diagonal, unos arriba de la diagonal y ceros en las demás entradas.

Una *matriz de Jordan* J tiene la forma

$$J = \begin{pmatrix} B_1(\lambda_1) & 0 & \cdots & 0 \\ 0 & B_2(\lambda_2) & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & B_r(\lambda_r) \end{pmatrix},$$

donde cada $B_j(\lambda_j)$ es una matriz de bloques de Jordan, es decir es una matriz que en la diagonal tiene bloques de Jordan y ceros en las otras entradas.

Teorema A.1.5. *Sea A una matriz real o compleja de $n \times n$. Entonces existe una matriz C compleja invertible de $n \times n$ tal que*

$$C^{-1}AC = J,$$

donde J es una matriz de Jordan cuyos elementos en la diagonal son los valores característicos de A. Más aún, la matriz de Jordan es única, excepto por el orden en el que aparecen los bloques de Jordan.

Definición A.1.9. La matriz J del Teorema A.1.5 se denomina *forma canónica de Jordan* de A.

Teorema A.1.6. *Suponga que A una matriz de 2×2 tiene un valor característico λ de multiplicidad algebraica 2 y mutiplicidad geométrica 1. Sea $\boldsymbol{v_1}$ un vector característico correspondiente a λ, entonces existe un vector $\boldsymbol{v_2}$ que satisface la ecuación*

$$(A - \lambda I)\boldsymbol{v_2} = \boldsymbol{v_1}. \tag{A.1.2}$$

Definición A.1.10. Sea A una matriz con un solo valor característico λ que tiene multiplicidad geométrica 1. Sea $\boldsymbol{v_1}$ un vector característico de A, entonces el vector $\boldsymbol{v_2}$ definido por (A.1.2) se denomina *vector característico generalizado* de A correspondiente al valor característico λ.

Teorema A.1.7. *Suponga que* $A, \lambda, \boldsymbol{v1}$ *y* $\boldsymbol{v_2}$ *están definidos como en el teorema y sea* C *la matriz cuyas columnas son* $\boldsymbol{v_1}$ *y* $\boldsymbol{v_2}$*. Entonces* $C^{-1}AC = J$*, donde* J *es la forma canónica de Jordan de* A*.*

A seguir se introducen algunos conceptos y propiedades teóricas de las EDO y SEDO, métodos de solución analítica y numérica para SEDO, destacando propiedades y características de los mismos.

A.2. Teoría analítica de SEDO

En esta sección se dan a conocer algunas definiciones de SEDO, las cuales han sido tomadas de [3, 23] y [35].

Definición A.2.1. Se llama *Ecuación Diferencial Ordinaria* (EDO) a una ecuación que relaciona la variable independiente x, la función incógnita $y = y(x)$ y sus derivadas $y', y'', \ldots, y^{(n)}$; es decir, una ecuación de la forma

$$F(x, y, y', y'', \ldots, y^{(n)}) = 0,$$

donde F es una función real con $n+2$ variables, $x,\ \ y,\ \ y', \ldots,\ \ y^{(n)}$, y $y^{(n)} = \frac{d^n y}{dx^n}$.

Definición A.2.2. La *solución de una EDO* es una función $\phi(t)$ definida en un intervalo I que tiene al menos n derivadas continuas en I, las cuales al sustituirlas en la EDO la reducen a una identidad.

Definición A.2.3. Un *Sistemas de Ecuaciones Diferenciales Ordinarias* (SEDO) de primer orden es de la forma

$$\begin{cases} \dfrac{dy_1}{dx} = f_1(x, y_1, y_2, \ldots, y_n) \\ \dfrac{dy_2}{dx} = f_2(x, y_1, y_2, \ldots, y_n) \\ \quad \vdots \\ \dfrac{dy_n}{dx} = f_n(x, y_1, y_2, \ldots, y_n), \end{cases} \tag{A.2.1}$$

donde cada $f_i(x, y_1, y_2, \ldots, y_n)$ es una función escalar, $i = 1 \ldots n$.

Para reducir una EDO de orden superior a un SEDO se realiza un cambio de variable. Así, considere la EDO de n-ésimo orden

$$y^{(n)} = f(x, y, y', y'', \ldots, y^{(n-1)}). \tag{A.2.2}$$

Se introducen las variables dependientes $y_1,\ \ y_2, \ldots,\ \ y_n$, tales que

$$y_1 = y,\ \ y_2 = y',\ \ y_3 = y'', \ldots,\ \ y_n = y^{(n-1)}. \tag{A.2.3}$$

Luego $y_1' = y' = y_2$, $y_2' = y'' = y_3$, y así sucesivamente hasta $y_{n-1}' = y^{(n-1)} = y_n$. Por tanto, la sustitución de (A.2.3) en la ecuación (A.2.2) proporciona el SEDO

$$\begin{cases} y_1' = y_2 \\ y_2' = y_3 \\ \quad \vdots \\ y_{n-1}' = y_n \\ y_n' = f(x, y_1, y_2, ..., y_n). \end{cases}$$

Al trabajar con un SEDO es conveniente utilizar una notación vectorial por ser más manejable y compacta. Sean

$$\boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \quad \text{y} \quad \boldsymbol{f}(x, \boldsymbol{y}) = \begin{pmatrix} f_1(x, \boldsymbol{y}) \\ f_2(x, \boldsymbol{y}) \\ \vdots \\ f_n(x, \boldsymbol{y}) \end{pmatrix},$$

entonces el sistema (A.2.1) se puede escribir como

$$\boldsymbol{y'} = \boldsymbol{f}(x, \boldsymbol{y}). \tag{A.2.4}$$

Definición A.2.4. Una *solución de un SEDO* (A.2.4) es un conjunto de funciones suficientemente derivables $x_1 = \phi_1(t), x_2 = \phi_2(t), \ldots, x_n = \phi_3(t)$, que satisface cada ecuación del sistema en algún intervalo I.

Observe que se denota en negrita las variables que hacen referencia a vectores.

Existen diferentes métodos de solución analítica tanto para EDO y SEDO, como el método de eliminación sistemática, este método es útil en la solución de EDO con coeficientes constantes y se basa en el principio algebraico de eliminación de variables. Ver con más detalle en [3] y [35].

Definición A.2.5. El método de *eliminación sistemática* consiste en la eliminación de una incógnita en un sistema de ecuaciones diferenciales lineales se facilita al rescribir cada ecuación del sistema en notación de operador diferencial. Una ecuación lineal

$$a_n y^{(n)} + a_{n-1} y^{(n-1)} + \cdots + a_1 y' + a_0 y = g(t),$$

donde las a_i, $i = 0, 1, \ldots, n$ son constantes, puede escribirse como

$$a_n D^n + a_{n-1} D^{(n-1)} + \cdots + a_1 D' + a_0 D = g(t).$$

Si el operador diferencial de n-ésimo orden $a_n D^n + a_{n-1} D^{(n-1)} + \cdots + a_1 D + a_0$ se factoriza en operadores diferenciales de menor orden, entonces los factores conmutan.

En la siguiente sección se introducen aspectos teóricos de las cuadraturas gaussianas y los polinomios ortogonales, conceptos esenciales a tener en cuenta en la deducción de los métodos RKI.

A.3. Cuadraturas gaussianas y polinomios ortogonales

Una regla de la *cuadratura gaussiana* es una aproximación de la integral definida de una función y su construcción se basa en la teoría de *polinomios ortogonales*. A continuación se presentan elementos básicos de la teoría de polinomios ortogonales y las fórmulas de cuadratura, se recomienda seguir [8, 19] y [21].

Se quiere aproximar el valor de la integral $\int_a^b f(x)d(x)$ mediante una fórmula de integración numérica o fórmula de cuadratura

$$\int_a^b f(x)d(x) \approx w_1 f(x_1) + \cdots + w_n f(x_n), \tag{A.3.1}$$

donde la integral de la función $f(x)$ respecto a la medida $d(x)$ es aproximada por una suma finita que involucra n valores de la función en distintos *nodos* x_i seleccionados adecuadamente.

Se consideran las fórmulas de *tipo interpolatorio* en las cuales la aproximación se obtiene integrando el polinomio de interpolación, es decir

$$\int_a^b f(x)d(x) \approx \int_a^b p(x)d(x),$$

donde $p(x)$ es el polinomio de interpolación de Lagrange de $f(x)$ en $x_1, \ldots, x_n$. En la base de Lagrange $p(x)$ viene expresado como

$$p(x) = f(x_1)l_1(x) + \cdots + f(x_n)l_n(x).$$

La expresión (A.3.1) es de tipo interpolatorio si los pesos w_i vienen dados por

$$w_i = \int_a^b l_i(x)d(x), \quad i = 1, \ldots, n.$$

Teorema A.3.1. *Dados los nodos $x_1, \ldots, x_n$, la fórmula de cuadratura*

$$\int_a^b f(x)d(x) \approx w_1 f(x_1) + \cdots + w_n f(x_n)$$

es de tipo interpolatorio si y sólo si es exacta para los polinomios $1, x, \cdots, x^{n-1}$.

La fórmula de cuadratura tiene grado de precisión n si es exacta para todo polinomio de grado menor o igual que n, una fórmula interpolatoria con n nodos posee al menos grado de precisión

$n-1$. Según la literatura no se puede construir una fórmula de tipo interpolatorio con n nodos y grado de precisión $2n$, el grado de precisión óptimo usando n nodos es $2n-1$, grado de precisión alcanzado por las *fórmulas de cuadratura gaussianas*.

La regla de cuadratura vista en (A.3.1) generalmente es expresada como una suma ponderada de valores de la función en puntos específicos dentro del dominio de integración. Una fórmula de cuadratura gaussiana de n puntos, llamada así por Carl Friedrich Gauss, es una regla de cuadratura construida para producir un resultado exacto para polinomios de grado $2n-1$ o menos mediante una elección adecuada de los puntos x_i y los pesos w_i. El dominio de integración para tal regla se toma convencionalmente como $[-1,1]$, por lo que la regla se establece como

$$\int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i). \tag{A.3.2}$$

La cuadratura gaussiana (A.3.2) producirá resultados precisos si $f(x)$ está bien aproximada por una función polinomial dentro del rango $[-1,1]$. La función integrada se puede escribir como $f(x) = W(x)g(x)$, donde $g(x)$ es aproximadamente polinomial y $W(x)$ se conocen, luego hay pesos alternativos w_i tales que

$$\int_{-1}^{1} f(x)dx = \int_{-1}^{1} W(x)g(x)dx \approx \sum_{i=1}^{n} w_i f(x_i).$$

La función $f(x)$ es aproximada por un polinomio denominado *polinomio ortogonal*. Existe una variedad de estos polinomios, pero en este trabajo se abordan los *polinomios de Legendre* por su importancia que tienen en la deducción de los métodos RKI. Estos polinomios satisfacen que $W(x) = 1$ y además los valores de x_i son sus raíces, para complementar se recomienda [21].

Los polinomios de Legendre se representan por medio de $P_n(x)$ pertenecientes al intervalo $[-1,1]$, donde n indica el grado del polinomio. $P_n^*(x)$ son los polinomios de Legendre en el intervalo $[0,1]$, se definen $P_n^*(x) = P_n(2x-1)$ que son los polinomios que se usan en la deducción de los métodos RKI en el Capítulo 2. Ver más en [8] y [21].

Estos polinomios se pueden obtener de su forma general

$$P_n(x) = \frac{1}{2^n} \sum_{k=0}^{n} \binom{n}{k}^2 (x+1)^{n-k} (x-1)^k$$

o a partir de la fórmula de Rodrigues

$$P_n(x) = (2^n n!)^{-1} \left(\frac{d}{dx}\right)^n (x^2-1)^n.$$

Los primeros polinomios de Legendre en el intervalo $[0,1]$ son los siguientes:

- $P_0^*(x) = 1$.
- $P_1^*(x) = 2x - 1$.
- $P_2^*(x) = 6x^2 - 6x + 1$.
- $P_3^*(x) = 20x^3 - 30x^2 + 12x - 1$.

Los resultados que se muestran a continuación resumen la teoría básica acerca de las fórmulas de cuadratura gaussianas y los polinomios ortogonales. Estos teoremas pueden ser vistos en [21].

Una manera de establecer una única sucesión de polinomios ortogonales para cada medida es especificar que todos ellos sean *mónicos*, es decir con coeficiente director igual a 1.

Teorema A.3.2. *Sea $p_n(x)$ el polinomio ortogonal (mónico) de grado n. Entonces, las n raíces de $p_n(x)$ son reales, simples y pertenecientes al intervalo abierto (a, b).*

Teorema A.3.3. *Sean $x_1, \ldots, x_n$ las raíces del polinomio $p_n(x)$ de grado n para la medida $d(x)$ en (a, b). Supongamos que se hallan los pesos $w_1, \ldots, w_n$ imponiendo la exactitud para los polinomios de grado menor o igual que $n - 1$, es decir que se construye la fórmula de tipo interpolatorio*

$$\int_a^b f(x)d(x) \approx w_1 f(x_1) + \cdots + w_n f(x_n).$$

Entonces, dicha fórmula tiene grado de precisión $2n - 1$.

El teorema a seguir afirma que no es posible hallar por otro procedimiento otra fórmula de tipo interpolatorio con grado de precisión $2n - 1$.

Teorema A.3.4. *Si una fórmula*

$$\int_a^b f(x)d(x) \approx w_1 f(x_1) + \cdots + w_n f(x_n)$$

tiene grado de precisión $2n - 1$, entonces los puntos x_i deben ser los ceros del polinomio ortogonal $p_n(x)$ para la medida $d(x)$ en (a, b).

El siguiente teorema afirma la positividad de los pesos.

Teorema A.3.5. *En una fórmula de cuadratura gaussiana, todos los pesos w_i son positivos.*

A seguir se ilustran algunas implementaciones de los métodos numéricos trabajados realizadas en Lenguaje C.

A.4. Implementaciones

A continuación, se muestran las implementaciones de algunos métodos implícitos que se hicieron durante el desarrollo de este trabajo para la solución numérica de SEDO.

En el siguiente apartado se presenta la implementación del método Euler implícito, declarando las funciones necesarias para su aplicación. Estas funciones están escritas con relación al Problema 4.3, por tanto sus funciones y derivadas se escriben de forma particular.

A.4.1. Método Euler implícito.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

//Se definen f1 y f2, funciones no lineales del SEDO a solucionar con Newton.
double f1(double x, double h, double *y, double *z){
  return z[0]-y[0]-h*(-80.6*z[0]+119.4*z[1]);
}
double f2(double x, double *y, double h, double *z){
   return z[1]-y[1]-h*(79.6*z[0]-120.4*z[1]);
}

//Declaración de derivadas parciales de f1 y f2.
double df11(double x, double h, double *y, double *z){ //f1' con respecto a z[0]
   return 1+h*80.6;
}

double df12(double x, double h, double *y, double *z){ //f1' con respecto a z[1]
   return -h*119.4;
}

double df21(double x, double h, double *y,double *z){ //f2' con respecto a z[0]
   return -h*79.6;
}

double df22(double x, double h, double *y, double *z){ //f2' con respecto a z[1]
   return 1+120.4*h;
}
```

```
/*En esta sección se definen funciones que se utilizan en la aplicación de Euler
    implícito, matriz inversa, normas vectoriales, operaciones entre matrices y vectores
    y lectura de archivos*/

//Función método de Newton para un sistema de ecuaciones.
double Newton(double x, double h, double *y, double *z){
   //Declaración de variables, vectores y matrices.
   int i, j, itmax, n1;
   double TOL, N_E, *sol, *F, *MV, **JF;

   TOL = 1e-10; // Tolerancia.
   itmax = 100; // Número máximo de iteraciones.
   *z = *y; // Igualación de vectores, condición inicial de *z.
   i = 0;
   //Iteraciones en Newton
   do{
      i = i + 1; // Conteo de iteraciones.

      //F vector de la evaluación de las funciones f.
      //JF matriz jacobiana y JF^{-1} su inversa.

      //Multiplicación matriz por vector
      MV = JF^{-1} * F;

      //Cálculo de la solución según la iteración de Newton.
      for(j=0; j<n1; j++){
         sol[j] = z[j] - MV[j];
      }

      //Cálculo de la norma Euclidiana mediante función.
      N_E = ||z,sol||;

      *z = *sol; //igualación de vectores
   }
   while (N_E>TOL && i<itmax);

  //Datos de salida
   if (i>itmax){
      printf("El método fracasó");
      exit(1);
   }
```

```
   else{
      return *z; //Soluciones del sistema no lineal
   }
}

/*Función Euler implícito. Esta función recibe parámetros como los valores y la cantidad
    de subdivisiones del intervalo, el tamaño de paso, las condiciones iniciales y
    retorna las aproximaciones buscadas en el punto determinado.*/

double Euler_Imp(double a, double b, int n, double h, double *y, double *z){
      //Declaración de variables
      int i;
      double x;

      //Ciclo que recorre el intervalo de integración.
      for(i=1; i<=n; i++){
         x = a + (i * h);
         //Aproximaciones en el punto x de las soluciones del sistema no lineal, con z el
             vector de salida.
         Newton(x, h, y, z);
         *y = *z; //Igualación de vectores, nuevas condiciones iniciales.
      }
      //Aproximación de la solución del SEDO en el extremo final del intervalo (b).
      return *z;
}

int main(int argc, char**argv){
   //Declaración de variables y vectores.
   int n, d1, d2;
   double a, b, h, xn, *Int_n, *y, *z;

   //Asignación de memoria dinámica, con d1 y d2 el tamaño de los vectores.
   Int_n = (double*) malloc (d1*sizeof(double)); //Intervalo y cantidad de particiones.
   y = (double*) malloc (d2*sizeof(double)); //Condiciones iniciales de las soluciones.
   z = (double*) malloc (d2*sizeof(double)); //Vector temporal.

   // Creación y configuración de archivos.
   FILE *arch_A, *arch_B, *arch_salida;
   arch_A = fopen(argv[1],"r"); //Datos del intervalo y cantidad de particiones.
   arch_B = fopen(argv[2],"r"); //Datos condiciones iniciales de las soluciones.
   arch_salida = fopen("Resultados.txt","w"); //Archivo para imprimir los datos de salida.
```

```
  //Lectura de archivos. Ingresa datos de un archivo a un vector.
  Read_file(arch_A,Int_n,d1);
  Read_file(arch_B,y,d2);

  a = Int_n[0]; //Extremo inicial del intervalo de integración.
  b = Int_n[1]; //Extremo final del intervalo de integración.
  n = Int_n[2]; //Cantidad de subdivisiones del intervalo.
  h = (b - a) / n; //Cálculo del tamaño de paso.
  xn = a + h * n;

  //Aproximaciones del SEDO con Euler implícito en b, con z el vector de salida.
  Euler_Imp(a, b, h, n, y, z);

  //Iteración final de Euler implícito en el caso de ser xn diferente de b.
  if (fabs(xn-b)>1e-10){
     *y = *z;
     Euler_Imp(xn, b, b-xn, 1, y, z);
  }

  //Impresión de resultados en un archivo, aproximaciones del SEDO.
  fprintf(arch_salida,"\%.10lf \t \%.10lf\n", z[0], z[1]);

 //Cierre de archivos
  fclose(arch_A);
  fclose(arch_B);
  fclose(arch_salida);
}
```

En el siguiente apartado se muestra la función del método de RKI21, donde para su aplicación se deben declarar sus respectivas funciones similar al método anterior. Esta función recibe ciertos parámetros como los valores y la cantidad de subdivisiones del intervalo, el tamaño de paso, las condiciones iniciales y retorna las aproximaciones buscadas en el punto determinado.

A.4.2. Método Runge-Kutta implícito, RKI21.

```
//Función método Runge-Kutta orden dos con un estados RKI21.
double RKI21(double a, double b, int n, double h, double *y, *k1, double *Sol){
     //Declaración de variables y vectores
     int i;
     double x1, x2, *k2;
```

```
      //Ciclo que recorre el intervalo de integración.
      for (i=0; i<n; i++){
         x1 = a + (i * h);
         x2 = x1 + 0.5 * h;

         // Aproximaciones en el punto x2 de las soluciones del sistema no lineal, datos
             de salida *k2.
         Newton(x2,h,y,k1,k2);

         // Iteración método RKI21
         Sol[0] = y[0] + h * k2[0];
         Sol[1] = y[1] + h * k2[1];

         // Igualación de vectores, nuevas condiciones iniciales.
         *y = *Sol;
         *k1 = *k2;
      }
      //Aproximación de la solución del SEDO en el extremo final del intervalo (b) y
          nuevas condiciones para *k1.
      return *k1, *Sol;
}
```

En el siguiente apartado se muestra la función del método AM2, donde para su aplicación se deben declarar sus respectivas funciones similar a la implementación del método de Euler implícito. Además, se debe agregar la función del método de Runge-Kutta de tercera orden para el cálculo de los valores previos a la aproximación requerida. La función de AM2 recibe ciertos parámetros como los valores y la cantidad de subdivisiones del intervalo, el tamaño de paso, las condiciones iniciales y retorna las aproximaciones buscadas en el punto determinado.

A.4.3. Método Adams-Moulton, AM2.

```
//Función método de Adams-Moulton con dos pasos AM2.
double AM2(double a, double b, int n, double h, double *k, double *y, double *z){
      //Declaración de variables y vectores
      int i;
      double *x;

      //Ciclo que recorre el intervalo de integración.
      for (i=2; i<=n; i++){
```

```
        x[0] = a + (i - 2) * h;
        x[1] = a + (i - 1) * h;
        x[2] = a + (i * h);

        //Aproximaciones en el punto x[2] de las soluciones del sistema no lineal, con z
            el vector de salida.
        Newton(x, h, k, y, z);

        //Igualación de vectores, nuevas condiciones iniciales.
        *k = *y;
        *y = *z;
    }
    //Aproximación de la solución del SEDO en el extremo final del intervalo (b) y
        nuevas condiciones para *k.
    return *k, *z;
}
```

Referencias

[1] Akinfenwa, O. A., Jator, S. N., y Yao, N. M. (2013). *Continuous block backward diferentiation formula for solving stiff ordinary differential equations*. Computers and Matematics with Applications, 65(7), 996-1005. DOI: 10.1016/j.camwa.2012.03.111.

[2] Alberdi, E. (2013). *Métodos numéricos para ecuaciones diferenciales rígidas. Aplicación a la semidiscretización del método de Elementos Finitos* (Tesis doctoral). Universidad del País Vasco, País Vasco, España.

[3] Alpala, J. (2017). *Soluciones numéricas para un modelo lineal y otro no-lineal aplicados a la diabetes* (Tesis de pregrado). Universidad de Nariño, Pasto, Colombia.

[4] Amat, S., Legaz, M. J., y Ruiz, J. (2019). *On a variational method for stiff differential equations arising from chemistry kinetics*. Mathematics, 7(5), 459. DOI: 10.3390/math7050459.

[5] Bolaños, C. (2016). *Análisis teórico y computacional sobre matrices esparzas* (Tesis de pregrado). Universidad de Nariño, Pasto, Colombia.

[6] Bui, T. D., y Bui, T. R. (1979). *Numerical methods for extremely stiff systems of ordinary differential equations*. Applied Mathematical Modelling, 3(5), 355-358. DOI: 10.1016/s0307-904x(79)80042-6.

[7] Burden, R. L., y Douglas, J. (2002). *Análisis numérico*, 7ma ed. México: International Thomson Editores, S. A.

[8] Butcher, J. C. (2008). *Numerical methods for ordinary differential equations*, 2da ed. England: John Wiley & Sons, Ltd.

[9] Celaya, E. A., Aguirrezabala, J. A., y Chatzipantelidis, P. (2014). *Implementation of an adaptive BDF2 formula and comparison with the MATLAB ode15s*. In ICCS, 29, 1014-1026. DOI: 10.1016/j.procs.2014.05.091.

[10] Darvishi, M. T., Khani, F., y Soliman, A. A. (2007). *The numerical simulation for stiff system of ordinary differential for stiff system of ordinary differential equations*. Computers and Matematics with Applications, 54(7-8), 1055-1063. DOI: 10.1016/j.camwa.2006.12.072.

[11] Ezquerro, J. A. (2012). *Iniciación a los métodos numéricos*. España: Universidad de La Rioja, Servicio de Publicaciones.

[12] Fatunla, S. O. (2014). *Numerical methods for initial value problems in ordinary differential equations*. New York: Academic Press, Inc.

[13] Fogler, H. S. (2001). *Elementos de ingeniería de las reacciones químicas*, 3ra ed. México: Pearson Educación.

[14] Grossman, S. I., y Flores, J. J. (2012). *Álgebra lineal*, 7ma ed. México: McGraw-Hill.

[15] Hairer, E., Nørsett, S. P., y Wanner, G. (1993). *Solving ordinary differential equations I. Nonstiff problems*, 2da ed. New York: Springer-Verlag Berlin Heidelberg.

[16] Hairer, E., y Wanner, G. (1996). *Solving ordinary differential equations II: Stiff and differential-algebraic problems*, 2da ed. New York: Springer-Verlag Berlin Heidelberg.

[17] Kolman, B., y Hill, D. R. (2006). *Álgebra lineal*, 8va ed. México: Pearson Educación.

[18] Lambert, J. D. (1973). *Computational methods in ordinary differential equations*. New York: John Wiley & Sons.

[19] Laurie, D. P. (2001). *Computation of Gauss-type quadrature formulas*. Journal of Computational and Applied Mathematics, 127(1-2), 201-217. DOI: 10.1016/S0377-0427(00)00506-9.

[20] Loch, G. G. (2016). *Sistemas rígidos associados a cadeias de decaimento radioactivo* (Tesis de Maestría). IME-USP, São Paulo, Brasil.

[21] Martínez, J. J. (2001). *Polinomios ortogonales, cuadratura Gaussiana y problemas de valores propios*. Margarita mathematica en memoria de José Javier (Chicho) Guadalupe Hernández, pp. 595-606.

[22] Migoni G. (2010). *Simulación por cuantificación de sistemas stiff* (Tesis doctoral). Universidad Nacional de Rosario, Rosario, Argentina.

[23] Pistala C. (2017). *Estudio teórico y computacional de métodos numéricos para ecuaciones diferenciales ordinarias* (Tesis de pregrado). Universidad de Nariño, Pasto, Colombia.

[24] Roma, A. M., y Nós, R. L. (2012). *Tratamento numérico de equações diferenciais* (Notas de aula). IME-USP, São Paulo, Brasil.

[25] Sehnem, R., Quadros, R. S., y Buske, D. (2018). *Método numérico para solução de EDOs rígidas na modelagem de reações químicas*. Revista Mundi Engenharia, Tecnologia e Gestão (ISSN: 2525-4782), 3(2). DOI: 10.21575/25254782rmetg2018vol3n2581.

[26] Shokri, A., Mehdizadeh, M., y Molayi, M. (2019). *The new high approximation of stiff systems of first order IVPs arising from chemical reactions by k-step L-stable hybrid methods.* Iranian Journal of Mathematical Chemistry, 10(2), 181-193. DOI: 10.22052/ijmc.2018.111016.1335.

[27] Smith, J. M. (1991). *Ingeniería de la cinética química*, 6ta ed. México: CECSA.

[28] Sotolongo, A., y Jiménez, J. C. (2014). *Construcción y estudio de códigos adaptativos de linealización local para ecuaciones diferenciales ordinarias.* Revista de Matemática: Teoría y Aplicaciones, 21(1), 21-53. DOI: 10.15517/RMTA.V21I1.14136.

[29] Spijker, M. N. (1996). *Stiffness in numerical initial-value problems.* Journal of Computacional and Applied Mathrmatics, 72(2), 393-406. DOI: 10.1016/0377-0427(96)00009-X.

[30] Süli, E. (2014). *Numerical solution of ordinary differential equations* (Notas de aula). Mathematical Institute, University of Oxford, Oxford, Inglaterra.

[31] Süli, E., y Mayers, D. F. (2003). *An introduction to numerical analysis.* New York: Cambridge university press.

[32] Terán, J. (2018). *Generalización del método de Newton y sus aplicaciones* (Tesis de pregrado). Universidad de Nariño, Pasto, Colombia.

[33] Thohura, S., y Rahman, A. (2013). *Numerical approach for solving stiff differential equations: A comparative study.* J Sci Front Res Math Decision Sci, 13, 7-18.

[34] Valencia, E. (2019). *Estudio numérico para ecuaciones diferenciales ordinarias rígidas utilizando ode45, ode23, ode15s y ode23s* (Tesis de maestría). Universidad Pontifica Bolivariana, Medellín, Colombia.

[35] Zill, D. G., y Cullen, M. R. (2009). *Ecuaciones diferenciales con problemas con valores en la frontera*, 7ma ed. México: Cengage Learning.