
**MÉTODOS ITERATIVOS EFICIENTES
PARA LA RESOLUCIÓN
DE SISTEMAS NO LINEALES**

María Penkova Vassileva

Departamento de Matemática Aplicada
Universitat Politècnica de València

A la memoria de Nikolay Sukhomlin

Agradecimientos

Es un placer tener la oportunidad para poder expresar en este espacio mi gratitud a todas aquellas personas que con su valiosa colaboración contribuyeron, de una u otra manera, en el desarrollo de esta investigación y que sin su aporte hubiese sido imposible llevarlo a un feliz término.

Quedo especialmente agradecida a Dra. Alicia Cordero y Dr. Juan Ramon Torregrosa, mis directores de Tesis. Les agradezco sinceramente su confianza y todo el apoyo, por sus recomendaciones, sugerencias, su asesoramiento científico su predisposición permanente a aclarar mis dudas y sus substanciales sugerencias durante la redacción de la Tesis y paciente labor de su corrección. Pos su amistad....

A Dios por haberme dado un esposo y hijos maravillosos, por su comprensión, apoyo incondicional y estar siempre a mi lado.

Finalmente, a quienes siempre me apoyaron, mis padres.

A todos gracias!!

Santo Domingo, 2011

Índice general

Agradecimientos	v
1. INTRODUCCIÓN	3
2. CONCEPTOS PREVIOS	7
2.1. Ecuaciones y sistemas de ecuaciones no lineales	7
2.1.1. Caso unidimensional	7
2.1.2. Caso multidimensional	13
2.2. Integración numérica	21
2.2.1. Cuadratura de Newton-Cotes	21
2.2.2. Cuadratura gaussiana	22
2.3. Problemas de valor inicial	23
2.3.1. Conceptos básicos	24
2.3.2. Métodos de un sólo paso	26
2.3.3. Métodos de Taylor y Runge-Kutta	27
2.3.4. Métodos multipaso	29
3. MÉTODOS ITERATIVOS PARA ECUACIONES NO LINEALES BASADOS EN LA CUADRATURA GAUSSIANA	31
3.1. Introducción	31
3.2. Método de Newton como predictor	32
3.2.1. Descripción de los métodos	32
3.2.2. Análisis de la convergencia de los métodos	35
3.2.3. Resultados numéricos	39
3.3. Métodos iterativos basados en cuadratura de Gauss con otros predictores	44
3.3.1. Método de Traub como predictor	45
3.3.2. Método de Ostrowski como predictor	47

3.3.3. Método de la derivada congelada como predictor	50
3.4. Generalización del predictor: pseudocomposición	53
3.5. Pruebas numéricas	56
4. MÉTODOS ITERATIVOS ÓPTIMOS PARA ECUACIONES NO LINEALES	66
4.1. Introducción	66
4.2. Métodos iterativos multipaso óptimos	67
4.2.1. Familia de métodos de orden cuatro óptimos con extensiones a orden mayor	67
4.2.2. Familia de métodos combinación del método de Ostrowski y de Chun	73
4.2.3. Pruebas numéricas	86
5. MÉTODOS ITERATIVOS PARA SISTEMAS NO LINEALES	92
5.1. Diseño de métodos iterativos empleando la cuadratura de Gauss	93
5.1.1. Métodos iterativos provenientes de la cuadratura de Gauss con predictor el método de Newton	94
5.1.2. Métodos iterativos con la cuadratura de Gauss y predictor el método de Traub	101
5.1.3. Pseudocomposición para sistemas	104
5.2. Otros métodos iterativos	108
5.3. Pruebas numéricas	127
5.3.1. Determinación de la órbita preliminar de un satélite artificial	144
5.3.2. Aplicación a la resolución de problemas de frontera	149
6. PROBLEMAS DE VALOR INICIAL Y MÉTODOS ITERATIVOS	154
6.1. Métodos provenientes de la cuadratura gaussiana	155
6.1.1. Orden de precisión y consistencia de los métodos obtenidos	158
6.1.2. Cotas de error	160
6.1.3. Estabilidad	161
6.2. Métodos de resolución de problemas de valor inicial no provenientes de fórmulas de cuadratura	162
6.2.1. Método Euler-Traub	164
6.2.2. Método de derivada congelada	166
7. LÍNEAS FUTURAS Y CONCLUSIONES	170
8. ANEXOS	173

INTRODUCCIÓN

*Sans les mathématiques on ne pénètre point au fond de la philosophie.
Sans la philosophie on ne pénètre point au fond des mathématiques.
Sans les deux on ne pénètre au fond de rien.*

Leibniz

En una situación práctica, un problema matemático no es más que la modelización de un fenómeno físico. Este modelo matemático es la imagen matemática del fenómeno de estudio y se representa mediante una gran variedad de ecuaciones: trascendentes, en derivadas ordinarias o en derivadas parciales, sujetos a condiciones iniciales o de frontera (o combinación de ambas), ecuaciones integrales o integro-diferenciales, todos ellos en dimensión finita o infinita, en su mayoría no lineales. En los casos en los que no somos capaces de encontrar la solución analítica o ésta es muy costosa, optamos por obtener una aproximación de dicha solución mediante procesos numéricos que aproximan de manera eficiente y con suficiente precisión, las soluciones de problemas expresados matemáticamente. El modelo matemático implementado en algoritmos lógico-numéricos en un ordenador, permite estudiar las cualidades del proceso original uniendo las ventajas de la teoría y del experimento. Al trabajar con un modelo matemático y no con el fenómeno de estudio, se pueden estudiar y pronosticar sus propiedades de estado (ventaja teórica) en forma relativamente rápida y a bajo costo. Al mismo tiempo los algoritmos numéricos permiten, apoyándose en la potencia de cálculo de los ordenadores, verificar las cualidades del fenómeno de estudio en una forma no accesible para los enfoques teóricos (ventaja del experimento). El gran impulso de la modelización matemática se obtuvo con la aparición de los ordenadores en los años 40 - 50 del siglo XX y con las investigaciones realizadas para la creación de escudos de defensa antiaérea contra misiles nucleares en Estados Unidos y Unión Soviética. El éxito que tuvo en este caso la modelización matemática, cumpliendo con todas las expectativas, contribuyó a su desarrollo hasta los niveles actuales posicionándola en el núcleo estructural de la sociedad de la información.

Históricamente, uno de los problemas que ha estado presente en todas las civilizaciones desde la egipcia hasta nuestros días es el cálculo de raíces cuadradas. Uno de los algoritmos más común y frecuentemente usado en computadores o calculadoras es aquel que permite averiguar una aproximación de las raíces cuadradas de un número real positivo. Este método aproximado, denominado método babilónico, data de alrededor de 1700 años antes de Cristo. Aunque no existe evidencia que muestre el uso del cálculo aproximado de $\sqrt{2}$ por los babilónicos con la tablilla YBC 7289 [36] que pertenece a la Yale Babylonian Collection, sí se conoce la famosa regla de Herón acotando el radical por exceso y por defecto mediante un proceso no recurrente. En la misma época para el cálculo del perímetro de un círculo multiplicaban su diámetro por 3, lo que equivale aproximar al número π por 3 (ver [15]).

El problema de la resolución de ecuaciones y sistemas de ecuaciones no lineales figura entre los más importantes en la teoría y la práctica, no sólo de las matemáticas aplicadas, sino también de muchas ramas de las ciencias, la ingeniería, la física, la informática, la astronomía, las finanzas, . . . Un vistazo a la bibliografía y a la lista de grandes matemáticos que han trabajado en este tema muestran un alto nivel de interés contemporáneo. Aunque el rápido desarrollo de las computadoras digitales llevó a la aplicación efectiva de muchos métodos numéricos, en

la realización práctica, es necesario resolver varios problemas tales como la eficiencia computacional basado en el tiempo usado por el procesador, el diseño de métodos iterativos que posean una rápida convergencia a la solución deseada, el control de errores de redondeo, la información sobre los límites de error de la solución aproximada obtenida, indicando las condiciones iniciales de cómputo verificables que garantizan una convergencia segura, etc. Dichos problemas constituyen el punto de partida de esta memoria.

Un caso particular de este problema es la aproximación de las raíces de un polinomio. Desde tiempos muy remotos se encontraban con éxito las raíces de polinomios de primer y segundo grado. En 1540 los matemáticos Scipione, Tartaglia y Cardano resolvieron la ecuación cúbica. En 1545 Ferrari resolvió la ecuación de cuarto grado. Muchos matemáticos de los siglos posteriores trataron de resolver ecuaciones de quinto grado y superior. A principios del siglo XIX Abel y Galois demostraron que es imposible obtener solución por radicales de una ecuación de grado mayor que cuatro. En consecuencia, para calcular las raíces de polinomios de grado mayor que cuatro se usan técnicas numéricas. A partir de este momento, la construcción de métodos numéricos para resolver ecuaciones no lineales ha atraído la atención de matemáticos puros y aplicados.

Los métodos numéricos consisten en hallar, mediante un proceso iterativo, y a partir de una aproximación inicial x_0 , una sucesión $\{x_n\}$ de aproximaciones a la solución de la ecuación, con la exigencia de que exista $\lim_{n \rightarrow \infty} x_n = \xi$, siendo ξ la solución de la ecuación no lineal, bajo ciertos criterios de error. Una vez que el algoritmo está propiamente formulado, el siguiente objetivo que se plantea es conocer exactamente cuáles son las condiciones bajo las que el esquema iterativo aproxima una solución del problema considerado, es decir, bajo qué condiciones la sucesión de aproximaciones generada es convergente a una solución de la ecuación correspondiente. Los resultados de convergencia que se establecen pueden ser de tres tipos: local, semilocal y global, dependiendo de cuáles sean las condiciones que se impongan; sobre la solución, sobre la aproximación inicial x_0 , o simplemente sobre la función que define la ecuación, respectivamente. En esta memoria analizaremos la convergencia local de los métodos desarrollados.

Consideremos el problema de encontrar un cero de la función de naturaleza no lineal $f : I \subseteq \mathbb{R} \rightarrow \mathbb{R}$, es decir, una solución $\xi \in I$ de la ecuación no lineal

$$f(x) = 0. \quad (1.1)$$

En la actualidad existen numerosos métodos iterativos para resolver la ecuación no lineal (1.1). Esta solución puede ser obtenida como un punto fijo de alguna función $g : \mathbb{R} \rightarrow \mathbb{R}$ mediante el método iterativo de punto fijo $x_{k+1} = g(x_k)$, $k = 0, 1, \dots$, donde x_0 es la aproximación inicial. El método más conocido por ser muy simple y efectivo es el método de Newton, dado por

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \dots \quad (1.2)$$

Su generalización a sistemas de ecuaciones fue propuesta por Ostrowski [72] y en operadores en espacios de Banach por L.V.Kantorovic [53]. Así, este método proporciona una herramienta poderosa para investigaciones teóricas y numéricas sobre ecuaciones definidas mediante operadores no lineales.

Otras cuestiones que se plantean sobre el comportamiento de un esquema iterativo son la velocidad de convergencia con la que la sucesión converge a una solución y el error cometido al aproximar esa solución. Existen distintos indicadores para medir la velocidad de convergencia de una sucesión como son el orden de convergencia teórico y la tasa de convergencia práctica. Al estudiar un método iterativo es muy importante considerar dos aspectos: la velocidad de convergencia y el coste del mismo. Los métodos de un sólo paso como, por ejemplo, el método de Newton, son muy eficaces, pero aumentar su velocidad de convergencia implica evaluar sucesivas derivadas de la función no lineal, de orden creciente, por lo que su utilidad en problemas prácticos se ve limitada.

Como consecuencia de la búsqueda de variantes del método clásico de Newton para resolver ecuaciones no lineales con una convergencia acelerada y un número reducido de operaciones o evaluaciones funcionales en cada paso del proceso iterativo, se han desarrollado los métodos multipaso. Estos métodos pertenecen a la clase de métodos más poderosos que superan las limitaciones de los métodos de un punto respecto al orden de convergencia y la eficiencia computacional. Ellos nacieron en la década de 1960 pero su rápido desarrollo ha comenzado en la primera década del siglo XXI. La clase más importante de los métodos multipaso son los métodos óptimos en el sentido de la conjetura de Kung-Traub.

Generalmente, el aumento del orden de un método iterativo conlleva un aumento del número de evaluaciones funcionales por paso. El índice de eficiencia es una medida del equilibrio entre las dos cantidades: el número de evaluaciones funcionales por iteración y el orden de convergencia de cualquier método multipaso sin memoria. Por lo tanto, el orden óptimo de un método con tres evaluaciones funcionales por paso es 4. La familia de métodos de King [55], de la que el método de Ostrowski [72] es un caso particular y el método Jarratt [51], junto con algunas de las familias de métodos multipaso estudiados extensamente en el libro de Traub [81] son métodos óptimos de cuarto orden, ya que sólo realizan tres evaluaciones funcionales por paso. En los últimos años, como muestra la amplia bibliografía, ha aumentado de nuevo el interés en la búsqueda de métodos multipaso con el fin de conseguir una convergencia de orden óptimo y así una mejor eficiencia.

En el presente, se sigue investigando en el tema y progresivamente surgen nuevos métodos iterativos que modifican los métodos clásicos con el fin de acelerar la convergencia o para reducir el número de operaciones y las evaluaciones funcionales en cada paso del proceso iterativo. Se han desarrollado una gran cantidad de técnicas numéricas para la aproximación de soluciones de ecuaciones no lineales, basadas en el método de punto fijo, en particular aquellas que modifican el método clásico de Newton.

El objetivo general de esta memoria radica en la búsqueda de nuevos y eficientes métodos iterativos para ecuaciones y sistemas de ecuaciones no lineales. El origen es el trabajo realizado por Weerakoon y Fernando ([82]) en el que desarrollan en dimensión uno la variante del método de Newton que utiliza la fórmula de cuadratura trapezoidal, consiguiendo orden de convergencia tres. Özban en [73] amplió esta idea, y obtuvo algunos métodos nuevos con convergencia de tercer orden. Por otra parte, dichos métodos son casos particulares de la familia de variantes del método de Newton de orden tres definida por M. Frontini y E. Sormani en [38], utilizando una fórmula de cuadratura interpolatoria genérica de nodos equiespaciados. Así, en primer lugar, basándonos en la idea de M. Frontini y E. Sormani, utilizamos la fórmula de cuadratura gaussiana genérica y desarrollamos en el Capítulo 3 un conjunto de familias de métodos iterativos para ecuaciones. Todos los métodos del conjunto son de tipo predictor-corrector donde la predicción se realiza inicialmente con el método de Newton. Demostramos que su orden de convergencia es tres bajo ciertas condiciones impuestas a los polinomios ortogonales que definen la familia de cuadratura gaussiana correspondiente y cinco dependiendo del comportamiento en la solución de la derivada segunda de la función que define la ecuación no lineal. Buscando mejorar el orden de convergencia del conjunto de métodos iterativos desarrollado, modificamos los algoritmos obtenidos utilizando otros métodos de predicción que superan el orden de convergencia dos del predictor Newton. Inicialmente usamos como predictor el método de Traub ([81]) que tiene orden de convergencia tres, obteniendo un conjunto de familias de métodos iterativos de orden cinco (u once bajo ciertas condiciones impuestas a los polinomios ortogonales y la derivada segunda de la función que define la ecuación). Posteriormente, usamos como predictores los métodos iterativos de Ostrowski [72] y de Chun [11] cuyo orden de convergencia es cuatro, recibiendo en ambos casos el conjunto de familias de métodos iterativos de orden 6 y 11 bajo las mismas condiciones.

Siguiendo la tendencia actual en el desarrollo de nuevos métodos iterativos, en el Capítulo 4 se han desarrollado varios métodos de orden de convergencia alto e índice de eficiencia óptimos según la conjetura de Kung-Traub.

El problema de resolver un sistema de ecuaciones no lineales se evita en la medida de lo posible. Por lo general, se aproxima el sistema no lineal mediante un sistema de ecuaciones lineales. Cuando esto no es satisfactorio, el problema debe enfrentarse de forma directa. El punto de vista más directo consiste en adaptar los métodos que aproximan las soluciones de una sola ecuación no lineal en una variable, la cual se reemplaza por un problema vectorial que incorpora todas las variables. De aquí surge la mayor dificultad para obtener métodos iterativos nuevos para sistemas no lineales, ya que no siempre los métodos de ecuaciones no lineales son extensibles a sistemas.

El Capítulo 5 está dedicado a la búsqueda de nuevos métodos iterativos para sistemas de ecuaciones no lineales. La extensión de las variantes del método de Newton descritas por Weerakoon y Fernando en [82], por Özban en [73] y Gerlach en [39], a las funciones de varias variables se han desarrollado en [17], [18] y [38], basado en el uso de fórmulas de cuadratura de la interpolación, y también en [68], incluyendo la segunda derivada parcial de la función no lineal o su estimación. Otra técnica de aceleración conocida es la composición de dos métodos iterativos de órdenes de p_1 y p_2 , respectivamente, y la obtención de un método de orden $p_1 p_2$ ([81]). Por lo general, para aumentar el orden de convergencia se necesitan nuevas evaluaciones de la matriz Jacobiana y de la función no lineal. En este sentido, usamos algunos de los métodos desarrollados para ecuaciones en el

Capítulo 3 y los adaptamos para sistemas, mientras que otros son desarrollados especialmente para sistemas.

Los Capítulos 3, 4 y 5 finalizan con secciones en las que se presentan pruebas numéricas que confirman los resultados teóricos. Los resultados numéricos obtenidos en la aplicación de los nuevos métodos que surgen y su comparación con los ya existentes, se han obtenido en MATLAB 7.10 (64-bit), en la computadora HP Pavilion dv6000, Intel(R) Core(TM)2 Duo CPU, 4GB RAM, todos con un esquema común: cada iteración se obtiene de la anterior por medio de la expresión iterativa: $x_{k+1} = x_k - A^{-1}b$, donde $x_k \in \mathbb{R}$ para ecuaciones no lineales y $x^{(k+1)} = x^{(k)} - A^{-1}b$, donde $x^{(k)} \in \mathbb{R}^n$ para sistemas de ecuaciones no lineales. Por otro lado, en algunos de los métodos obtenidos de alto orden de convergencia para ecuaciones no lineales, el análisis de su orden de convergencia involucra un volumen de cálculo algebraico considerable. En este caso se ha usado Mathematica 7 como manipulador simbólico.

El Capítulo 6 está dedicado a la relación entre los métodos iterativos de resolución de ecuaciones no lineales y problemas de valor inicial. Las fórmulas de cuadratura en general, y las de Gauss en particular, nos permiten determinar numéricamente las soluciones, tanto de ecuaciones o sistemas de ecuaciones no lineales (generando métodos iterativos como los descritos en la presente memoria), como de problemas de valor inicial, lineales o no. De este modo, las fórmulas de cuadratura constituyen el nexo de unión entre ambos problemas. Grau-Sánchez et al. en [41] y [42] describe la forma de "trasladar" métodos de resolución de problemas de valor inicial para obtener métodos nuevos de resolución de ecuaciones no lineales. Estas ideas se pueden aplicar a métodos sencillos, como Euler, o a otros más complejos, como los de Adams o a métodos implícitos. Nos planteamos en este punto si este proceso es reversible: dado un método iterativo de resolución de ecuaciones no lineales (no proveniente de una fórmula de cuadratura), ¿es posible encontrar un método de resolución de problemas de valor inicial? Dados diferentes métodos numéricos de ecuaciones no lineales, de los cuales conocemos el índice de eficiencia, ¿se mantiene la relación de eficiencia cuando trasladamos estos métodos a problemas de valor inicial?

En el Capítulo 7 presentamos las conclusiones sobre los resultados obtenidos en esta memoria que nos permiten plantear futuras líneas de investigación.

Terminamos esta memoria con el listado de referencias que, en mayor o menor medida, han sido utilizadas en el desarrollo de la misma.

CONCEPTOS PREVIOS

En este capítulo vamos a introducir las herramientas matemáticas que utilizaremos a lo largo de toda la memoria. En primer lugar, consideramos los métodos iterativos de punto fijo para determinar las raíces de una ecuación no lineal y de un sistema no lineal de n ecuaciones con n incógnitas. Tras el planteamiento del problema, se discute la convergencia y velocidad de convergencia de los métodos iterativos clásicos, introduciendo los términos orden de convergencia, índice de eficiencia e índice de eficiencia computacional. Asimismo, se describen las técnicas que se utilizarán en las demostraciones de la convergencia de los nuevos métodos desarrollados, el concepto de método óptimo para ecuaciones y su extensión, a modo de conjetura, para sistemas. En la primera sección se analizan por separado el caso unidimensional, $n = 1$, y el caso multidimensional, $n > 1$. En la siguiente sección se describen las fórmulas de cuadratura utilizadas en los capítulos 3, 4 y 5. Finalizamos este capítulo introduciendo los elementos necesarios para describir y resolver un problema de valor inicial, haciendo especial énfasis en aquellos métodos numéricos que permiten estimar su solución y que pueden deducirse empleando fórmulas de cuadratura. Posteriormente, se describen algunos métodos multipunto más complejos.

2.1. Ecuaciones y sistemas de ecuaciones no lineales

2.1.1. Caso unidimensional

Encontrar de manera eficiente los ceros de una ecuación no lineal de variable real de la forma

$$f(x) = 0, \tag{2.1}$$

es un problema antiguo y muy interesante en el análisis numérico con muchas aplicaciones en diversos campos de la ingeniería o de la matemática aplicada. En los últimos años, se han desarrollado una gran cantidad de diferentes métodos iterativos para resolver dicha ecuación no lineal (2.1). La esencia de estos métodos es la siguiente: si se conoce un entorno suficientemente pequeño, que contiene una raíz única ξ de la ecuación (2.1), seleccionamos un punto x_0 - estimación inicial a la raíz - lo suficientemente cerca de la raíz buscada ξ y con la ayuda de algunas relaciones de recurrencia, construimos una sucesión de puntos $x_1, x_2, \dots, x_k, \dots$, que converge a la raíz ξ . La convergencia de la sucesión se garantiza con la elección apropiada de la función g y la aproximación inicial x_0 . Eligiendo de distintas maneras la función g ,

$$x = g(x) \tag{2.2}$$

que depende de $f(x)$ y puede que también de sus derivadas, podemos obtener diferentes métodos iterativos, cuya expresión iterativa sería

$$x_{k+1} = g(x_k), \quad k = 0, 1, \dots \tag{2.3}$$

El método descrito por la función de iteración $g : I \subseteq \mathbb{R} \rightarrow \mathbb{R}$ según (2.3) con x_0 dado, concentra una gran cantidad de esquemas iterativos. Éstos difieren entre sí por la forma de definir la función de iteración g . El dominio de definición de g también puede elegirse en otros espacios, como \mathbb{R}^n por ejemplo, lo que da origen a métodos de resolución de sistemas de n ecuaciones con n incógnitas.

Para el estudio de la convergencia de los métodos iterativos, así como para probar la existencia de la raíz de la ecuación $f(x) = 0$ se usan extensamente las llamadas funciones contractivas.

Definición 2.1.1 Consideramos \mathbb{R} un espacio métrico y $g(x)$ una función definida en este espacio, que transforma \mathbb{R} en si mismo: $g : I \subset \mathbb{R} \rightarrow \mathbb{R}$. Entonces, g será una función lipschitziana si existe un número positivo $L > 0$, tal que para cualquier par de elementos $x, y \in I$, se cumple la desigualdad

$$|g(x) - g(y)| \leq L|x - y|. \quad (2.4)$$

Hay un tipo de funciones lipschitzianas de interés especial - las funciones contractivas - que son funciones lipschitzianas para las cuales L toma valores $0 < L < 1$.

Definición 2.1.2 Un punto fijo de una función $g(x)$ es un número real ξ tal que $\xi = g(\xi)$ y representa geoméricamente el punto de intersección de la curva $y = g(x)$ con la recta $y = x$.

Teorema 2.1.1 (Teorema de punto fijo, Kolmogorov y Fomin [56]). Toda función contractiva definida en el espacio métrico completo de los números reales $f : \mathbb{R} \rightarrow \mathbb{R}$ tiene un único punto fijo.

En la práctica si g tiene un punto fijo ξ , entonces $f(x) = g(x) - x$ tiene un cero en ξ . Del mismo modo, si f tiene una raíz ξ , entonces $g(x) = x - f(x)$ tiene un punto fijo en ξ .

Los siguientes resultados aseguran la existencia y unicidad del punto fijo, bajo ciertas condiciones.

Teorema 2.1.2 (Existencia y unicidad del punto fijo, Burden y Faires [8]). Sea $g \in C[a, b]$ tal que $g : [a, b] \rightarrow [a, b]$. Entonces,

- a) g tiene un punto fijo en $[a, b]$.
Si, además, g' existe, entonces
- b) existe $L \in (0, 1)$ tal que $|g'(x)| \leq L$ para todo $x \in (a, b)$;
- c) el punto fijo de g en $[a, b]$ es único.

El resultado que sigue establece bajo qué hipótesis está asegurada la convergencia global de (2.2) y proporciona una cota del error.

Teorema 2.1.3 (Ostrowski [72]) Supongamos que $g : [a, b] \rightarrow \mathbb{R}$ es contractiva en el intervalo $[a, b]$ tal que $g([a, b]) \subseteq [a, b]$. Sea L la constante de Lipschitz para g en $[a, b]$. Bajo estas condiciones, (2.3) es globalmente convergente en $[a, b]$ al único punto fijo $\xi \in (a, b)$ de g . Además, se tienen las siguientes estimaciones de error:

$$|x_k - \xi| \leq L^k |x_0 - \xi| \quad \text{para todo } k \geq 0, \quad x_0 \in [a, b] \quad (2.5)$$

$$|x_k - \xi| \leq \frac{L^k}{1 - L} |x_1 - x_0| \quad \text{para todo } k \geq 0, \quad x_0 \in [a, b]. \quad (2.6)$$

Demostración: Las hipótesis del Teorema 2.1.3 garantizan la existencia de una solución única ξ de $x = g(x)$ según el Teorema 2.1.2. Para probar la convergencia de la sucesión $\{x_k\}_{k \geq 0}$, vamos a estimar la diferencia $|x_k - \xi|$. Por definición, $x_k - \xi = g(x_{k-1}) - \xi = g(x_{k-1}) - g(\xi)$ y, por tanto, por la condición de Lipschitz, $|x_k - \xi| \leq L|x_{k-1} - \xi|$. Aplicando consecutivamente la misma desigualdad, obtenemos

$$|x_k - \xi| \leq L^k |x_0 - \xi|.$$

Dado que $0 < L < 1$, $\lim_{k \rightarrow \infty} L^k = 0$, de donde se deduce que $\lim_{k \rightarrow \infty} |x_k - \xi| = 0$, es decir, $\lim_{k \rightarrow \infty} x_k = \xi$, con que completamos la primera parte de la prueba.

Siguiendo el procedimiento que nos ha permitido deducir (2.5) podemos obtener que

$$|x_{k+1} - x_k| \leq L^k |x_1 - x_0|, \quad \forall k \geq 0.$$

Sea k un entero positivo fijo y $m > k$ cualquiera.

$$x_m - x_k = x_m - x_{m-1} + x_{m-1} - x_{m-2} + \cdots + x_{k+1} - x_k.$$

Por la desigualdad triangular

$$\begin{aligned} |x_{m+1} - x_k| &\leq |x_{m+1} - x_m| + |x_m - x_{m-1}| + \cdots + |x_{k+1} - x_k| \\ &\leq L^m |x_1 - x_0| + L^{m-1} |x_1 - x_0| + \cdots + L^k |x_1 - x_0| \\ &= (L^m + L^{m-1} + \cdots + L^k) |x_1 - x_0| \\ &= L^k (1 + L + \cdots + L^{m-k}) |x_1 - x_0| \\ &= L^k \frac{1}{1-L} |x_1 - x_0|. \end{aligned}$$

Tomando límites para m , se tiene

$$|\xi - x_k| \leq L^k \frac{1}{1-L} |x_1 - x_0|$$

con lo que se ha completado la demostración. \square

Observación: De la cota del error (2.5) se deduce que (2.3) converge más rápidamente cuanto más próxima a cero sea L .

Un concepto muy importante de los métodos iterativos es su orden de convergencia, que proporciona una medida de la velocidad de convergencia de la sucesión de iterados.

Definición 2.1.3 Sea $\{x_k\}_{k \geq 0}$ una sucesión de números reales tal que $\lim_{k \rightarrow \infty} x_k = \xi$. La convergencia se llama:

✓ lineal, si existen M , $0 < M < 1$ y k_0 tales que $|x_{k+1} - \xi| \leq M|x_k - \xi|$ para todo $k \geq k_0$;

✓ cuadrática, si existen $M > 0$, y k_0 tales que $|x_{k+1} - \xi| \leq M|x_k - \xi|^2$ para todo $k \geq k_0$;

✓ cúbica, si existen $M > 0$ y k_0 tales que $|x_{k+1} - \xi| \leq M|x_k - \xi|^3$ para todo $k \geq k_0$;

.....

✓ de orden p , si existen $M > 0$ y k_0 tales que $|x_{k+1} - \xi| \leq M|x_k - \xi|^p$ para todo $k \geq k_0$.

Denotamos por $e_k = x_k - \xi$ el error de la k -ésima iteración. La ecuación $e_{k+1} = Me_k^p + O[e_k^{p+1}]$, se llama ecuación de error donde p es el orden de la convergencia. En la práctica, como el límite ξ es desconocido, se analizan para cada p el comportamiento de los cocientes:

$$\frac{|x_{k+1} - x_k|}{|x_k - x_{k-1}|^p}, \quad k = 1, 2, \dots$$

donde $p = 1, 2, 3, \dots$, y se llama tasa o razón de convergencia.

◦ Si $p = 1$ y la razón de la convergencia eventualmente tiende a C_L , $0 < C_L < 1$, se dice que la sucesión $\{x_k\}_{k \geq 0}$ converge linealmente a ξ .

◦ Si $p = 2$ y existe C_C , $C_C > 0$, tal que la razón de la convergencia eventualmente tiende a C_C , se dice que la sucesión $\{x_k\}_{k \geq 0}$ es cuadráticamente convergente a ξ .

◦ De manera similar para $p = 3$ (convergencia cúbica C_{CU}) y así sucesivamente.

Observación: En el proceso de la demostración del Teorema 2.1.2 se prueba que $|x_k - \xi| \leq L|x_{k-1} - \xi|$, $k \geq 1$, con $0 \leq L < 1$, y por tanto, el método (2.2) tiene orden de convergencia al menos lineal. De hecho, en muchas ocasiones es exactamente lineal. Así, por ejemplo, si bajo las condiciones del Teorema 2.1.2 suponemos que además existe $g'(x)$ para todo $x \in (a, b)$, y la función g' es continua en (a, b) entonces para todo $k \geq 1$,

$|x_k - \xi| = |g(x_{k-1}) - g(\xi)| = |g'(\zeta)(x_{k-1} - \xi)|$, con ζ un punto entre x_k y ξ ; con lo que, si suponemos que $x_k \neq \xi$ para todo $k \geq 0$, tenemos $\frac{|x_k - \xi|}{|x_{k-1} - \xi|} = |g'(\zeta)|$ para todo $k \geq 0$, y en consecuencia, como $x_k \rightarrow \xi$ implica que $\zeta \rightarrow \xi$ cuando $k \rightarrow \infty$, obtenemos $\lim_{k \rightarrow \infty} \frac{|x_k - \xi|}{|x_{k-1} - \xi|} = |g'(\xi)|$, de modo que si $g'(\xi) \neq 0$, la convergencia del método (2.2) a la solución ξ es exactamente lineal.

Sea $g : [a, b] \rightarrow \mathbb{R}$ continua, tal que $g([a, b]) \subseteq [a, b]$, y existe la derivada $g'(x)$ en todo punto de (a, b) . Supongamos que $g'(x) \leq L < 1$, para todo $x \in (a, b)$. En tal caso, sabemos que el método (2.3) es convergente a la única solución ξ de la ecuación de punto fijo (2.2). Además, suponiendo que x_{k-1} y x_k son distintas, es decir, que no se ha llegado a ξ , se tiene

$$x_{k+1} - x_k = g(x_k) - g(x_{k-1}) = g'(\zeta)(x_k - x_{k-1}),$$

con ζ un punto intermedio entre x_{k-1} y x_k .

En estas condiciones, si $g'(x) > 0$ para todo $x \in (a, b)$, de la igualdad precedente tenemos

$$\operatorname{sgn}(x_{k+1} - x_k) = \operatorname{sgn}(x_k - x_{k-1}),$$

$$x_{k+1} - x_k < x_k - x_{k-1},$$

con lo cual, geoméricamente, el proceso de acercamiento de x_k a ξ es convergente en escalera.

Análogamente, si $g'(x) < 0$ para todo $x \in (a, b)$, se tiene

$$\operatorname{sgn}(x_{k+1} - x_k) \neq \operatorname{sgn}(x_k - x_{k-1}),$$

$$x_{k+1} - x_k > x_k - x_{k-1},$$

con lo cual, geoméricamente, el proceso de acercamiento de x_k a ξ es convergente en espiral.

El teorema siguiente generaliza estos resultados.

Teorema 2.1.4 (Convergencia local del método de punto fijo, Ostrowski [72]). *Supongamos que son dados $x_0 \in \mathbb{R}$, $\rho > 0$ y $g : [x_0 - \rho, x_0 + \rho] \rightarrow \mathbb{R}$ una función contractiva en el intervalo $[x_0 - \rho, x_0 + \rho]$. Sea $L \in (0, 1)$ la constante de Lipschitz para g en $[x_0 - \rho, x_0 + \rho]$, y supongamos que $|x_0 - g(x_0)| \leq (1 - L)\rho$, entonces,*

i) *existe $\lim_{k \rightarrow \infty} x_k = \xi$, el número ξ es punto fijo de (2.2), es decir, $g(\xi) = \xi$, y además, $|x_k - \xi| \leq L^k \rho$ para todo $k \geq 0$,*

ii) *ξ es la única solución en el intervalo $[x_0 - \rho, x_0 + \rho]$ de $g(x) = x$.*

Sea la función $f : I \subset \mathbb{R} \rightarrow \mathbb{R}$ dos veces continuamente derivable en el intervalo $I = [a, b]$ y $f'(x) \neq 0$ para todo $x \in I$. Si la ecuación $f(x) = 0$ tiene una única solución $x = \xi$ en (a, b) y la función de $g(x)$ es continua en un entorno de $x = \xi$, entonces la ecuación

$$g(x) = x - \varphi(x)f(x) \tag{2.7}$$

también tiene un punto fijo en $x = \xi$, para cualquier función $\varphi \in C^1([a, b])$.

Se puede determinar una función $\varphi(x)$ de modo que la sucesión generada por (2.7) converja a ξ cuadráticamente. A vista del Teorema 2.1.4, parece natural pedir que se tenga $g \in C^2[\xi - \rho, \xi + \rho]$ (con $\rho > 0$ tal que $[\xi - \rho, \xi + \rho] \subseteq [a, b]$ y $g'(\xi) = 0$). Entonces, derivando obtenemos

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

y teniendo en cuenta que $f(\xi) = 0$, deducimos que φ debe satisfacer la condición: $\varphi(\xi)f'(\xi) = 1$. Si $f'(x) \neq 0$ para cualquier $x \in [a, b]$ un enfoque razonable es suponer que $\varphi(x) = \frac{1}{f'(x)}$ y entonces, el procedimiento natural para conseguir la convergencia cuadrática será

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad k = 0, 1, 2, \dots \tag{2.8}$$

Esta fórmula iterativa se conoce como el método de Newton.

Dicho método tiene una interpretación geométrica sencilla, que en realidad está en el origen histórico del mismo. En efecto, en cada etapa k , el valor x_{k+1} corresponde a la abscisa del punto de corte con el eje OX de la recta tangente a la curva $y = f(x)$ en el punto $(x_k, f(x_k))$. Esta interpretación geométrica justifica que este método reciba también el nombre de método de la tangente.

Una condición para conseguir convergencia de orden dos en el método (2.8) es que el valor de la estimación inicial esté “suficientemente cerca” de la solución ξ . El siguiente resultado especifica explícitamente un intervalo en el que puede ser tomada la primera aproximación y describe las condiciones que garantizan la convergencia cuadrática del método (2.8).

Teorema 2.1.5 (Convergencia local del método de Newton [8]). *Supongamos que $f : \mathbb{R} \rightarrow \mathbb{R}$ es una función de clase $C^2([a, b])$ en un entorno del número ξ para el cual $f(\xi) = 0$, $f'(\xi) \neq 0$. Si x_0 se selecciona suficientemente cerca de ξ , entonces, existe un δ tal que las iteraciones del método de Newton generan una sucesión $\{x_k\}_{k \geq 0}$ que converge a ξ para cualquier aproximación inicial $x_0 \in [\xi - \delta, \xi + \delta]$ con orden de convergencia $p = 2$:*

$$\lim_{k \rightarrow \infty} \frac{|\xi - x_{k+1}|}{(\xi - x_k)^2} = \frac{1}{2} \left| \frac{f''(\xi)}{2f'(\xi)} \right|.$$

A continuación estudiaremos la convergencia global del método de Newton en el intervalo $[a, b]$. Se tiene:

Teorema 2.1.6 (Convergencia global del método de Newton). *Sea $f \in C^2[a, b]$ tal que verifica*

- i) $f(a)f(b) < 0$,
- ii) $f'(x) \neq 0$, para $x \in [a, b]$,
- iii) $f''(x) \leq 0$ ó $f''(x) \geq 0$, $\forall x \in [a, b]$,
- iv) Si c denota el extremo de $[a, b]$ para el que $|f'(x)|$ es más pequeño, entonces

$$\left| \frac{f(c)}{f'(c)} \right| \leq b - a$$

Entonces, el método de Newton converge a la única solución $\xi \in [a, b]$ de $f(x) = 0$, para cualquiera elección de $x_0 \in [a, b]$. Además, la convergencia es al menos cuadrática.

Demostración: La condición (i) afirma que $f(a)$ y $f(b)$ tienen signos distintos, por lo que, por el Teorema de Bolzano, la ecuación $f(x) = 0$ tiene al menos una solución. En virtud de la condición (ii) esta solución es única y la denotamos por ξ . La condición (iii) indica que la función f es cóncava o convexa. Finalmente, la condición (iv) nos permite afirmar que la tangente a la curva en c corta al eje x en el intervalo $[a, b]$. Para demostrar el resto de la tesis podemos considerar 4 casos:

- (a) $f(a) < 0$, $f(b) > 0$, $f''(x) \leq 0$ ($c = b$);
- (b) $f(a) > 0$, $f(b) < 0$, $f''(x) \geq 0$ ($c = b$);
- (c) $f(a) < 0$, $f(b) > 0$, $f''(x) \geq 0$ ($c = a$);
- (d) $f(a) > 0$, $f(b) < 0$, $f''(x) \leq 0$ ($c = a$).

Los casos (b) y (d) se pueden reducir a los casos (a) y (c), respectivamente, considerando la función $-f$ en lugar de f (este cambio no afecta a la sucesión $\{x_k\}$). El caso (c) se reduce al caso (a) sin más que reemplazar x por $-x$ (con este cambio cambia se obtiene $\{-x_k\}$ en lugar de $\{x_k\}$, y la solución $-\xi$ en lugar de ξ). Por tanto, es suficiente demostrar el Teorema en el caso (a).

Supongamos en primer lugar que $a \leq x_0 \leq \xi$. Dado que $f(x_0) \leq 0$, se tiene

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \geq x_0.$$

Veamos que $x_k \leq \xi$ y $x_{k+1} \geq x_k$, para todo k por inducción sobre k . Para $k = 0$ el resultado es cierto. Supongamos que lo es para k y veámoslo para $k + 1$. Si $x_k \leq \xi$, por el teorema del valor medio

$$-f(x_k) = f(\xi) - f(x_k) = (\xi - x_k)f'(x_k^*),$$

donde $x_k \leq x_k^* \leq \xi$. Como $f''(x) \leq 0$, f' es decreciente y por tanto, $f'(x_k^*) \leq f'(x_k)$. Así,

$$-f(x_k) \leq (\xi - x_k)f'(x_k)$$

y

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \leq x_k + (\xi - x_k) = \xi.$$

Por lo tanto, $f(x_{k+1}) \leq 0$ y $x_{k+2} = x_{k+1} - \frac{f(x_{k+1})}{f'(x_{k+1})} \geq x_{k+1}$, lo que completa el proceso de inducción.

Es conocido que toda sucesión monótona acotada tiene límite, luego $\lim_{k \rightarrow \infty} x_k = m \leq \xi$. Tomando límites en la expresión iterativa del método de Newton y teniendo en cuenta que f y f' con continuas, se tiene

$$m = m - \frac{f(m)}{f'(m)}$$

luego $f(m) = 0$ y por tanto $m = \xi$.

Supongamos ahora que $\xi < x_0 \leq b$. De nuevo, por el Teorema del valor medio, $f(x_0) = f'(x_0^*)(x_0 - \xi)$, donde $\xi < x_0^* < x_0$, y puesto que f' es decreciente, $f(x_0) \geq (x_0 - \xi)f'(x_0)$. Se sigue que

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \leq x_0 - (x_0 - \xi) = \xi.$$

Por otra parte, $f(x_0) = f(b) - (b - x_0)f'(x_0')$ donde $x_0 \leq x_0' \leq b$, y de aquí

$$f(x_0) \leq f(b) - (b - x_0)f'(b).$$

A partir de la condición *iv*) tenemos

$$\begin{aligned} x_1 &= x_0 - \frac{f(x_0)}{f'(x_0)} \geq x_0 - \frac{f(x_0)}{f'(b)} \geq x_0 - \frac{f(b)}{f'(b)} + (b - x_0) \\ &\geq x_0 - (b - a) + (b - x_0) = a. \end{aligned}$$

Por tanto, $a \leq x_1 \leq \xi$ y siguiendo los mismos pasos que para el caso $a \leq x_0 \leq \xi$ concluimos que la sucesión $\{x_k\}$ converge a ξ . Veamos ahora que la convergencia es al menos cuadrática. A partir de la fórmula iterativa

$$f(x_k) + f'(x_k)(x_{k+1} - x_k) = 0, \quad \forall k \geq 0.$$

Por otra parte, utilizando el desarrollo de Taylor de f hasta orden 2

$$0 = f(\xi) = f(x_k) + f'(x_k)(\xi - x_k) + \frac{1}{2}f''(\zeta_k)(\xi - x_k)^2, \quad \forall k \geq 0,$$

donde ζ_k es un punto situado entre ξ y x_k . Restando las dos últimas expresiones, resulta:

$$f'(x_k)(\xi - x_{k+1}) + \frac{1}{2}f''(\zeta_k)(\xi - x_k)^2 = 0, \quad \forall k \geq 0.$$

Si $p = \min_{x \in [a, b]} |f'(x)|$ y $q = \max_{x \in [a, b]} |f''(x)|$, de la expresión anterior obtenemos:

$$|x_k - \xi| \leq \frac{q}{2p}|x_k - \xi|^2, \quad \forall k \geq 0,$$

lo que demuestra la convergencia cuadrática del método. \square

Observación: La fórmula iterativa de Newton (2.8) es aplicable no sólo para encontrar las raíces reales de una ecuación $f(x) = 0$, sino también sus raíces complejas. En el campo complejo x y, por consiguiente, $f(x)$ son números complejos. Partiendo de la base de que es posible definir la derivada sobre funciones de variable compleja del mismo modo que se realiza con funciones de variable real, el método de Newton se puede expandir al campo complejo, con aproximación inicial x_0 también un número complejo. Cabe destacar que cualquier número en el campo complejo se puede expresar como una combinación de dos números reales, $z = a + bi$, por lo que mientras que un punto en el campo real se puede localizar en una recta, en el campo complejo se debe localizar en un plano, lo que conlleva una geometría más complicada. El método iterativo comenzará con un punto z_0 y generará una sucesión de puntos z_k de los cuales se espera que converjan a la raíz compleja. La representación gráfica más habitual consiste en generar las regiones de atracción en el mismo sentido que en el campo real, es decir, a cada raíz se le asigna un color y a cada punto del plano se le aplica el proceso iterativo, obteniendo la raíz compleja a la que converge la sucesión $\{z_k\}$. Se pinta el punto del color que le corresponda a dicha raíz.

Con el fin de comparar los diferentes métodos, se utiliza el índice de eficiencia, $I = p^{1/d}$ (ver [72]), donde recordamos que p es el orden de convergencia y d es el número total de nuevas evaluaciones funcionales (por iteración) que requiere el método. Este es el índice más utilizado, pero no es el único. En [75] Romero et al. utilizan un índice operacional definido como $C = p^{1/op}$, donde op es el número de operaciones por iteración. Además uno de los objetivos de esta memoria es la búsqueda de métodos iterativos óptimos según en los que se obtiene convergencia de orden 2^{d-1} usando d evaluaciones funcionales en cada iteración y su índice de eficiencia computacional es $2^{\frac{d-1}{d}}$ según la conjetura de Kung-Traub [64]:

Conjetura de Kung-Traub: *El orden de convergencia de cualquier método multipaso sin memoria no puede superar 2^{d-1} (llamado orden óptimo), donde d es el número de evaluaciones funcionales por iteración, con índice de eficiencia $2^{(d-1)/d}$ (llamado índice óptimo).*

Por lo tanto, el orden 2^{d-1} es el orden óptimo.

El análisis de la convergencia de los métodos diseñados en esta memoria se llevará a cabo fundamentalmente mediante desarrollos de Taylor. Si bien existen otras técnicas de demostración, los desarrollos de Taylor permiten determinar el orden de convergencia de manera relativamente sencilla.

2.1.2. Caso multidimensional

Consideremos un sistema de ecuaciones no lineales

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) = 0 \\ f_2(x_1, x_2, \dots, x_n) = 0 \\ \vdots \\ f_n(x_1, x_2, \dots, x_n) = 0 \end{cases}$$

donde las funciones f_i , $i = 1, 2, \dots, n$, son funciones de $\Omega \subseteq \mathbb{R}^n$ en \mathbb{R} . Este sistema se puede expresar de la forma

$$F(x) = 0, \quad (2.9)$$

donde F es una función vectorial de \mathbb{R}^n en \mathbb{R}^n que tiene como funciones coordenadas

$$F(x) = (f_1(x), \dots, f_n(x))^T, \quad (2.10)$$

es decir, las f_i , $i = 1, 2, \dots, n$, anteriores con primeras derivadas parciales continuas en $\Omega \subset \mathbb{R}^n$. Los conceptos y la demostración de los resultados que se muestran en continuación pueden encontrarse en [71] y [76].

La búsqueda de la solución del sistema $F(x) = 0$ es un problema mucho más complejo que el de encontrar una solución de una ecuación escalar $f(x) = 0$. Al igual que en el caso escalar, podemos transformar el sistema

(2.9) en un sistema equivalente de la forma:

$$x = G(x), \quad (2.11)$$

para una determinada función vectorial $G : \mathbb{R}^n \rightarrow \mathbb{R}^n$, cuyas funciones coordenadas las vamos a denotar por g_i , $i = 1, 2, \dots, n$. Partiendo de una aproximación inicial $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})^T \in \mathbb{R}^n$, podemos generar una sucesión de vectores de \mathbb{R}^n mediante la fórmula iterativa

$$x^{(k)} = G(x^{(k-1)}), \quad k = 1, 2, \dots, n, \quad (2.12)$$

donde $x^{(k)} = (x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)})^T \in \mathbb{R}^n$, $k = 1, 2, \dots$

Decimos que el proceso es convergente si $\{x^{(k)}\} \rightarrow \xi$, cuando $k \rightarrow \infty$ donde $\xi = (\xi_1, \xi_2, \dots, \xi_n)^T \in \mathbb{R}^n$ será, bajo ciertas condiciones de la función G , una solución del sistema $x = G(x)$. Al vector ξ se le llama punto fijo de la función G y al algoritmo descrito por la ecuación (2.12) método de punto fijo.

A continuación daremos algunas definiciones de términos utilizados frecuentemente más adelante.

Definición 2.1.4 *Llamamos la función $F : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ F -diferenciable en un punto x en el interior de Ω si existe una aplicación lineal $A : \mathbb{R}^n \rightarrow \mathbb{R}^n$ tal que verifique, para cualquier $h \in \mathbb{R}^n$,*

$$\lim_{h \rightarrow 0} \frac{1}{\|h\|} \|F(x+h) - F(x) - A(h)\| \quad (2.13)$$

donde $\|\cdot\|$ norma vectorial en \mathbb{R}^n . La aplicación lineal A es una matriz $n \times n$ que depende del punto x , es decir, $A = A(x)$ recibe el nombre de derivada de Fréchet de F en x , denotándose por $F'(x)$.

Consideremos de nuevo que las funciones coordenadas de F , $f_j(x)$, $j = 1, \dots, n$ tienen derivadas parciales primeras continuas en Ω . Denotamos por $a_{ij}(x)$ el i, j -ésimo elemento de A . Dado que la convergencia en norma implica la convergencia componente a componente, podemos reescribir (2.13) con $h = te_j$, donde $e_j = (0, \dots, 0, 1, 0, \dots, 0)^T$ es el j -ésimo vector unitario, da

$$\lim_{t \rightarrow 0} \frac{1}{t} |f_i(x + te_j) - f_i(x) - ta_{ij}| = 0 \quad \text{para } 1 \leq i \leq n. \quad (2.14)$$

Sin embargo, $\lim_{t \rightarrow 0} \frac{1}{t} (f_i(x + te_j) - f_i(x)) = \frac{\partial f_i(x)}{\partial x_j}$. Por lo tanto, (2.14) implica que $a_{ij}(x) = \frac{\partial f_i(x)}{\partial x_j}$, $1 \leq i \leq n$, $1 \leq j \leq n$. Así, A es una matriz $n \times n$ cuyas componentes son las derivadas parciales primeras de la función F en x la llamamos matriz Jacobiana y la denotamos por $J_F(x)$.

Definición 2.1.5 *Diremos que una función $G : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ tiene un punto fijo $\xi \in \Omega \subset \mathbb{R}^n$ si $G(\xi) = \xi$.*

Definición 2.1.6 *Un conjunto Ω_0 se dice que es convexo si $\lambda x + (1 - \lambda)y \in \Omega_0$ para $\forall x, y \in \Omega_0$ y $\lambda \in [0, 1]$.*

Lema 2.1.1 ([70]) *Supongamos que $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ es continuamente diferenciable en un conjunto convexo $\Omega_0 \subset \mathbb{R}^n$ con funciones coordenadas $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, 2, \dots, n$. Entonces, para cualesquiera $x, y \in \Omega_0$, se verifica:*

$$F(y) - F(x) = \int_0^1 J_F(x + t(y-x))(y-x) dt.$$

Definición 2.1.7 *Se dice que una aplicación $G : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ es una contracción sobre un conjunto $\Omega_0 \subset \Omega$ si existe un escalar $0 < L < 1$ tal que $\|G(x) - G(y)\| \leq L\|x - y\|$ para todo $x, y \in \Omega_0$. La constante L recibe el nombre de constante de Lipschitz de G .*

El siguiente resultado establece no sólo las condiciones bajo las cuales el método iterativo (2.12) es convergente sino también de la unicidad del punto fijo.

Teorema 2.1.7 ([70]). Sea $G : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ una contracción sobre un conjunto cerrado $\Omega_0 \subset \Omega$ tal que $G(\Omega_0) \subset \Omega_0$. Entonces, podemos afirmar que G tiene un único punto fijo $\xi \in \Omega_0$. Por otro lado, para cualquier $x^{(0)} \in \Omega_0$, la sucesión $\{x^{(k)}\}$ definida por (2.12) converge a ξ . Además el error cometido en la k -ésima iteración está acotado por la desigualdad $\|x^{(k)} - \xi\| \leq \frac{L}{1-L} \|x^{(k)} - x^{(k-1)}\|$ y además, $\|x^{(k)} - \xi\| \leq \frac{L^k}{1-L} \|x^{(1)} - x^{(0)}\|$ donde L es la misma constante de Lipschitz de G .

Demostración: Sea $x^{(0)}$ un vector arbitrario en Ω_0 y $\{x^{(k)}\}$ la secuencia generada por (2.12). Como, por hipótesis, $G(\Omega_0) \subseteq \Omega_0$, la sucesión de iterados $\{x^{(k)}\}_{n \geq 0} \subset \Omega_0$ está bien definida y se encuentra en Ω_0 . Por la Definición 2.1.7,

$$\|x^{(k+1)} - x^{(k)}\| = \|G(x^{(k)}) - G(x^{(k-1)})\| \leq L \|x^{(k)} - x^{(k-1)}\| \quad \text{para } k \geq 1. \quad (2.15)$$

Aplicando sucesivamente (2.15) obtenemos

$$\begin{aligned} \|x^{(k+p)} - x^{(k)}\| &\leq \sum_{i=1}^p \|x^{(k+i)} - x^{(k+i-1)}\| \\ &= \|x^{(k+1)} - x^{(k)}\| + \|x^{(k+2)} - x^{(k+1)}\| + \dots + \|x^{(k+p)} - x^{(k+p-1)}\| \\ &\leq \|x^{(k+1)} - x^{(k)}\| + L \|x^{(k+1)} - x^{(k)}\| + \dots + L \|x^{(k+p-1)} - x^{(k+p-2)}\| \\ &= (1 + L + L^2 + \dots + L^{p-1}) \|x^{(k+1)} - x^{(k)}\| \\ &= \frac{1}{1-L} \|x^{(k+1)} - x^{(k)}\| \\ &\leq \frac{L^k}{1-L} \|x^{(1)} - x^{(0)}\| \end{aligned} \quad (2.16)$$

Además, cuando $p \rightarrow \infty$ se verifica $\|x^{(k)} - \xi\| \leq \frac{L^k}{1-L} \|x^{(1)} - x^{(0)}\|$. A partir (2.16) se deduce que la sucesión de iterados $\{x^{(k)}\}_{n \geq 0}$ es una sucesión de Cauchy en el conjunto cerrado $\Omega_0 \subset \mathbb{R}^n$, con respecto a cualquier norma, y por lo tanto tiene un límite ξ en Ω_0 : $\lim_{k \rightarrow \infty} x^{(k)} = \xi$. Además, por la continuidad de G

$$\begin{aligned} \|\xi - G(x^{(k)})\| &= \|\xi - x^{(k+1)} + G(x^{(k)}) - G(\xi)\| \\ &\leq \|\xi - x^{(k+1)}\| + \|G(x^{(k)}) - G(\xi)\| \\ &\leq \|\xi - x^{(k+1)}\| + L \|x^{(k)} - \xi\|. \end{aligned} \quad (2.17)$$

por lo que $\|\xi - G(x^{(k)})\| \rightarrow 0$ cuando $k \rightarrow \infty$. Así, ya que G es continua, es decir, ξ es punto fijo de G . Además, el límite es único, ya que si hay dos puntos fijos $\xi_1 \neq \xi_2$ en Ω_0 , entonces $\|\xi_1 - \xi_2\| = \|G(\xi_1) - G(\xi_2)\| \leq L \|\xi_1 - \xi_2\| < \|\xi_1 - \xi_2\|$, con lo que queda probada la unicidad. \square

Teorema 2.1.8 Si $G : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ es una contracción sobre un espacio cerrado $\Omega \subset \mathbb{R}^n$, el método de punto fijo converge hacia una solución de $F(x) = 0$.

Demostración: Aplicando el Teorema 2.1.7, existe un único punto fijo $\xi \in \Omega$ tal que es solución del sistema de ecuaciones $x = G(x)$. Por otra parte, al ser G contractiva, converge a ξ . Además, al ser la ecuación vectorial $x = G(x)$ equivalente a $F(x) = 0$, es evidente que ξ es solución de la ecuación $F(x) = 0$. \square

Observaciones:

(1) Si la función G es diferenciable, utilizando el teorema de valor medio, se obtiene $n = \sup_{x \in \Omega} \|J_G(x)\|$, siendo $\|\cdot\|$ la norma matricial subordinada a la norma vectorial que se utiliza en el Teorema 2.1.7.

(2) Si G es una contracción de constante L , designando por ξ a su punto fijo, se tiene que:

$$\|x^{(k+1)} - \xi\| = \|G(x^{(k)}) - G(\xi)\| \leq L \|x^{(k)} - \xi\|$$

por lo que el método presenta al menos convergencia lineal.

Lema 2.1.2 *Supongamos que $G : [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}^n$ es continua. Entonces, $\left\| \int_a^b G(t) dt \right\| \leq \int_a^b \|G(t)\| dt$.*

Demostración: Puesto que cualquier norma es una función continua en \mathbb{R}^n , ambas integrales existen y para cualquier $\epsilon > 0$ existe una partición $a < t_0 < t_1 < \dots < t_n < b$ de $[a, b]$ tal que

$$\left\| \int_a^b G(t) dt - \sum_{i=1}^n G(t_i)(t_i - t_{i-1}) \right\| < \epsilon$$

y

$$\left| \int_a^b \|G(t)\| dt - \sum_{i=1}^n \|G(t_i)\|(t_i - t_{i-1}) \right| < \epsilon.$$

Así,

$$\begin{aligned} \left\| \int_a^b G(t) dt \right\| &\leq \left\| \sum_{i=1}^n G(t_i)(t_i - t_{i-1}) \right\| + \epsilon \\ &\leq \sum_{i=1}^n \|G(t_i)\|(t_i - t_{i-1}) + \epsilon \leq \int_a^b \|G(t)\| dt + 2\epsilon \end{aligned}$$

y, como ϵ es arbitrario, queda probada la tesis del Lema. \square

Teorema 2.1.9 *Sean las funciones $G(x)$ y $J_G(x)$ continuas en el dominio Ω (convexo, acotado y cerrado), cumpliéndose la desigualdad*

$$\|J_G(x)\| = \max_{x \in \Omega} \left(\max_{i \in \{1, \dots, n\}} \sum_{j=1}^n \left| \frac{\partial g^i(x)}{\partial x_j} \right| \right) \leq L < 1$$

donde L es una constante. Si las aproximaciones sucesivas $x^{(k)} = G(x^{(k-1)})$, donde $k = 1, 2, \dots$ están contenidas en Ω , el proceso iterativo converge y el vector límite $\xi = \lim_{k \rightarrow \infty} x^{(k)}$ es la única solución del sistema $x = G(x)$ en el dominio Ω .

Demostración: Aplicando el Lema 2.1.1, podemos escribir $G(y) - G(x) = \int_0^1 J_G(w)(y - x) dt$, donde $J_G(w)$ es la matriz jacobiana de G evaluada en $w = x - t(y - x)$. Por lo tanto, por ser G continua

$$\|G(y) - G(x)\| = \left\| \int_0^1 J_G(w)(y - x) dt \right\| \leq \sup_{0 \leq t \leq 1} \|J_G(w)\| \cdot \|y - x\| \leq \|J_G(x)\| \cdot \|y - x\| \quad (2.18)$$

la jacobiana está acotada por hipótesis por lo que

$$\|G(y) - G(x)\| \leq L\|y - x\|, \quad 0 \leq L \leq 1. \quad (2.19)$$

lo que demuestra que G es una contracción y, aplicando el Teorema 2.1.8, se completa la demostración. \square

De manera análoga que al caso unidimensional, buscamos métodos de orden de convergencia al menos dos, lo que nos lleva a buscar una nueva forma de aproximar ξ . Si denotamos por $e_k = \xi - x^{(k)}$ y hacemos uso del desarrollo de Taylor hasta orden uno obtenemos que: $0 = F(\xi) \approx F(x^{(k)}) + e_k J_F(x^{(k)})$ con $J_F(x^{(k)})$ no singular, de donde $e_k \approx \xi - x^{(k)} = -[J_F(x^{(k)})]^{-1} F(x^{(k)})$ y, por tanto, $\xi \approx x^{(k)} - [J_F(x^{(k)})]^{-1} F(x^{(k)})$. Esta aproximación es la que utilizaremos para definir el nuevo iterado, es decir,

$$x^{(k+1)} = x^{(k)} - [J_F(x^{(k)})]^{-1} F(x^{(k)}), \quad (2.20)$$

que es la expresión iterativa del método de Newton. Es sencillo mostrar que el método de Newton se puede considerar como un método de punto fijo

$$x^{(k+1)} = x^{(k)} - [J_F(x^{(k)})]^{-1} F(x^{(k)}) = G(x^{(k)}).$$

De la expresión (2.20) podemos deducir la siguiente interpretación geométrica: cada componente f_i de F puede aproximarse por $f_i(x) \approx f_i(x^{(k)}) + f'_i(x^{(k)})(x - x^{(k)})$, expresión que describe el hiperplano tangente de f_i en $x^{(k)}$ y por tanto estamos considerando el iterado $x^{(k+1)}$ como la intersección de los n hiperplanos definidos por las componentes de F con el hiperplano $x = 0$.

Respecto a la convergencia local del método de Newton, Ortega [70] establece los siguientes resultados previos al teorema de convergencia local del método de Newton.

Lema 2.1.3 (Banach) *Sea A una matriz no singular. Supongamos que B es una matriz tal que $\|A^{-1}\| \|B\| < 1$. Entonces, $A + B$ es no singular, y*

$$\|(A + B)^{-1}\| \leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\| \|B\|}. \quad (2.21)$$

Demostración: $A + B = A(I + A^{-1}B)$, pero $\| -A^{-1}B \| \leq \|A^{-1}\| \|B\| < 1$. Por lo tanto, $\rho(-A^{-1}B) < 1$, así que aplicando el Lema de Neumann $(I + A^{-1}B)$ es no singular, de lo cual sigue que $A + B$ es no singular. Además,

$$\|(I + A^{-1}B)^{-1}\| = 1 + \|A^{-1}B\| + \|(A^{-1}B)^2\| + \dots = \frac{1}{1 - \|A^{-1}B\|}.$$

Como $(A + B)^{-1} = (I + A^{-1}B)^{-1} A^{-1}$, se sigue que:

$$\|(A + B)^{-1}\| \leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\| \|B\|}.$$

□

Teorema 2.1.10 ([71]) *Supongamos que $F : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ es F -diferenciable en $\xi \in \Omega$ tal que $F(\xi) = 0$. Sea $A : \Omega_0 \subset \Omega \rightarrow \mathcal{L}(\mathbb{R}^n)$ una aplicación definida en el entorno abierto Ω_0 de ξ , continua en ξ y con imagen en el conjunto de matrices reales de orden n . Supongamos, además, que $A(\xi)$ es no singular. Entonces, existe la bola cerrada $\Omega^* = \{x \in \Omega_0 / \|x - \xi\| \leq \delta\} \subset \Omega_0$, $\delta > 0$ sobre la cual la aplicación $G : \Omega^* \rightarrow \mathbb{R}^n$,*

$$G(x) = x - A^{-1}(x)F(x)$$

está bien definida en un entorno de ξ y es diferenciable en ξ , verificándose:

$$J_G(\xi) = I - (A(\xi))^{-1} J_F(\xi).$$

Demostración: Demostraremos en primer lugar que $A(x)$ es no singular en Ω^* : sea $\beta = \|A^{-1}(\xi)\|$ y sea $\varepsilon > 0$ tal que $0 < \varepsilon < \frac{1}{2\beta}$. Por continuidad de A en ξ existe un $\delta > 0$ tal que

$$\|A(x) - A(\xi)\| \leq \varepsilon \quad (2.22)$$

donde $x \in \Omega^*$. Así, el Lema de Banach 2.1.3 nos asegura que $A(x)$ es no singular y además

$$\|A(x)^{-1}\| \leq \frac{\beta}{1 - \beta\varepsilon} < 2\beta \quad \text{para todo } x \in \Omega^*, \quad (2.23)$$

por lo que G está bien definida en Ω^* .

A continuación, por ser F diferenciable en ξ , podemos suponer que δ es suficientemente pequeño para que se verifique

$$\|F(x) - F(\xi) - J_F(\xi)(x - \xi)\| \leq \varepsilon \|x - \xi\| \quad \text{para toda } x \in \Omega^*. \quad (2.24)$$

Obviamente, $\xi = G(\xi)$ y aplicando (2.22) y (2.24),

$$\begin{aligned} & \|G(x) - G(\xi) - (I - A(\xi)^{-1}J_F(\xi))(x - \xi)\| \\ &= \|A(\xi)^{-1}J_F(\xi)(x - \xi) - A(x)^{-1}F(x)\| \\ &\leq \|A(x)^{-1}[F(x) - F(\xi) - J_F(\xi)(x - \xi)]\| + \|A^{-1}(x)(A(\xi) - A(x))A(\xi)^{-1}J_F(\xi)(x - \xi)\| \\ &\leq (2\beta\varepsilon + 2\beta^2\varepsilon\|J_F(\xi)\|)\|x - \xi\| \end{aligned}$$

para todo $x \in \Omega^*$. Como ε es arbitrario y tanto $\|J_F(\xi)\|$ como β son constantes se demuestra que G es diferenciable en ξ y por lo tanto se cumple

$$J_G(\xi) = I - A(\xi)^{-1}J_F(\xi).$$

□

Teorema 2.1.11 (*Ostrowski*) *Supongamos que $G : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ es F -diferenciable en el punto fijo ξ en el interior de Ω y que $\rho(J_G(\xi)) = \sigma < 1$. Entonces, ξ es punto de atracción de $x^{(k)} = G(x^{(k-1)})$, $k = 1, 2, \dots$*

Demostración: Sea $\epsilon > 0$. Entonces, existe una norma en \mathbb{R}^n tal que $\|J_G(\xi)\| \leq \sigma + \epsilon$. Además, por ser G F -diferenciable, existe un $\delta > 0$ tal que si $\Omega^* = \{x : \|x - \xi\| \leq \delta\}$, entonces

$$\|G(x) - G(\xi) - J_G(\xi)(x - \xi)\| \leq \epsilon\|x - \xi\|$$

cuando $x \in \Omega^*$. Así, para todo $x \in \Omega^*$,

$$\|G(x) - G(\xi)\| \leq \|G(x) - G(\xi) - J_G(\xi)(x - \xi)\| + \|J_G(\xi)(x - \xi)\| \leq (\sigma + 2\epsilon)\|x - \xi\|.$$

Como $\sigma < 1$, podemos suponer que $\epsilon > 0$ es elegido de manera que $L = \sigma + 2\epsilon < 1$. Así, si $x^{(0)} \in \Omega^*$, se sigue de la desigualdad anterior que:

$$\|x^{(1)} - \xi\| = \|G(x^{(0)}) - G(\xi)\| \leq L\|x^{(0)} - \xi\|.$$

Por tanto, $x^{(1)} \in \Omega^*$ y por inducción se puede afirmar que todos los iterados $x^{(k)} \in \Omega^*$ y además, que

$$\|x^{(k)} - \xi\| \leq L\|x^{(k-1)} - \xi\| \leq \dots \leq L^k\|x^{(0)} - \xi\|$$

de donde se deduce que $x^{(k)} \rightarrow \xi$ cuando $k \rightarrow \infty$. □

Observación: Un caso especial, en el que $\rho(J_G(\xi)) = 0$, es cuando $A(x) = J_F(x)$. Esto corresponde a la iteración $x^{(k+1)} = x^{(k)} - [J_F(x^{(k)})]^{-1}F(x^{(k)})$ para $k = 0, 1, \dots$ que es el método de Newton.

Teorema 2.1.12 *Supongamos que $F : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ es F -diferenciable en el entorno abierto $\Omega_0 \subset \Omega$ del punto $\xi \in \Omega_0$ para que $F(\xi) = 0$. Además, supongamos que $J_F(x)$ es continua en ξ y que $J_F(\xi)$ es no singular. Entonces, ξ es un punto de atracción del método de Newton (2.20).*

Demostración: Según Teorema 2.1.11, con $A(x) = J_F(x)$ para $x \in \Omega^*$, concluimos que $G(x) = x - [J_F(x)]^{-1}F(x)$ está bien definida en alguna bola $\Omega^*(\xi, \delta) \subset \Omega_0$, $\delta > 0$. Además, $\rho(J_G(\xi)) = \sigma = 0$. Entonces, ξ es punto de atracción del método de Newton (2.20). □

Lema 2.1.4 *Supongamos que $F : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ es F -diferenciable en el conjunto convexo $\Omega_0 \subset \Omega$ y supongamos que existe una constante $\alpha \geq 0$, tal que $J_F(x)$ satisface*

$$\|J_F(u) - J_F(v)\| \leq \alpha\|u - v\| \quad \text{para todo } u, v \in \Omega_0.$$

Entonces, para cualesquiera $x, y \in \Omega_0$

$$\|F(y) - F(x) - J_F(x)(y - x)\| \leq \frac{\alpha}{2}\|x - y\|^2.$$

Demostración: Aplicando el Lema 2.1.1 tenemos que:

$$F(y) - F(x) = \int_0^1 J_F(x + t(y-x))(y-x)dt,$$

entonces, dado que $J_F(x)(y-x)$ es constante respecto a la variable de la integración,

$$F(y) - F(x) - J_F(x)(y-x) = \int_0^1 (J_F(x + t(y-x)) - J_F(x))(y-x)dt.$$

Tomando normas a ambos lados y aplicando el Lema 2.1.2

$$\begin{aligned} \|F(y) - F(x) - J_F(x)(y-x)\| &= \left\| \int_0^1 (J_F(x + t(y-x)) - J_F(x))(y-x)dt \right\| \\ &\leq \int_0^1 \|(J_F(x + t(y-x)) - J_F(x))(y-x)\| dt \\ &\leq \int_0^1 \|(J_F(x + t(y-x)) - J_F(x))\| dt \|(y-x)\|. \end{aligned}$$

Aplicando la desigualdad de las hipótesis e integrando obtenemos:

$$\|F(y) - F(x) - J_F(x)(y-x)\| \leq \int_0^1 \alpha \|y-x\| dt \|(y-x)\| \leq \frac{1}{2} \alpha \|x-y\|^2$$

con lo que queda demostrado el lema. \square

Teorema 2.1.13 (*Convergencia local del método de Newton*) *Supongamos que las hipótesis del Teorema 2.1.12 se satisfacen. Entonces,*

$$\lim_{k \rightarrow \infty} \frac{\|x^{(k+1)} - \xi\|}{\|x^{(k)} - \xi\|} = 0. \quad (2.25)$$

Además, si $J_F(x)$ es Lipschitz

$$\|J_F(x) - J_F(\xi)\| \leq \alpha \|x - \xi\|, \quad (2.26)$$

para todo x en algún entorno de ξ , entonces, existe una constante c positiva tal que

$$\|x^{(k+1)} - \xi\| \leq c \|x^{(k)} - \xi\|^2. \quad (2.27)$$

para todo $k \geq k_0$, donde k_0 depende de la elección de x_0 .

Demostración: Como el método de Newton es un método de punto fijo, aplicando el Teorema 2.1.10 su función de iteración $G(x) = x - [J_F(x)]^{-1}F(x)$ está bien definida en algún entorno de ξ . G es diferenciable en ξ tal que $J_F(\xi) = I - [J_F(\xi)]^{-1}J_F(\xi) = 0$. Así, por el Teorema 2.1.11, ξ es punto de atracción de G . Entonces, para $x^{(k)}$ en la bola de atracción, $x^{(k+1)} = G(x^{(k)})$ implica

$$\begin{aligned} &\lim_{k \rightarrow \infty} \frac{\|G(x^{(k)}) - G(\xi) - J_G(\xi)(x^{(k)} - \xi)\|}{\|x^{(k)} - \xi\|} \\ &= \lim_{k \rightarrow \infty} \frac{\|G(x^{(k)}) - G(\xi) - (I - [J_F(\xi)]^{-1}J_F(\xi))(x^{(k)} - \xi)\|}{\|x^{(k)} - \xi\|} \\ &= \lim_{k \rightarrow \infty} \frac{\|G(x^{(k)}) - G(\xi)\|}{\|x^{(k)} - \xi\|} \\ &= \lim_{k \rightarrow \infty} \frac{\|x^{(k+1)} - \xi\|}{\|x^{(k)} - \xi\|} = 0 \end{aligned}$$

con lo que se demuestra (2.25). Supongamos que se verificá (2.26) para todo x en el entorno de ξ . Por un argumento análogo a la demostración del Lema 2.1.4:

$$\|F(x) - F(\xi) - J_F(\xi)(x - \xi)\| \leq \frac{1}{2} \alpha \|x - \xi\|^2. \quad (2.28)$$

En consecuencia,

$$\begin{aligned}
\|x^{(k+1)} - \xi\| &= \|G(x^{(k)}) - \xi\| = \|x^{(k)} - [J_F(x^{(k)})]^{-1}F(x^{(k)}) - \xi\| \\
&\leq \| [J_F(x^{(k)})]^{-1} [F(x^{(k)}) - F(\xi) - J_F(\xi)(x^{(k)} - \xi)] \| \\
&\quad + \| [J_F(x^{(k)})]^{-1} [J_F(x^{(k)}) - J_F(\xi)](x^{(k)} - \xi) \| \\
&\leq \| [J_F(x^{(k)})]^{-1} \| \left(\frac{\alpha}{2} + \alpha \right) \| (x^{(k)} - \xi) \|^2 \\
&\leq 2\beta \left(\frac{\alpha}{2} + \alpha \right) \| (x^{(k)} - \xi) \|^2 = c \| (x^{(k)} - \xi) \|^2
\end{aligned}$$

a partir de (2.26) y (2.28) y del Teorema 2.1.10 que nos permite afirmar que $[J_F(x^{(k)})]^{-1} \leq 2\beta$. Por lo tanto siempre que, para una estimación inicial dada $x^{(0)}$ se elija un k_0 tal que $x^{(k)}$ permanezca en el entorno de ξ definido por $c = 3\beta\alpha$ para todo $k \geq k_0$. \square

En el estudio de la convergencia de los diferentes métodos utilizamos el siguiente resultado.

Teorema 2.1.14 ([81]) *Sea $G(x)$ una función de punto fijo continuamente diferenciable hasta orden p . El método iterativo $x^{(k+1)} = G(x^{(k)})$ es de orden p si*

- i) $G(\xi) = \xi$;
- ii) $\frac{\partial^{(k)} g_i(\xi)}{\partial x_{j_1} \partial x_{j_2} \cdots \partial x_{j_k}} = 0$, para todo $1 \leq k \leq p-1$, $1 \leq i, j_1, \dots, j_k \leq n$;
- iii) $\frac{\partial^{(p)} g_i(\xi)}{\partial x_{j_1} \partial x_{j_2} \cdots \partial x_{j_p}} \neq 0$, al menos para un valor de $i, j_1, \dots, j_p \in \{1, \dots, n\}$

donde g_i , $i = 1, 2, \dots, n$, son las funciones coordenadas de G .

Como el objetivo principal de la memoria es el desarrollo de métodos iterativos efectivos de orden alto, las demostraciones del orden de convergencia utilizando el Teorema 2.1.14 son tediosas. Combinamos esta técnica con el desarrollo en serie de Taylor en varias variables. A continuación, introducimos la notación necesaria para estas demostraciones.

Sea $F : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ suficientemente diferenciable en Ω . La q -ésima derivada de F en $u \in \mathbb{R}^n$, $q \geq 1$, es la función q -lineal $F^{(q)}(u) : \mathbb{R}^n \times \cdots \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ tal que $F^{(q)}(u)(v_1, \dots, v_q) \in \mathbb{R}^n$. Es fácil observar que

1. $F^{(q)}(u)(v_1, \dots, v_{q-1}, \cdot) \in \mathcal{L}(\mathbb{R}^n)$.
2. $F^{(q)}(u)(v_{\sigma(1)}, \dots, v_{\sigma(q)}) = F^{(q)}(u)(v_1, \dots, v_q)$, para toda permutación σ de $\{1, 2, \dots, q\}$.

Partiendo de las propiedades anteriores podemos introducir la siguiente notación:

- (a) $F^{(q)}(u)(v_1, \dots, v_q) = F^{(q)}(u)v_1 \cdots v_q$,
- (b) $F^{(q)}(u)v^{q-1}F^{(p)}v^p = F^{(q)}(u)F^{(p)}v^{q+p-1}$.

Por otro lado, para $\xi + h \in \mathbb{R}^n$ acotado en un entorno de la solución ξ de $F(x) = 0$, podemos aplicar la expansión de Taylor y suponiendo que la matriz jacobiana $J_F(\xi)$ sea no singular, tenemos

$$F(\xi + h) = J_F(\xi) \left[h + \sum_{q=2}^{p-1} C_q h^q \right] + O[h^p], \quad (2.29)$$

donde $C_q = \frac{1}{q!} [J_F(\xi)]^{-1} F^{(q)}(\xi)$, $q \geq 2$. Observamos que $C_q h^q \in \mathbb{R}^n$ ya que $F^{(q)}(\xi) \in \mathcal{L}(\mathbb{R}^n \times \cdots \times \mathbb{R}^n, \mathbb{R}^n)$ y $[J_F(\xi)]^{-1} \in \mathcal{L}(\mathbb{R}^n)$. Además, podemos expresar $J_F(\xi + h)$ como:

$$J_F(\xi + h) = J_F(\xi) \left[I + \sum_{q=2}^{p-1} q C_q h^{q-1} \right] + O[h^p], \quad (2.30)$$

donde I es la matriz identidad. Por lo tanto, $qC_q h^{q-1} \in \mathcal{L}(\mathbb{R}^n)$.

Denotamos por $e_k = x^{(k)} - \xi$ al error en la iteración k -ésima y la ecuación del error puede expresarse como $e_{k+1} = L e_k^p + O[e_k^{p+1}]$, donde L es la función p -lineal $L \in \mathcal{L}(\mathbb{R}^n \times \cdots \times \mathbb{R}^n, \mathbb{R}^n)$, y p es el orden de convergencia del método.

Con el fin de comparar los distintos métodos, consideramos el orden de convergencia, p , y el índice de eficiencia, I , tal como fueron definidos en sección anterior. Además, utilizamos $IC = p^{1/(d+op)}$, que llamamos índice de eficiencia computacional, donde op es el número de operaciones - productos y cocientes, - por iteración. Observamos que cada iteración del método de Newton se reduce al cálculo del vector $h = x^{(k+1)} - x^{(k)}$ correspondiente y éste no es más que la solución del sistema de ecuaciones lineales: $J_F(x^{(k)})h + F(x^{(k)}) = 0$. Recordamos que el número de productos y cocientes que necesitamos para resolver un sistema lineal, mediante la eliminación de Gauss, es $\frac{1}{3}n^3 + n^2 - \frac{1}{3}n$ donde n es el tamaño del sistema.

2.2. Integración numérica

Los métodos numéricos de integración se basan en la interpretación de la integral definida como medida del área entre la curva que define la función, el eje de abscisa y las imágenes de los extremos del intervalo por la función.

Numéricamente, aproximaremos la integral $I = \int_a^b f(x)dx$ mediante una suma ponderada $\sum_{i=0}^n a_i f(x_i)$ de los valores de la función en ciertos puntos x_i donde $i = 0, 1, 2, \dots, n$ del intervalo de integración $[a, b]$ llamados nodos. La fórmula resultante

$$I = \int_a^b f(x)dx = \sum_{i=0}^n a_i f(x_i) + E_n(f(x)) \quad (2.31)$$

recibe el nombre de fórmula de cuadratura y los coeficientes de dicha fórmula a_i llamados pesos son tales que $a_0 + a_1 + a_2 + \cdots + a_n = b - a$. El problema clásico consiste en elegir los nodos x_i y los coeficientes a_i de manera que el error de integración $E_n(f(x)) = I - \sum_{i=0}^n a_i f(x_i)$ sea lo más pequeño posible. Una manera de medir el error es mediante el grado de precisión, que es el mayor número natural $p \in \mathbb{N}$ tal que $E_n(x^k) = 0$, $k = 1, 2, \dots, p$ pero $E_n(x^{k+1}) \neq 0$. De este modo si una fórmula tiene grado de precisión p , todos los polinomios de grado máximo p son integrados exactamente. En el caso de nodos equiespaciadas, $h = \frac{b-a}{n}$ el error es de orden m cuando $E_n(f(x)) = O(h^m)$.

2.2.1. Cuadratura de Newton-Cotes

Seleccionemos un conjunto de nodos equidistantes y evaluemos sobre ellos la función integrando. Si aplicamos la interpolación de Lagrange, entonces, el polinomio de interpolación es:

$$P_n(x) = f(x_0)L_0(x) + f(x_1)L_1(x) + \cdots + f(x_n)L_n(x),$$

donde los polinomios de Lagrange $L_i(x)$, $i = 0, 1, \dots, n$ pueden expresarse de la forma: $L_i(x) = \frac{z(x)}{(x-x_i)z'(x_i)}$ y $z(x) = (x-x_0)(x-x_1)\cdots(x-x_n)$. Además,

$$\int_a^b P_n(x)dx = \sum_{i=0}^n \int_a^b f(x_i)L_i(x)dx = \sum_{i=0}^n f(x_i) \int_a^b L_i(x)dx = \sum_{i=0}^n a_i f(x_i),$$

donde los coeficientes $a_i = \int_a^b L_i(x)dx$, $i = 0, 1, \dots, n$ no dependen de la función sino sólo del dominio de integración. El error de las fórmulas de cuadraturas así definidas viene dado por $E(f) = \int_a^b \prod_{i=0}^n (x -$

$x_i)f^{(n+1)}(\zeta)dx$. De este modo se obtienen las fórmulas de Newton-Cotes abiertas y cerradas. La fórmula cerrada de $(n + 1)$ puntos de Newton-Cotes incluye los extremos del intervalo cerrado $[a, b]$ como nodos: $x_i = x_0 + ih$, para $i = 0, 1, \dots, n$, donde $x_0 = a$, $x_n = b$ y $h = \frac{b-a}{n}$. A continuación, en la Tabla 2.1 se incluyen algunas de las fórmulas cerradas de Newton-Cotes más conocidas.

Tabla 2.1: Fórmulas cerradas comunes de Newton-Cotes

Método	Fórmula
n=1: regla del trapecio	$\int_{x_0}^{x_1} f(x)dx = \frac{h}{2}[f(x_0) + f(x_1)]$
n=2: regla de Simpson	$\int_{x_0}^{x_2} f(x)dx = \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)]$
n=3: regla de 3/8 de Simpson	$\int_{x_0}^{x_3} f(x)dx = \frac{3h}{8}[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)]$

En las fórmulas abiertas de Newton-Cotes, los nodos: $x_i = x_0 + (i + 1)h$ se usan para cada $i = 0, 1, \dots, n$, donde $x_0 = a + h$ y $h = \frac{b-a}{n+2}$. Esto implica que $x_n = b - h$, por lo cual denotamos los extremos por $x_{-1} = a$ y $x_{n+1} = b$. Las fórmulas abiertas emplean únicamente los nodos dentro del intervalo abierto (a, b) . A continuación, en la Tabla 2.2 se incluyen algunas de las fórmulas abiertas de Newton-Cotes más comunes.

Tabla 2.2: Fórmulas abiertas comunes de Newton-Cotes

Método	Fórmula
n=0: regla de punto medio	$\int_{x_{-1}}^{x_1} f(x)dx = 2hf(x_0)$
n=1	$\int_{x_{-1}}^{x_2} f(x)dx = \frac{3h}{2}[f(x_0) + f(x_1)]$
n=2	$\int_{x_{-1}}^{x_3} f(x)dx = \frac{4h}{3}[2f(x_0) - f(x_1) + 2f(x_2)]$

2.2.2. Cuadratura gaussiana

Seleccionamos un conjunto de nodos $\tau_1, \tau_2, \dots, \tau_n$ en intervalo $[a, b]$ de manera óptima. Estos nodos, junto con sus coeficientes $\omega_1, \omega_2, \dots, \omega_n$, se eligen para reducir en lo posible el error cometido al efectuar la aproximación $\int_a^b f(x)dx \approx \sum_{i=1}^n \omega_i f(\tau_i)$. Sea $P(x)$ el polinomio que interpola los puntos $(\tau_i, f(\tau_i))$, $i = 1, 2, \dots, n$, se tiene que

el error

$$E_n(f(x)) = \int_a^b (f(x) - P(x))dx = \int_a^b z_n(x) \frac{f^{(n)}(\zeta)}{n!} dx, \quad (2.32)$$

donde $z_n(x) = (x - x_1)(x - x_2) \cdots (x - x_n)$ es el polinomio fundamental de interpolación. Si $f(x)$ es un polinomio de grado menor o igual que $n - 1$, entonces el error (2.32) será cero por utilizar n nodos y, por lo tanto, el grado de precisión será $n - 1$, por lo menos. El objetivo es calcular los puntos τ_i de tal manera que el error (2.32) se anule cuando $f(x)$ es un polinomio de grado $n + p$ con $p = 0, 1, \dots, t$ con t lo más grande posible. Si derivamos un polinomio de grado $n + p$ n veces, se obtiene un polinomio de grado p , entonces la condición necesaria y suficiente de que el error (2.32) se anule para todos los polinomios de grado $n + p$ es $\int_a^b z_n(x)x^p dx = 0, \quad p = 0, 1, \dots, t.$

Teorema 2.2.1 *La condición necesaria y suficiente para que la fórmula de integración*

$$\int_a^b f(x)dx = \sum_{i=1}^n \omega_i f(\tau_i)$$

tenga grado máximo de precisión igual a $2n-1$ es que los nodos $\tau_i, i = 1, 2, \dots, n$, sean los ceros del polinomio ortogonal en $[a, b]$ de grado n , $z(x)$. En este caso la expresión del error es

$$E_n(f(x)) = \int_a^b \frac{f^{(n)}(\zeta)}{n!} z(x) dx.$$

Los coeficientes ω_i se determinan una vez que se haya definido la familia de polinomios ortogonales y calculado los nodos $\tau_i, i = 1, 2, \dots, n$. Fácilmente se puede demostrar que estos coeficientes son positivos.

La forma estándar de la cuadratura de Gauss viene dada por:

$$\int_a^b f(x)dx = \frac{b-a}{2} \int_{-1}^1 f\left(\frac{(b-a)\tau_i + (b+a)}{2}\right) d\tau_i = \sum_{i=1}^n \omega_i f\left(\frac{(b-a)\tau_i + (b+a)}{2}\right)$$

Por lo tanto, sólo tenemos que considerar el intervalo $[-1, 1]$, ya que si ω_i y $\tau_i, 0 \leq i \leq n$, son los pesos y nodos en $[-1, 1]$, entonces ω_i y $((b-a)\tau_i + (b+a))/2, 0 \leq i \leq m$ son pesos y nodos de $[a, b]$.

En el Tabla 2.3 se presentan los nodos y los pesos de los polinomios ortogonales correspondientes a la cuadratura gaussiana que se han usado en el los capítulos 3, 4 y 5 de esta memoria.

2.3. Problemas de valor inicial

Las ecuaciones diferenciales aparecen en la mayor parte de los problemas de las ciencias y la ingeniería modelando los fenómenos que estudiamos. Estos modelos pueden ser muy sencillos (una sola ecuación diferencial para una función desconocida) o más complejos formando sistemas de ecuaciones diferenciales. La mayoría de las ecuaciones diferenciales de importancia práctica no se pueden resolver mediante métodos analíticos. En estos casos los métodos numéricos son la única alternativa posible para su resolución.

La ecuación $F(x, y, y', \dots, y^{(n)}) = 0$ se llama ecuación diferencial ordinaria de orden n , si F está definida y es continua en alguna región de $\Omega \in \mathbb{R}^{n+2}$, ($n > 1$) y, en todo caso, dependerá de la derivada n -ésima de y . Su solución es cualquier función $y(x)$ que satisfaga esta ecuación para todo x en un intervalo finito o infinito. La ecuación diferencial puede ser presentada en la forma

$$y^{(n)} = f(x, y, y', \dots, y^{(n-1)}). \quad (2.33)$$

Una solución de (2.33) en el intervalo $I = [a, b]$ es una función $y(x)$ tal que

Tabla 2.3: Nodos y pesos de algunos polinomios ortogonales

Polinomio	n	Nodos, τ_i	Pesos, ω_i
Tschebychev	1	$\tau_1 = 0$	$\omega_1 = \pi$
	2	$\tau_1 = -\frac{\sqrt{2}}{2}, \tau_2 = \frac{\sqrt{2}}{2}$	$\omega_1 = \omega_2 = \frac{\pi}{2}$
	3	$\tau_1 = -\frac{\sqrt{3}}{2}, \tau_2 = 0, \tau_3 = \frac{\sqrt{3}}{2}$	$\omega_1 = \omega_2 = \omega_3 = \frac{\pi}{3}$
Legendre	1	$\tau_1 = 0$	$\omega_1 = 2$
	2	$\tau_1 = -\frac{\sqrt{3}}{3}, \tau_2 = \frac{\sqrt{3}}{3}$	$\omega_1 = \omega_2 = 1$
	3	$\tau_1 = -\frac{\sqrt{3}}{5}, \tau_2 = 0, \tau_3 = \frac{\sqrt{3}}{5}$	$\omega_1 = \omega_3 = \frac{5}{9}, \omega_2 = \frac{8}{9}$
Lobatto	1	$\tau_1 = 0$	$\omega_1 = 2$
	2	$\tau_1 = -1, \tau_2 = 1$	$\omega_1 = \omega_2 = 1$
	3	$\tau_1 = -1, \tau_2 = 0, \tau_3 = 1$	$\omega_1 = \omega_3 = \frac{1}{3}, \omega_2 = \frac{4}{3}$
Radau	1	$\tau_1 = -1$	$\omega_1 = 2$
	2	$\tau_1 = -1, \tau_2 = \frac{1}{3}$	$\omega_1 = 0.5, \omega_2 = 1.5$
	3	$\tau_1 = -1, \tau_2 = \frac{1-\sqrt{6}}{5}, \tau_3 = \frac{1+\sqrt{6}}{5}$	$\omega_1 = \frac{2}{9}, \omega_2 = \frac{16+\sqrt{6}}{18}, \omega_3 = \frac{16-\sqrt{6}}{18}$

1. $y(x) \in C^n[a, b]$,
2. $(x, y(x), \dots, y^{(n-1)}(x)) \in \text{Dom}(f)$, para todo $x \in I$,
3. $y^{(n)}(x) = f(x, y, y', \dots, y^{(n-1)})$, para todo $x \in I$.

Habitualmente estas ecuaciones diferenciales están acompañadas por condiciones adicionales que especifican el estado del sistema que se estudia en un tiempo o posición inicial. Esto se conoce como la condición inicial y junto con la ecuación diferencial forman el llamado problema de valor inicial o problema de Cauchy.

Cualquier ecuación de tipo (2.33) se puede reducir a un sistema de ecuaciones diferenciales de primer orden equivalente.

2.3.1. Conceptos básicos

Resolver un problema de valor inicial (PVI ó el problema de Cauchy) consiste en encontrar una solución de una ecuación diferencial que satisfaga ciertas condiciones iniciales en el extremo inicial del intervalo donde está definida la variable independiente.

Más concretamente, siendo I un intervalo de \mathbb{R} de la forma $[a, b]$ y siendo f una función continua definida en $I \times \mathbb{R}^n$ y con valores en \mathbb{R}^n puede considerarse el problema siguiente:

Hallar una función continua y diferenciable $y(x)$ definida en I y con valores en \mathbb{R}^n verificando:

$$\begin{aligned} y'(x) &= f(x, y(x)) \quad \text{para todo } x \in I; \\ y(a) &= y_0 \in \mathbb{R}. \end{aligned} \tag{2.34}$$

Si denotamos por $f_i : I \times \mathbb{R}^n \rightarrow \mathbb{R}$ a las funciones coordenadas de f , resolver el PVI (2.34) implica encontrar n funciones $y_i(x)$, $i = 1, \dots, n$ definidas en $[a, b]$ tales que:

$$\begin{aligned} y_i'(x) &= f(x, y_1(x), \dots, y_n(x)) \quad 1 \leq i \leq n, \quad a \leq x \leq b; \\ y_i(a) &= y_{0,i}. \end{aligned} \quad (2.35)$$

La solución de este tipo de problemas, formalmente, estará dada por:

$$y(x) = y_0 + \int_a^x f(s, y(s)) ds. \quad (2.36)$$

Para resolver el problema (2.34) numéricamente será necesario comprobar previamente si hay solución y si ésta es única. Para ello es conveniente recordar el teorema de existencia y unicidad del problema de valor inicial de una ecuación diferencial ordinaria de primer orden. Como se sabe, hay distintos resultados en los que se definen las condiciones suficientes para la existencia de solución única. En ellos se van debilitando las condiciones impuestas a la función $f(x, y(x))$.

Teorema 2.3.1 ([80]) (Existencia local y unicidad) Si existe un número positivo γ tal que f_1, f_2, \dots, f_n son continuas y tienen derivadas parciales primeras continuas con respecto a las componentes y_1, y_2, \dots, y_n de y , para

$$|x - a| < \gamma, \quad |y_1 - y_{0,1}| < \gamma, \quad |y_2 - y_{0,2}| < \gamma, \dots, \quad |y_n - y_{0,n}| < \gamma, \quad (2.37)$$

entonces existe $\eta > 0$ tal que el sistema (2.34) tiene solución única para $|x - a| \leq \eta$.

Definición 2.3.1 Se dice que f satisface la condición de Lipschitz en y siempre que

$$\|f(x, y) - f(x, y^*)\| \leq L\|y - y^*\|, \quad (2.38)$$

donde L es la constante de Lipschitz independiente de x, y e y^* .

Si se debilita la condición de que la derivada parcial de f respecto de y sea continua en el dominio y se exige que se satisfaga la condición de Lipschitz para la variable y : $\|f(x, y_1) - f(x, y_2)\| \leq L\|y_1 - y_2\|$ en los puntos del dominio, se consigue una versión más fuerte del teorema, que recibe el nombre de teorema de Picard-Lindelöf.

Teorema 2.3.2 ([14]) Sea $f(x, y) = (f_1(x, y), \dots, f_n(x, y))^T$ una función vectorial continua definida en

$$S = \{(x, y) | x \in [a, b] \subset \mathbb{R}, y \in \mathbb{R}^n, \|y\| < \infty\}. \quad (2.39)$$

Si f satisface la condición de Lipschitz en y , entonces, (2.34) tiene solución única para $x \in [a, b]$.

Nota: Notamos que si f es Lipschitz con respecto a una norma en \mathbb{R}^n , entonces, por la equivalencia de las normas vectoriales, es Lipschitz con respecto a cualquier otra norma en \mathbb{R}^n pero con otra constante de Lipschitz.

Además, si cada $\frac{\partial f_i}{\partial y_j}$, $1 \leq i \leq n$, $1 \leq j \leq n$ es continua y acotada en S , entonces, la aplicación del teorema de valor medio sobre cada componente

$$f_i(x, y) - f_i(x, y^*) = \sum_{j=1}^n \frac{\partial f_i(x, c_i)}{\partial y_j} (y_j - y_j^*)$$

para algunos puntos $c_i \in \mathbb{R}^n$ en el segmento de línea en \mathbb{R}^n entre y y y^* . Al permitir que

$$L = \max_{x, y \in S} \max_i \sum_{j=1}^n \left| \frac{\partial f_i(x, y)}{\partial y_j} \right|$$

el teorema de valor medio multivariable da la condición (2.38) para $\|\cdot\| = \|\cdot\|_\infty$.

En esta sección nos concentramos en la solución numérica de (2.34), es decir, vamos a buscar aproximaciones a $y(x)$, en un conjunto de puntos discreto $x_0, x_1, \dots, x_N \in [a, b]$, donde N es un número positivo. Aunque en la práctica los x no necesitan ser equidistantes, vamos a suponer aquí que lo son para simplificar la presentación del análisis teórico. Con ese supuesto, definimos el tamaño del paso como: $h = \frac{b-a}{N}$. Los nodos del esquema numérico se definen como puntos equidistantes $x_k = a + hk$, $0 \leq k \leq N$, es decir, $x_0 = a, x_1 = a+h, \dots, x_N = b$ y $x_{k+1} - x_k = h$. Con el algoritmo numérico se generan vectores y_0, y_1, \dots, y_N , que se aproximarán a la solución $y(x)$ en cada uno de los nodos, es decir, $y_k \approx y(x_k)$.

Al resolver numéricamente una ecuación diferencial aparecen varios tipos de errores. Si suponemos que la condición inicial $y(x_0) = y_0$, $x_0 = a$, es exacta, entonces $e_0 = 0$. Entonces, podemos clasificar los errores en:

1. Error de discretización local es el error que se comete en un solo paso cuando reemplazamos un proceso infinito por uno finito, es inherente a cualquier algoritmo que podamos escoger y en gran medida es independiente del error de redondeo.

$$\varepsilon_{k+1} = y(x_{k+1}) - y(x_k) - h\Phi(x_k, y(x_k), h), \quad 0 \leq k \leq N-1. \quad (2.40)$$

2. Error de redondeo local es el provocado por la limitada precisión de los ordenadores y su magnitud depende del número de bits en la mantisa usando aritmética de coma flotante.
3. Error de discretización global define la diferencia entre la solución exacta y la calculada con el método en el punto correspondiente: es la acumulación de los errores de discretización local, y está asociado al método y es independiente del ordenador.

$$e_k = y(x) - y(x_k) \text{ para } k = 0, 1, \dots, N. \quad (2.41)$$

4. Error de redondeo global es la acumulación de los errores de redondeo local.
5. Error total es la suma de los errores de truncamiento global y redondeo global.

Por simplicidad, se desarrollaran los métodos numéricos que configuren este tema en el caso de problemas de valor inicial en los que interviene una única ecuación diferencial ordinaria. La extensión de los métodos numéricos a problemas en los que intervenga un sistema de ecuaciones diferenciales ordinarias es automática y, en cuanto a su análisis, serán válidos los teoremas y propiedades que se presenten con la precaución de emplear la métrica correspondiente en \mathbb{R}^n .

Podemos clasificar en general los métodos numéricos como métodos de un sólo paso cuando la determinación de y_{k+1} involucra solamente un único valor de y_k , es decir, $(x_0, y_0) \rightarrow (x_1, y_1) \rightarrow (x_2, y_2) \dots$ y de multipaso cuando para calcular cada paso se utilizan varios de los anteriores, es decir, $(x_0, y_0), \dots, (x_i, y_i) \rightarrow (x_{i+1}, y_{i+1})$. Por otra parte, un método puede ser explícito si para determinar y_{k+1} se utilizan valores anteriores de y_k , por ejemplo (un paso), $y_{k+1} = y_k + h\Phi(x_k, y_k, h)$ e implícito si se utiliza el valor de la función y_{k+1} , por ejemplo (un paso), $y_{k+1} = y_k + h\Phi(x_k, y_k, y_{k+1}, h)$.

2.3.2. Métodos de un sólo paso

La característica fundamental de todos estos métodos es que partiendo de la condición inicial $y(x_0) = y_0$ en cada paso se considera un nuevo PVI:

$$PVI_k = \begin{cases} y' = f(x, y) \\ y(x_k) = y_k \end{cases}$$

para $k = 0, 1, 2, \dots$ donde $x_k = x_{k-1} + h = x_0 + kh$. Aproximamos $y(x_{k+1})$ mediante

$$y_{k+1} = y_k + h\Phi(x_k, y_k, h), \quad \text{donde } y_0 = y(x_0), \quad k = 0, 1, \dots \quad (2.42)$$

La expresión (2.42) es la fórmula general de los métodos un paso que se diferencian entre sí en la forma de calcular $\Phi(x_k, y_k, h)$.

Puesto que por definición $\frac{dy(x)}{dx} = \lim_{h \rightarrow 0} \frac{y(x+h) - y(x)}{h}$, el enfoque más simple y más evidente para la resolución de la ecuación diferencial $\frac{dy(x)}{dx} = f(x, y(x))$ es aproximarla por

$$\frac{y(x+h) - y(x)}{h} \approx f(x, y(x)), \quad (2.43)$$

donde h es un tamaño de paso muy pequeño pero no nulo.

A partir del valor inicial dado $y(x_0) = y_0$, podemos usar (2.43) para la aproximación de y en el instante $x = h$, después en $x = 2h$, y así sucesivamente. De este modo tenemos el método de aproximación de Euler:

$$\begin{aligned} y(x_0) &= y_0, \\ y(x_{k+1}) &= y(x_k) + hf(x_k, y(x_k)), \quad k = 0, 1, \dots \end{aligned}$$

Si $y(x)$ es solución exacta del problema (2.34) y aproximamos la integral en el lado derecho de la expresión (2.36) por la regla de punto medio, se obtiene

$$\begin{aligned} y(x_0) &= y_0, \\ y(x_{k+1}) &= y(x_k) + hf \left(x_k + \frac{h}{2}, y(x_k) + \frac{h}{2} f(x_k, y(x_k)) \right). \end{aligned}$$

Definición 2.3.2 : Un método de un solo paso (2.42) es consistente con la ecuación diferencial $y' = f(x, y)$ si $\Phi(x, y, 0) = f(x, y)$.

Definición 2.3.3 : Decimos que un método de un solo paso (2.42) tiene orden de precisión p si p es el mayor entero para el que se verifica:

$$y(x+h) - [y(x) + h\Phi(x, y, h)] = O(h^{p+1}). \quad (2.44)$$

Usando estas definiciones es fácil de mostrar que los métodos de Euler y punto medio son consistentes y tienen orden de precisión uno y dos, respectivamente.

2.3.3. Métodos de Taylor y Runge-Kutta

Supongamos que la solución $y(x)$ del problema de valor inicial (2.34) tiene $(n+1)$ derivadas continuas. Si se desarrolla la solución $y(x)$ en función del n -ésimo polinomio de Taylor alrededor de x_k y la evaluamos en x_{k+1} obtenemos

$$y(x_{k+1}) = y(x_k) + hy'(x_k) + \frac{h^2}{2}y''(x_k) + \dots + \frac{h^n}{n!}y^{(n)}(x_k) + \frac{h^{n+1}}{(n+1)!}y^{(n+1)}(\zeta_k), \quad (2.45)$$

para algún ζ_k en (x_k, x_{k+1}) .

La derivación sucesiva de la solución $y(x)$ aplicando la regla de la cadena proporciona

$$\begin{aligned} y'(x) &= f(x, y(x)), \\ y''(x) &= f'(x, y(x)), \end{aligned}$$

y en general

$$y^{(n)}(x) = f^{(n-1)}(x, y(x)).$$

Al sustituir estos resultados en la expresión (2.45), obtenemos

$$\begin{aligned} y(x_{k+1}) &= y(x_k) + hf(x_k, y(x_k)) + \frac{h^2}{2}f'(x_k, y(x_k)) + \dots \\ &+ \frac{h^n}{n!}f^{(n-1)}(x_k, y(x_k)) + \frac{h^{n+1}}{(n+1)!}f^{(n)}(\zeta_k, y(\zeta_k)). \end{aligned} \quad (2.46)$$

El método de la ecuación en diferencias que surge de la expresión (2.46) es el método de Taylor de orden n y se obtiene suprimiendo el término residual que contiene ζ_k .

$$y_0 = y(x_0), y(x_{k+1}) = y(x_k) + \dots + \frac{h^n}{n!} f^{(n-1)}(x_k, y(x_k)). \quad (2.47)$$

Notamos que el método de Taylor de orden uno es el método de Euler. El error local de discretización en el método de Euler es de orden 2, dado que ese es el orden del error en el polinomio de Taylor utilizado. Este error es el que se comete al avanzar desde el valor real de la función en un punto al siguiente. El error de discretización global, que es el cometido en toda la aproximación de la función, es de orden 1 (siempre es una unidad menos que el local).

El método de Taylor tiene la desventaja de requerir el cálculo y la evaluación de las derivadas de $f(x, y(x))$. Una característica de los métodos de Runge-Kutta es que hacen uso de la función f (cada evaluación funcional de f es una etapa) pero no de sus derivadas, lo que representa una gran ventaja respecto a los esquemas de Taylor. Exponemos a continuación el método de segundo orden de dos etapas:

$$\begin{aligned} y_0 &= y(x_0), \\ y_{k+1} &= y(x_k) + h[c_1 K_1 + c_2 K_2]. \end{aligned} \quad (2.48)$$

El primer término en los corchetes es siempre la pendiente al principio del intervalo, $K_1 = f(x_k, y_k) = y'(x_k)$. En cuanto al segundo,

$$K_2 = f(x_k + \lambda h, y_k + \mu h K_1), \quad (2.49)$$

donde $0 < \lambda \leq 1$. Los parámetros c_1 , c_2 , λ y μ se fijan imponiendo que el algoritmo sea compatible con un desarrollo de Taylor de orden 2, y se ha impuesto la forma $\mu h K_1$ del segundo incremento para facilitar dicha comparación. Desarrollando K_2 ,

$$f(x_k + \lambda h, y_k + \mu h f(x_k, y_k)) = K_1 + \frac{\partial f(x_k, y_k)}{\partial x} \lambda h + \frac{\partial f(x_k, y_k)}{\partial y} \mu h K_1 + O(h^2),$$

resulta

$$y_{k+1} = y_k + h(c_1 + c_2)K_1 + h^2 c_2 \left[\frac{\partial f(x_k, y_k)}{\partial x} \lambda + \frac{\partial f(x_k, y_k)}{\partial y} \mu K_1 \right] + O(h^3). \quad (2.50)$$

Comparamos ahora esta última expresión con el desarrollo de Taylor de $y(x_{k+1}) = y(x_k + h)$,

$$y(x_{k+1}) = y(x_k) + h y'(x_k) + \frac{h^2}{2} y''(x_k) + \dots$$

Sustituyendo $y(x_k)$ por su valor aproximado y_k y notando que

$$y''(x_k) = \frac{\partial f(x_k, y_k)}{\partial x} + \frac{\partial f(x_k, y_k)}{\partial y} f(x_k, y_k)$$

resulta

$$y_{k+1} \approx y_k + h f(x_k, y_k) + \frac{h^2}{2} \left[\frac{\partial f(x_k, y_k)}{\partial x} \lambda + \frac{\partial f(x_k, y_k)}{\partial y} \mu f(x_k, y_k) \right]. \quad (2.51)$$

Comparando las expresiones (2.50) y (2.51), se llega al sistema $c_1 + c_2 = 1$, $c_2 \lambda = 1/2$, $c_2 \mu = 1/2$, que es indeterminado. Dejando libre c_2 , resulta $c_1 = 1 - c_2$, $\lambda = \mu = 1/2c_2$. Si $c_2 = 1$, lo que proporciona un método particular Runge-Kutta, obtenemos $c_1 = 0$ y $\lambda = \mu = 1/2$. En definitiva,

$$y_{k+1} = y_k + h f \left(x_k + \frac{h}{2}, y_k + \frac{h}{2} K_1 \right).$$

Este método de Runge-Kutta de orden dos se conoce como el método de punto medio. Si tomamos $c_1 = c_2 = \frac{1}{2}$ y $\lambda = \mu = 1$ obtenemos otro método de Runge-Kutta de orden dos conocido como método de Euler modificado ó método de trapecio:

$$y_{k+1} = y_k + \frac{h}{2} [f(x_k, y(x_k)) + f(x_k + h, y_k + h K_1)].$$

Otro caso interesante del método de Runge-Kutta de orden 2 es el método de Heun. Se define para los siguientes valores de los parámetros: $c_1 = \frac{1}{4}$, $c_2 = \frac{3}{4}$, $\lambda = \mu = \frac{2}{3}$. La expresión del método es:

$$y_{k+1} = y_k + \frac{h}{4} \left[f(x_k, y(x_k)) + 3f \left(x_k + \frac{2h}{3}, y_k + \frac{2h}{3}K_1 \right) \right].$$

Cabe resaltar que el método de Heun es el método de dos etapas que minimiza el error de aproximación de la solución al PVI. Por otra parte, un método de Runge-Kutta de tercer orden viene dado por

$$y_{k+1} = y_j + h[c_1K_1 + c_2K_2 + c_3K_3],$$

con

$$\begin{aligned} K_1 &= f(x_k, y_k), \\ K_2 &= f(x_k + \lambda h, y_k + \mu h K_1), \\ K_3 &= f(x_k + \lambda_2 h, y_k + \mu_2 K_2 + (\lambda_2 - \mu_2)h K_1). \end{aligned}$$

Los parámetros, c_1 , c_2 , c_3 , λ , μ , λ_2 y μ_2 se calculan desarrollando K_2 y K_3 en serie de Taylor de dos variables hasta orden h^3 , e identificando los factores que multiplican a h , h^2 y h^3 con los correspondientes del desarrollo de Taylor de una variable de $y(x_k + h)$. El sistema de ecuaciones así obtenido es indeterminado, y sus distintas soluciones corresponden a diferentes esquemas Runge-Kutta (ver [9]).

Los métodos de Runge-Kutta de cuarto orden conjugan bien la precisión con el esfuerzo de computación. Uno de ellos (el más conocido) es

$$\begin{aligned} y_{k+1} &= y_k + \frac{h}{6} [K_1 + 2K_2 + 2K_3 + K_4], \\ K_1 &= f(x_k, y_k), \\ K_2 &= f(x_k + \frac{h}{2}, y_k + \frac{h}{2}K_1), \\ K_3 &= f(x_k + \frac{h}{2}, y_k + \frac{h}{2}K_2), \\ K_4 &= f(x_k + h, y_k + hK_3). \end{aligned}$$

2.3.4. Métodos multipaso

La idea de extender el método de Euler para permitir que la solución aproximada en un punto dependa de los valores de la solución y de los valores de las derivadas en varios puntos anteriores fue propuesta originalmente por Bashforth y Adams (1883). La teoría moderna de los métodos multipaso se desarrolló en gran medida por Dahlquist [26] en 1956, fue dada a conocer en los textos de Henrici [46] y [47] publicados en 1962 y 1963.

Los métodos de Adams son métodos multipaso y se pueden clasificar en dos grandes clases: los métodos de Adams-Bashforth y los métodos de Adams-Moulton. Estos se pueden combinar para formar los métodos predictor-corrector de Adams-Bashforth-Moulton. La idea fundamental del método de Adams-Bashforth de n pasos es usar un polinomio de interpolación de $f(x, y(x))$ que pasa por los n puntos: $(x_i, f(x_i, y(x_i))), (x_{i-1}, f(x_{i-1}, y(x_{i-1}))), \dots, (x_{i-n+1}, f(x_{i-n+1}, y(x_{i-n+1})))$. La idea fundamental del método de Adams-Moulton de n pasos es usar un polinomio de interpolación de $f(x, y(x))$ que pasa por los $n+1$ puntos: $(x_{i+1}, f(x_{i+1}, y(x_{i+1}))), (x_i, f(x_i, y(x_i))), (x_{i-1}, f(x_{i-1}, y(x_{i-1}))), \dots, (x_{i-n+1}, f(x_{i-n+1}, y(x_{i-n+1})))$.

Los métodos de Adams-Bashforth de n pasos tienen la forma general

$$y_{i+1} = y_i + \beta h \sum_{j=1}^n \alpha_j f(x_{i-j+1}, y_{i-j+1}(x_{i-j+1})),$$

tienen orden n y sus coeficientes se muestran en la Tabla 2.4.

Tabla 2.4: Coeficientes del método de Adams-Bashforth de n pasos

n	β	α_1	α_2	α_3	α_4
1	1	1			
2	1/2	3	-1		
3	1/12	23	-16	5	
4	1/24	55	-59	37	-9

Los métodos de Adams-Moulton de n pasos tienen la forma general

$$y_{i+1} = y_i + \beta h \sum_{j=0}^n \alpha_j f(x_{i-j+1}, y_{i-j+1}(x_{i-j+1})),$$

son de orden $n + 1$ y sus coeficientes se muestran en la Tabla 2.5. En la práctica los métodos multipaso

Tabla 2.5: Coeficientes del método de Adams-Moulton de n pasos

n	β	α_0	α_1	α_2	α_3
0	1	1			
1	1/2	1	1		
2	1/12	5	8	-1	
3	1/24	9	19	-15	1

implícitos (por ejemplo: el método de Adams-Moulton), no se puede usar directamente. Estos métodos sirven para mejorar las aproximaciones obtenidas con los métodos explícitos. La combinación de un método explícito con un método implícito del mismo orden se denomina método predictor-corrector.

Método predictor-corrector de cuarto orden de Adams-Bashforth-Moulton:

1. La fórmula predictora es la de Adams-Bashforth (explícito):

$$y_{i+1} = y_i + \frac{h}{24}(55f(x_i, y(x_i)) - 59f(x_{i-1}, y(x_{i-1})) + 37f(x_{i-2}, y(x_{i-2})) - 9f(x_{i-3}, y(x_{i-3}))),$$

2. La fórmula correctora es la de Adams-Moulton (implícito):

$$y_{i+1} = y_i + \frac{h}{24}(9f(x_{i+1}, y(x_{i+1})) + 19f(x_i, y(x_i)) - 5f(x_{i-1}, y(x_{i-1})) + f(x_{i-2}, y(x_{i-2})));$$

Nota: Para usar la fórmula predictora se requiere que se conozcan los valores y_0, y_1, y_2, y_3 , para obtener y_4 . Sabemos que y_0 es la condición inicial dada y como el método de Adams-Bashforth-Moulton es de orden 4, los valores y_1, y_2, y_3 se suelen calcular con un método de igual orden, es decir de orden 4, como el método de Runge Kutta de orden 4.

MÉTODOS ITERATIVOS PARA ECUACIONES NO LINEALES BASADOS EN LA CUADRATURA GAUSSIANA

3.1. Introducción

Nuestro objetivo en esta parte de la memoria es encontrar un cero real de una función no lineal $f : I \subseteq \mathbb{R} \rightarrow \mathbb{R}$, es decir, una solución real ξ de la ecuación no lineal

$$f(x) = 0. \quad (3.1)$$

Para ello, emplearemos métodos iterativos (véase Sección 2.1.1) que utilizan evaluaciones funcionales de f y su derivada primera f' . Es bien conocido que la ecuación (3.1) se puede resolver iterativamente por medio del método de punto fijo $x_{k+1} = g(x_k)$, $k = 0, 1, 2, \dots$, donde $g : \mathbb{R} \rightarrow \mathbb{R}$ y x_0 es la estimación inicial. El más conocido método de punto fijo es el método de clásico de Newton (véase (2.8)) que tiene convergencia cuadrática en el entorno de una raíz simple bajo ciertas condiciones ([72]).

El método iterativo de Newton es un método de un sólo paso y representa la referencia clásica para resolver tales ecuaciones. En los últimos años diferentes investigadores han desarrollado nuevos métodos iterativos para resolver la ecuación (3.1) que mejoran, en cierta forma, la precisión del método de Newton. Se han obtenido algunas variantes del método con convergencia de tercer orden que no requieren el cálculo de derivadas de segundo orden como, por ejemplo, en los trabajos de Chun en [10] y de Kou en [57]. En otras clases de métodos se usan distintas técnicas para aumentar el orden de convergencia. Algunas de ellas son: la descomposición de Adomian en [1], la perturbación homotopía en [44] ó el análisis de homotopía [10]. A partir de la identidad

$$f(x) = f(x_k) + \int_{x_k}^x f'(t) dt. \quad (3.2)$$

es facil obtener la expresión iterativa del método de Newton aproximando la integral por el valor del integrando en x_k . Esta idea ha sido empleada por diferentes autores para generar variantes del método de Newton. Weerakoon y Fernando en [82], usando la regla trapezoidal para calcular la integral en (3.2), obtienen la siguiente modificación de método de Newton:

$$x_{k+1} = x_k - \frac{2f(x_k)}{f'(x_k) + f'(x_k - f(x_k)/f'(x_k))}. \quad (3.3)$$

Por otra parte, sobre la base de la regla de punto medio para el cálculo de la integral en (3.2), Frontini y Sormani en [38] y Özban en [73] derivan la fórmula iterativa

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k - f(x_k)/(2f'(x_k)))}, \quad (3.4)$$

Homeier en [45] considerando el método de Newton sobre la función inversa, obtiene el esquema iterativo

$$x_{k+1} = x_k - \frac{f(x_k)}{2} \left[\frac{1}{f'(x_k)} + \frac{1}{f'(x_k - f(x_k)/f'(x_k))} \right]. \quad (3.5)$$

La expresión (3.5) también ha sido derivada de forma independiente por Özban en [73]. Kou, Li y Wang en [57] obtienen un método con la siguiente fórmula iterativa:

$$x_{k+1} = x_k - \frac{f(x_k + f(x_k)/f'(x_k)) - f(x_k)}{f'(x_k)}. \quad (3.6)$$

Cualquiera de los métodos (3.3) a (3.6) no exigen evaluación de derivadas de orden superior a uno. Todos ellos convergen cúbicamente.

En [52], Jiang y Han construyen dos familias de métodos iterativos dependientes de un parámetro basándose en la construcción geométrica de fórmulas iterativas, donde los métodos de Euler, Tschebychev, Halley y Super Halley son casos especiales.

Al estudiar un método iterativo es muy importante considerar dos aspectos: la velocidad de convergencia y el coste computacional del mismo. Los métodos de un sólo paso son muy eficaces, pero su convergencia es lenta y su utilidad en procesos prácticos se ve limitada. Los métodos que aceleran la convergencia intentan transformar una sucesión que converge lentamente en otra que converja más rápido. Una de las técnicas de aceleración conocidas es la composición de dos métodos iterativos con órdenes de convergencia p_1 y p_2 , respectivamente, para obtener un método de orden $p_1 p_2$, según el teorema siguiente (véase [81]):

Teorema 3.1.1 Sean $g_1(x)$ y $g_2(x)$ dos funciones de punto fijo asociadas a la ecuación no lineal $f(x) = 0$. Sean los métodos iterativos $x_{k+1} = g_1(x_k)$ y $x_{k+1} = g_2(x_k)$ de ordenes p_1 y p_2 , respectivamente. Entonces, el orden de convergencia del método iterativo que corresponde a la función de punto fijo $g(x) = g_2(g_1(x))$ es $p_1 p_2$.

Este procedimiento incrementa sustancialmente el número de evaluaciones funcionales por paso. Para mejorar el índice de eficiencia del método compuesto, se introducen aproximaciones que reduzcan el número de evaluaciones funcionales y mantengan el orden de convergencia tan alto como sea posible.

Numerosos trabajos usan esta técnica, obteniendo nuevos métodos, como por ejemplo en [66] Martínez et al. componen la familia de métodos de tercer y cuarto orden obtenida por Kou et al. en [60] con el método de Newton obteniéndose una nueva familia de métodos con orden seis o siete.

Inicialmente, en este capítulo se desarrolla el conjunto de familias de métodos iterativos basado en la cuadratura de Gauss. Dichos métodos iterativos son multipaso (ó predictor-corrector) habiéndose empleado como predictor diferentes métodos, determinando el orden de convergencia para cada caso. La generalización de esta técnica, a la que llamaremos *pseudocomposición*, será analizada en términos de convergencia y eficiencia.

3.2. Método de Newton como predictor

En esta sección se obtiene un conjunto de familias de variantes del método de Newton a partir de la cuadratura de Gauss, y se demuestra que la convergencia es de orden tres (bajo ciertas condiciones impuestas a los polinomios ortogonales que definen la familia de cuadratura gaussiana correspondiente) o cinco, dependiendo del comportamiento en la solución de la derivada segunda de la función que define la ecuación no lineal.

3.2.1. Descripción de los métodos

Sea $f : I \subseteq \mathbb{R} \rightarrow \mathbb{R}$, una función suficientemente diferenciable y $\xi \in I$ un cero simple de la ecuación no lineal $f(x) = 0$. Entonces, para $x, y \in I$, f satisface

$$f(y) - f(x) = \int_x^y f'(t) dt \quad (3.7)$$

Una vez obtenida la k -ésima iteración x_k , sustituyéndola en (3.7) se obtiene:

$$f(y) = f(x_k) + \int_{x_k}^y f'(t) dt. \quad (3.8)$$

Aproximamos la integral en (3.8) utilizando la cuadratura de Gauss:

$$\int_a^b f'(x)dx \approx \frac{b-a}{2} \sum_{i=1}^n \omega_i f' \left(\frac{(b-a)\tau_i + (b+a)}{2} \right),$$

donde ω_i son los pesos de la cuadratura de Gauss y τ_i los nodos correspondientes, $i = 1, \dots, n$ definidos en la Sección 2.2.2. Sustituimos ésta aproximación en (3.8) y obtenemos:

$$f(y) \approx f(x_k) + \frac{y-x_k}{2} \sum_{i=1}^n \omega_i f' \left(\frac{(y-x_k)\tau_i + (y+x_k)}{2} \right).$$

Tomando $y = \xi$,

$$f(\xi) = 0 \approx f(x_k) + \frac{\xi-x_k}{2} \sum_{i=1}^n \omega_i f' \left(\frac{(\xi-x_k)\tau_i + (\xi+x_k)}{2} \right),$$

de donde

$$\xi = x_k - \frac{2f(x_k)}{\sum_{i=1}^n \omega_i f' \left(\frac{(1+\tau_i)\xi + (1-\tau_i)x_k}{2} \right)},$$

y por lo tanto, una nueva aproximación x_{k+1} de ξ viene dada por

$$x_{k+1} = x_k - \frac{2f(x_k)}{\sum_{i=1}^n \omega_i f' \left(\frac{(1+\tau_i)x_{k+1} + (1-\tau_i)x_k}{2} \right)}.$$

Con el fin de evitar el problema implícito que esta ecuación supone predecimos el $(k+1)$ -ésimo iterado del lado derecho de ésta ecuación con el método de Newton y obtenemos la fórmula de iteración del conjunto de familias de variantes del método de Newton que llamaremos Newton-Gauss (NeG)

$$x_{k+1} = x_k - \frac{2f(x_k)}{\sum_{i=1}^n \omega_i f'(\eta_{i,k})}, \quad (3.9)$$

donde

$$\eta_{i,k} = \frac{(1+\tau_i)z_k + (1-\tau_i)x_k}{2}$$

y

$$z_k = x_k - \frac{f(x_k)}{f'(x_k)}.$$

Esta fórmula iterativa es de la forma predictor-corrector, donde el predictor es el método clásico de Newton. Usando las cuadraturas de Gauss-Tschebyshev, Gauss-Legendre, Gauss-Lobatto y Gauss-Radau se obtienen cuatro subfamilias que llamamos Newton-Gauss-Tschebyshev (NeG-T), Newton-Gauss-Legendre (NeG-LD), Newton-Gauss-Lobatto (NeG-LT), y Newton-Gauss-Radau (NeG-R), respectivamente.

Caso I: Cuadratura de Gauss-Tschebyshev. Usando un solo nodo (véase la Tabla 2.3) obtenemos la expresión iterativa del método que denotaremos por NeG-T1:

$$x_{k+1} = x_k - \frac{2}{\pi} \frac{f(x_k)}{f' \left(\frac{x_k+z_k}{2} \right)}. \quad (3.10)$$

Empleando dos y tres nodos obtenemos los métodos NeG-T2 y NeG-T3, cuyas fórmulas iterativas son respectivamente:

$$x_{k+1} = x_k - \frac{4}{\pi} \frac{f(x_k)}{f' \left(\frac{(2+\sqrt{2})}{4} x_k + \frac{(2-\sqrt{2})}{4} z_k \right) + f' \left(\frac{(2-\sqrt{2})}{4} x_k + \frac{(2+\sqrt{2})}{4} z_k \right)}, \quad (3.11)$$

$$x_{k+1} = x_k - \frac{6}{\pi} \frac{f(x_k)}{f' \left(\frac{(2+\sqrt{3})}{4} x_k + \frac{(2-\sqrt{3})}{4} z_k \right) + f' \left(\frac{x_k+z_k}{2} \right) + f' \left(\frac{(2-\sqrt{3})}{4} x_k + \frac{(2+\sqrt{3})}{4} z_k \right)}. \quad (3.12)$$

Caso II: *Cuadratura de Gauss-Legendre.* En este caso, tomando un solo nodo obtenemos la fórmula iterativa que denotaremos por NeG-LD1:

$$x_{k+1} = x_k - \frac{f(x_k)}{f' \left(\frac{x_k+z_k}{2} \right)},$$

que es la misma fórmula iterativa que se obtiene al aproximar la integral (3.8) usando la regla de punto medio (véase la expresión (3.4) en la Sección 3.1). Si usamos dos nodos, obtenemos la variante que denotamos por NeG-LD2:

$$x_{k+1} = x_k - \frac{2f(x_k)}{f' \left(\frac{(3+\sqrt{3})}{6} x_k + \frac{(3-\sqrt{3})}{6} z_k \right) + f' \left(\frac{(3-\sqrt{3})}{6} x_k + \frac{(3+\sqrt{3})}{6} z_k \right)}. \quad (3.13)$$

En el caso del uso de tres nodos, la fórmula iterativa del método NeG-LD3 tendrá la forma:

$$x_{k+1} = x_k - \frac{18f(x_k)}{5f' \left(\frac{(1+\sqrt{3/5})}{2} x_k + \frac{(1+\sqrt{3/5})}{2} z_k \right) + 8f' \left(\frac{x_k+z_k}{2} \right) + 5f' \left(\frac{(1-\sqrt{3/5})}{2} x_k + \frac{(1+\sqrt{3/5})}{2} z_k \right)}. \quad (3.14)$$

Caso III: *Cuadratura de Gauss-Lobatto.* El proceso de obtención de la fórmula de iteración es el mismo de los casos anteriores. Para el caso de un solo nodo, la fórmula iterativa del método que denotaremos por NeG-LT1 será:

$$x_{k+1} = x_k - \frac{f(x_k)}{f' \left(\frac{x_k+z_k}{2} \right)},$$

que coincide con método NeG-LD1 ó el método de punto medio (3.4). En el caso de dos nodos, tenemos la fórmula iterativa NeG-LT2 que es el método de trapecios (3.3):

$$x_{k+1} = x_k - \frac{2f(x_k)}{f'(x_k) + f'(z_k)}.$$

Si utilizamos tres nodos, obtenemos la fórmula iterativa de NeG-LT3:

$$x_{k+1} = x_k - \frac{6f(x_k)}{f'(x_k) + 4f' \left(\frac{x_k+z_k}{2} \right) + f'(z_k)}. \quad (3.15)$$

Ésta expresión es la misma que se obtiene estimando la integral en (3.8) basándose en el método de Simpson en [17].

Caso IV: *Cuadratura de Gauss-Radau.* La fórmula iterativa que aparece al utilizar un solo nodo y que denotaremos por NeG-R1, es:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)},$$

que coincide con el método clásico de Newton. Usando dos y tres nodos obtenemos, respectivamente:

$$x_{k+1} = x_k - \frac{4f(x_k)}{f'(x_k) + 3f' \left(\frac{x_k+2z_k}{3} \right)}, \quad (3.16)$$

$$x_{k+1} = x_k - \frac{36f(x_k)}{(16 + \sqrt{6})f' \left(\frac{(4+\sqrt{6})}{10} x_k + \frac{(6-\sqrt{6})}{10} z_k \right) + 4f'(x_k) + (16 - \sqrt{6})f' \left(\frac{(4-\sqrt{6})}{10} x_k + \frac{(6+\sqrt{6})}{10} z_k \right)}. \quad (3.17)$$

Estas variantes del método de Newton las denotamos por NeG-R2 y NeG-R3, respectivamente. La variante NeG-R2 es el método desarrollado por Noor en [69] utilizando una fórmula de cuadratura cerrada-abierta.

3.2.2. Análisis de la convergencia de los métodos

En primer lugar, introducimos las siguientes notaciones:

$$\sum_{i=1}^n \omega_i = \sigma, \tag{3.18}$$

$$\sum_{i=1}^n \frac{\omega_i \tau_i^j}{\sigma} = \sigma_j \quad \text{con } j = 1, 2, \dots$$

que nos permitirán simplificar las demostraciones del teorema que analiza las condiciones bajo las cuales convergen los métodos descritos.

Teorema 3.2.1 *Sea $\xi \in I$ un cero simple de la función $f : I \subseteq \mathbb{R} \rightarrow \mathbb{R}$, suficientemente diferenciable en el intervalo abierto I y tomemos una estimación inicial x_0 suficientemente cercana a ξ . Consideramos el conjunto de familias iterativas (3.9) con predictor el método clásico de Newton. Entonces,*

1. Si $\sigma = 2$, las variantes del método de Newton tienen convergencia, al menos, de orden dos;
2. Si, además $\sigma_1 = 0$, la convergencia es, al menos, de orden tres;
3. Si, además, $f''(\xi) = 0$ y $\sigma_2 = 1/3$ el orden de convergencia es, al menos, cuatro y
4. Finalmente, si además, $\sigma_3 = 0$, la convergencia será de orden cinco.

Demostración: Como f es suficientemente diferenciable, mediante la expansión de Taylor de $f(x_k)$ y $f'(x_k)$ alrededor de ξ , tenemos:

$$f(x_k) = f'(\xi) [e_k + c_2 e_k^2 + c_3 e_k^3 + c_4 e_k^4 + c_5 e_k^5] + O[e_k^6], \tag{3.19}$$

$$f'(x_k) = f'(\xi) [1 + 2c_2 e_k + 3c_3 e_k^2 + 4c_4 e_k^3 + 5c_5 e_k^4] + O[e_k^5], \tag{3.20}$$

donde $c_j = \frac{1}{j!} \frac{f^{(j)}(\xi)}{f'(\xi)}$, $j = 2, 3, \dots$ y $e_k = x_k - \xi$. Como los términos entre corchetes son polinomios en e_k , la división directa nos da:

$$\begin{aligned} \frac{f(x_k)}{f'(x_k)} &= e_k - c_2 e_k^2 + 2(c_2^2 - c_3) e_k^3 + (7c_2 c_3 - 4c_2^3 - 3c_4) e_k^4 \\ &\quad + (8c_2^4 - 20c_2^2 c_3 + 6c_3^2 + 10c_2 c_4 - 4c_5) e_k^5 + O[e_k^6] \end{aligned} \tag{3.21}$$

y entonces, la expresión del predictor z_k es:

$$\begin{aligned} z_k &= \xi + c_2 e_k^2 - 2(c_2^2 - c_3) e_k^3 - (7c_2 c_3 - 4c_2^3 - 3c_4) e_k^4 \\ &\quad + (-8c_2^4 + 20c_2^2 c_3 - 6c_3^2 - 10c_2 c_4 + 4c_5) e_k^5 + O[e_k^6]. \end{aligned} \tag{3.22}$$

Sean $x_k = \xi + e_k$ y $z_k = \xi + A$ con

$$\begin{aligned} A &= c_2 e_k^2 - 2(c_2^2 - c_3) e_k^3 - (7c_2 c_3 - 4c_2^3 - 3c_4) e_k^4 \\ &\quad + (-8c_2^4 + 20c_2^2 c_3 - 6c_3^2 - 10c_2 c_4 + 4c_5) e_k^5 + O[e_k^6]. \end{aligned}$$

Denotamos por

$$\eta_{i,k} = \left(\frac{(1 + \tau_i) z_k + (1 - \tau_i) x_k}{2} \right),$$

con lo que podemos expresar:

$$\eta_{i,k} - \xi = \frac{1}{2} [(A + e_k) + \tau_i (A - e_k)].$$

Expandiendo $f'(\eta_{i,k})$ alrededor de ξ , obtenemos:

$$f'(\eta_{i,k}) = f'(\xi) [B + C\tau_i + D\tau_i^2 + E\tau_i^3 + F\tau_i^4 + \dots],$$

donde

$$\begin{aligned} B &= 1 + c_2 e_k + \left(c_2^2 - \frac{3}{4}c_3\right) e_k^2 + \left(-2c_2^3 + \frac{7}{2}c_2 c_3 + \frac{1}{2}c_4\right) e_k^3 \\ &+ \left(-\frac{37}{4}c_2^2 c_3 + 5c_2^4 + \frac{9}{2}c_2 c_4 + 3c_3^3 + \frac{5}{16}c_5\right) e_k^4, \\ C &= -c_2 e_k + \left(c_2^2 - \frac{3}{2}c_3\right) e_k^2 + \left(-2c_2^3 + 2c_2 c_3 + \frac{3}{2}c_4\right) e_k^3 \\ &+ \left(-\frac{11}{4}c_2^2 c_3 + 5c_2^4 + \frac{3}{2}c_2 c_4 - \frac{5}{4}c_5\right) e_k^4, \\ D &= \frac{3}{4}c_3 e_k^2 + \left(-\frac{3}{2}c_2 c_3 + \frac{3}{2}c_4\right) e_k^3 + \left(\frac{15}{4}c_2 c_3^2 - 3c_3^3 - \frac{3}{2}c_2 c_4 + \frac{15}{8}c_5\right) e_k^4, \\ E &= -\frac{1}{2}c_4 e_k^3 + \left(\frac{3}{2}c_2 c_4 - \frac{5}{4}c_5\right) e_k^4, \\ F &= \frac{5}{16}c_5 e_k^4. \end{aligned}$$

Desarrollamos a continuación $K = \sum_{i=1}^n \omega_i f'(\eta_{i,k})$ en términos de potencias del error e_k y agrupamos en potencias de τ_i :

$$\begin{aligned} K &= f'(\xi) \left[B \sum_{i=1}^n \omega_i + C \sum_{i=1}^n \omega_i \tau_i + D \sum_{i=1}^n \omega_i \tau_i^2 + E \sum_{i=1}^n \omega_i \tau_i^3 + F \sum_{i=1}^n \omega_i \tau_i^4 \right] + O[e_k^5] \quad (3.23) \\ &= f'(\xi) \left[B \sum_{i=1}^n \omega_i + C \sum_{i=1}^n \omega_i \tau_i \frac{\sum_{i=1}^n \omega_i}{\sum_{i=1}^n \omega_i} + D \sum_{i=1}^n \omega_i \tau_i^2 \frac{\sum_{i=1}^n \omega_i}{\sum_{i=1}^n \omega_i} + E \sum_{i=1}^n \omega_i \tau_i^3 \frac{\sum_{i=1}^n \omega_i}{\sum_{i=1}^n \omega_i} + F \sum_{i=1}^n \omega_i \tau_i^4 \frac{\sum_{i=1}^n \omega_i}{\sum_{i=1}^n \omega_i} \right] \\ &+ O[e_k^5]. \end{aligned}$$

O equivalentemente, usando la notación (3.18) obtenemos

$$\begin{aligned} K &= f'(\xi) [B\sigma + C\sigma\sigma_1 + D\sigma\sigma_2 + E\sigma\sigma_3 + F\sigma\sigma_4] + O[e_k^5] \\ &= \sigma f'(\xi) [B + C\sigma_1 + D\sigma_2 + E\sigma_3 + F\sigma_4] + O[e_k^5]. \end{aligned}$$

A continuación, calculamos:

$$\begin{aligned} L &= \frac{2f(x_k)}{K} = \frac{2f'(\xi) [e_k + c_2 e_k^2 + c_3 e_k^3 + c_4 e_k^4 + c_5 e_k^5] + O[e_k^6]}{\sigma f'(\xi) [B + C\sigma_1 + D\sigma_2 + E\sigma_3 + F\sigma_4] + O[e_k^5]} \\ &= \frac{2}{\sigma} e_k + \frac{2}{\sigma} \sigma_1 c_2 e_k^2 + \frac{2}{\sigma} \left((-1 - 2\sigma_1 + \sigma_2) c_2^2 + \frac{1}{4} (1 + 6\sigma_1 - 3\sigma_2) c_3 \right) e_k^3 \\ &+ \frac{2}{\sigma} \left((3 + 2\sigma_1 - 4\sigma_2 + \sigma_3) c_2^3 - \frac{1}{4} (15 + 16\sigma_1 - 21\sigma_2 + 6\sigma_3) c_2 c_3 \right) e_k^4 \\ &+ \frac{2}{\sigma} \left(\frac{1}{2} (1 + 3\sigma_1 - 3\sigma_2 + \sigma_3) c_4 \right) e_k^4 \\ &+ \frac{2}{\sigma} \left((-6 + 2\sigma_1 + 9\sigma_2 - 6\sigma_3 + \sigma_4) c_2^4 - \frac{1}{4} (-54 - 2\sigma_1 + 75\sigma_2 - 48\sigma_3 + 9\sigma_4) c_2^2 c_3 \right) e_k^5 \\ &+ \frac{2}{\sigma} \left(\frac{3}{16} (-17 - 4\sigma_1 + 30\sigma_2 - 12\sigma_3 + 3\sigma_4) c_3^2 + (-5 - 3\sigma_1 + 6\sigma_2 - 5\sigma_3 + \sigma_4) c_2 c_4 \right) e_k^5 \\ &+ \frac{2}{\sigma} \left(\frac{1}{16} (11 + 20\sigma_1 - 30\sigma_2 + 20\sigma_3 - 5\sigma_4) c_5 \right) e_k^5 + O[e_k^6]. \end{aligned}$$

La ecuación del error queda:

$$\begin{aligned}
 e_{k+1} &= e_k - L \\
 &= e_k - \frac{2}{\sigma}e_k - \frac{2}{\sigma}\sigma_1c_2e_k^2 - \frac{2}{\sigma}\left((-1 - 2\sigma_1 + \sigma_2)c_2^2 + \frac{1}{4}(1 + 6\sigma_1 - 3\sigma_2)c_3\right)e_k^3 \\
 &\quad - \frac{2}{\sigma}\left((3 + 2\sigma_1 - 4\sigma_2 + \sigma_3)c_2^3 - \frac{1}{4}(15 + 16\sigma_1 - 21\sigma_2 + 6\sigma_3)c_2c_3\right)e_k^4 \\
 &\quad - \frac{2}{\sigma}\left(\frac{1}{2}(1 + 3\sigma_1 - 3\sigma_2 + \sigma_3)c_4\right)e_k^4 \\
 &\quad - \frac{2}{\sigma}\left((-6 + 2\sigma_1 + 9\sigma_2 - 6\sigma_3 + \sigma_4)c_2^4 - \frac{1}{4}(-54 - 2\sigma_1 + 75\sigma_2 - 48\sigma_3 + 9\sigma_4)c_2^2c_3\right)e_k^5 \\
 &\quad - \frac{2}{\sigma}\left(\frac{3}{16}(-17 - 4\sigma_1 + 30\sigma_2 - 12\sigma_3 + 3\sigma_4)c_3^2 + (-5 - 3\sigma_1 + 6\sigma_2 - 5\sigma_3 + \sigma_4)c_2c_4\right)e_k^5 \\
 &\quad - \frac{2}{\sigma}\left(\frac{1}{16}(11 + 20\sigma_1 - 30\sigma_2 + 20\sigma_3 - 5\sigma_4)c_5\right)e_k^5 + O[e_k^6]. \tag{3.24}
 \end{aligned}$$

Según (3.24), en general, todas las variantes del método clásico de Newton desarrolladas mediante la aproximación de la integral en (3.8) usando cuadraturas de Gauss, tendrán convergencia, al menos, lineal. Si imponemos la condición $\sigma = 2$, el orden de convergencia de los métodos obtenidos será, al menos 2, en cuyo caso la expresión (3.24) se puede reescribir como:

$$\begin{aligned}
 e_{k+1} &= -\sigma_1c_2e_k^2 - \left((-1 - 2\sigma_1 + \sigma_2)c_2^2 + \frac{1}{4}(1 + 6\sigma_1 - 3\sigma_2)c_3\right)e_k^3 \\
 &\quad - \left((3 + 2\sigma_1 - 4\sigma_2 + \sigma_3)c_2^3 - \frac{1}{4}(15 + 6\sigma_1 - 21\sigma_2 + 6\sigma_3)c_2c_3 + \frac{1}{2}(1 + 3\sigma_1 - 3\sigma_2 + \sigma_3)c_4\right)e_k^4 \\
 &\quad - \left((-6 + 2\sigma_1 + 9\sigma_2 - 6\sigma_3 + \sigma_4)c_2^4 - \frac{1}{4}(-54 - 2\sigma_1 + 75\sigma_2 - 48\sigma_3 + 9\sigma_4)c_2^2c_3\right)e_k^5 \\
 &\quad - \left(\frac{3}{16}(-17 - 4\sigma_1 + 30\sigma_2 - 12\sigma_3 + 3\sigma_4)c_3^2 + (-5 - 3\sigma_1 + 6\sigma_2 - 5\sigma_3 + \sigma_4)c_2c_4\right)e_k^5 \\
 &\quad - \frac{1}{16}(11 + 20\sigma_1 - 30\sigma_2 + 20\sigma_3 - 5\sigma_4)c_5e_k^5 + O[e_k^6]. \tag{3.25}
 \end{aligned}$$

Notemos que si $\sigma_1 = 0$, tenemos convergencia cúbica:

$$\begin{aligned}
 e_{k+1} &= \left((1 - \sigma_2)c_2^2 - \frac{1}{4}(1 - 3\sigma_2)c_3\right)e_k^3 \\
 &\quad - \left((3 - 4\sigma_2 + \sigma_3)c_2^3 - \frac{1}{4}(15 - 21\sigma_2 + 6\sigma_3)c_2c_3 + \frac{1}{2}(1 - 3\sigma_2 + \sigma_3)c_4\right)e_k^4 \\
 &\quad - \left((-6 + 9\sigma_2 - 6\sigma_3 + \sigma_4)c_2^4 - \frac{1}{4}(-54 + 75\sigma_2 - 48\sigma_3 + 9\sigma_4)c_2^2c_3\right)e_k^5 \\
 &\quad - \left(\frac{3}{16}(-17 + 30\sigma_2 - 12\sigma_3 + 3\sigma_4)c_3^2 + (-5 + 6\sigma_2 - 5\sigma_3 + \sigma_4)c_2c_4\right)e_k^5 \\
 &\quad - \frac{1}{16}(11 - 30\sigma_2 + 20\sigma_3 - 5\sigma_4)c_5e_k^5 + O[e_k^6].
 \end{aligned}$$

Para que los métodos tengan convergencia superior a tres, además de que $\sigma = 2$ y $\sigma_1 = 0$, se debe cumplir:

i) $c_2 = 0$, (es decir, $f''(\xi) = 0$) y $\sigma_2 = 1/3$; en cuyo caso la convergencia será de orden cuatro:

$$e_{k+1} = -\frac{1}{2}\sigma_3c_4e_k^4 + \frac{1}{16}((21 + 36\sigma_3 - 9\sigma_4)c_3^2 - (1 + 20\sigma_3 - 5\sigma_4)c_5)e_k^5 + O[e_k^6].$$

ii) Si, además, $\sigma_3 = 0$ el método tendrá orden de convergencia cinco con ecuación del error

$$e_{k+1} = \frac{1}{16}((21 - 9\sigma_4)c_3^2 - (1 - 5\sigma_4)c_5)e_k^5 + O[e_k^6].$$

□

Si analizamos las diferentes cuadraturas de Gauss utilizadas en esta sección (véase Tabla 3.1), podemos observar que los pesos y nodos de Gauss-Tschebyshev no cumplen ninguna de las tres condiciones del Teorema 3.2.1, por lo que los métodos iterativos obtenidos en las expresiones (3.10), (3.11) y (3.12) serán tan solo de orden 1. En cuanto al resto de cuadraturas, Gauss-Legendre, Gauss-Lobatto y Gauss-Radau, sus nodos y pesos cumplen las condiciones 1 y 2 del teorema anterior, por lo que los métodos iterativos a los que dan lugar son métodos que tienen orden de convergencia 3, excepto el que se deduce de la cuadratura de Gauss-Radau con un único nodo que tiene orden 2, ya que, como hemos visto, el método iterativo que se obtiene es el método de Newton. Las condiciones 3 y 4 del Teorema 3.2.1 se cumplen por las variantes NeG-LD2, NeG-LT3, NeG-R2 y NeG-R3 y alcanzarán orden de convergencia 5 si se cumple la condición $f''(\xi) = 0$. La variante NeG-R1 (Newton) no cumple las condiciones 2 y 3, pero el orden de convergencia aumentará hasta 3 si $f'''(\xi) = 0$ (véase (3.25)).

Tabla 3.1: Valores de σ , σ_1 , σ_2 y σ_3 para las distintas cuadraturas de Gauss

Cuadratura	nodos	σ	σ_1	σ_2	σ_3
Gauss-Tshebychev	1	π	0	0	0
	2	π	0	1/2	0
	3	π	0	1/2	0
Gauss-Legendre	1	2	0	0	0
	2	2	0	1/3	0
	3	2	0	1/15	0
Gauss-Lobatto	1	2	0	0	0
	2	2	0	1	0
	3	2	0	1/3	0
Gauss-Radau	1	2	-1	0	0
	2	2	0	1/3	0
	3	2	0	1/3	0

En la Tabla 3.2 se presentan el orden de convergencia p , el número de evaluaciones funcionales d por iteración y el número de operaciones op (número total de productos y cocientes por iteración) así mismo el índice de eficiencia I , índice de eficiencia operacional IO y el índice de eficiencia computacional IC de algunos métodos del conjunto de familias NeG: NeG-LD, NeG-LT y NeG-R todos con 1, 2 y 3 nodos.

Tabla 3.2: Características de los métodos iterativos obtenidos

Métodos	p	d	op	I	IO	IC
NeG-LD1	3	3	2	1.4422	1.7321	1.2457
NeG-LD2	3	4	2	1.3161	1.7321	1.2009
NeG-LD3	3	5	2	1.2457	1.7321	1.1699
NeG-LT1	3	3	2	1.4422	1.7321	1.2457
NeG-LT2	3	3	2	1.4422	1.7321	1.2457
NeG-LT3	3	4	2	1.3161	1.7321	1.2009
NeG-R1(NC)	2	2	1	1.4142	2.0000	1.2599
NeG-R2	3	3	2	1.4422	1.7321	1.2457
NeG-R3	3	4	2	1.3161	1.7321	1.2009

Según la definición de Ostrowski, el índice de eficiencia es $I = p^{\frac{1}{d}}$. Examinando el conjunto de familias de variantes del método clásico de Newton de la Tabla 3.2 vemos que se obtiene orden de convergencia, en caso general tres, que es mayor que el del método clásico de Newton, pero no es óptimo según la conjetura de Kung-Traub. Las variantes: NeG-LD1 y NeG-LT1 (punto medio), NeG-LT2 (trapezios) y NeG-R2 (Noor) tienen tres evaluaciones funcionales por iteración - dos derivadas y una función, y sus índices de eficiencia tienen el mismo valor, 1.4422 que siendo no óptimo según la conjetura de Kung-Traub es superior al de Newton y mayor que de las restantes variantes por tener éstas mayor número de evaluaciones funcionales. El método iterativo NeG-LD3 con cinco evaluaciones funcionales tiene el menor índice de eficiencia $I = 1.2457$. Todos los métodos

de la familia NeG tienen igual índice operacional $IO = 1.7321$ excepto el método NeG-R1 que es el método de Newton.

Los métodos iterativos NeG-LD1, NeG-LT1 y 2 y NeG-R2 tienen el mismo índice de eficiencia computacional $IC = 1.2457$ muy cerca del índice del método de Newton $IC = 1.2599$.

Es importante observar que el orden de convergencia de los métodos resultantes es, en general, independiente del número de nodos utilizado en la fórmula de cuadratura. Es por ello que, desde el punto de vista del índice de eficiencia, no resulta interesante utilizar más de dos o tres nodos, pues en caso contrario el orden no aumenta pero sí lo hace el número de evaluaciones funcionales, con lo que se disminuye la eficiencia del método. En las pruebas numéricas más adelante serán utilizados por estas razones sólo las variantes NeG-LD1 (NeG-LT1), NeG-LD2, NeG-LT2 y NeG-R2.

3.2.3. Resultados numéricos

A continuación presentamos algunos ejemplos para ilustrar la eficacia de los métodos desarrollados y compararlos entre sí (NeG-LD1, NeG-LD2, NeG-LT2 y NeG-R2), con el fin de calcular los ceros de las siguientes funciones no lineales.

1. $f_1(x) = \sin x - x^2 + 1$, $\xi \approx 1.404492$.
2. $f_2(x) = x^2 - \exp x - 3x + 2$, $\xi \approx 0.257530$.
3. $f_3(x) = \cos x - x$, $\xi \approx 0.739085$.
4. $f_4(x) = (x - 1)^3 - 1$, $\xi = 2$.
5. $f_5(x) = x^3 - 10$, $\xi \approx 2.154435$.
6. $f_6(x) = \cos x - x \exp x + x^2$, $\xi \approx 0.639154$.
7. $f_7(x) = \exp x - 1.5 - \arctan x$, $\xi \approx 0.767653$.
8. $f_8(x) = x^3 + 4x^2 - 10$, $\xi \approx 1.365230$.
9. $f_9(x) = 8x - \cos x - 2x^2$, $\xi \approx 0.128077$.
10. $f_{10}(x) = \arctan x$, $\xi = 0$.
11. $f_{11}(x) = \exp x - 4x^2$, $\xi \approx 0.714806$.
12. $f_{12}(x) = \left(\sin x - \frac{x}{2}\right)^2$, $\xi = 0$, doble.
13. $f_{13}(x) = x \exp x^2 - \sin x^2 + 3 \cos x + 5$, $\xi \approx -1.207648$.
14. $f_{14}(x) = \sqrt{x^2 + 2x + 5} - 2 \sin x - x^2 + 3$, $\xi \approx 2.331968$.
15. $f_{15}(x) = x^4 + \sin \frac{\pi}{x^2} - 5$, $\xi = \sqrt{2}$.
16. $f_{16}(x) = 10x \exp(-x^2) - 1$, $\xi \approx 1.679631$.
17. $f_{17}(x) = \exp(-x) + \cos x$, $\xi \approx 1.746140$.
18. $f_{18}(x) = \sqrt{x^4 + 8} \sin \frac{\pi}{x^2 + 2} + \frac{x^3}{x^4 + 1} - \sqrt{6} + \frac{8}{17}$, $\xi \approx -2$.

Para su comparación usaremos el método clásico de Newton. Asimismo utilizaremos el método de Traub (TR) [81] conocido también en la bibliografía como método de Potra-Pták [74]

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)} \\ x_{k+1} &= x_k - \frac{f(x_k) + f(y_k)}{f'(x_k)}, \end{aligned} \quad (3.26)$$

con convergencia de orden 3 y el método de Jarratt (JT) [51],

$$\begin{aligned} y_k &= x_k - \frac{2}{3} \frac{f(x_k)}{f'(x_k)} \\ x_{k+1} &= x_k - \frac{1}{2} \frac{3f'(y_k) + f'(y_k) \frac{f(x_k)}{f'(x_k)}}{3f'(y_k) - f'(y_k) \frac{f(x_k)}{f'(x_k)}}, \end{aligned} \quad (3.27)$$

que tiene orden de convergencia 4. Los cálculos numéricos se han realizado en MATLAB 7.1, utilizando aritmética de precisión variable que usa representación en coma flotante de 2000 dígitos decimales de mantisa.

Tabla 3.3: Comparación de los índices de los métodos iterativos

Métodos	p	d	op	I	IO	IC
NeG-LD1	3	3	2	1.4422	1.7321	1.2457
NeG-LD2	3	4	2	1.3161	1.7321	1.2009
NeG-LT2	3	3	2	1.4422	1.7321	1.2457
NeG-R2	3	3	2	1.4422	1.7321	1.2457
NC	2	2	1	1.4142	2.0000	1.2599
TR	3	3	2	1.4422	1.7321	1.2457
JT	4	3	3	1.5874	1.5874	1.2599

Según los resultados de la Tabla 3.3 entre los métodos presentados el mayor índice de eficiencia lo tiene el método de Jarratt con $I = 1.5874$ y le siguen los métodos NeG-LD1, NeG-LT2, NeG-R2 y Traub con $I = 1.4422$. El mejor índice de eficiencia operacional lo tiene Newton (2.0000) y le siguen todos los restantes métodos (1.7321) excepto el método de Jarratt que tiene $IO = 1.5874$. Computacionalmente los métodos más efectivos son Newton y Jarratt ($IC = 1.2599$), menos efectivo es el método NeG-LD1 con $IC = 1.2009$ y todos los restantes métodos tienen igual índice de eficiencia computacional ($IC = 1.2457$). Se puede decir, que los métodos de la familia de NeG no son óptimos según la conjetura de Kung-Traub, operacionalmente son mejores que el método de Jarratt y su eficiencia computacional está muy cerca de la eficiencia computacional de los dos mejores métodos en este sentido: Newton y Jarratt. Podemos ordenar los métodos según su índice de eficiencia computacional de la manera siguiente:

$$\begin{aligned} IC_{\text{NeG-LD2}} &= 1.2009 < IC_{\text{NeG-LD1}} = IC_{\text{NeG-LT2}} = IC_{\text{NeG-R2}} \\ &= IC_{\text{TR}} = 1.2457 < IC_{\text{NC}} = IC_{\text{JT}} = 1.2599. \end{aligned}$$

El criterio de parada utilizado es $|x_{k+1} - x_k| + |f(x_k)| < 10^{-400}$. Por tanto, estamos garantizando que los iterados convergen a un límite y que ese límite es una solución de la ecuación no lineal. En la Tabla 3.4 presentamos los resultados obtenidos al utilizar los métodos anteriormente descritos para estimar los ceros de las funciones de (1) a (18). Para cada ejemplo mostramos los siguientes datos: la aproximación inicial x_0 y, para cada método, el número de iteraciones necesario para obtener $|x_{k+1} - x_k| + |f(x_k)| < \varepsilon$, el valor de la función f en la última iteración, la distancia entre las dos últimas iteraciones, el tiempo medio transcurrido (ϵ) después de 100 actuaciones del programa, calculado mediante el comando "tic, . . . , toc" de Matlab, y el orden de convergencia p (véase [17]) aproximado por la fórmula:

$$p \approx \rho = \frac{\ln(|x_{k+1} - x_k|)/(|x_k - x_{k-1}|)}{\ln(|x_k - x_{k-1}|)/(|x_{k-1} - x_{k-2}|)}. \quad (3.28)$$

Tabla 3.4: Ejemplos numéricos

	Métodos	ξ	$ x_{k+1} - x_k $	$ f(x_{k+1}) $	Iter	ρ	ϵ, s
$f_1,$ $x_0 = 1$	NeG-LD1	1.409624	$1.9e-130$	0	6	3.0000	0.2255
	NeG-LD2	1.409624	$8.6e-130$	0	6	3.0000	0.2932
	NeG-LT2	1.409624	$1.6e-128$	0	6	3.0000	0.2479
	NeG-R2	1.409624	$1.1e-129$	0	6	3.0000	0.2381
	NC	1.409624	$1.9e-273$	0	10	2.0000	0.2990
	TR	1.409624	$1.0e-198$	0	7	3.0000	0.2373
	JT	1.409624	0	0	5	4.0000	0.2290
$f_2,$ $x_0 = 1$	NeG-LD1	0.257530	$4.9e-257$	0	6	3.0000	0.2588
	NeG-LD2	0.257530	$1.1e-119$	0	5	3.0000	0.3036
	NeG-LT2	0.257530	$3.7e-202$	0	6	3.0000	0.2892
	NeG-R2	0.257530	$1.2e-120$	0	5	3.0000	0.2551
	NC	0.257530	$4.5e-190$	0	9	2.0000	0.2616
	TR	0.257530	$3.7e-125$	0	5	3.0000	0.2211
	JR	0.257530	0	0	4	4.0000	0.2710
$f_3,$ $x_0 = 1$	NeG-LD1	0.739085	$1.3e-296$	0	6	3.0000	0.2176
	NeG-LD2	0.739085	$1.5e-316$	0	6	3.0000	0.3003
	NeG-LT2	0.739085	$8.0e-148$	0	5	3.0000	0.1999
	NeG-R2	0.739085	$6.1e-317$	0	5	3.0000	0.2688
	NC	0.739085	$7.1e-167$	0	8	2.0000	0.2105
	TR	0.739085	$4.3e-284$	0	6	3.0000	0.2198
	JT	0.739085	0	0	5	4.0000	0.2487
$f_4,$ $x_0 = 2.5$	NeG-LD1	2.000000	$8.1e-110$	0	6	3.0000	0.2219
	NeG-LD2	2.000000	$3.4e-318$	0	7	3.0000	0.3395
	NeG-LT2	2.000000	$1.6e-300$	0	7	3.0000	0.2916
	NeG-R2	2.000000	$3.4e-318$	0	7	3.0000	0.3031
	NC	2.000000	$7.9e-224$	0	10	2.0000	0.2903
	TR	2.000000	$6.3e-250$	0	7	3.0000	0.2748
	JT	2.000000	0	0	5	4.0000	0.2390
$f_5,$ $x_0 = 2$	NeG-LD1	2.154435	$3.0e-278$	0	6	3.0000	0.2809
	NeG-LD2	2.154435	$1.9e-273$	0	6	3.0000	0.3663
	NeG-LT2	2.154435	$4.1e-265$	0	6	3.0000	0.3058
	NeG-R2	2.154435	$1.9e-273$	0	7	3.0000	0.3264
	NC	2.154435	$4.5e-288$	0	9	2.0000	0.3583
	TR	2.154435	$3.5e-233$	0	6	3.0000	0.2866
	JT	2.154435	0	0	5	4.0000	0.3014
$f_6,$ $x_0 = 1$	NeG-LD1	0.639154	$4.7e-152$	0	6	3.0000	0.2809
	NeG-LD2	0.639154	$3.1e-144$	0	6	3.0000	0.3663
	NeG-LT2	0.639154	$1.3e-131$	0	6	3.0000	0.3058
	NeG-R2	0.639154	$4.7e-144$	0	6	3.0000	0.3264
	NC	0.639154	$2.3e-303$	0	10	2.0000	0.3583
	TR	0.639154	$4.5e-177$	0	6	3.0000	0.3736
	JR	0.639154	0	0	5	4.0000	0.3014
$f_7,$ $x_0 = 1$	NeG-LD1	0.767653	$5.0e-184$	0	6	3.0000	0.2770
	NeG-LD2	0.767653	$6.0e-181$	0	6	3.0000	0.4049
	NeG-LT2	0.767653	$2.8e-175$	0	6	3.0000	0.3089
	NeG-R2	0.767653	$6.7e-181$	0	7	3.0000	0.3110
	NC	0.767653	$1.4e-190$	0	9	2.0000	0.3167
	TR	0.767653	$5.2e-152$	0	6	3.0000	0.2883
	JT	0.767653	0	0	5	4.0000	0.2699

Tabla 3.4 Ejemplos numéricos, continuación....

	Métodos	ξ	$ x_{k+1} - x_k $	$ f(x_{k+1}) $	<i>Iter</i>	ρ	ϵ
$f_8,$ $x_0 = 1$	NeG-LD1	1.365230	$7.5e-171$	0	6	3.0000	0.3386
	NeG-LD2	1.365230	$4.0e-167$	0	6	3.0000	0.3951
	NeG-LT2	1.365230	$2.0e-160$	0	6	3.0000	0.3530
	NeG-R2	1.365230	$4.0e-167$	0	6	3.0000	0.3428
	NC	1.365230	$4.0e-176$	0	9	2.0000	0.3681
	TR	1.365230	$2.9e-117$	0	6	3.0000	0.3390
	JT	1.365230	0	0	5	4.0000	0.3062
$f_9,$ $x_0 = 1$	NeG-LD1	0.128077	$9.0e-159$	0	6	3.0000	0.3897
	NeG-LD2	0.128077	$5.9e-161$	0	6	3.0000	0.4246
	NeG-LT2	0.128077	$1.5e-165$	0	6	3.0000	0.4025
	NeG-R2	0.128077	$1.9e-161$	0	6	3.0000	0.3357
	NC	0.128077	$9.0e-169$	0	9	2.0000	0.4161
	TR	0.128077	$6.8e-318$	0	7	3.0000	0.3252
	JT	0.128077	0	0	5	4.0000	0.4006
$f_{10},$ $x_0 = 1$	NeG-LD1	0.000000	$1.8e-310$	0	7	3.0000	0.2545
	NeG-LD2	0.000000	$4.0e-195$	0	5	5.0000	0.3175
	NeG-LT2	0.000000	$7.2e-243$	0	7	3.0000	0.3180
	NeG-R2	0.000000	$1.6e-142$	0	5	5.0000	0.2059
	NC	0.000000	$9.7e-249$	0	8	3.0000	0.2544
	TR	0.000000	$8.9e-238$	0	6	5.0000	0.2875
	JT	0.000000	0	0	5	5.0000	0.2202
$f_{11},$ $x_0 = 1$	NeG-LD1	0.714806	$3.9e-178$	0	6	3.0000	0.3285
	NeG-LD2	0.714806	$2.4e-180$	0	6	3.0000	0.4084
	NeG-LT2	0.714806	$4.1e-185$	0	6	3.0000	0.2920
	NeG-R2	0.714806	$2.1e-180$	0	6	3.0000	0.4048
	NC	0.714806	$5.3e-190$	0	9	2.0000	0.3585
	TR	0.714806	$2.3e-152$	0	6	3.0000	0.3173
	JT	0.714806	0	0	5	4.0000	0.2685
$f_{12},$ $x_0 = 0.5$	NeG-LD1	0.000000	$4.5e-162$	0	338	1.0000	22.0802
	NeG-LD2	0.000000	$4.2e-162$	0	338	1.0000	31.3793
	NeG-LT2	0.000000	$3.4e-162$	0	338	1.0000	24.8764
	NeG-R2	0.000000	$3.1e-162$	0	338	1.0000	27.4508
	NC	0.000000	$1.6e-162$	0	536	1.0000	26.3145
	TR	0.000000	$3.1e-162$	0	376	1.0000	30.6062
	JT	0.000000	0	0	267	1.0000	24.7335
$f_{13},$ $x_0 = -1$	NeG-LD1	-1.201576	$5.8e-150$	0	6	3.0000	0.4220
	NeG-LD2	-1.201576	$3.6e-132$	0	6	3.0000	0.5278
	NeG-LT2	-1.201576	0	0	7	3.0000	0.4916
	NeG-R2	-1.201576	$1.5e-133$	0	6	3.0000	0.4407
	NC	-1.201576	$9.6e-278$	0	10	2.0000	0.4728
	TR	-1.201576	$2.0e-245$	0	7	3.0000	0.4996
	JT	-1.201576	0	0	5	4.0000	0.4417
$f_{14},$ $x_0 = 2$	NeG-LD1	2.331968	$1.8e-110$	0	5	3.0000	0.5477
	NeG-LD2	2.331968	$1.1e-132$	0	5	3.0000	0.8154
	NeG-LT2	2.331968	$5.3e-275$	0	6	3.0000	0.6666
	NeG-R2	2.331968	$1.4e-137$	0	5	3.0000	0.6134
	NC	2.331968	$3.7e-211$	0	8	2.0000	0.5500
	TR	2.331968	$6.4e-123$	0	5	3.0000	0.5980
	JT	2.331968	0	0	4	4.0000	0.5150

Tabla 3.4 Ejemplos numéricos, continuación...

	Métodos	ξ	$ x_{k+1} - x_k $	$ f(x_{k+1}) $	<i>Iter</i>	ρ	ϵ
$f_{15},$ $x_0 = 1$	NeG-LD1	1.414214	$1.5e-135$	0	6	3.0000	0.3774
	NeG-LD2	1.414214	$1.6e-214$	0	6	3.0000	0.5247
	NeG-LT2	1.414214	$1.4e-114$	0	6	3.0000	0.4110
	NeG-R2	1.414214	$1.0e-182$	0	6	3.0000	0.4365
	NC	1.414214	$2.9e-215$	0	9	2.0000	0.4070
	TR	1.414214	$4.9e-214$	0	6	3.0000	0.4322
	JT	1.414214	0	0	5	4.0000	0.3990
$f_{16},$ $x_0 = 2$	NeG-LD1	1.679631	$2.0e-309$	0	7	3.0000	0.3599
	NeG-LD2	1.679631	$2.2e-300$	0	7	3.0000	0.4938
	NeG-LT2	1.679631	$1.5e-286$	0	7	3.0000	0.3846
	NeG-R2	1.679631	$1.3e-297$	0	7	3.0000	0.3718
	NC	1.679631	$4.7e-219$	0	10	2.0000	0.4178
	TR	1.679631	$2.0e-188$	0	8	3.0000	0.3779
	JT	1.679631	0	0	5	4.0000	0.3308
$f_{17},$ $x_0 = 2$	NeG-LD1	1.7461395	$4.1e-287$	0	6	3.0000	0.2312
	NeG-LD2	1.7461395	$1.0e-332$	0	6	3.0000	0.2946
	NeG-LT2	1.7461395	$2.2e-111$	0	5	3.0000	0.2208
	NeG-R2	1.7461395	$1.5e-321$	0	6	3.0000	0.2510
	NC	1.7461395	$9.6e-170$	0	8	2.0000	0.2394
	TR	1.7461395	$2.9e-279$	0	6	3.0000	0.2390
	JT	1.7461395	0	0	5	4.0000	0.2392
$f_{18},$ $x_0 = -1.8$	NeG-LD1	≈ -2	$2.5e-200$	0	6	3.0000	0.7711
	NeG-LD2	≈ -2	$3.6e-226$	0	6	3.0000	1.2243
	NeG-LT2	≈ -2	$8.6e-109$	0	5	3.0000	0.8089
	NeG-R2	≈ -2	$3.4e-227$	0	6	3.0000	1.0544
	NC	≈ -2	$1.1e-239$	0	9	2.0000	0.9038
	TR	≈ -2	$1.5e-174$	0	6	3.0000	1.1127
	JT	≈ -2	0	0	5	4.0000	1.0364

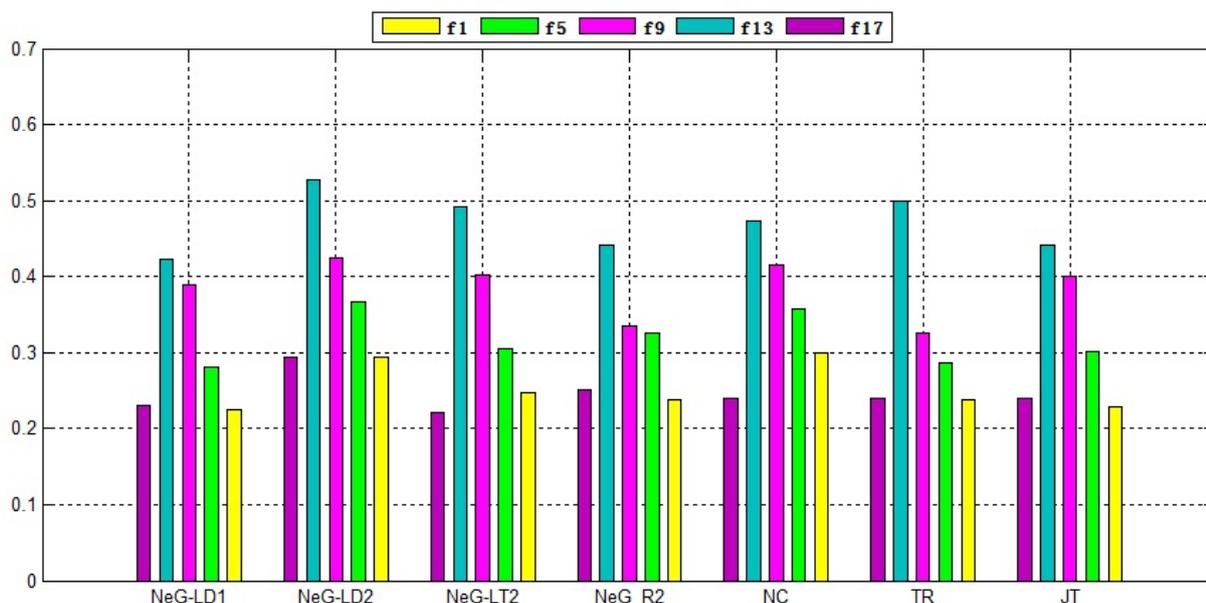


Figura 3.1: E-time de las funciones 1, 5, 9, 13 y 17

Los resultados numéricos obtenidos están en concordancia con la teoría desarrollada en esta sección. Notamos dos casos particulares: $f_{10}(x) = 0$ y $f_{12}(x) = 0$. En el caso de $f_{10}(x) = 0$ notamos que los métodos NGLD2 y NGLT2 conservan su orden de convergencia y todos los restantes métodos tienen orden de convergencia mejor que en otros casos. Estudiando esta particularidad, resulta que $f''_{10}(\xi) = 0$. Por lo tanto, dado que los métodos NeG-LD2 y NeG-LT2 no cumplen con las condiciones 3 y 4 del Teorema 3.2.1 su orden de convergencia es tres. Sin embargo, los métodos NeG-LD1 y NeG-R2 tienen orden de convergencia cinco, el método de Newton orden tres y los métodos de Traub y Jarratt órdenes 4 y 5, respectivamente. En el caso de $f_{12}(x) = 0$, que tiene una raíz doble, todos los métodos estudiados muestran orden de convergencia uno con elevado número de iteraciones. El método de Jarratt emplea el menor número de iteraciones (267) y obtiene mayor precisión que los restantes métodos, mientras el método de Newton alcanza la raíz con la mayor cantidad de iteraciones (536).

La Figura 3.1 presenta las gráficas de ϵ obtenidos con todos los métodos presentados en la Tabla 3.4 para cinco de las funciones estudiadas f_1 , f_5 , f_9 , f_{13} y f_{17} , escogidas al azar. En general, el menor ϵ para alcanzar los resultados mostrados en Tabla 3.4 lo necesitan los métodos NeG-LD1 y Jarratt, y el mayor NeG-LD2. Estos resultados coinciden con los índices de eficiencia mostrados en la Tabla 3.3. La excepción es el método de Newton. Tales resultados no concuerdan por la mayor cantidad ($\approx 50\%$ más) de iteraciones que necesita el método de Newton.

3.3. Métodos iterativos basados en cuadratura de Gauss con otros predictores

En la sección anterior se obtuvieron métodos iterativos usando el método de Newton como predictor y corrigiendo el resultado obtenido con la cuadratura gaussiana. Así fue diseñado el conjunto de familias de métodos iterativos que tienen orden de convergencia tres si la suma de los pesos de la familia de cuadratura gaussiana correspondiente es dos.

El objetivo de esta sección es el desarrollo de métodos iterativos usando distintos predictores con el fin de obtener mayor orden de convergencia y mejor índice de eficiencia.

3.3.1. Método de Traub como predictor

Introducimos como predictor el método de Traub [81] con fórmula iterativa (3.26) y convergencia de orden 3. Introducimos las mismas notaciones de la Sección 3.2.2 determinados por las expresiones (3.18) que, para mayor claridad, repetimos:

$$\sum_{i=1}^n \omega_i = \sigma,$$

$$\sum_{i=1}^n \frac{\omega_i \tau_i^j}{\sigma} = \sigma_j \quad \text{con } j = 1, 2, \dots$$

En este caso podemos enunciar el siguiente resultado:

Teorema 3.3.1 *Sea $\xi \in I$ un cero simple de la función $f : I \subseteq \mathbb{R} \rightarrow \mathbb{R}$ suficientemente diferenciable en el intervalo abierto I y consideremos que la estimación inicial x_0 este suficientemente cerca de ξ . Si aplicamos el método de Traub como predictor y las cuadraturas de Gauss como corrector, se obtiene la fórmula iterativa:*

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)} \\ z_k &= x_k - \frac{f(x_k) + f(y_k)}{f'(x_k)}, \\ x_{k+1} &= y_k - \frac{2f(y_k)}{\sum_{i=1}^n \omega_i f'(\eta_{i,k})} \end{aligned} \quad (3.29)$$

del método que denotamos como Traub-Gauss (TrG), donde $\eta_{i,k} = \frac{(1 + \tau_i)z_k + (1 - \tau_i)y_k}{2}$. Entonces,

1. Si $\sigma = 2$, el conjunto de familias Traub-Gauss tienen convergencia de orden cuatro;
2. Si, además $\sigma_1 = 0$, la convergencia es de orden cinco,
3. Si $f''(\xi) = 0$ y $\sigma = 2$ el orden es nueve.
4. Si, además, $\sigma_1 = 0$, $\sigma_2 = 1/3$ entonces, la convergencia es de orden once.

Demostración: Calculamos de la misma manera que en el Teorema 3.2.1 las cantidades expresadas con las fórmulas (3.19), (3.20) y (3.21). Entonces, $y_k = \xi + A_1$, donde

$$A_1 = c_2 e_k^2 - 2(c_2^2 - c_3) e_k^3 - (7c_2 c_3 - 4c_2^3 - 3c_4) e_k^4 - (8c_2^4 - 20c_2^2 c_3 + 6c_3^2 + 10c_2 c_4 - 4c_5) e_k^5 + O[e_k^6],$$

representa el error en la k -ésima iteración para y_k . Expandiendo $f(y_k)$ en serie de Taylor alrededor de ξ obtenemos:

$$f(y_k) = f'(\xi) [c_2 e_k^2 + 2(c_3 - c_2^2) e_k^3 + (4c_2^3 + 3c_4 - 7c_2 c_3) e_k^4 + (20c_2^2 c_3 - 8c_2^5 - 6c_3^2 - 10c_2 c_4 + 4c_5) e_k^5] + O[e_k^6].$$

Así:

$$\frac{f(x_k) + f(y_k)}{f'(x_k)} = e_k - 2c_2^2 e_k^3 + (9c_2^3 - 7c_2 c_3) e_k^4 + 2(-15c_2^4 + 22c_2 c_3 - 3c_3^2 - 5c_2 c_4) e_k^5 + O[e_k^6]$$

y entonces, la expresión para z_k será:

$$z_k = x_k - \frac{f(x_k) + f(y_k)}{f'(x_k)} = \xi + A_2,$$

donde $A_2 = 2c_2^2 e_k^3 + (-9c_2^3 + 7c_2 c_3) e_k^4 + (30c_2^4 - 44c_2 c_3 + 6c_3^2 + 10c_2 c_4) e_k^5 + O[e_k^6]$ representa el error en la k -ésima iteración para z_k .

Reescribimos $\eta_{i,k}$ en función de los errores de y_k y z_k :

$$\eta_{i,k} = \frac{(1 + \tau_i)z_k + (1 - \tau_i)y_k}{2} = \xi + \frac{1}{2}(A_2 + A_1) + \frac{1}{2}\tau_i(A_2 - A_1),$$

y expandiendo $f'(\eta_{i,k})$ alrededor de ξ obtenemos:

$$\begin{aligned} f'(\eta_{i,k}) &= f'(\xi) [1 + (1 - \tau_i)c_2 e_k^2 + 2(2\tau_i c_2^3 + (1 - \tau_i)c_2 c_3) e_k^3] \\ &+ f'(\xi) \left[\left(\frac{1}{4}(3 + 5\tau_i + 3\tau_i^2)c_2^2 c_3 - (5 + 13\tau_i)c_2^4 + 3(1 - \tau_i)c_2 c_4 \right) e_k^4 \right] \\ &+ f'(\xi) [(22 + 38\tau_i c_2^5 - (12 + 19\tau_i + 3\tau_i^2)c_2^3 c_3 + 3(1 - \tau_i)^2 c_3^2 + 20\tau_i c_2 c_4 + 4(1 - \tau_i)c_5)] e_k^5 + O[e_k^6]. \end{aligned}$$

Usando el procedimiento introducido en el Teorema 3.2.1 para obtención de la expresión de K e introduciendo las notaciones (3.18), obtenemos

$$\begin{aligned} K &= \sum_{i=1}^n \omega_i f'(\eta_{i,k}) \tag{3.30} \\ &= f'(\xi) \sum_{i=1}^n \omega_i [1 + (1 - \sigma_1)c_2 e_k^2 + 2(2\sigma_1 c_2^3 + (1 - \sigma_1)c_2 c_3) e_k^3] \\ &+ f'(\xi) \sum_{i=1}^n \omega_i \left[\left(\frac{1}{4}(3 + 5\sigma_1 + 3\sigma_2)c_2^2 c_3 - (5 + 13\sigma_1)c_2^4 + 3(1 - \sigma_1)c_2 c_4 \right) e_k^4 \right] \\ &+ f'(\xi) \sum_{i=1}^n \omega_i [(22 + 38\sigma_1)c_2^5 - (12 + 19\sigma_1 + 3\sigma_2)c_2^3 c_3 + 3(1 - 2\sigma_1 + \sigma_2)c_3^2] e_k^5 \\ &+ f'(\xi) \sum_{i=1}^n \omega_i [20\sigma_1 c_2 c_4 + 4(1 - \sigma_1)c_5] e_k^5 + O[e_k^6]. \end{aligned}$$

Ahora calculamos

$$\begin{aligned} L &= \frac{2f(y_k)}{K} \\ &= \frac{2}{\sigma} c_2 e_k^2 + \frac{2}{\sigma} (2c_3 - c_2^2) e_k^3 + \frac{2}{\sigma} [(4 + \sigma_1)c_2^3 - 7c_2 c_3 + 3c_4] e_k^4 \\ &+ \frac{2}{\sigma} [-2(5 + 3\sigma_1)c_2^4 + 4(5 + \sigma_1)c_2^2 c_3 - 6c_3^2 - 10c_2 c_4 + 4c_5] e_k^5 + O[e_k^6]. \end{aligned}$$

Como $x_{k+1} = y_k - L$, la estimación del error resulta

$$\begin{aligned} e_{k+1} &= \left(1 - \frac{2}{\sigma}\right) c_2 e_k^2 + 2 \left(1 - \frac{2}{\sigma}\right) (c_3 - 2c_2^2) e_k^3 + \left[\frac{4}{\sigma} (\sigma - 2 - \frac{\sigma_1}{2}) c_2^3 - \left(1 - \frac{2}{\sigma}\right) (7c_2 c_3 - 3c_4) \right] e_k^4 \\ &+ \frac{2}{\sigma} [2(5 - 2\sigma + 3\sigma_1)c_2^4 - 2(10 + 5\sigma - 2\sigma_1)c_2^2 c_3 + (\sigma - 2)(3c_3^2 + 5c_2 c_4 - 2c_5)] e_k^5 + O[e_k^6]. \end{aligned}$$

Notamos que para tener orden de convergencia mayor que dos debe verificarse $\sigma = 2$. Entonces, la expresión del error queda:

$$e_{k+1} = -\sigma_1 c_2^3 e_k^4 + 2[(1 + 3\sigma_1)c_2^4 - 2\sigma_1 c_2^2 c_3] e_k^5 + O[e_k^6],$$

y si además $\sigma_1 = 0$ obtenemos:

$$e_{k+1} = 2c_2^4 e_k^5 + O[e_k^6].$$

Por otra parte, si $f''(\xi) = 0$ y $\sigma = 2$ la expresión de e_{k+1} será:

$$\begin{aligned} e_{k+1} &= 2(-1 - 6\sigma_1 + 3\sigma_2)c_3^4 e_k^9 + 9(-1 - 6\sigma_1 + 3\sigma_2)c_3^3 c_4 e_k^{10} \\ &+ \left[18(3 + 6\sigma_1 - 5\sigma_2)c_3^5 - \frac{27}{2}(1 + 6\sigma_1 - 3\sigma_2)c_3^2 c_4^2 - 12(1 - 6\sigma_1 + 3\sigma_2)c_3^3 c_5 \right] e_k^{11} + O[e_k^{12}]. \end{aligned}$$

En caso de que además $\sigma_1 = 0$ y $\sigma_2 = 1/3$, la expresión del error queda: $e_{k+1} = 24c_3^5 e_k^{11} + O[e_k^{12}]$. \square

Del conjunto de familias de variantes del método de TrG únicamente el método Traub-Gauss-Radau de un sólo nodo (TrG-R1), según el Teorema 3.3.1 (véase la Tabla 3.1), tendrá orden de convergencia 4. Todas las restantes variantes verifican las condiciones 1 y 2 del Teorema 3.3.1 y tendrán orden cinco. En caso de que $c_2 = f''(\xi) = 0$ los métodos TrG-LD1, TrG-LT2 y TrG-R1 tendrán orden 9, y TrG-LD2 y TrG-R2 tendrán orden 11.

Tabla 3.5: Características de los métodos iterativos

Métodos	p	d	op	I	IO	IC
TrG-LD1	5	4	3	1.4953	1.7100	1.2585
TrG-LD2	5	5	3	1.3797	1.7100	1.2228
TrG-LT2	5	5	3	1.3797	1.7100	1.2228
TrG-R2	5	5	3	1.3797	1.7100	1.2228

En la Tabla 3.5 se muestran el orden de convergencia, el número de evaluaciones funcionales y el número de operaciones por iteración y los índices de eficiencia, operacionales y de eficiencia computacional de los métodos desarrollados. Vemos que el método TrG-LD1 tiene una evaluación menos que todos los restantes métodos y esta es la razón de que tenga mejores índice de eficiencia (1.4953) e índice de eficiencia computacional (1.2585). Todos los restantes métodos tienen los mismos índices de eficiencia (1.3797), operacionales (1.7100) y de eficiencia computacional (1.2228).

3.3.2. Método de Ostrowski como predictor

A continuación estudiamos el método iterativo que se origina cuando utilizamos el método de Ostrowski [72] como predictor. La expresión iterativa del método de Ostrowski es:

$$\begin{aligned}
 y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\
 x_{k+1} &= y_k - \frac{f(y_k)}{f(x_k) - 2f(y_k)} \frac{f(x_k)}{f'(x_k)},
 \end{aligned}
 \tag{3.31}$$

que tiene orden de convergencia 4 y es óptimo según la conjetura de Kung-Traub.

Teorema 3.3.2 *Sea $\xi \in I$ un cero simple de la función $f : I \subseteq \mathbb{R} \rightarrow \mathbb{R}$ suficientemente diferenciable en el intervalo abierto I y consideremos que la estimación inicial x_0 sea suficientemente cerca a ξ . Si aplicamos el método de Ostrowski (3.31) como predictor y las cuadraturas de Gauss como corrector, se obtiene el método iterativo:*

$$\begin{aligned}
 y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\
 z_k &= y_k - \frac{f(y_k)}{f(x_k) - 2f(y_k)} \frac{f(x_k)}{f'(x_k)}, \\
 x_{k+1} &= y_k - \frac{2f(y_k)}{\sum_{i=1}^n \omega_i f'(\eta_{i,k})},
 \end{aligned}
 \tag{3.32}$$

del método que denotamos como Ostrowski-Gauss (OsG), donde $\eta_{i,k} = \frac{(1 + \tau_i)z_k + (1 - \tau_i)y_k}{2}$. Entonces, si las cuadraturas satisfacen las condiciones:

1. $\sigma = 2$, el conjunto de familias OsG tienen convergencia de orden cuatro;

2. Si, además $\sigma_1 = 0$, la convergencia es de orden seis,
3. Si $f''(\xi) = 0$ y $\sigma = 2$ el orden es nueve.
4. Si, además, $\sigma_1 = 0$, $\sigma_2 = 1/3$ entonces, la convergencia es de orden once.

Demostración: Calculamos de la misma manera que en el Teorema 3.2.1 las siguientes cantidades:

$$f(x_k) = f'(\xi) [e_k + c_2 e_k^2 + c_3 e_k^3 + c_4 e_k^4 + c_5 e_k^5 + c_6 e_k^6] + O[e_k^7], \quad (3.33)$$

$$f'(x_k) = f'(\xi) [1 + 2c_2 e_k + 3c_3 e_k^2 + 4c_4 e_k^3 + 5c_5 e_k^4 + 6c_6 e_k^5] + O[e_k^6]. \quad (3.34)$$

Como los términos entre corchetes son polinomios en términos de e_k , la división directa nos da:

$$\begin{aligned} \frac{f(x_k)}{f'(x_k)} &= e_k - c_2 e_k^2 + 2(c_2^2 - c_3) e_k^3 + (7c_2 c_3 - 4c_2^3 - 3c_4) e_k^4 \\ &+ (8c_2^5 - 20c_2^2 c_3 + 6c_3^2 + 10c_2 c_4 - 4c_5) e_k^5 \\ &+ (52c_2^3 - 16c_2^5 - 28c_2^2 c_4 + 17c_3 c_4 + c_2(13c_5 - 33c_3^2) - 5c_6) e_k^6 + O[e_k^7], \end{aligned} \quad (3.35)$$

y entonces, la expresión para y_k es $y_k = \xi + A_1$ con

$$\begin{aligned} A_1 &= c_2 e_k^2 - 2(c_2^2 - c_3) e_k^3 - (7c_2 c_3 - 4c_2^3 - 3c_4) e_k^4 \\ &- (8c_2^5 - 20c_2^2 c_3 + 6c_3^2 + 10c_2 c_4 - 4c_5) e_k^5 \\ &- (52c_2^3 - 16c_2^5 - 28c_2^2 c_4 + 17c_3 c_4 + c_2(13c_5 - 33c_3^2) - 5c_6) e_k^6 + O[e_k^7]. \end{aligned} \quad (3.36)$$

Expandiendo y_k en serie de Taylor alrededor de ξ obtenemos:

$$\begin{aligned} f(y_k) &= f'(\xi) [c_2 e_k^2 + 2(c_3 - c_2^2) e_k^3 + (4c_2^3 + 3c_4 - 7c_2 c_3) e_k^4] \\ &+ f'(\xi) (20c_2^2 c_3 - 8c_2^5 - 6c_3^2 - 10c_2 c_4 + 4c_5) e_k^5 \\ &+ f'(\xi) [16c_2^5 - 52c_2^3 + 28c_2^2 c_4 - 17c_3 c_4 - c_2(13c_5 - 33c_3^2) + 5c_6] e_k^6 + O[e_k^7]. \end{aligned} \quad (3.37)$$

Calculamos:

$$\begin{aligned} \frac{f(y_k)}{f(x_k) - 2f(y_k)} &= c_2 e_k + (2c_3 - c_2^2) e_k^2 + (3c_4 - 2c_2 c_3) e_k^3 + (2c_2^4 - 3c_2^2 c_3 - 2c_2 c_4 + 4c_5) e_k^4 \\ &+ (14c_2^3 c_3 - 4c_2^5 - 5c_2^2 c_4 + 2c_3 c_4 - c_2(9c_3^2 + 2c_5) + 5c_6) e_k^5 + O[e_k^6], \end{aligned}$$

con lo que, la expresión para z_k es:

$$\begin{aligned} z_k &= y_k - \frac{f(y_k)}{f(x_k) - 2f(y_k)} \frac{f(x_k)}{f'(x_k)} \\ &= \xi + (c_2^3 - c_2 c_3) e_k^4 - 2(2c_2^4 - 4c_2^2 c_3 + c_3^2 + c_2 c_4) e_k^5 \\ &+ [10c_2^5 - 30c_2^3 c_3 + 12c_2^2 c_4 - 7c_3 c_4 + 3c_2(6c_3^2 - c_5)] e_k^6 + O[e_k^7]. \end{aligned}$$

Sean $y_k = \xi + A_1$ y $z_k = \xi + A_2$ con

$$\begin{aligned} A_2 &= (c_2^3 - c_2 c_3) e_k^4 - 2(2c_2^4 - 4c_2^2 c_3 + c_3^2 + c_2 c_4) e_k^5 \\ &+ (10c_2^5 - 30c_2^3 c_3 + 12c_2^2 c_4 - 7c_3 c_4 + 3c_2(6c_3^2 - c_5)) e_k^6 + O[e_k^7]. \end{aligned}$$

Tomando $\eta_{i,k} = \frac{(1 + \tau_i)z_k + (1 - \tau_i)y_k}{2}$, obtenemos $\eta_{i,k} = \xi + \frac{1}{2}(A_2 + A_1) + \frac{1}{2}\tau_i(A_2 - A_1)$ y expandiendo $f'(\eta_{i,k})$ alrededor de ξ obtenemos:

$$f'(\eta_{i,k}) = f'(\xi) [B + C\tau_i + D\tau_i^2 + E\tau_i^3 + \dots],$$

donde

$$\begin{aligned}
 B &= 1 + c_2^2 e_k^2 + 2(c_2 c_3 - c_2^3) e_k^3 + \left(5c_4 - \frac{29}{4} c_2^2 c_3 + 3c_2 c_4\right) e_k^4 \\
 &+ (-12c_2^5 + 25c_2^3 c_3 - 5c_2 c_3^2 - 12c_2^2 c_4 + 4c_2 c_5) e_k^5 \\
 &+ \left(26c_2^6 - \frac{143}{2} c_2^4 c_3 + 33c_2^2 c_3^2 + 3c_3^3 + \frac{81}{2} c_2^3 c_4 - \frac{39}{2} c_2 c_3 c_4 - 16c_2^2 c_5 + 5c_2 c_5\right) e_k^6, \\
 C &= -c_2^2 e_k^2 + 2(c_2^3 - c_2 c_3) e_k^3 + \left(-3c_2^4 + \frac{9}{2} c_2^2 c_3 - 3c_2 c_4\right) e_k^4 \\
 &+ (4c_2^5 - 6c_2^3 c_3 - 2c_2 c_3^2 + 8c_2^2 c_4 - 4c_2 c_5) e_k^5 \\
 &+ \left(-6c_2^6 + 4c_2^4 c_3 + 18c_2^2 c_3^2 - 6c_3^3 + \frac{35}{2} c_2^3 c_4 + c_2 c_3 c_4 + 10c_2^2 c_5 - 5c_2 c_5\right) e_k^6, \\
 D &= \frac{3}{4} c_2^2 c_3 e_k^4 + (-3c_2^3 c_3 + 3c_2 c_3^2) e_k^5 + \left(\frac{15}{4} c_2^4 c_3 - 15c_2^2 c_3^2 + 3c_3^3 + \frac{3}{2} c_2^3 c_4 + \frac{9}{2} c_2 c_2 c_4\right) e_k^6, \\
 E &= -\frac{1}{2} c_2^2 c_4 e_k^6.
 \end{aligned}$$

Usando el procedimiento introducido en el Teorema 3.2.1 para la obtención de la expresión de K , agrupando e introduciendo las notaciones (3.18), calculamos:

$$\begin{aligned}
 K &= \sum_{i=1}^n \omega_i f'(\eta_{i,k}) \\
 &= \sigma f'(\xi) [B + C\sigma_1 + D\sigma_2 + E\sigma_3 + F\sigma_4] + O[e_k^6].
 \end{aligned}$$

Por lo tanto, obtenemos la expresión de L en la forma:

$$\begin{aligned}
 L &= \frac{2f(y_k)}{K} \\
 &= \frac{2}{\sigma} c_2^2 e_k^2 + \frac{4}{\sigma} (c_3 - c_2^2) e_k^3 + \frac{2}{\sigma} [(4 + \sigma_1) c_2^3 - 7c_2 c_3 + 3c_4] e_k^4 \\
 &+ \frac{2}{\sigma} [-4(2 + \sigma_1) c_2^4 + 4(5 + \sigma_1) c_2^2 c_3 - 6c_3^2 - 10c_2 c_4 + 4c_5] e_k^5 \\
 &+ \frac{2}{\sigma} \left[(15 + 10\sigma_1 + \sigma_2) c_2^5 - \frac{1}{4} (203 + 78\sigma_1 + 3\sigma_2) c_2^3 c_3 + 2(14 + 3\sigma_1) c_2^2 c_4 \right] e_k^6 \\
 &+ \frac{2}{\sigma} [-17c_3 c_4 + (33 + 4\sigma_1) c_2 c_3^2 - 13c_2 c_5 + 5c_6] e_k^6 + O[e_k^7].
 \end{aligned}$$

Como $x_{k+1} = y_k - L$, calculamos $e_{k+1} = A_1 - L$:

$$\begin{aligned}
 e_{k+1} &= \left(1 - \frac{2}{\sigma}\right) c_2 e_k^2 + 2 \left(1 - \frac{2}{\sigma}\right) (c_2^2 - c_3) e_k^3 \\
 &+ \left[\left(4 - \frac{2}{\sigma}(4 + \sigma_1)\right) c_2^3 + \left(1 - \frac{2}{\sigma}\right) (3c_4 - 7c_2 c_3) \right] e_k^4 \\
 &+ \left[-8 \left(1 - \frac{2 + \sigma_1}{\sigma}\right) c_2^4 + 4 \left(5 - \frac{10 + 2\sigma_1}{\sigma}\right) c_2^2 c_3 + \left(1 - \frac{2}{\sigma}\right) (4c_5 - 6c_3^2 - 10c_2 c_4) \right] e_k^5 \\
 &+ \left[\left(16 - \frac{2(15 + 10\sigma_1 + \sigma_2)}{\sigma}\right) c_2^5 + \left(-52 + \frac{203 + 78\sigma_1 + 3\sigma_2}{2\sigma}\right) c_2^3 c_3 + \left(28 - \frac{4(14 + 3\sigma_1)}{\sigma}\right) c_2^2 c_4 \right] e_k^6 \\
 &+ \left[\left(33 - \frac{2(33 + 4p)}{\sigma}\right) c_2 c_3^2 + \left(1 - \frac{2}{\sigma}\right) (17c_3 c_4 + 13c_2 c_5 - 5c_6) \right] e_k^6 + O[e_k^7].
 \end{aligned}$$

Según este resultado, en general, todos los métodos, OsG, tendrán convergencia al menos cuadrática. Si imponemos la condición $\sigma = 2$, el orden de convergencia de los métodos obtenidos será cuatro. Entonces, la expresión del error queda:

$$\begin{aligned}
 e_{k+1} &= -\sigma_1 c_2^3 e_k^4 - 4\sigma_1 c_2^2 (c_2^2 - c_3) e_k^5 \\
 &- \frac{1}{4} [4(1 - 10\sigma_1 - \sigma_2) c_2^5 + (5 - 78\sigma_1 - 3\sigma_2) c_2^3 c_3 + 16\sigma_1 c_2 c_3^2 + 24\sigma_1 c_2^2 c_4] e_k^6 + O[e_k^7].
 \end{aligned}$$

Si imponemos la condición de que $\sigma_1 = 0$ la convergencia será de orden seis:

$$e_{k+1} = \left[(1 - \sigma_2)c_2^5 - \frac{1}{4}(5 - 3\sigma_2)c_2^3c_3 \right] e_k^6 + O[e_k^7].$$

Si además $c_2 = f''(\xi) = 0$ la expresión de e_{k+1} será:

$$e_{k+1} = 2(-1 + 3\sigma_2)c_3^4e_k^9 + 9(-1 + 3\sigma_2)c_3^3c_4e_k^{10} + \left[(-1 + 7\sigma_2)c_3^5 - \frac{3}{4}(1 - 3\sigma_2)(9c_3^2c_4^2 + 8c_3^4c_5) \right] e_k^{11} + O[e_k^{12}].$$

Si también $\sigma_2 = 1/3$, entonces, $e_{k+1} = -\frac{5}{3}c_3^5e_k^{11} + O[e_k^{12}]$. □

La variante OsG-R1 no verifica la condición 2 del Teorema 3.3.2 por lo que su orden de convergencia es cuatro y nueve si $f''(\xi) = 0$. Todas las restantes variantes verifican la condición 2 del Teorema 3.3.2 y tendrán orden de convergencia seis en el caso general y once si $f''(\xi) = 0$. En caso de que $c_2 = f''(\xi) = 0$, los métodos OsG-LD1, OsG-LT2 y OsG-R1 tendrán orden 9, mientras que OsG-LD2 y OsG-R2 tendrán orden 11.

Tabla 3.6: Comparación de los índices de los métodos iterativos

Métodos	p	d	op	I	IO	IC
OsG-LD1	6	4	4	1.5651	1.5651	1.2510
OsG-LD2	6	5	4	1.4310	1.5651	1.2203
OsG-LT2	6	5	4	1.4310	1.5651	1.2203
OsG-R2	6	5	4	1.4310	1.5651	1.2203

La Tabla 3.6 muestra el orden de convergencia, el número de evaluaciones funcionales y el número de operaciones por iteración y los índices de eficiencia, operacionales y de eficiencia computacional de los métodos desarrollados. Vemos de nuevo que el método OsG-LD1 tiene una evaluación menos que todos los restantes métodos y mejores índice de eficiencia (1.5651) e índice de eficiencia computacional (1.2510). Todos los restantes métodos tienen los mismos índices de eficiencia (1.4310), operacionales (1.5651) y computacional (1.2203). Además si comparamos estos resultados con los mostrados en la Tabla 3.5 vemos que el mayor orden de convergencia de estos métodos le garantiza un índice de eficiencia mayor, pero el mayor número de operaciones por iteración hace que sus índices de eficiencia operacional e índices de eficiencia computacional sean menores que los de los métodos generados con predictor Traub.

3.3.3. Método de la derivada congelada como predictor

A continuación empleamos como predictor otro método de orden 4, un caso particular del método introducido por Hueso et al en [50], que llamaremos derivada congelada DC . Este método tiene cuatro evaluaciones funcionales: una derivada y tres funciones, no es óptimo según la conjetura de Kung-Traub y su expresión iterativa es:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\ z_k &= y_k - \frac{f(y_k)}{f'(x_k)}, \\ x_{k+1} &= x_k - \frac{f(z_k)}{f'(x_k)}. \end{aligned} \tag{3.38}$$

El siguiente resultado muestra las condiciones bajo las cuales el método resultante DC-Gauss (DCG) tiene orden de convergencia 6 ó 17.

Teorema 3.3.3 Sea $\xi \in I$ un cero simple de la función $f : I \subseteq \mathbb{R} \rightarrow \mathbb{R}$ suficientemente diferenciable en el intervalo abierto I y consideremos que la estimación inicial x_0 esté suficientemente cerca de ξ . Si aplicamos el método (3.38) como predictor y las cuadraturas de Gauss como corrector, se obtiene la fórmula iterativa del método DCG:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\ z_k &= y_k - \frac{f(y_k)}{f'(x_k)}, \\ u_k &= z_k - \frac{f(z_k)}{f'(x_k)}, \\ x_{k+1} &= z_k - \frac{2f(z_k)}{\sum_{i=1}^n \omega_i f'(\eta_{i,k})} \end{aligned} \quad (3.39)$$

donde $\eta_{i,k} = \frac{(1 + \tau_i)u_k + (1 - \tau_i)z_k}{2}$. Entonces, si las cuadraturas satisfacen las condiciones:

1. Si $\sigma = 2$, las variantes del método de DCG tienen convergencia de orden seis;
2. Si, además $\sigma_1 = 0$, dichas variantes del método de DCG tienen convergencia de orden siete;
3. Si $f''(\xi) = 0$ y $\sigma = 2$ el orden es 15.
4. Si, además, $\sigma_1 = 0$, $\sigma_2 = 1/3$ entonces, la convergencia es de orden 17.

Demostración: La demostración de este teorema comienza con la obtención de las expresiones (3.33), (3.34), (3.35), (3.36) y (3.37) del mismo modo que en el Teorema 3.3.2. A partir de ellas:

$$\begin{aligned} \frac{f(y_k)}{f'(x_k)} &= f'(\xi)[c_2 e_k^2 + 2(c_3 - 2c_2^2)e_k^3 + (13c_2^3 + 3c_4 - 14c_2c_3)e_k^4 \\ &+ (-38c_2^4 + 64c_2^2c_3 - 20c_2c_4 - 12c_2^3 + 4c_5)e_k^5 \\ &+ (104c_2^5 - 240c_2^3c_3 + 90c_2^2c_4 - 34c_3c_4 + 103c_2c_3^2 - 26c_2c_5 + 5c_6)e_k^6] + O[e_k^7]. \end{aligned}$$

Entonces, la expresión para z_k será:

$$\begin{aligned} z_k &= y_k - \frac{f(y_k)}{f'(x_k)} \\ &= \xi + A_1 \\ &= \xi + 2c_2^2 e_k^3 + (7c_2c_3 - 9c_2^3)e_k^4 + (30c_2^4 - 44c_2^2c_3 + 6c_2^3 + 10c_2c_4)e_k^5 \\ &+ (-88c_2^5 + 188c_2^3c_3 - 62c_2^2c_4 + 17c_3c_4 - 70c_2c_3^2 + 13c_2c_5)e_k^6 \\ &+ 2(120c_2^6 - 336c_2^4c_3 - 18c_2^3 + 132c_2^3c_4 + 6c_2^4 + 8c_2^2(27c_2^3 - 5c_5))e_k^7 \\ &+ 2(11c_3c_5 + c_2(8c_6 - 98c_3c_4))e_k^7 + O[e_k^8]. \end{aligned}$$

Ahora podemos obtener la expresión para $f(z_k)$:

$$\begin{aligned} f(z_k) &= 2c_2^2 e_k^3 + (7c_2c_3 - 9c_2^3)e_k^4 + (30c_2^4 - 44c_2^2c_3 + 6c_2^3 + 10c_2c_4)e_k^5 \\ &+ (-80c_2^5 + 188c_2^3c_3 - 62c_2^2c_4 + 17c_3c_4 - 70c_2c_3^2 + 13c_2c_5)e_k^6 \\ &+ 2(120c_2^6 - 336c_2^4c_3 - 18c_2^3 + 132c_2^3c_4 + 6c_2^4 + 8c_2^2(27c_2^3 - 5c_5))e_k^7 \\ &+ 2(11c_3c_5 + c_2(8c_6 - 98c_3c_4))e_k^7 + O[e_k^8]. \end{aligned}$$

De esta manera la expresión de u_k será:

$$\begin{aligned} u_k &= z_k - \frac{f(z_k)}{f'(x_k)} \\ &= \xi + A_2 \\ &= \xi + 4c_2^3 e_k^4 + (-26c_2^4 + 20c_2^2c_3)e_k^5 + c_2(108c_2^4 - 167c_2^2c_3 + 33c_2^3 + 28c_2c_4)e_k^6 \\ &+ (-356c_2^6 + 850c_2^4c_3 + 18c_2^3 - 232c_2^3c_4 + 92c_2c_3c_4 + c_2^2(-398c_2^3 + 36c_5))e_k^7 + O[e_k^8]. \end{aligned}$$

Tomando $\eta_{k,i} = \frac{(1 + \tau_i)u_k + (1 - \tau_i)z_k}{2}$, obtenemos $\eta_{k,i} = \xi + \frac{1}{2}(A_2 + A_1) + \frac{1}{2}\tau_i(A_2 - A_1)$ y expandiendo $f'(\eta_{k,i})$ alrededor de ξ obtenemos:

$$\begin{aligned} f'(\eta_{k,i}) &= f'(\xi)[1 - 2(\tau_i - 1)c_2^2e_k^3 - 2c_2^2((13\tau_i - 5)c_2^2 - 7(\tau_i - 1)c_3)e_k^4] \\ &+ 2f'(\xi)c_2[(2 - 28\tau_i)c_2^4 + 4(-3 + 8\tau_i)c_2^2c_3 - 3(\tau_i - 1)c_3^2 - 5(\tau_i - 1)c_2c_4]e_k^5 \\ &+ f'(\xi)c_2[4(49\tau_i + 5)c_2^5 + (24 - 361\tau_i + 3\tau_i^2)c_2^3c_3 + 2(45\tau_i - 17)c_2^2c_4]e_k^6 \\ &+ f'(\xi)c_2[-17(2\tau_i - 1)c_3c_4 + c_2((103\tau_i - 37)c_3^2 - 13(\tau_i - 1)c_5)]e_k^6 \\ &+ f'(\xi)c_2[-4(29 + 149\tau_i)c_2^6 + (163 + 1576\tau_i - 39\tau_i^2)c_2^4c_3 + 16(2 - 31\tau_i)c_2^3c_4]e_k^7 \\ &+ f'(\xi)c_2^3[(55 - 872\tau_i + 21\tau_i^2)c_2^2c_4 + 4(29\tau_i - 11)c_5]e_k^7 \\ &+ 2f'(\xi)c_2[9(3\tau_i - 1)c_3^3 - 6(\tau_i - 1)c_4^2 - 11(\tau_i - 1)c_3c_5]e_k^7 \\ &+ 8f'(\xi)c_2[(36\tau_i^2 - 13)c_3c_3c_4 - 2(\tau_i - 1)c_6]e_k^7 + O[e_k^8]. \end{aligned}$$

Usando el procedimiento introducido en el Teorema 3.2.1 para la obtención de la expresión de K , agrupando e introduciendo las notaciones (3.18), se obtiene la expresión de K y calculamos L :

$$\begin{aligned} L &= \frac{2f(y_k)}{K} \\ &= \frac{2}{\sigma}[2c_2^2e_k^3 + (7c_2^2c_3 - 9c_3^2)e_k^4 + 2(15c_2^4 - 22c_2^2c_3^2 + 3c_3^2 + 5c_2c_4)e_k^5] \\ &+ \frac{2}{\sigma}[4(\sigma_1 - 22)c_2^5 + 188c_2^3c_3 - 62c_2^2c_4 + 17c_3c_4 + c_2(13c_5 - 70c_3^2)]e_k^6 \\ &+ \frac{4}{\sigma}[(116 - 22\sigma_1)c_2^6 + 14(\sigma_1 - 24)c_2^4c_3 + 132c_2^3c_4]e_k^7 \\ &+ \frac{4}{\sigma}[6c_4^2 + 8c_2^2(27c_3^2 - 5c_5) + 11c_3c_5 + c_2(8c_6 - 98c_3c_4)]e_k^7 \\ &+ O[e_k^8]. \end{aligned}$$

Como $x_{k+1} = z_k - L$, calculamos e_{k+1}

$$\begin{aligned} e_{k+1} &= 2\left(1 - \frac{2}{\sigma}\right)c_2^2e_k^3 + \left(1 - \frac{2}{\sigma}\right)[7c_2c_3 - 9c_3^2]e_k^4 + 2\left(1 - \frac{2}{\sigma}\right)[15c_2^4 - 22c_2^2c_3 + 3c_3^2 + 5c_2c_4]e_k^5 \\ &+ \left[\left(-88 - \frac{8}{\sigma}(\sigma_1 - 22)\right)c_2^5 + \left(1 - \frac{2}{\sigma}\right)(188c_2^3c_3 - 62c_2^2c_4 + 17c_3c_4 - 70c_2c_3^2 + 13c_2c_5)\right]e_k^6 \\ &+ 2\left[\left(120 - \frac{2}{\sigma}(116 - 22\sigma_1)\right)c_2^6 - \left(336 - \frac{28}{\sigma}(\sigma_1 - 24)\right)c_2^4c_3\right]e_k^7 \\ &+ 2\left(1 - \frac{2}{\sigma}\right)[6c_4^2 + 8(27c_3^2 - 5c_5)c_2^2 + 11c_3c_5 + c_2(8c_6 - 98c_3c_4)]e_k^7 + O[e_k^8]. \end{aligned}$$

Notamos que para tener orden de convergencia seis es necesario que $\sigma = 2$ (véase la Tabla 3.1). Entonces, la expresión de la ecuación del error e_{k+1} queda:

$$e_{k+1} = -4\sigma_1c_2^5e_k^6 + [(8 + 44\sigma_1)c_2^6 - 28\sigma_1c_2^4c_3]e_k^7 + O[e_k^8]$$

y si, además, la fórmula de cuadratura verifica $\sigma_1 = 0$ obtenemos:

$$e_{k+1} = 8c_2^6e_k^7 + O[e_k^8]. \quad (3.40)$$

Finalmente, si $f''(\xi) = 0$ la expresión (3.40) de e_{k+1} será:

$$\begin{aligned} e_{k+1} &= 54(3\sigma_2 - 1)c_3^7e_k^{15} + 459(3\sigma_2 - 1)c_3^6c_4e_k^{16} \\ &- \frac{9}{2}[432(2\sigma_2 - 1)c_3^8 - (3\sigma_2 - 1)(361c_3^5c_4^2 + 132c_3^6c_5)]e_k^{17} + O[e_k^{18}]. \end{aligned}$$

Si, además, $\sigma_2 = 1/3$, entonces,

$$e_{k+1} = 648c_3^8e_k^{17} + O[e_k^{18}].$$

□

La variante DCG-R1 no cumple con la condición 2 del Teorema 3.3.3 por lo que tendrá orden de convergencia 6 ó 15 si $f''(\xi) = 0$. Todas las restantes variantes verifican la condición 2 del Teorema 3.3.3 y tendrán orden de convergencia siete en el caso general. En caso de que $c_2 = f''(\xi) = 0$ los métodos DCG-LT1 y DCG-LT2 tendrán orden 15, mientras que DCG-LD2 y DCG-R2 tendrán orden 17.

Tabla 3.7: Datos de los nuevos métodos iterativos

Métodos	p	d	op	I	IO	IC
DCG-LD1	7	5	4	1.4758	1.6266	1.2414
DCG-LD2	7	6	4	1.3831	1.6266	1.2148
DCG-LT2	7	6	4	1.3831	1.6266	1.2148
DCG-R2	7	6	4	1.3831	1.6266	1.2148

De la misma manera que en las secciones anteriores, en la Tabla 3.7 se muestra el orden de convergencia, el número de evaluaciones funcionales y el número de operaciones por iteración, así como los índices de eficiencia, operacional y computacional de los métodos desarrollados. Nótese que todos los métodos tienen el mismo índice operacional. El método DCG-LD1 tiene una evaluación menos que los restantes métodos. Por esta razón tiene mejores índices de eficiencia (1.6266) y computacional (1.2414). Todos los restantes métodos tienen los mismos índices de eficiencia (1.3831) y computacionales (1.2148).

3.4. Generalización del predictor: pseudocomposición

En esta sección vamos a introducir una técnica, a la que llamaremos pseudocomposición, que permite considerar todos los casos vistos hasta ahora como casos particulares. Demostraremos que el orden final del método dependerá, entre otros factores, del orden de los dos últimos pasos del predictor.

Sea un método iterativo de orden p cuyo penúltimo paso es de orden q ; el penúltimo y el último puntos obtenidos a través de este método iterativo:

$$\begin{aligned} y_k &= \xi + m_q e_k^q + m_{q+1} e_k^{q+1} + \dots + m_{2q} e_k^{2q} + m_{2q+1} e_k^{2q+1} + \dots + O[e_k^{5q}] \\ &= \xi + \sum_{j=q}^{5q-1} m_j e_k^j + O[e_k^{5q}], \end{aligned} \quad (3.41)$$

$$\begin{aligned} z_k &= \xi + n_p e_k^p + n_{p+1} e_k^{p+1} + \dots + n_{2p} e_k^{2p} + n_{2p+1} e_k^{2p+1} + \dots + O[e_k^{5p}] \\ &= \xi + \sum_{j=p}^{5p-1} n_j e_k^j + O[e_k^{5p}]. \end{aligned} \quad (3.42)$$

Usaremos este método iterativo como predictor y como corrector introducimos el paso de Gauss:

$$x_{k+1} = y_k - \frac{2f(y_k)}{\sum_{i=1}^n \omega_i f'(\eta_{i,k})}, \quad (3.43)$$

donde $\eta_{i,k} = \frac{(1 + \tau_i)z_k + (1 - \tau_i)y_k}{2}$. Para simplificar los cálculos, usaremos la siguiente notación:

$$\begin{aligned} \sum_{j=q}^{5q-1} m_j e_k^j &= A_{1(q)}, \\ \sum_{j=p}^{5p-1} n_j e_k^j &= A_{2(p)}. \end{aligned}$$

donde en el subíndice de la expresión entre paréntesis denota el valor de la potencia más pequeña que toma j en la suma. Usando los pasos y_k y z_k calculamos:

$$\begin{aligned}\eta_{k,i} &= \frac{1}{2}[(1 + \tau_i)z_k + (1 - \tau_i)y_k] \\ &= \xi + \frac{1}{2}(A_{2(p)} + A_{1(q)}) + \frac{1}{2}\tau_i(A_{2(p)} - A_{1(q)}) \\ &= \xi + \frac{1}{2}(R + \tau_i S)_{(q)},\end{aligned}$$

donde obtenemos $R_{(q)} = \frac{1}{2}(A_{2(p)} + A_{1(q)})$ y $S_{(q)} = \frac{1}{2}\tau_i(A_{2(p)} - A_{1(q)})$. Mediante la expansión de $f(y_k)$ y $f'(\eta_{k,i})$ en serie de Taylor alrededor de ξ , obtenemos:

$$\begin{aligned}f(y_k) &= f'(\xi) \left[A_{1(q)} + c_2 A_{1(2q)}^2 + c_3 A_{1(3q)}^3 + c_4 A_{1(4q)}^4 \right] + O[e_k^{5q}], \\ f'(\eta_{k,i}) &= f'(\xi) \left[1 + 2c_2(\eta_{k,i} - \xi)_{(q)} + 3c_3(\eta_{k,i} - \xi)_{(2q)}^2 + 4c_4(\eta_{k,i} - \xi)_{(3q)}^3 \right] + O[e_k^{4q}] \\ &= f'(\xi) \left[1 + (B + C\tau_i + D\tau_i^2 + E\tau_i^3 + F\tau_i^4)_{(q)} \right] + O[e_k^{4q}].\end{aligned}$$

Desarrollamos la expresión $K = \sum_{i=1}^n \omega_i f'(\eta_{k,i})$:

$$\begin{aligned}K &= \sum_{i=1}^n \omega_i f'(\eta_{k,i}) = \sum_{i=1}^n \omega_i f'(\xi) \left[1 + (B + C\tau_i + D\tau_i^2 + E\tau_i^3 + F\tau_i^4)_{(q)} \right] + O[e_k^{4q}] \\ &= f'(\xi) \left[\sum_{i=1}^n \omega_i + \left(B \sum_{i=1}^n \omega_i + C \sum_{i=1}^n \omega_i \tau_i + D \sum_{i=1}^n \omega_i \tau_i^2 + E \sum_{i=1}^n \omega_i \tau_i^3 + F \sum_{i=1}^n \omega_i \tau_i^4 \right)_{(q)} \right] + O[e_k^{4q}].\end{aligned}$$

Empleando la notación (3.18) obtenemos:

$$K = \sigma f'(\xi) [1 + (B + C\sigma_1 + D\sigma_2 + E\sigma_3 + F\sigma_4)] + O[e_k^{4q}].$$

Calculamos $L = \frac{2f(y_k)}{K}$:

$$L = \frac{2}{\sigma} [A_{1(q)} + Q_{1(2q)} + Q_{2(3q)} + Q_{3(4q)}] + O[e_k^{5q}],$$

donde

$$\begin{aligned}Q_{1(2q)} &= c_2 [A_1^2 - A_1(R + \sigma_1 S)]_{(2q)}, \\ Q_{2(3q)} &= \left[c_3 [A_1^3 - \frac{3}{4}A_1(R^2 + 2\sigma_1 RS + \sigma_2 S^2)] - c_2^2 (R + \sigma_1 S) [A_1^2 - A_1(R + \sigma_1 S)] \right]_{(3q)}, \\ Q_{3(4q)} &= c_4 \left[A_1^4 - \frac{1}{2}A_1(R^3 + 3\sigma_1 R^2 S + 3\sigma_2 RS^2 + \sigma_3 S^3) \right]_{(4q)} \\ &\quad - c_2 c_3 \left[\frac{3}{4}(R^2 + 2\sigma_1 RS + \sigma_2 S^2) [A_1^2 - A_1(R + \sigma_1 S)] \right]_{(4q)} \\ &\quad - c_2 c_3 \left[(R + \sigma_1 S) [A_1^3 - \frac{3}{4}A_1(R^2 + 2\sigma_1 RS + \sigma_2 S^2)] \right]_{(4q)} \\ &\quad + c_2^3 [(R^2 + 2\sigma_1 RS + \sigma_2 S^2) [A_1^2 - A_1(R + \sigma_1 S)]]_{(4q)}.\end{aligned}$$

Como $x_{k+1} = y_k - L$, calculamos la ecuación del error:

$$e_{k+1} = A_{1(q)} - \frac{2}{\sigma} A_{1(q)} - \frac{2}{\sigma} [Q_{1(2q)} + Q_{2(3q)} + Q_{3(4q)}] + O[e_k^{5q}]. \quad (3.44)$$

Notamos que si $\sigma = 2$ obtenemos orden de convergencia $2q$. Entonces, la expresión de e_{k+1} queda:

$$e_{k+1} = - [Q_{1(2q)} + Q_{2(3q)} + Q_{3(4q)}] + O[e_k^{5q}]. \quad (3.45)$$

De $Q_{1(2q)}$ depende la posibilidad de tener orden de convergencia mayor que $2q$. Si lo desarrollamos,

$$\begin{aligned} Q_{1(2q)} &= (A_1^2)_{(2q)} - [A_1(q)(R + \sigma_1 S)]_{(2q)} \\ &= (A_1^2)_{(2q)} - A_{1(q)}[(A_2 + A_1) + \sigma_1(A_2 - A_1)]_{(q)} \\ &= -(A_1 A_2)_{(q+p)} - \sigma_1 [A_1(A_2 - A_1)]_{(2q)}, \end{aligned} \quad (3.46)$$

notamos que si $\sigma_1 = 0$ entonces, la expresión de e_{k+1} queda:

$$e_{k+1} = -c_2(A_1 A_2)_{(q+p)} - Q_{2(3q)} - Q_{3(4q)} + O[e_k^{5q}]. \quad (3.47)$$

Nos encontramos con dos posibles casos: si $q + p \geq 3q$ el orden de convergencia será $3q$ y si $q + p < 3q$ el orden será $q + p$. En este último caso tener mayor orden de convergencia que $q + p$ dependerá de $A_1 A_2$ que representa el producto de las expresiones de los errores en el penúltimo y en el último paso del predictor. En estas circunstancias, para tener orden de convergencia al menos $3q$ la derivada segunda evaluada en la raíz buscada debe anularse ($c_2 = 0$). En este caso tendremos la siguiente expresión para la ecuación del error:

$$\begin{aligned} e_{k+1} &= -c_3 \left[A_1^3 - \frac{3}{4} A_1(R^2 + \sigma_2 S^2) \right]_{(3q)} \\ &\quad - c_4 \left[A_1^4 - \frac{1}{2} A_1(R^3 + 3\sigma_2 R S^2 + \sigma_3 S^3) \right]_{(4q)} + O[e_k^{5q}]. \end{aligned} \quad (3.48)$$

Vemos que el orden de convergencia que se obtiene en el caso de que la derivada segunda evaluada en la raíz se anule, depende completamente de las expresiones A_1 y A_2 , de su suma o diferencia y de los valores que tomen σ_2 y σ_3 .

A partir de estos comentarios se puede enunciar el siguiente resultado:

Teorema 3.4.1 *Sea $\xi \in I$ un cero simple de la función $f : I \subseteq \mathbb{R} \rightarrow \mathbb{R}$ suficientemente diferenciable en el intervalo abierto I y consideramos que la estimación inicial x_0 está cerca de ξ . Consideremos un método iterativo de orden de convergencia p , donde y_k y z_k (véase (3.41, 3.42)) son penúltimo y último pasos del método. Tomando este esquema como predictor y a Gauss como corrector obtenemos una nueva aproximación x_{k+1} de ξ dada por*

$$x_{k+1} = y_k - \frac{2f(y_k)}{\sum_{i=1}^n \omega_i f'(\eta_{i,k})},$$

donde $\eta_{i,k} = \frac{1}{2} [(1 + \tau_i)z_k + (1 - \tau_i)y_k]$ y τ_i y ω_i son los nodos y los pesos del polinomio ortogonal correspondiente a la cuadratura gaussiana utilizada. Entonces:

1. El conjunto de familias obtenidas tendrá orden de convergencia al menos q ;
2. Si se verifica la condición $\sigma = 2$ el orden de convergencia será $2q$;
3. Si, además, $\sigma_1 = 0$ el orden de la convergencia será $\min\{q + p, 3q\}$;
4. En cualquier caso, aumentar el orden de convergencia dependerá del producto $A_1 A_2$ y
5. cuando $f''(\xi) = 0$, es decir, $c_2 = 0$, el orden de convergencia es al menos $3q$.

El orden de la convergencia que tendrá el método en el caso de que se cumple la última condición dependerá de las expresiones A_1 y A_2 , de su suma y diferencia y de los valores de σ_2 y σ_3 en algunos casos.

Para ilustrar este resultado consideramos como casos particulares del mismo los métodos en las secciones anteriores.

1. Caso Newton-Gauss: $q = 1$, $p = 2$, entonces
 - a) si $\sigma = 2$, orden $2q = 2$,
 - b) si $\sigma = 2$ y $\sigma_1 = 0$, orden $q + p = 3$ ($q + p = 3q$);
2. Caso Traub-Gauss: $q = 2$, $p = 3$, entonces
 - a) si $\sigma = 2$, orden $2q = 4$,
 - b) si $\sigma = 2$ y $\sigma_1 = 0$, orden $q + p = 5$ ($q + p < 3q$);
3. Caso Ostrowski-Gauss: $q = 2$, $p = 4$, entonces
 - a) si $\sigma = 2$, orden $2q = 4$,
 - b) si $\sigma = 2$ y $\sigma_1 = 0$, orden $q + p = 6$ ($q + p = 3q$);
4. Caso Derivada congelada-Gauss: $q = 3$, $p = 4$, entonces
 - a) si $\sigma = 2$, orden $2q = 6$,
 - b) si $\sigma = 2$ y $\sigma_1 = 0$, orden $q + p = 7$ ($q + p < 3q$).

3.5. Pruebas numéricas

A continuación, utilizamos los mismos ejemplos anteriores

1. $f_1(x) = \sin x - x^2 + 1$, $\xi \approx 1.404492$.
2. $f_2(x) = x^2 - \exp x - 3x + 2$, $\xi \approx 0.257530$.
3. $f_3(x) = \cos x - x$, $\xi \approx 0.739085$.
4. $f_4(x) = (x - 1)^3 - 1$, $\xi = 2$.
5. $f_5(x) = x^3 - 10$, $\xi \approx 2.154435$.
6. $f_6(x) = \cos x - x \exp x + x^2$, $\xi \approx 0.639154$.
7. $f_7(x) = \exp x - 1.5 - \arctan x$, $\xi \approx 0.767653$.
8. $f_8(x) = x^3 + 4x^2 - 10$, $\xi \approx 1.365230$.
9. $f_9(x) = 8x - \cos x - 2x^2$, $\xi \approx 0.128077$.
10. $f_{10}(x) = \arctan x$, $\xi = 0$.
11. $f_{11}(x) = \exp x - 4x^2$, $\xi \approx 0.714806$.
12. $f_{12}(x) = \left(\sin x - \frac{x}{2}\right)^2$, $\xi = 0$, doble.
13. $f_{13}(x) = x \exp x^2 - \sin x^2 + 3 \cos x + 5$, $\xi \approx -1.207648$.
14. $f_{14}(x) = \sqrt{x^2 + 2x + 5} - 2 \sin x - x^2 + 3$, $\xi \approx 2.331968$.
15. $f_{15}(x) = x^4 + \sin \frac{\pi}{x^2} - 5$, $\xi = \sqrt{2}$.
16. $f_{16}(x) = 10x \exp(-x^2) - 1$, $\xi \approx 1.679631$.

17. $f_{17}(x) = \exp(-x) + \cos x$, $\xi \approx 1.746140$.
 18. $f_{18}(x) = \sqrt{x^4 + 8} \sin \frac{\pi}{x^2 + 2} + \frac{x^3}{x^4 + 1} - \sqrt{6} + \frac{8}{17}$, $\xi \approx -2$.

Para ilustrar la eficacia de los métodos desarrollados TrG, OsG y DCG la compararemos con el método clásico de Newton (NC) y el método de Ostrowski (OS) [72]. Los métodos desarrollados tienen orden de convergencia 4, 5 y 6, respectivamente. Además utilizaremos en el estudio comparativo otros tres métodos de órdenes 5, 6 y 8 respectivamente. Recientemente, sobre la base de Ostrowski o Jarratt han sido propuestas algunas familias de métodos de orden superior a cuatro para la resolución de ecuaciones no lineales. Por ejemplo, Cordero et al. en [21] presentan una familia biparamétrica de métodos, aplicable tanto a ecuaciones como a sistemas de ecuaciones, basada en la composición de los métodos de Newton y Jarratt y una posterior modificación de la derivada. Su fórmula iterativa es:

$$\begin{aligned} y_k &= x_k - \frac{2}{3} \frac{f(x_k)}{f'(x_k)} \\ z_k &= x_k - \frac{1}{2} \frac{3f'(y_k) + f'(x_k)}{3f'(y_k) - f'(x_k)} \frac{f(x_k)}{f'(x_k)} \\ x_{k+1} &= z_k - \frac{f(z_k)}{\alpha f'(x_k) + \beta f'(y_k)}. \end{aligned}$$

Para tener orden de convergencia cinco los parámetros de ésta familia deben cumplir la condición $\alpha + \beta = 1$. En este caso escogemos $\alpha = 1$ y $\beta = 0$ y lo denotamos por R1. En el caso de convergencia de orden seis los parámetros deben tener valores específicos $\alpha = -\frac{1}{2}$ y $\beta = \frac{3}{2}$ y en este caso el método lo denotamos por RR1.

Por otro lado, Liu y Wang han desarrollado en [65] una familia de variantes del método Ostrowski con orden de convergencia ocho por el método de funciones de peso. La expresión iterativa de esta familia, a la que nos referiremos como LW8, es:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)} \\ z_k &= x_k - \frac{f(x_k) - f(y_k)}{f(x_k) - 2f(y_k)} \frac{f(x_k)}{f'(x_k)} \\ x_{k+1} &= z_k - \left[\left(\frac{f(x_k) - f(y_k)}{f(x_k) - 2f(y_k)} \right)^2 + \frac{f(z_k)}{f(y_k) - \alpha f(z_k)} + G(\mu_k) \right], \end{aligned} \tag{3.49}$$

donde α es un parámetro, $\mu_k = f(z_k)/f(x_k)$ y $G(\mu_k)$ es una función real. Para los estudios comparativos emplearemos este método con $\alpha = 1$ y $G(t) = 4t$ como representante de los métodos de orden ocho.

En la Tabla 3.8 comparamos los distintos índices de eficiencia de los métodos empleados en las pruebas numéricas

Tabla 3.8: Datos de los nuevos métodos iterativos

Métodos	p	d	op	I	IO	IC
TrG-LD1	5	4	3	1.4953	1.7100	1.2585
OsG-LD1	6	4	4	1.5651	1.5651	1.2510
DCG-LD1	7	5	4	1.4758	1.6266	1.2414
NC	2	2	1	1.4142	2.0000	1.2599
OS	4	4	3	1.5874	1.5874	1.2599
DC	5	4	3	1.4142	1.5874	1.2190
RR1	6	4	4	1.5651	1.5651	1.2510
LW8	8	4	6	1.6818	1.4142	1.2311

Obviamente, el mayor índice de eficiencia lo tienen los dos métodos óptimos: LW8 (1.6818) y OS (1.5874). Les siguen los métodos OsG-LD1 y RR1 (0.5651). El mejor índice operacional es el de NC y le siguen los métodos:

$TrG - LD1$ (1.7100) y $DCG - LD1$ (1.6266). Comparando los métodos a partir de los índices de eficiencia computacional notamos que los mejores son los métodos NC y OS . Les siguen los métodos $TrG - LD1$, $OsG - LD1$, $RR1$ y $DCG - LD1$. Notamos que el método $LW8$ es el menos eficiente computacionalmente.

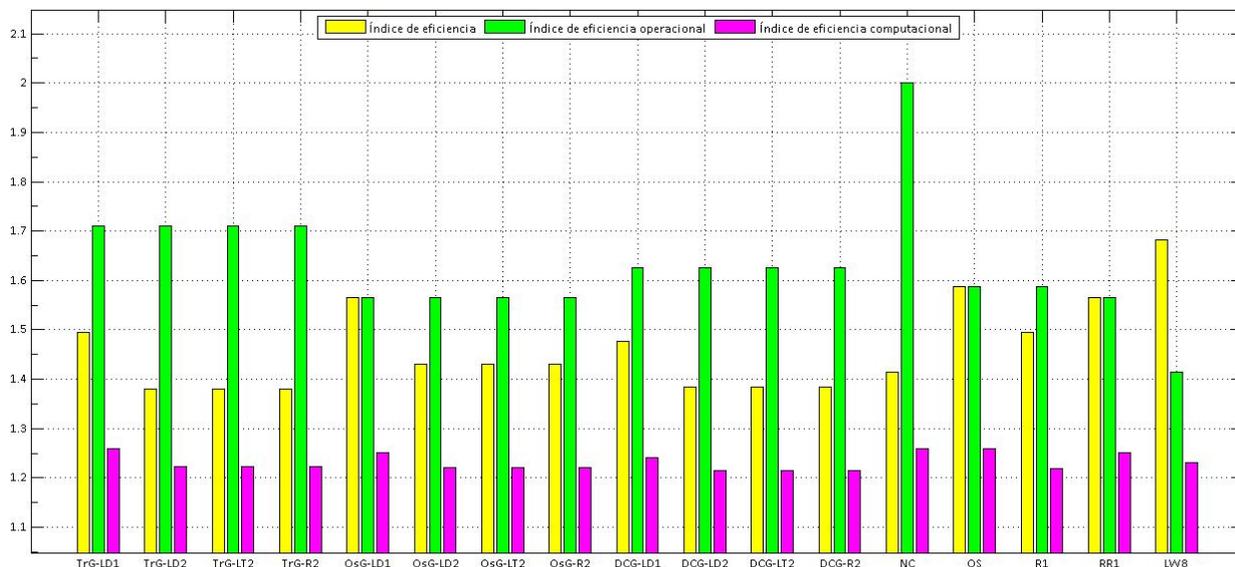


Figura 3.2: Comparación de los índices

En la Figura 3.2 se muestra una comparación de los diferentes índices de eficiencia alcanzados por los métodos mencionados anteriormente.

En la Tabla 3.9 presentamos los resultados obtenidos al utilizar los métodos descritos en esta sección para estimar los ceros de las funciones de f_1 a f_{18} . Todos los cálculos se han realizado bajo las mismas premisas que se utilizaron para obtener la Tabla 3.4.

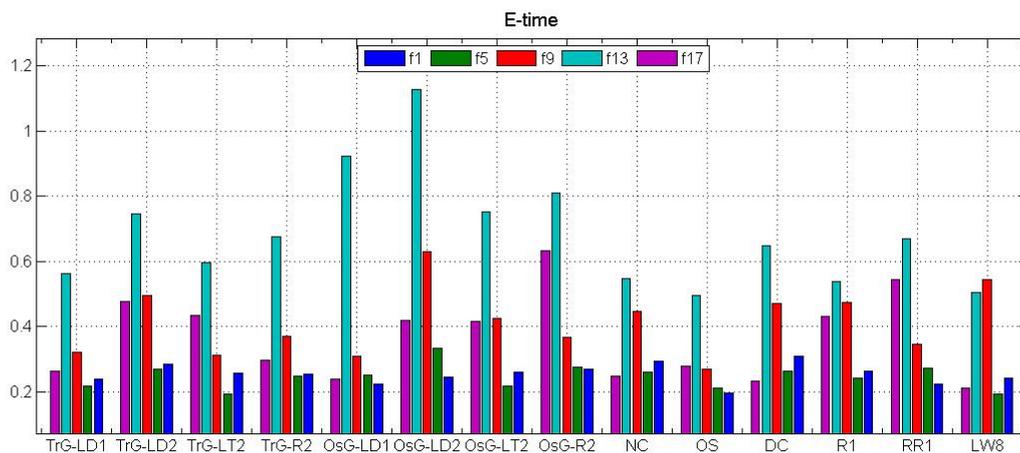


Figura 3.3: Comparación de los tiempos de ejecución

Los resultados numéricos están en concordancia con la teoría desarrollada en esta sección. De nuevo notamos los dos casos particulares: $f_{10}(x) = 0$ y $f_{12}(x) = 0$. En el caso de $f_{10}(x) = 0$, $c_2 = 0$ y notamos que los métodos de las familias TrG , OsG y DCG se comportan según lo previsto en los Teoremas 3.3.1, 3.3.2 y 3.3.3. En el caso de $f_{12}(x) = 0$, que tiene una raíz doble, todos los métodos estudiados muestran convergencia de

orden uno con elevado número de iteraciones. Los métodos de la familia OsG y DCG tienen menor número de iteraciones (208) y el método de Newton emplea la mayor cantidad de iteraciones (536). Los métodos de Ostrowski y TrG-LD1 obtienen la raíz con el menor tiempo computacional $\epsilon \approx 18.5s$ mientras que el método LW8 no converge. La precisión de todos los métodos es de orden $\approx 10^{-162}$.

La Figura 3.3 muestra el tiempo de ejecución ϵ empleado por todos los métodos presentados en la Tabla 3.9 para las funciones f_1, f_5, f_9, f_{13} y f_{17} escogidas al azar. En general, de entre los métodos introducidos en esta sección, el más rápido es TrG-LD1.

Tabla 3.9: Ejemplos numéricos

	Métodos	ξ	$ x_{k+1} - x_k $	$ f(x_{k+1}) $	Iter	ρ	ϵ, s
$f_1,$ $x_0 = 1$	TrG-LD1	1.409624	$6.7e-248$	0	5	5.0000	0.2141
	TrG-LD2	1.409624	$4.4e-248$	0	5	5.0000	0.2779
	TrG-LT2	1.409624	$1.9e-248$	0	5	5.0000	0.2324
	TrG-R2	1.409624	$4.9e-248$	0	5	5.0000	0.2558
	OsG-LD1	1.409624	$1.0e-116$	0	4	5.9999	0.1914
	OsG-LD2	1.409624	$1.2e-116$	0	4	5.9999	0.2469
	OsG-LT2	1.409624	$1.8e-116$	0	4	5.9999	0.2013
	OsG-R2	1.409624	$1.2e-116$	0	4	5.9999	0.2232
	DCG-LD1	1.409624	$1.3e-102$	0	4	7.0022	0.2242
	DCG-LD2	1.409624	$1.5e-107$	0	4	7.0022	0.2462
	DCG-LT2	1.409624	$2.1e-102$	0	4	7.0022	0.2606
	DCG-R2	1.409624	$1.5e-102$	0	4	7.0022	0.2696
	NC	1.409624	$1.9e-273$	0	10	2.0000	0.2933
	OS	1.409624	$7.3e-139$	0	5	4.0000	0.1977
	DC	1.409624	$5.4e-271$	0	6	4.0000	0.3098
	R1	1.409624	0	0	5	4.9996	0.2653
	RR1	1.409624	0	0	4	6.0215	0.2257
LW8	1.409624	0	0	4	8.1069	0.2435	
$f_2,$ $x_0 = 1$	TrG-LD1	0.257530	$2.0e-193$	0	4	5.0000	0.2329
	TrG-LD2	0.257530	$4.9e-192$	0	4	5.0000	0.3002
	TrG-LT2	0.257530	$1.0e-189$	0	4	5.0000	0.2433
	TrG-R2	0.257530	$4.9e-192$	0	4	5.0000	0.2628
	OsG-LD1	0.257530	$8.9e-256$	0	4	6.0000	0.2364
	OsG-LD2	0.257530	$6.4e-256$	0	4	6.0000	0.3197
	OsG-LT2	0.257530	$2.0e-257$	0	4	6.0000	0.2549
	OsG-R2	0.257530	$6.4e-256$	0	4	6.0000	0.2714
	DCG-LD1	0.257530	$2.2e-078$	0	3	6.5924	0.2904
	DCG-LD2	0.257530	$2.2e-078$	0	3	6.5923	0.3703
	DCG-LT2	0.257530	$2.1e-078$	0	3	6.5922	0.2835
	DCG-R2	0.257530	$2.2e-078$	0	3	6.5923	0.3239
	NC	0.257530	$4.5e-190$	0	8	2.0000	0.2721
	OS	0.257530	$1.8e-139$	0	5	4.0000	0.2659
	DC	0.257530	$5.7e-102$	0	4	4.0000	0.2086
	R1	0.257530	0	0	4	4.8658	0.2866
	RR1	0.257530	0	0	4	6.1725	0.2866
LW8	0.257530	$1.1e-201$	0	3	8.1108	0.2580	

Tabla 3.9 Ejemplos numéricos, continuación...

	Métodos	ξ	$ x_{k+1} - x_k $	$ f(x_{k+1}) $	<i>Iter</i>	ρ	ϵ, s
$f_3,$ $x_0 = 1$	TrG-LD1	0.739085	$1.0e-154$	0	4	5.0000	0.1967
	TrG-LD2	0.739085	$8.5e-155$	0	4	5.0000	0.2524
	TrG-LT2	0.739085	$5.6e-155$	0	4	5.0000	0.1995
	TrG-R2	0.739085	$8.5e-155$	0	4	5.0000	0.1989
	OsG-LD1	0.739085	$4.7e-260$	0	4	6.0000	0.1900
	OsG-LD2	0.739085	$3.2e-262$	0	4	6.0000	0.2485
	OsG-LT2	0.739085	$1.6e-267$	0	4	6.0000	0.1921
	OsG-R2	0.739085	$3.2e-262$	0	4	6.0000	0.2186
	DCG-LD1	0.739085	$1.3e-057$	0	3	6.9188	0.1523
	DCG-LD2	0.739085	$1.3e-057$	0	3	6.9188	0.1919
	DCG-LT2	0.739085	$1.3e-057$	0	3	6.9187	0.1519
	DCG-R2	0.739085	$1.3e-057$	0	3	6.9188	0.1884
	NC	0.739085	$7.1e-177$	0	8	2.0000	0.2405
	OS	0.739085	$4.2e-296$	0	5	4.0000	0.2295
	DC	0.739085	$5.7e-288$	0	5	4.0000	0.2023
	R1	0.739085	0	0	4	4.9985	0.2209
	RR1	0.739085	0	0	4	6.0041	0.2689
LW8	0.739085	$7.7e-286$	0	3	8.0668	0.1797	
$f_4,$ $x_0 = 2.5$	TrG-LD1	2.000000	$3.8e-246$	0	4	5.0000	0.2448
	TrG-LD2	2.000000	$8.0e-246$	0	4	5.0000	0.3246
	TrG-LT2	2.000000	$3.4e-245$	0	4	5.0000	0.2659
	TrG-R2	2.000000	$8.0e-246$	0	4	5.0000	0.2968
	OsG-LD1	2.000000	$3.8e-102$	0	4	5.9998	0.3017
	OsG-LD2	2.000000	$4.2e-100$	0	4	5.9998	0.3673
	OsG-LT2	2.000000	$1.4e-096$	0	4	5.9998	0.2532
	OsG-R2	2.000000	$4.2e-100$	0	4	5.9998	0.2965
	DCG-LD1	2.000000	$7.2e-121$	0	4	6.9984	0.2588
	DCG-LD2	2.000000	$8.1e-121$	0	4	6.9984	0.3352
	DCG-LT2	2.000000	$1.0e-120$	0	4	6.9984	0.2963
	DCG-R2	2.000000	$8.1e-121$	0	4	6.9984	0.3025
	NC	2.000000	$7.9e-224$	0	10	2.0000	0.3104
	OS	2.000000	$4.1e-123$	0	5	4.0000	0.2288
	DC	2.000000	$4.7e-309$	0	6	4.0000	0.3084
	R1	2.000000	0	0	5	5.0006	0.3118
	RR1	2.000000	0	0	4	6.0374	0.2771
LW8	2.000000	0	0	5	7.8971	0.2892	
$f_5,$ $x_0 = 2$	TrG-LD1	2.154435	$2.6e-130$	0	4	5.0000	0.2195
	TrG-LD2	2.154435	$2.4e-130$	0	4	5.0000	0.2700
	TrG-LT2	2.154435	$1.9e-130$	0	4	5.0000	0.1950
	TrG-R2	2.154435	$2.4e-130$	0	4	5.0000	0.2475
	OsG-LD1	2.154435	$1.3e-253$	0	4	6.0000	0.2532
	OsG-LD2	2.154435	$4.7e-251$	0	4	6.0000	0.3346
	OsG-LT2	2.154435	$8.0e-247$	0	4	6.0000	0.2188
	OsG-R2	2.154435	$4.7e-251$	0	4	6.0000	0.2751
	DCG-LD1	2.154435	$2.1e-047$	0	3	7.0465	0.1660
	DCG-LD2	2.154435	$2.1e-047$	0	3	7.0465	0.2583
	DCG-LT2	2.154435	$2.2e-047$	0	3	7.0466	0.1708
	DCG-R2	2.154435	$2.1e-047$	0	3	7.0465	0.2264
	NC	2.154435	$4.5e-288$	0	9	2.0000	0.2616
	OS	2.154435	$1.1e-303$	0	5	4.0000	0.2125
	DC	2.154435	$2.7e-232$	0	5	4.0000	0.2635
	R1	2.154435	0	0	4	4.9985	0.2418
	RR1	2.154435	0	0	4	6.0000	0.2747
LW8	2.154435	7.4^{-288}	0	3	8.0065	0.1942	

Tabla 3.9 Ejemplos numéricos, continuación...

	Métodos	ξ	$ x_{k+1} - x_k $	$ f(x_{k+1}) $	Iter	ρ	ϵ, s
$f_6,$ $x_0 = 1$	TrG-LD1	0.639154	$4.7e-068$	0	4	4.9998	0.3275
	TrG-LD2	0.639154	$6.2e-068$	0	4	4.9998	0.3632
	TrG-LT2	0.639154	$1.1e-068$	0	4	4.9998	0.3539
	TrG-R2	0.639154	$6.2e-068$	0	4	4.9998	0.3211
	OsG-LD1	0.639154	$5.4e-293$	0	5	6.0000	0.4500
	OsG-LD2	0.639154	$5.2e-259$	0	5	6.0000	0.4975
	OsG-LT2	0.639154	$3.5e-218$	0	5	6.0000	0.4777
	OsG-R2	0.639154	$1.3e-257$	0	5	6.0000	0.5687
	DCG-LD1	0.639154	$5.6e-167$	0	4	7.0000	0.3531
	DCG-LD2	0.639154	$6.8e-167$	0	4	7.0000	0.4657
	DCG-LT2	0.639154	$9.8e-167$	0	4	7.0000	0.4267
	DCG-R2	0.639154	$7.8e-167$	0	4	7.0000	0.3956
	NC	0.639154	$2.3e-303$	0	10	2.0000	0.3829
	OS	0.639154	$3.9e-187$	0	5	4.0000	0.3654
	DC	0.639154	$1.8e-113$	0	5	4.0000	0.3794
	R1	0.639154	$7.6e-313$	0	4	5.0000	0.3428
	RR1	0.639154	0	0	4	6.0047	0.3768
	LW8	0.639154	$6.7e-172$	0	3	7.9880	0.3358
$f_7,$ $x_0 = 1$	TrG-LD1	0.767653	$2.2e-086$	0	4	4.9999	0.2541
	TrG-LD2	0.767653	$2.4e-086$	0	4	4.9999	0.4004
	TrG-LT2	0.767653	$2.9e-086$	0	4	4.9999	0.2769
	TrG-R2	0.767653	$2.4e-086$	0	4	4.9999	0.3029
	OsG-LD1	0.767653	$1.9e-167$	0	4	6.0000	0.2751
	OsG-LD2	0.767653	$6.2e-166$	0	4	6.0000	0.5157
	OsG-LT2	0.767653	$3.0e-163$	0	4	6.0000	0.2642
	OsG-R2	0.767653	$6.2e-166$	0	4	6.0000	0.3417
	DCG-LD1	0.767653	$4.6e-216$	0	4	7.0000	0.2900
	DCG-LD2	0.767653	$4.8e-216$	0	4	7.0000	0.3827
	DCG-LT2	0.767653	$5.3e-216$	0	4	7.0000	0.4081
	DCG-R2	0.767653	$4.8e-216$	0	4	7.0000	0.4191
	NC	0.767653	$1.4e-190$	0	9	2.0000	0.3170
	OS	0.767653	$1.6e-200$	0	5	4.0000	0.2922
	DC	0.767653	$7.5e-150$	0	5	4.0000	0.3064
	R1	0.767653	0	0	4	5.0485	0.3157
	RR1	0.767653	0	0	4	6.0054	0.3583
	LW8	0.767653	$3.1e-191$	0	3	7.9730	0.3020
$f_8,$ $x_0 = 1$	TrG-LD1	1.365230	$1.1e-072$	0	4	5.0002	0.3368
	TrG-LD2	1.365230	$7.8e-073$	0	4	5.0002	0.3986
	TrG-LT2	1.365230	$4.2e-073$	0	4	5.0002	0.4055
	TrG-R2	1.365230	$7.8e-073$	0	4	5.0002	0.3183
	OsG-LD1	1.365230	$3.6e-156$	0	4	6.0000	0.4263
	OsG-LD2	1.365230	$1.7e-154$	0	4	6.0000	0.4345
	OsG-LT2	1.365230	$1.5e-151$	0	4	6.0000	0.4277
	OsG-R2	1.365230	$1.7e-154$	0	4	6.0000	0.3421
	DCG-LD1	1.365230	$1.0e-166$	0	4	7.0001	0.4415
	DCG-LD2	1.365230	$1.5e-166$	0	4	7.0001	0.4523
	DCG-LT2	1.365230	$3.0e-166$	0	4	7.0001	0.4148
	DCG-R2	1.365230	$1.5e-166$	0	4	7.0001	0.4551
	NC	1.365230	$4.0e-176$	0	9	2.0000	0.4000
	OS	1.365230	$1.5e-187$	0	5	4.0000	0.3112
	DC	1.365230	$3.2e-112$	0	5	4.0000	0.3206
	R1	1.365230	$2.3e-315$	0	4	5.0000	0.4039
	RR1	1.365230	0	0	4	6.0080	0.3969
	LW8	1.365230	$4.1e-169$	0	3	8.0336	0.3468

Tabla 3.9 Ejemplos numéricos, continuación...

	Métodos	ξ	$ x_{k+1} - x_k $	$ f(x_{k+1}) $	<i>Iter</i>	ρ	ϵ, s
$f_9,$ $x_0 = 1$	TrG-LD1	0.128077	$7.1e-068$	0	4	5.0005	0.3218
	TrG-LD2	0.128077	$7.6e-068$	0	4	5.0005	0.4959
	TrG-LT2	0.128077	$8.6e-068$	0	4	5.0005	0.3117
	TrG-R2	0.128077	$7.3e-068$	0	4	5.0005	0.3702
	OsG-LD1	0.128077	$3.7e-139$	0	4	6.0000	0.3099
	OsG-LD2	0.128077	$3.4e-139$	0	4	6.0000	0.6294
	OsG-LT2	0.128077	$2.8e-139$	0	4	6.0000	0.4268
	OsG-R2	0.128077	$3.8e-139$	0	4	6.0000	0.3680
	DCG-LD1	0.128077	$1.9e-151$	0	4	7.0002	0.3231
	DCG-LD2	0.128077	$1.6e-151$	0	4	7.0002	0.4780
	DCG-LT2	0.128077	$1.1e-151$	0	4	7.0002	0.5303
	DCG-R2	0.128077	$1.5e-151$	0	4	7.0002	0.3910
	NC	0.128077	$9.0e-169$	0	9	2.0000	0.4455
	OS	0.128077	$8.3e-163$	0	5	4.0000	0.2704
	DC	0.128077	$3.7e-101$	0	5	4.0000	0.4717
	R1	0.128077	$7.5e-269$	0	4	4.9999	0.4742
	RR1	0.128077	0	0	4	6.0127	0.3453
LW8	0.128077	0	0	4	8.0575	0.5445	
$f_{10},$ $x_0 = 0.5$	TrG-LD1	0.000000	$1.7e-038$	0	3	9.0803	0.2336
	TrG-LD2	0.000000	$2.6e-051$	0	3	10.9540	0.2648
	TrG-LT2	0.000000	$4.4e-038$	0	3	9.0029	0.2581
	TrG-R2	0.000000	$3.0e-051$	0	3	10.9558	0.2060
	OsG-LD1	0.000000	$4.7e-044$	0	3	8.9265	0.1478
	OsG-LD2	0.000000	$4.7e-056$	0	3	10.9779	0.2436
	OsG-LT2	0.000000	$1.1e-038$	0	3	8.8950	0.2296
	OsG-R2	0.000000	$4.0e-056$	0	3	10.9767	0.3104
	DCG-LD1	0.000000	$7.8e-090$	0	3	15.0524	0.2069
	DCG-LD2	0.000000	$3.3e-109$	0	3	16.9669	0.2239
	DCG-LT2	0.000000	$2.3e-089$	0	3	15.0000	0.2660
	DCG-R2	0.000000	$3.3e-109$	0	3	16.9669	0.3439
	NC	0.000000	$3.3e-289$	0	7	3.0000	0.2206
	OS	0.000000	$4.1e-302$	0	5	5.0000	0.2567
	DC	0.000000	$3.1e-115$	0	4	7.0000	0.2700
	R1	0.000000	0	0	4	6.9609	0.3629
	RR1	0.000000	0	0	4	7.0100	0.2947
LW8	0.000000	$4.8e-207$	0	3	8.9388	0.2302	
$f_{11},$ $x_0 = 1$	TrG-LD1	0.714806	$2.8e-086$	0	4	4.9998	0.3275
	TrG-LD2	0.714806	$2.6e-086$	0	4	4.9998	0.3632
	TrG-LT2	0.714806	$2.3e-086$	0	4	4.9998	0.3539
	TrG-R2	0.714806	$2.6e-086$	0	4	4.9998	0.3211
	OsG-LD1	0.714806	$7.6e-157$	0	4	6.0000	0.3845
	OsG-LD2	0.714806	$2.1e-157$	0	4	6.0000	0.5095
	OsG-LT2	0.714806	$1.4e-158$	0	4	6.0000	0.2707
	OsG-R2	0.714806	$2.1e-157$	0	4	6.0000	0.4036
	DCG-LD1	0.714806	$5.9e-217$	0	4	7.0000	0.3365
	DCG-LD2	0.714806	$5.8e-217$	0	4	7.0000	0.4270
	DCG-LT2	0.714806	$5.5e-217$	0	4	7.0000	0.3865
	DCG-R2	0.714806	$5.8e-217$	0	4	7.0000	0.4097
	NC	0.714806	$5.3e-190$	0	9	2.0000	0.3829
	OS	0.714806	$3.6e-184$	0	5	4.0000	0.3654
	DC	0.714806	$8.9e-151$	0	5	4.0000	0.2379
	R1	0.714806	0	0	4	5.0429	0.3428
	RR1	0.714806	0	0	4	6.0069	0.3768
LW8	0.714806	$3.5e-191$	0	3	7.9076	0.3358	

Tabla 3.9 Ejemplos numéricos, continuación...

	Métodos	ξ	$ x_{k+1} - x_k $	$ f(x_{k+1}) $	Iter	ρ	ϵ, s
$f_{12},$ $x_0 = 0.5$	TrG-LD1	0.000000	$8.1e-162$	0	241	1.0000	18.8305
	TrG-LD2	0.000000	$8.1e-162$	0	239	1.0000	24.8066
	TrG-LT2	0.000000	$8.1e-162$	0	239	1.0000	20.0354
	TrG-R2	0.000000	$8.1e-162$	0	239	1.0000	20.7758
	OsG-LD1	0.000000	$2.9e-162$	0	208	1.0000	28.6165
	OsG-LD2	0.000000	$2.9e-162$	0	208	1.0000	23.4529
	OsG-LT2	0.000000	$2.9e-162$	0	208	1.0000	23.8791
	OsG-R2	0.000000	$2.9e-162$	0	208	1.0000	31.1729
	DCG-LD1	0.000000	$1.5e-161$	0	208	1.0000	20.3960
	DCG-LD2	0.000000	$1.5e-161$	0	208	1.0000	26.5028
	DCG-LT2	0.000000	$1.5e-161$	0	208	1.0000	26.7585
	DCG-R2	0.000000	$1.5e-161$	0	208	1.0000	28.8762
	NC	0.000000	$1.7e-162$	0	536	1.0000	21.0063
	OS	0.000000	$6.0e-162$	0	266	1.0000	18.4757
	DC	0.000000	$2.3e-162$	0	313	1.0000	22.8873
	R1	0.000000	$1.2e-163$	0	243	1.0000	21.3531
	RR1	0.000000	$6.0e-163$	0	220	1.0000	20.5260
LW8	<i>n.c.</i>	<i>n.c.</i>	<i>n.c.</i>	<i>n.c.</i>	<i>n.c.</i>	<i>n.c.</i>	<i>n.c.</i>
$f_{13},$ $x_0 = -1$	TrG-LD1	-1.20158	$6.1e-268$	0	5	5.0000	0.5633
	TrG-LD2	-1.20158	$1.9e-272$	0	5	5.0000	0.7442
	TrG-LT2	-1.20158	$1.2e-282$	0	5	5.0000	0.5959
	TrG-R2	-1.20158	$9.9e-273$	0	5	5.0000	0.6745
	OsG-LD1	-1.20158	$4.9e-146$	0	4	6.0000	0.9235
	OsG-LD2	-1.20158	$4.6e-183$	0	4	6.0000	1.1262
	OsG-LT2	-1.20158	$1.7e-129$	0	4	6.0000	0.7520
	OsG-R2	-1.20158	$5.8e-179$	0	4	6.0000	0.8081
	DCG-LD1	-1.20158	$6.2e-122$	0	4	7.0002	0.6883
	DCG-LD2	-1.20158	$1.7e-120$	0	4	7.0002	1.9068
	DCG-LT2	-1.20158	$7.4e-118$	0	4	7.0003	0.8373
	DCG-R2	-1.20158	$1.5e-120$	0	4	7.0002	0.9691
	NC	-1.20158	$9.6e-278$	0	10	2.0000	0.5465
	OS	-1.20158	$2.3e-266$	0	5	4.0000	0.4954
	DC	-1.20158	$1.9e-315$	0	6	4.0000	0.6466
	R1	-1.20158	0	0	4	4.9326	0.5390
	RR1	-1.20158	0	0	4	6.0173	0.6690
LW8	-1.20158	$1.1e-259$	0	3	7.9547	0.5051	
$f_{14},$ $x_0 = 2$	TrG-LD1	2.331968	$7.5e-198$	0	4	5.0000	0.7356
	TrG-LD2	2.331968	$3.3e-198$	0	4	5.0000	1.0443
	TrG-LT2	2.331968	$6.0e-199$	0	4	5.0000	0.6954
	TrG-R2	2.331968	$3.3e-198$	0	4	5.0000	0.7905
	OsG-LD1	2.331968	$2.4e-302$	0	4	6.0000	1.2524
	OsG-LD2	2.331968	$3.5e-305$	0	4	6.0000	1.1990
	OsG-LT2	2.331968	$7.4e-313$	0	4	6.0000	0.9025
	OsG-R2	2.331968	$3.6e-303$	0	4	6.0000	1.1327
	DCoG-LD1	2.331968	$1.4e-074$	0	3	6.8540	0.8153
	DCoG-LD2	2.331968	$1.4e-074$	0	3	6.8540	0.9250
	DCoG-LT2	2.331968	$1.4e-074$	0	3	6.8541	0.9676
	DCoG-R2	2.331968	$1.4e-074$	0	3	6.8540	0.9882
	NC	2.331968	$3.7e-211$	0	8	2.0000	0.7165
	OS	2.331968	$3.5e-323$	0	5	3.9998	0.7312
	DC	2.331968	$1.5e-094$	0	4	4.0000	0.7792
	R1	2.331968	0	0	4	5.0099	0.8912
	RR1	2.331968	0	0	4	6.0909	0.8630
LW8	2.331968	$1.1e-280$	0	5	8.0286	0.8091	

Tabla 3.9 Ejemplos numéricos, continuación...

	Métodos	ξ	$ x_{k+1} - x_k $	$ f(x_{k+1}) $	<i>Iter</i>	ρ	ϵ, s
$f_{15},$ $x_0 = 1$	TrG-LD1	1.41424	$1.2e-111$	0	4	5.0000	0.5672
	TrG-LD2	1.41424	$2.5e-110$	0	4	5.0000	0.8070
	TrG-LT2	1.41424	$4.1e-108$	0	4	5.0000	0.6599
	TrG-R2	1.41424	$2.6e-110$	0	4	5.0000	0.7323
	OsG-LD1	1.41424	$1.5e-144$	0	4	6.0000	0.6496
	OsG-LD2	1.41424	$8.0e-146$	0	4	6.0000	1.2676
	OsG-LT2	1.41424	$9.5e-145$	0	4	5.9999	0.7735
	OsG-R2	1.41424	$8.1e-146$	0	4	6.0000	1.0227
	DCG-LD1	1.41424	$1.4e-314$	0	4	7.0000	0.8053
	DCG-LD2	1.41424	$9.8e-315$	0	4	7.0000	0.9631
	DCG-LT2	1.41424	$5.1e-315$	0	4	7.0000	0.8807
	DCG-R2	1.41424	$9.8e-315$	0	4	7.0000	0.9265
	NC	1.41424	$2.9e-215$	0	9	2.0000	0.4989
	OS	1.41424	$7.0e-156$	0	5	4.0000	0.6759
	DC	1.41424	$1.4e-229$	0	5	4.0000	0.7364
	R1	1.41424	$1.3e-307$	0	4	4.9998	0.6539
	RR1	1.41424	0	0	4	6.1807	0.6755
LW8	1.41424	0	0	4	8.0235	0.7447	
$f_{16},$ $x_0 = 2$	TrG-LD1	1.679631	$2.1e-156$	0	5	5.0000	0.7006
	TrG-LD2	1.679631	$1.4e-158$	0	5	5.0000	0.7276
	TrG-LT2	1.679631	$4.2e-162$	0	5	5.0000	0.6322
	TrG-R2	1.679631	$4.9e-157$	0	5	5.0000	0.6440
	OsG-LD1	1.679631	$5.9e-096$	0	4	5.9997	0.6282
	OsG-LD2	1.679631	$4.9e-097$	0	4	5.9997	1.0873
	OsG-LT2	1.679631	$1.5e-099$	0	4	5.9997	0.5799
	OsG-R2	1.679631	$2.2e-097$	0	4	5.9997	1.1589
	DCG-LD1	1.679631	$4.0e-060$	0	4	7.0257	0.4846
	DCG-LD2	1.679631	$3.3e-059$	0	4	7.0272	0.8095
	DCG-LT2	1.679631	$7.7e-058$	0	4	7.0296	0.7794
	DCG-R2	1.679631	$8.9e-059$	0	4	7.0279	0.9003
	NC	1.679631	$4.7e-219$	0	10	2.0000	0.6829
	OS	1.679631	$9.6e-118$	0	5	4.0000	0.4352
	DC	1.679631	$1.8e-179$	0	6	4.0000	0.8262
	R1	1.679631	0	0	5	4.9986	0.0891
	RR1	1.679631	0	0	4	6.0442	0.4856
LW8	1.679631	0	0	5	8.1613	0.7079	
$f_{17},$ $x_0 = 2$	TrG-LD1	1.7461395	$9.7e-155$	0	4	5.0000	0.2633
	TrG-LD2	1.7461395	$2.2e-154$	0	4	5.0000	0.4775
	TrG-LT2	1.7461395	$1.1e-153$	0	4	5.0000	0.4357
	TrG-R2	1.7461395	$2.2e-154$	0	4	5.0000	0.2973
	OsG-LD1	1.7461395	$2.9e-252$	0	4	6.0000	0.2390
	OsG-LD2	1.7461395	$1.2e-252$	0	4	6.0000	0.4185
	OsG-LT2	1.7461395	$7.0e-271$	0	4	6.0000	0.4175
	OsG-R2	1.7461395	$1.2e-256$	0	4	6.0000	0.6323
	DCG-LD1	1.7461395	$4.3e-056$	0	3	7.1286	0.2575
	DCG-LD2	1.7461395	$4.2e-056$	0	3	7.1273	0.4327
	DCG-LT2	1.7461395	$4.0e-056$	0	3	7.1278	0.4056
	DCG-R2	1.7461395	$4.2e-056$	0	3	7.1283	0.2007
	NC	1.7461395	$9.6e-170$	0	8	2.0000	0.2503
	OS	1.7461395	$9.0e-280$	0	5	4.0000	0.2799
	DC	1.7461395	$4.4e-279$	0	5	4.0000	0.2327
	R1	1.7461395	0	0	4	5.0092	0.4305
	RR1	1.7461395	0	0	4	5.9937	0.5446
LW8	1.7461395	$5.9e-273$	0	3	7.9370	0.2134	

Tabla 3.9 Ejemplos numéricos, continuación...

	Métodos	ξ	$ x_{k+1} - x_k $	$ f(x_{k+1}) $	Iter	ρ	ϵ, s
$f_{18},$ $x_0 = -1.8$	TrG-LD1	≈ -2	$1.4e-103$	0	4	5.0000	1.8215
	TrG-LD2	≈ -2	$8.3e-103$	0	4	5.0000	2.5532
	TrG-LT2	≈ -2	$2.2e-101$	0	4	5.0000	1.8037
	TrG-R2	≈ -2	$8.2e-103$	0	4	5.0000	1.7785
	OsG-LD1	≈ -2	$1.5e-172$	0	4	6.0000	2.5743
	OsG-LD2	≈ -2	$1.5e-172$	0	4	6.0000	3.3033
	OsG-LT2	≈ -2	$1.2e-188$	0	4	6.0000	2.5429
	OsG-R2	≈ -2	$1.6e-176$	0	4	6.0000	2.7048
	DCG-LD1	≈ -2	$1.7e-243$	0	4	7.0000	2.1811
	DCG-LD2	≈ -2	$5.2e-244$	0	4	7.0000	2.8524
	DCG-LT2	≈ -2	$4.6e-245$	0	4	7.0000	2.7699
	DCG-R2	≈ -2	$5.2e-244$	0	4	7.0000	2.7903
	NC	≈ -2	$1.1e-239$	0	9	2.0000	1.5642
	OS	≈ -2	$3.1e-188$	0	5	4.0000	1.7762
	DC	≈ -2	$3.8e-168$	0	5	4.0000	1.9822
	R1	≈ -2	0	0	4	5.0060	1.9358
RR1	≈ -2	0	0	4	5.9974	2.3122	
LW8	≈ -2	$1.5e-179$	0	3	7.9840	1.7753	

MÉTODOS ITERATIVOS ÓPTIMOS PARA ECUACIONES NO LINEALES

4.1. Introducción

Como consecuencia de la búsqueda de variantes del método de Newton se han desarrollado como hemos visto en el capítulo anterior, numerosos métodos multipaso. Desde el punto de vista de la eficiencia computacional, se deberían diseñar métodos óptimos en el sentido de la conjetura de Kung-Traub, como son los métodos de Ostrowski y Traub. Así, en los últimos años, se han publicado algunos métodos óptimos de orden 4 y 8. En [83], basándose en el método de Newton, Wang et al. obtienen una versión modificada del método de Ostrowski con una convergencia de octavo orden que requiere tres evaluaciones de la función y una evaluación de su primera derivada, lo que implica que su eficiencia es óptima de acuerdo con la conjetura de Kung y Traub. En [63] Kou et al., presentan variantes del método Ostrowski con orden de convergencia ocho y que requieren para cada iteración tres evaluaciones de la función y una evaluación de su primera derivada y por lo tanto son óptimos también.

King en [55] desarrolló una familia uniparamétrica de métodos óptimos de orden cuatro, cuya fórmula iterativa es

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\ x_{k+1} &= y_k - \frac{f(x_k) + \beta f(y_k)}{f(x_k) + (\beta - 2)f(y_k)} \frac{f(y_k)}{f'(x_k)}, \end{aligned} \quad (4.1)$$

donde $\beta \in \mathbb{R}$ es un parámetro. En particular, el método Ostrowski (véase [72]) es el miembro de esta familia que se obtiene cuando $\beta = 0$.

Recientemente, sobre la base del método de Ostrowski o de los métodos de King, se han propuesto algunos métodos multipaso de orden superior para resolver ecuaciones no lineales. Sharma et al. en [78] y Kou et al. en [48] utilizan la interpolación de Hermite para generar métodos óptimos de orden ocho. Por otra parte, Bi et al. en [7] también presentan una nueva familia de métodos de octavo orden con base los métodos de King y la familia de los métodos de la iteración de sexto orden desarrollada por Chun et al. en [13].

En este capítulo se obtienen nuevos métodos iterativos entre los cuales aparecen algunos óptimos según la conjetura de Kung-Traub con orden de convergencia cuatro y ocho.

4.2. Métodos iterativos multipaso óptimos

4.2.1. Familia de métodos de orden cuatro óptimos con extensiones a orden mayor

El índice de eficiencia (véase Capítulo 2, sección 2.1.1) del método clásico de Newton es 1.414. Todas las modificaciones hechas para mejorar el orden local de la convergencia de este método van en la dirección de incrementar sus índices de eficiencia. Por ejemplo, Frontini y Sormani en [37] desarrollaron nuevas modificaciones del método de Newton para diseñar métodos iterativos con orden de convergencia tres e índice de eficiencia 1.442. Con el mismo índice de eficiencia, Traub en [81] desarrolló un método de tercer orden que exige la evaluación de una función y dos derivadas primeras por iteración. Chen en [10] describió algunas nuevas fórmulas iterativas de orden de convergencia tres. En [43], Gutierrez et al. presentan una familia de métodos de tercer orden que requiere la evaluación de una función, una primera derivada y una derivada segunda por iteración. La expresión iterativa es:

$$x_{k+1} = x_k - \left(1 + \frac{1}{2} \frac{L_f(x_k)}{1 - \beta L_f(x_k)}\right) \frac{f(x_k)}{f'(x_k)}, \quad \text{donde } L_f(x_k) = \frac{f''(x_k)f(x_k)}{f'(x_k)^2}. \quad (4.2)$$

Fácilmente se ve que el método de Chebyshev ($\beta = 0$), el método de Halley ($\beta = \frac{1}{2}$) y el método Super-Halley ($\beta = 1$) son casos especiales de esta familia. Aunque sus índices de eficiencia son 1.442, implican la evaluación de una derivada de segundo orden que puede ser computacionalmente difícil de obtener. Ostrowski desarrolló dos métodos de orden tres y cuatro respectivamente y cada uno de ellos requiere la evaluación de dos funciones y una derivada por iteración. Recientemente, también han aparecido como extensiones de los métodos mencionados anteriormente una serie de métodos de sexto orden para resolver $f(x) = 0$: Neta desarrolló en [67] una familia de métodos de sexto orden que requiere las evaluaciones de tres funciones y una primera derivada por iteración; Sharma y Guha en [77] desarrollaron una familia uniparamétrica de métodos de sexto orden de cuatro pasos basado en el método de Ostrowski y requiere tres evaluaciones de la función no lineal y una evaluación de la derivada, por iteración. Chun en [12] presentó una familia uniparamétrica del métodos de variantes del método de Jarratt, mostrando que el orden de convergencia de la familia aumenta de cuatro hasta seis, aunque añade una evaluación de la función por iteración. Kou et al. en [59], [61] y [62], presentaron nuevas variantes de los métodos de Chebyshev-Halley. Estos métodos que tienen sexto orden de convergencia, añaden una evaluación de la función por iteración respecto al método original.

El objetivo de esta sección es desarrollar algunos métodos de orden 4 óptimos y extenderlos a órdenes superiores intentando mantener su carácter óptimo.

En las páginas 197-204 del texto de Traub, [81], son analizadas una clase de fórmulas iterativas del tipo:

$$x_{k+1} = x_k - a_1 w_1(x_k) - a_2 w_2(x_k) - a_3 w_3(x_k), \quad (4.3)$$

donde $w_1(x_k) = \frac{f(x_k)}{f'(x_k)}$, $w_2(x_k) = \frac{f(x_k)}{f'(y_k)}$, $w_3(x_k) = \frac{f(x_k)}{f'(z_k)}$, $y_k = x_k + \alpha w_1(x_k)$ y $z_k = x_k + \gamma w_1(x_k) + \delta w_2(x_k)$ con a_1 , a_2 , a_3 , α , γ y δ parámetros. Si $a_3 = 0$, algunos valores de los parámetros proporcionan métodos iterativos de orden tres que requieren tres evaluaciones funcionales por iteración. Si $a_3 \neq 0$ otros valores de los parámetros generan métodos de orden cuatro. En ningún caso se obtienen métodos óptimos. En [81], Jarratt propone una modificación de la propuesta de Traub que le permite obtener familias de métodos iterativos de orden cuatro pero con tres evaluaciones funcionales, lo que los convierte en métodos iterativos óptimos. La propuesta de Jarratt se presenta en la siguiente forma:

$$x_{k+1} = x_k - \phi_1(x_k) - \phi_2(x_k),$$

donde $\phi_1(x_k) = a_1 w_1(x_k) + a_2 w_2(x_k)$, $\phi_2(x_k) = \frac{f(x_k)}{b_1 f'(x_k) + b_2 f'(y_k)}$, $w_1(x_k) = \frac{f(x_k)}{f'(x_k)}$, $w_2(x_k) = \frac{f(x_k)}{f'(y_k)}$, e $y_k = x_k + \alpha w_1(x_k)$. Siguiendo esta propuesta introducimos el siguiente método iterativo multipaso en el que para disminuir las evaluaciones funcionales utilizamos combinaciones lineales de los valores de la función y aproximaciones de la derivada de la función a través de sus valores en los dos primeros puntos.

Teorema 4.2.1 Sea $f : I \subseteq \mathbb{R} \rightarrow \mathbb{R}$ suficientemente diferenciable en cada punto del intervalo abierto I tal que $\xi \in I$ es solución de la ecuación no lineal $f(x) = 0$. Supongamos que $f'(x)$ es diferente de cero en ξ . Entonces, la sucesión $\{x_k\}_{k \geq 0}$ obtenida usando las expresiones

$$\begin{aligned} y_k &= x_k + \alpha w_1(x_k), \\ z_k &= y_k + \beta w_1(x_k), \\ x_{k+1} &= z_k + w_2(x_k), \end{aligned} \tag{4.4}$$

donde $w_1(x_k) = f(x_k)/f'(x_k)$, $w_2(x_k) = f(x_k)/[b_1 f'(x_k) + b_2 f'(y_k)]$, converge a ξ con convergencia de orden cuatro si $\alpha = -\frac{2}{3}$, $\beta = \frac{1}{6}$, $b_1 = 1$ y $b_2 = -3$.

Demostración: Sea ξ un cero simple de f . Como f es una función suficientemente diferenciable, mediante la expansión de Taylor de $f(x_k)$ y $f'(x_k)$ alrededor de ξ y teniendo en cuenta que $f(\xi) = 0$, obtenemos:

$$\begin{aligned} f(x_k) &= f'(\xi) [e_k + c_2 e_k^2 + c_3 e_k^3 + c_4 e_k^4 + c_5 e_k^5 + c_6 e_k^6 + c_7 e_k^7 + c_8 e_k^8] + O[e_k^9], \\ f'(x_k) &= f'(\xi) [1 + 2c_2 e_k + 3c_3 e_k^2 + 4c_4 e_k^3 + 5c_5 e_k^4 + 6c_6 e_k^5 + 7c_7 e_k^6 + 8c_8 e_k^7] + O[e_k^8], \end{aligned}$$

donde $c_j = \frac{1}{j!} \frac{f^{(j)}(\xi)}{f'(\xi)}$ con $j = 2, 3, \dots$. Como los términos entre corchetes son polinomios en términos de e_k , la división directa nos da:

$$w_1 = \frac{f(x_k)}{f'(x_k)} = e_k - c_2 e_k^2 + (2c_2^2 - 2c_3) e_k^3 + (7c_2 c_3 - 4c_2^3 - 3c_4) e_k^4 + O[e_k^5].$$

Por lo tanto, la expresión para $y_k = x_k + \alpha w_1(x_k)$ es:

$$y_k = \xi + (1 + \alpha) e_k - \alpha c_2 e_k^2 + 2\alpha(c_2^2 - c_3) e_k^3 + \alpha(4c_2^3 - 3c_4 + 7c_2 c_3) e_k^4 + O[e_k^5]$$

y la expresión para $z_k = y_k + \beta w_1(x_k)$ es:

$$z_k = \xi + (1 + \alpha + \beta) e_k - (\alpha + \beta) c_2 e_k^2 + 2(\alpha + \beta)(c_3 - c_2^2) e_k^3 + (\alpha + \beta)(4c_2^3 - 3c_4 + 7c_2 c_3) e_k^4 + O[e_k^5].$$

De nuevo, mediante la expansión de Taylor obtenemos $f'(y_k)$:

$$\begin{aligned} f'(y_k) &= f'(\xi) [1 + 2(1 + \alpha) c_2 e_k + (-2\alpha c_2^2 + 3(1 + \alpha)^2) c_3 e_k^2 + (4\alpha c_2^3 - 2(5 + 3\alpha)\alpha c_2 c_3 - 4(-1 - \alpha)^3 c_4) e_k^3] \\ &+ f'(\xi) [-8\alpha c_2^4 + \alpha(26 + 15\alpha) c_2^2 c_3 - 6\alpha(3 + 4\alpha + 2\alpha^2) c_2 c_4 + (-1 - \alpha)(12\alpha c_3^2 + 5(-1 - \alpha)^3 c_5)] e_k^4 \\ &+ O[e_k^5]. \end{aligned}$$

Determinamos $w_2(x_k)$:

$$\begin{aligned} w_2(x_k) &= \frac{f(x_k)}{b_1 f'(x_k) + b_2 f'(y_k)} = D_1 e_k + D_2 e_k^2 + D_3 e_k^3 + D_4 e_k^4 + O[e_k^5], \\ D_1 &= \frac{1}{b_1 + b_2}, \\ D_2 &= -\frac{(b_1 + b_2 + 2b_2 \alpha) c_2}{(b_1 + b_2)^2}, \\ D_3 &= \frac{2(b_1^2 + 2b_1(b_2 + 2b_2 \alpha) + b_2^2(1 + 4\alpha + 2\alpha^2)) c_2^2 - (2b_1^2 + b_2^2(2 + 6\alpha + 3\alpha^2) + b_1 b_2(4 + 6\alpha + 3\alpha^2)) c_3}{(b_1 + b_2)^3}, \\ D_4 &= \frac{(-4b_1^3 - 2b_1^2 b_2(6 + 13\alpha) - 4b_1 b_2^2(3 + 13\alpha + 7\alpha^2) - 2b_2^3(2 + 13\alpha + 14\alpha^2 + 4\alpha^3)) c_2^3}{(b_1 + b_2)^4} \\ &+ \frac{(7b_1^3 + b_1^2 b_2(21 + 38\alpha + 15\alpha^2) + b_2^3(7 + 38\alpha + 39\alpha^2 + 12\alpha^3) + b_1 b_2^2(21 + 76\alpha + 54\alpha^2 + 12\alpha^3)) c_2 c_3}{(b_1 + b_2)^4} \\ &- \frac{(b_1 + b_2)^2 (3b_1 + b_2(3 + 12\alpha + 12\alpha^2 + 4\alpha^3)) c_4}{(b_1 + b_2)^4}. \end{aligned}$$

Calculamos finalmente $z_k + w_2(x_k)$:

$$x_{k+1} = \xi + A_1 e_k + A_2 e_k^2 + A_3 e_k^3 + A_4 e_k^4 + O[e_k^5],$$

donde

$$\begin{aligned} A_1 &= 1 + \frac{1}{b_1 + b_2} + \alpha + \beta, \\ A_2 &= - \left[\frac{(b_1 + b_2 + 2b_2\alpha)}{(b_1 + b_2)^2} + \alpha + \beta \right] c_2, \\ A_3 &= 2 \left[\alpha + \beta + \frac{b_1^2 + 2b_1(b_2 + 2b_2\alpha) + b_2^2(1 + 4\alpha + 2\alpha^2)}{(b_1 + b_2)^3} \right] c_2^2 \\ &\quad - \left[2(\alpha + \beta) + \frac{2b_1^2 + b_2^2(2 + 6\alpha + 3\alpha^2) + b_1 b_2(4 + 6\alpha + 3\alpha^2)}{(b_1 + b_2)^3} \right] c_3, \\ A_4 &= -2 \left[2(\alpha + \beta) + \frac{2b_1^3 + b_1^2 b_2(6 + 13\alpha) + 2b_1 b_2^2(3 + 13\alpha + 7\alpha^2) + b_2^3(2 + 13\alpha + 14\alpha^2 + 4\alpha^3)}{(b_1 + b_2)^4} \right] c_2^3 \\ &\quad + \left[7(\alpha + \beta) + \frac{7b_1^3 + b_1^2 b_2(21 + 38\alpha + 15\alpha^2) + b_2^3(7 + 38\alpha + 39\alpha^2 + 12\alpha^3)}{(b_1 + b_2)^4} \right] c_2 c_3 \\ &\quad + \left[\frac{b_1 b_2^2(21 + 76\alpha + 54\alpha^2 + 12\alpha^3)}{(b_1 + b_2)^4} \right] c_2 c_3 - \left[3(\alpha + \beta) - \frac{(3b_1 + b_2(3 + 12\alpha + 12\alpha^2 + 4\alpha^3))}{(b_1 + b_2)^2} \right] c_4. \end{aligned}$$

Para obtener orden de convergencia cuatro se debe resolver el sistema de ecuaciones:

$$\begin{aligned} 1 + \frac{1}{b_1 + b_2} + \alpha + \beta &= 0 \\ \frac{(b_1 + b_2 + 2b_2\alpha)}{(b_1 + b_2)^2} + \alpha + \beta &= 0 \\ \alpha + \beta + \frac{b_1^2 + 2b_1(b_2 + 2b_2\alpha) + b_2^2(1 + 4\alpha + 2\alpha^2)}{(b_1 + b_2)^3} &= 0 \\ 2(\alpha + \beta) + \frac{2b_1^2 + b_2^2(2 + 6\alpha + 3\alpha^2) + b_1 b_2(4 + 6\alpha + 3\alpha^2)}{(b_1 + b_2)^3} &= 0 \end{aligned}$$

La solución del sistema proporciona los siguientes valores de los cuatro parámetros introducidos: $\alpha = -\frac{2}{3}$, $\beta = \frac{1}{6}$, $b_1 = 1$ y $b_2 = -3$ que garantizan convergencia de orden cuatro. La expresión de la ecuación del error queda:

$$e_{k+1} = \left(c_2^3 - c_2 c_3 + \frac{c_4}{9} \right) e_k^4 + O[e_k^5].$$

Con esto queda demostrado que para los valores dados de los parámetros α , β , b_1 , b_2 , se obtiene orden de convergencia 4 con tres evaluaciones funcionales: dos derivadas y una función. En este caso el método desarrollado es óptimo según la conjetura de Kung-Traub. La fórmula iterativa del método óptimo de orden 4 es:

$$\begin{aligned} y_k &= x_k - \frac{2}{3} \frac{f(x_k)}{f'(x_k)}, \\ z_k &= y_k + \frac{1}{6} \frac{f(x_k)}{f'(x_k)}, \\ x_{k+1} &= z_k + \frac{f(x_k)}{f'(x_k) - 3f'(y_k)}. \end{aligned} \tag{4.5}$$

□

Siguiendo la técnica desarrollada por Jarratt y Traub intentamos extender el método descrito en el teorema anterior con el objetivo de alcanzar un método de orden ocho. En el siguiente resultado observamos que, a pesar de que el objetivo se ha alcanzado, el orden del método resultante no es óptimo.

Teorema 4.2.2 *Bajo los mismos supuestos del teorema anterior, consideremos el esquema iterativo*

$$\begin{aligned}
y_k &= x_k - \frac{2}{3} \frac{f(x_k)}{f'(x_k)}, \\
z_k &= y_k + \frac{1}{6} \frac{f(x_k)}{f'(x_k)}, \\
u_k &= z_k + \frac{f(x_k)}{f'(x_k) - 3f'(y_k)}, \\
v_k &= z_k + w_3(x_k), \\
x_{k+1} &= z_k + w_3(x_k) + w_4(x_k).
\end{aligned} \tag{4.6}$$

donde

$$\begin{aligned}
w_3(x_k) &= \frac{a_1 f(x_k) + a_2 f(u_k)}{a_3 f'(x_k) + a_4 f'(y_k)}, \\
w_4(x_k) &= \frac{(r_1 f(x_k) + r_2 f(u_k) + r_3 f(v_k))(r_4 f'(x_k) + r_5 f'(y_k))}{r_6 f'(x_k)^2 + r_7 f'(x_k) f'(y_k) + r_8 f'(y_k)^2}.
\end{aligned}$$

La sucesión $\{v_k\}_{k \geq 0}$ converge a ξ con convergencia de orden seis para valores de los parámetros $a_1 = a_2/2$, $a_3 = a_2/2$, $a_4 = -3a_2/2$. Además, si $r_1 = r_2 = 0$, $r_4 = -[r_3 r_5 - r_6 r_7 - r_8]/r_3$, $r_3 = [-3r_6 - r_7 + r_8]/2r_5$ y cualquier valor de r_6 , r_7 y r_8 , la sucesión $\{x_k\}_{k \geq 0}$ converge a ξ con convergencia de orden 8.

Demostración: Como la primera parte de la demostración es la prueba del Teorema anterior, con valores de los parámetros conocidos continuamos con la demostración introduciendo los siguientes pasos. Expandimos $f(u_k)$ alrededor de ξ y obtenemos:

$$\begin{aligned}
f(u_k) &= f'(\xi) \left[c_2^3 - c_2 c_3 + \frac{c_4}{9} \right] e_k^4 + f'(\xi) \left[-4c_2^4 + 8c_2^2 c_3 - 2c_3^2 - \frac{20c_2 c_4}{9} + \frac{8c_5}{27} \right] e_k^5 \\
&+ \frac{2}{27} f'(\xi) [135c_2^5 - 405c_2^3 c_3 + 165c_2^2 c_4 - 99c_3 c_4 + 9c_2 (27c_3^2 - 5c_5) + 7c_6] e_k^6 \\
&+ O[e_k^7].
\end{aligned}$$

y buscamos $w_3(x_k) = \frac{a_1 f(x_k) + a_2 f'(u_k)}{a_3 f(x_k) + a_4 f'(z_k)}$ y $v_k = z_k + w_3(x_k)$ para la demostración de convergencia de orden 6. Por lo tanto:

$$v_k = \xi + E_1 e_k + E_2 e_k^2 + E_3 e_k^3 + E_4 e_k^4 + E_5 e_k^5 + E_6 e_k^6 + O[e_k^7],$$

donde

$$\begin{aligned}
E_1 &= \frac{1}{2} + \frac{a_1}{a_3 + a_4}, \\
E_2 &= \frac{1}{6} \left[3 + \frac{2a_1(-3a_3 + a_4)}{(a_3 + a_4)^2} \right] c_2, \\
E_3 &= \left[-1 + \frac{2a_1(9a_3^2 - 6a_3 a_4 - 7a_4^2)}{9(a_3 + a_4)^3} \right] c_2^2 + 3 \left[\frac{-3a_3^2 - 2a_3 a_4 + a_4^2}{9(a_3 + a_4)^3} \right] c_3, \\
E_4 &= \left[2 + \frac{27a_2(a_3 + a_4)^3 + 4a_1(-27a_3^3 + 36a_3^2 a_4 + 69a_3 a_4^2 + 22a_4^3)}{27(a_3 + a_4)^4} \right] c_2^3 \\
&- \left[\frac{7}{2} + \frac{-9a_2(a_3 + a_4)^2 + a_1(63a_3^2 - 42a_3 a_4 - 41a_4^2)}{9(a_3 + a_4)^3} \right] c_2 c_3 \\
&+ \left[\frac{3}{2} - \frac{a_1(81a_3 - 23a_4) - 3a_2(1(a_3 + a_4))}{27(a_3 + a_4)^2} \right] c_4.
\end{aligned}$$

Para tener orden de convergencia 6 se debe resolver el sistema de ecuaciones:

$$\begin{aligned}
\frac{1}{2} + \frac{a_1}{a_3 + a_4} &= 0 \\
3 + \frac{2a_1(-3a_3 + a_4)}{(a_3 + a_4)^2} &= 0 \\
-1 + \frac{2a_1(9a_3^2 - 6a_3a_4 - 7a_4^2)}{9(a_3 + a_4)^3} &= 0 \\
1 + \frac{2a_1(-3a_3^2 - 2a_3a_4 + a_4^2)}{3(a_3 + a_4)^3} &= 0 \\
2 + \frac{27a_2(a_3 + a_4)^3 + 4a_1(-27a_3^3 + 36a_3^2a_4 + 69a_3a_4^2 + 22a_4^3)}{27(a_3 + a_4)^4} &= 0 \\
\frac{7}{2} + \frac{-9a_2(a_3 + a_4)^2 + a_1(63a_3^2 - 42a_3a_4 - 41a_4^2)}{9(a_3 + a_4)^3} &= 0 \\
\frac{3}{2} - \frac{a_1(81a_3 - 23a_4) - 3a_2(a_3 + a_4)}{27(a_3 + a_4)^2} &= 0
\end{aligned}$$

Resolviendo el sistema obtenemos: $a_1 = a_3 = -\frac{a_4}{3}$ y $a_2 = -\frac{2a_4}{3}$. Por lo tanto:

$$e_{v_k} = v_k - \xi = \frac{1}{9} (2c_2^2 - c_3) (9c_2^3 - 9c_2c_3 + c_4) e_k^6 + O[e_k^7].$$

Hasta este paso el método tiene orden de convergencia 6 con cuatro evaluaciones funcionales: dos funciones y dos derivadas. Esto implica que no es óptimo según la conjetura de Kung-Traub. La fórmula iterativa queda:

$$\begin{aligned}
y_k &= x_k - \frac{2}{3} \frac{f(x_k)}{f'(x_k)}, \\
z_k &= y_k + \frac{1}{6} \frac{f(x_k)}{f'(x_k)}, \\
u_k &= z_k + \frac{f(x_k)}{f'(x_k) - 3f'(y_k)}, \\
v_k &= z_k + \frac{f(x_k) + 2f(u_k)}{f'(x_k) - 3f'(z_k)}.
\end{aligned} \tag{4.7}$$

Por último buscamos w_4 para obtener orden ocho. Expandimos de nuevo esta vez $f(v_k)$ alrededor de ξ . Entonces obtenemos para $f(v_k)$:

$$\begin{aligned}
f(v_k) &= f'(\xi) \left[\frac{1}{9} (2c_2^2 - c_3) (9c_2^3 - 9c_2c_3 + c_4) e_k^6 \right] \\
&+ f'(\xi) \left[-\frac{2}{81} (486c_2^6 - 121c_2^4c_3 - 81c_3^3 + 270c_2^3c_4 - 189c_2c_3c_4 + 8c_4^2 + 3c_2^2(243c_3^2 - 8c_5) + 12c_3c_5) e_k^7 \right] \\
&+ f'(\xi) \left[39c_2^7 - 140c_2^5c_3 + 48c_2^4c_4 + c_2^3 \left(136c_3^2 - \frac{277c_5}{27} \right) + c_2 \left(-33c_3^3 + \frac{385c_4^2}{81} + \frac{203c_3c_5}{27} \right) \right] e_k^8 \\
&- f'(\xi) \left[\frac{2c_2^2}{27} (849c_3c_4 - 14c_6) + \frac{1}{243} (2727c_2^2c_4 - 193c_4c_5 - 126c_3c_6) \right] e_k^8 + O[e_k^9].
\end{aligned}$$

Calculamos w_4 :

$$\begin{aligned}
w_4 &= \frac{(r_4 + r_5)r_1}{r_6 + r_7 + r_8} e_k \\
&+ \frac{(r_5(-7r_6 - 3r_7 + r_8) + r_4(-3r_6 + r_7 + 5r_8))r_1 c_2}{3(r_6 + r_7 + r_8)^2} e_k^2 \\
&+ \frac{2r_1 r_4(9r_4^2 + 6r_6 r_7 + 7r_7^2 - 38r_6 r_8 - 30r_7 r_8 - 15r_8^2)c_2^2}{9(r_6 + r_7 + r_8)^3} e_k^3 \\
&+ \frac{2r_1 r_4(33r_6^2 + 34r_6 r_7 + 9r_7^2 - 6r_6 r_8 - 6r_7 r_8 - 7r_8^2)c_2^2}{9(r_3 + r_4 + r_5)^3} e_k^3 \\
&- \frac{6r_1(r_4(3r_6 - r_7 - 5r_8) + r_5(7r_6 + 3r_7 - r_8))(r_6 + r_7 + r_8)c_3}{9(r_3 + r_4 + r_5)^3} e_k^3 + O[e_k^4].
\end{aligned}$$

Calculamos $x_{k+1} = v_k + w_4(x_k)$ y la expresión de la ecuación de error se presenta en la forma:

$$e_{k+1} = H_1 e_k + H_2 e_k^2 + H_3 e_k^3 + O[e_k^4],$$

donde

$$\begin{aligned}
H_1 &= \frac{(r_4 + r_5)r_1}{r_6 + r_7 + r_8}, \\
H_2 &= \frac{(r_5(-7r_6 - 3r_7 + r_8) + r_4(-3r_6 + r_7 + 5r_8))r_1 c_2}{3(r_6 + r_7 + r_8)^2}, \\
H_3 &= \frac{2r_1 r_4(9r_4^2 + 6r_6 r_7 + 7r_7^2 - 38r_6 r_8 - 30r_7 r_8 - 15r_8^2)c_2^2}{9(r_6 + r_7 + r_8)^3} \\
&+ \frac{2r_1 r_4(33r_6^2 + 34r_6 r_7 + 9r_7^2 - 6r_6 r_8 - 6r_7 r_8 - 7r_8^2)c_2^2}{9(r_3 + r_4 + r_5)^3} \\
&- \frac{6r_1(r_4(3r_6 - r_7 - 5r_8) + r_5(7r_6 + 3r_7 - r_8))(r_6 + r_7 + r_8)c_3}{9(r_3 + r_4 + r_5)^3}.
\end{aligned}$$

Notamos que los primeros tres elementos se eliminan si escogemos $r_1 = 0$. Simplificando la expresión de error obtenemos:

$$e_{k+1} = r_2 J_4 e_k^4 + r_2 J_5 e_k^5 + J_6 e_k^6 + O[e_k^7].$$

Si $r_2 = 0$ se obtiene orden de convergencia 6. Simplificando de nuevo, la expresión del error queda:

$$e_{k+1} = \frac{(r_3(r_4 + r_5) + r_6 + r_7 + r_8)(2c_2^2 - c_3)(9c_2^3 - 9c_2 c_3 + c_4)}{9(r_6 + r_7 + r_8)} e_k^6 + O[e_k^7].$$

Para obtener orden de convergencia 7, debemos resolver la ecuación: $\frac{r_3(r_4 + r_5) + r_6 + r_7 + r_8}{9(r_6 + r_7 + r_8)} = 0$. Expresamos r_4 como $r_4 = -\frac{r_3 r_5 + r_6 + r_7 + r_8}{r_3}$ con $r_3 \neq 0$ y lo sustituimos en la ecuación de error. Simplificando obtenemos:

$$e_{k+1} = -\frac{2c_2(2r_3 r_5 - 3r_6 - r_7 + r_8)(2c_2^2 - c_3)(9c_2^3 - 9c_2 c_3 + c_4)}{27(r_6 + r_7 + r_8)} e_k^7 + G e_k^8 + O[e_k^9].$$

Para tener convergencia de orden ocho resolvemos la ecuación $-\frac{2(2r_3 r_5 - 3r_6 - r_7 + r_8)}{27(r_6 + r_7 + r_8)} = 0$ despejando r_3 en función de los demás $r_3 = \frac{3r_6 + r_7 - r_8}{2r_5}$ con $r_5 \neq 0$, que sustituimos en la ecuación de error obteniendo:

$$e_{k+1} = \frac{(2c_2^2 - c_3)[(56r_6 + 30r_7 + 22r_8)c_2^2 - 9(r_6 + r_7 + r_8)c_3](9c_2^3 - 9c_2 c_3 + c_4)e_k^8}{81(r_6 + r_7 + r_8)} + O[e_k^9].$$

Así determinamos la familia de nuevos métodos iterativos con tres parámetros, orden de convergencia ocho y cinco evaluaciones funcionales: dos derivadas y tres funciones. La familia queda determinada por la fórmula iterativa:

$$\begin{aligned}
y_k &= x_k - \frac{2}{3} \frac{f(x_k)}{f'(x_k)}, \\
z_k &= y_k + \frac{1}{6} \frac{f(x_k)}{f'(x_k)}, \\
u_k &= z_k + \frac{f(x_k)}{f'(x_k) - 3f'(y_k)}, \\
v_k &= u_k + \frac{2f(u_k)}{f'(x_k) - 3f'(z_k)}, \\
x_{k+1} &= v_k - \frac{1}{2} \frac{((5r_6 + 3r_7 + r_8)f'(x_k) - (3r_6 + r_7 - r_8)f'(y_k))f(u_k)}{r_6f'(x_k)^2 + r_7f'(x_k)f'(y_k) + r_8f'(y_k)^2}.
\end{aligned} \tag{4.8}$$

□

Un caso particular de la familia (4.8) se obtiene si tomamos $r_7 = r_8 = 0$:

$$\begin{aligned}
y_k &= x_k - \frac{2}{3} \frac{f(x_k)}{f'(x_k)}, \\
z_k &= y_k + \frac{1}{6} \frac{f(x_k)}{f'(x_k)}, \\
u_k &= z_k + \frac{f(x_k)}{f'(x_k) - 3f'(z_k)}, \\
v_k &= u_k + \frac{2f(u_k)}{f'(x_k) - 3f'(z_k)}, \\
x_{k+1} &= v_k - \frac{1}{2} \left[\frac{5f'(x_k) - 3f'(y_k)}{f'(x_k)} \right] \frac{f(v_k)}{f'(x_k)}.
\end{aligned} \tag{4.9}$$

Llamamos a este método iterativo M8A.

4.2.2. Familia de métodos combinación del método de Ostrowski y de Chun

En esta sección se desarrollan familias de métodos iterativos multipaso, variantes del método de Newton y generados a partir de la combinación de los métodos de Chun y Ostrowski con el fin de obtener el orden de convergencia mayor posible y mínimo coste computacional, empleando para ello una técnica diferente a la desarrollada en la sección anterior.

Usando el método de Newton como predictor buscamos una combinación de los métodos de Ostrowski (3.32) y Chun (3.39) en la forma:

$$\begin{aligned}
y_k &= x_k - \alpha \frac{f(x_k)}{f'(x_k)}, \\
x_{k+1} &= y_k - \left[\frac{f(x_k)}{a_1f'(x_k) + a_2f'(y_k)} + \frac{b_1f(x_k) + b_2f(y_k)}{f'(x_k)} \right] \frac{f(y_k)}{f'(x_k)},
\end{aligned} \tag{4.10}$$

donde $\alpha, a_1, a_2, b_1, b_2 \in \mathbb{R}$. Buscamos sus valores de tal manera que el orden de la convergencia sea al menos 4.

Teorema 4.2.3 *Sea $f : I \subseteq \mathbb{R} \rightarrow \mathbb{R}$ suficientemente diferenciable en cada punto del intervalo abierto tal que $\xi \in I$ es una solución de la ecuación no lineal $f(x) = 0$. Supongamos que $f'(x)$ es continua y es diferente de cero en ξ . Entonces, la sucesión $\{x_k\}_{k \geq 0}$ obtenida usando las expresiones (4.10) converge a ξ con convergencia de orden cuatro si $\alpha = 1$, $a_2 = a_1^2(b_2 - 2)$, $b_1 = 1 - \frac{1}{a_1}$ y para todo a_1 y $b_2 \in \mathbb{R}$ con $a_1 \neq 0$.*

Demostración: Sea ξ un cero simple de f . Como f es una función suficientemente diferenciable, mediante la expansión de Taylor de $f(x_k)$ y $f'(x_k)$ alrededor de ξ y teniendo en cuenta que $f(\xi) = 0$, obtenemos:

$$f(x_k) = f'(\xi) [e_k + c_2 e_k^2 + c_3 e_k^3 + c_4 e_k^4] + O[e_k^5], \quad (4.11)$$

$$f'(x_k) = f'(\xi) [1 + 2c_2 e_k + 3c_3 e_k^2 + 4c_4 e_k^3] + O[e_k^4], \quad (4.12)$$

donde $c_j = \frac{1}{j!} \frac{f^{(j)}(\xi)}{f'(\xi)}$, con $j = 2, 3, \dots$. Como los términos entre corchetes son polinomios en términos de e_k , la división directa de (4.11) y (4.12) nos da:

$$\frac{f(x_k)}{f'(x_k)} = e_k - c_2 e_k^2 + 2(c_2^2 - c_3) e_k^3 + (7c_2 c_3 - 4c_2^3 - 3c_4) e_k^4 + O[e_k^5],$$

y por lo tanto, la expresión para $y_k = x_k - \alpha \frac{f(x_k)}{f'(x_k)}$ es:

$$y_k = \xi + (1 - \alpha) c_2 e_k^2 + 2\alpha (c_3 - c_2^2) e_k^3 + \alpha (4c_2^3 + 3c_4 - 7c_2 c_3) e_k^4 + O[e_k^5].$$

De nuevo, expandiendo $f(y_k)$ alrededor de ξ , obtenemos:

$$\begin{aligned} f(y_k) &= f'(\xi) [(1 - \alpha) e_k + (1 - \alpha + \alpha^2) c_2 e_k^2 + (-2\alpha^2 c_2^2 - (-1 + \alpha - 3\alpha^2 + \alpha^3) c_3) e_k^3 \\ &+ (5\alpha^2 c_2^3 + \alpha^2 (-10 + 3\alpha) c_2 c_3 + (1 - \alpha + 6\alpha^2 - 4\alpha^3 + \alpha^4 c_4) e_k^4 + O[e_k^5]]. \end{aligned} \quad (4.13)$$

Ahora calculamos:

$$\begin{aligned} H_1 &= \frac{f(x_k)}{a_1 f(x_k) + a_2 f(y_k)} \quad (4.14) \\ &= \frac{1}{a_1 + a_2 - a_2 \alpha} - \frac{a_2 \alpha^2}{(a_1 + a_2 - a_2 \alpha)^2} c_2 e_k \\ &+ \frac{a_2 \alpha^2 ((3a_1 + a_2 (3 - 3\alpha + \alpha^2)) c_2^2 + (\alpha - 3) (a_1 + a_2 - a_2 \alpha) c_3)}{(a_1 + a_2 - a_2 \alpha)^3} e_k^2 \\ &- \frac{a_2 \alpha^2 (8a_1^2 + a_2^2 (\alpha + 2)^2 (2 - 2\alpha + \alpha^2) + 2a_1 a_2 (8 - 8\alpha + 3\alpha^2))}{(a_1 + a_2 - a_2 \alpha)^4} c_2^3 e_k^3 \\ &- \frac{2(a_1^2 (2\alpha - 7) + a_1 a_2 (\alpha^3 - 7\alpha^2 + 18\alpha - 14) - a_2^2 (7 - 16\alpha + 14\alpha^2 - 6\alpha^3 + \alpha^4))}{(a_1 + a_2 - a_2 \alpha)^4} c_2 c_3 e_k^3 \\ &- \frac{(a_1 + a_2 - a_2 \alpha)^2 (6 - 4\alpha + \alpha^2)}{(a_1 + a_2 - a_2 \alpha)^4} c_4 e_k^3 + O[e_k^4] \end{aligned}$$

y

$$\begin{aligned} H_2 &= \frac{b_1 f(x_k) + b_2 f(y_k)}{f(x_k)} \quad (4.15) \\ &= (b_1 + b_2 - b_2 \alpha) + b_2 \alpha^2 c_2 e_k + b_2 \alpha^2 (3c_2^2 + (\alpha - 3) c_3) e_k^2 \\ &+ b_2 \alpha^2 (8c_2^3 + 2(2\alpha - 7) c_2 c_3 + (6 - 4\alpha + \alpha^2) c_4) e_k^3 + O[e_k^4]. \end{aligned}$$

Sustituimos (4.13) y (4.12) en el cociente:

$$\begin{aligned} \frac{f(y_k)}{f'(x_k)} &= (1 - \alpha) e_k - (1 - \alpha - \alpha^2) c_2 e_k^2 - (1 + \alpha) ((4\alpha - 2) c_2^2 + (2 - 4\alpha + \alpha^2) c_3) e_k^3 \\ &+ ((-4 + 4\alpha + 13\alpha^2) c_2^3 + (7 - 7\alpha - 19\alpha^2 + 5\alpha^3) c_2 c_3) e_k^4 \\ &+ (-3 + 3\alpha + 6\alpha^2 - 4\alpha^3 + \alpha^4) c_4 e_k^4 + O[e_k^5] \end{aligned}$$

y sumamos (4.14) y (4.15)

$$\begin{aligned}
H_1 + H_2 &= \left(b_1 + b_2 - b_2\alpha + \frac{1}{a_1 + a_2 - a_2\alpha} \right) + \alpha^2 \left[b^2 - \frac{a_2}{(a_1 + a_2 - a_2\alpha)^2} \right] c_2 e_k \\
&- \alpha^2 b_2 (3c_2^2 + (\alpha - 3)c_3) e_k^2 \\
&+ \frac{\alpha^2 (3a_1 a_2 + a_2^2 (3 - 3\alpha + \alpha^2)) c_2^2 + \alpha^2 a_2 (\alpha - 3) (a_1 + a_2 - a_2\alpha) c_3}{(a_1 + a_2 - a_2\alpha)^3} c_3 e_k^2 \\
&+ \alpha^2 (b_2 (8c_2^3 + 2(2\alpha - 7)c_2 c_3 + (6 - 4\alpha + \alpha^2)c_4)) e_k^3 \\
&- \frac{\alpha^2 a_2 (8a_1^2 + a_2^2 (\alpha - 2)^2 (2 - 2\alpha + \alpha^2) + 2a_1 a_2 (8 - 8\alpha + 3\alpha^2))}{(a_1 + a_2 - a_2\alpha)^4} c_2^3 e_k^3 \\
&+ \frac{2\alpha^2 a_2 (a_1^2 (2\alpha - 7) - a_1 a_2 (14 - 18\alpha + 7\alpha^2 - \alpha_3))}{(a_1 + a_2 - a_2\alpha)^4} c_2 c_3 e_k^3 \\
&- \frac{2\alpha^2 a_2^3 (7 - 16\alpha + 14\alpha^2 - 6\alpha^3 + \alpha^4)}{(a_1 + a_2 - a_2\alpha)^4} c_2 c_3 e_k^3 \\
&+ \frac{2\alpha^2 a_2 (6 - 4\alpha + \alpha^2)}{(a_1 + a_2 - a_2\alpha)^2} c_4 e_k^3 + O[e_k^5].
\end{aligned}$$

Por lo tanto, la ecuación del error queda:

$$\begin{aligned}
e_{k+1} &= (\alpha - 1) \left(-1 + b_1 + b_2 - b_2\alpha + \frac{1}{a_1 + a_2 - a_2\alpha} \right) e_k \\
&+ \left(\alpha - (1 - \alpha) \alpha^2 \left(b_2 - \frac{a_2}{(a_1 + a_2 - a_2\alpha)^2} \right) \right) c_2 e_k^2 \\
&+ (1 - \alpha - \alpha^2) \left(b_1 + b_2 - b_2\alpha + \frac{1}{a_1 + a_2 - a_2\alpha} \right) c_2 e_k^2 + O[e_k^3]. \tag{4.16}
\end{aligned}$$

Notamos que para que el orden de convergencia sea al menos 2 el valor de α debe ser la unidad. Sustituimos en (4.16), y obtenemos para la ecuación de error:

$$\begin{aligned}
e_{k+1} &= \left(1 - \frac{1}{a_1} - b_1 \right) c_2 e_k^2 \\
&+ \frac{((4a_1 + a_2 + a_1^2(-2 + 4b_1 - b_2))c_2^2 + 2a_1(-1 + a_1 - a_1 b_1)c_3)}{a_1^2} e_k^3 \\
&+ \frac{1}{a_1^3} (-(13a_1^2 + 7a_1 a_2 + a_2^2 + a_1^3(-4 + 13b_1 - 7b_2))c_2^3) e_k^4 \\
&+ \frac{1}{a_1^3} (a_1(14a_1 + 4a_2 + a_1^2(-7 + 14b_1 - 4b_2))c_2 c_3 + 3a_1^2(-1 + a_1 - a_1 b_1)c_4) e_k^4 + O[e_k^5].
\end{aligned}$$

Para que el orden de convergencia sea cuatro debemos resolver el sistema de ecuaciones:

$$\begin{aligned}
1 - \frac{1}{a_1} - b_1 &= 0, \\
\frac{4a_1 + a_2 + a_1^2(-2 + 4b_1 - b_2)}{a_1^2} &= 0, \\
\frac{-1 + a_1 - a_1 b_1}{a_1} &= 0.
\end{aligned}$$

Notamos que la primera y la tercera ecuaciones son iguales y su solución dependerá de dos de los parámetros:

$$\begin{aligned}
a_2 &= a_1^2(b_2 - 2), \\
b_1 &= \frac{a_1 - 1}{a_1},
\end{aligned}$$

por lo tanto, recalculando el error, obtenemos:

$$e_{k+1} = c_2 [(5 - a_1(-2 + b_2)^2 c_2^2 - c_3)] e_k^4 + O[e_k^5],$$

de donde concluimos que el máximo orden de convergencia para la familia biparamétrica es cuatro para todo a_1 y $b_2 \in \mathbb{R}$ con $a_1 \neq 0$. \square

Finalmente la fórmula iterativa obtenida del método Newton-Ostrowski-Chun (NOC) es:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\ x_{k+1} &= y_k - \frac{1}{a_1} \left[\frac{f(x_k)}{f(x_k) + a_1(b_2 - 2)f(y_k)} + \frac{(a_1 - 1)f(x_k) + a_1 b_2 f(y_k)}{f(x_k)} \right] \frac{f(y_k)}{f'(x_k)}, \end{aligned}$$

que define una familia biparamétrica de métodos óptimos de orden cuatro.

En caso de que $a_1 = 1$ la fórmula iterativa adopta la forma:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)} \\ x_{k+1} &= y_k - \left[\frac{f(x_k)}{f(x_k) + (b_2 - 2)f(y_k)} + \frac{b_2 f(y_k)}{f(x_k)} \right] \frac{f(y_k)}{f'(x_k)} \end{aligned}$$

con casos particulares: si $b_2 = 2$ tenemos el método de Chun (3.39) y si $b_2 = 0$ tenemos el método de Ostrowski (3.32).

A continuación nos preguntamos: ¿Podríamos obtener con uno o dos pasos más usando combinaciones de métodos, otras familias con mayor orden de convergencia? La respuesta la proporciona el siguiente teorema, en el que analizamos la convergencia de un método multipaso que parte del método de Ostrowski (orden 4) y alcanza orden 7 combinando Ostrowski consigo mismo.

Teorema 4.2.4 *Sea $f : I \subseteq \mathbb{R} \rightarrow \mathbb{R}$ suficientemente diferenciable en cada punto del intervalo abierto I y $\xi \in I$ una solución de la ecuación no lineal $f(x) = 0$. Supongamos que $f'(x)$ es continua y es diferente de cero en ξ . Entonces, la sucesión $\{x_k\}_{k \geq 0}$ obtenida usando el algoritmo:*

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\ z_k &= x_k - \left[1 + \frac{f(y_k)}{f(x_k) - 2f(y_k)} \right] \frac{f(x_k)}{f'(x_k)}, \\ x_{k+1} &= z_k - \left[m_1 + m_2 \frac{f(y_k)}{f(x_k) - 2f(y_k)} + m_3 \frac{f(z_k)}{a_1 f(y_k) + a_2 f(z_k)} \right]^n \frac{f(z_k)}{f'(x_k)}, \end{aligned} \tag{4.17}$$

converge a ξ con convergencia de orden siete si $m_1 = m_2 = 1$, $m_3 = \frac{1}{2}$ y $n = 2$ para todo $a_1, a_2 \in \mathbb{R}$ y $a_1 \neq 0$.

Demostración: Sea ξ un cero simple de f . Como f es una función suficientemente diferenciable, mediante la expansión de Taylor de $f(x_k)$ y $f'(x_k)$ alrededor de ξ , teniendo en cuenta que $f(\xi) = 0$, denotando por $c_j = \frac{1}{j!} \frac{f^{(j)}(\xi)}{f'(\xi)}$ con $j = 2, 3, \dots$ obtenemos:

$$\begin{aligned} f(x_k) &= f'(\xi) [e_k + c_2 e_k^2 + c_3 e_k^3 + c_4 e_k^4 + c_5 e_k^5 + c_6 e_k^6 + c_7 e_k^7] + O[e_k^8], \\ f'(x_k) &= f'(\xi) [1 + 2c_2 e_k + 3c_3 e_k^2 + 4c_4 e_k^3 + 5c_5 e_k^4 + 6c_6 e_k^5 + 7c_7 e_k^6] + O[e_k^7], \end{aligned}$$

y combinando ambos desarrollos:

$$\begin{aligned} f(x_k)/f'(x_k) &= e_k - c_2 e_k^2 + 2(c_2^2 - c_3) e_k^3 + (7c_2 c_3 - 4c_3^2 - 3c_4) e_k^4 \\ &+ (8c_2^4 - 20c_2^2 c_3 + 6c_3^2 + 10c_2 c_4 - 4c_5) e_k^5 \\ &+ (-16c_2^5 + 52c_2^3 c_3 - 28c_2^2 c_4 + 17c_3 c_4 + c_2(13c_5 - 33c_3^2) - 5c_6) e_k^6 \\ &+ 2(16c_2^6 - 64c_2^4 c_3 - 9c_3^3 + 36c_2^3 c_4 + 6c_4^2 + 9c_2^2(7c_3^2 - 2c_5) - 11c_3 c_5 + c_2(8c_6 - 46c_3 c_4) - 3c_7) e_k^7 \\ &+ O[e_k^8]. \end{aligned}$$

Así, el polinomio de Taylor asociado al primer paso del método es:

$$\begin{aligned}
y_k &= \xi + c_2 e_k^2 + (2c_3 - 2c_2^2) e_k^3 \\
&+ (4c_2^3 - 7c_2 c_3 + 3c_4) e_k^4 \\
&+ (20c_2^2 c_3 - 8c_2^4 - 6c_3^2 - 10c_2 c_4 + 4c_5) e_k^5 \\
&+ (16c_2^5 - 52c_2^3 c_3 + 28c_2^2 c_4 - 17c_3 c_4 + c_2 (33c_3^2 - 13c_5) + 5c_6) e_k^6 \\
&- (32c_2^6 - 128c_2^4 c_3 - 18c_3^3 + 72c_2^3 c_4 + 12c_4^2 + 18c_2^2 (7c_3^2 - 2c_5) + 11c_3 c_5 + c_2 (8c_6 - 46c_3 c_4)) e_k^7 \\
&- 3c_7 e_k^7 + O[e_k^8]
\end{aligned}$$

y el desarrollo de Taylor de su imagen:

$$\begin{aligned}
f(y_k) &= f'(\xi) [c_2 e_k^2 + (2c_3 - 2c_2^2) e_k^3 + (4c_2^3 - 7c_2 c_3 + 3c_4) e_k^4] \\
&+ f'(\xi) [(8c_2^4 + 20c_2^2 c_3 - 6c_3^2 - 10c_2 c_4 + 4c_5) e_k^5] \\
&+ f'(\xi) [(16c_2^5 - 52c_2^3 c_3 + 28c_2^2 c_4 - 17c_3 c_4 + c_2 (33c_3^2 - 13c_5) + 5c_6) e_k^6] \\
&- f'(\xi) [2(16c_2^6 - 64c_2^4 c_3 - 9c_3^3 + 36c_2^3 c_4 + 6c_4^2 + 9c_2^2 (7c_3^2 - 2c_5)) e_k^7] \\
&+ f'(\xi) [(11c_3 c_5 + c_2 (8c_6 - 46c_3 c_4) - 3c_7) e_k^7] + O[e_k^8]
\end{aligned}$$

lo que nos permiten calcular el cociente H_1 :

$$\begin{aligned}
H_1 &= \frac{f(y_k)}{f(x_k) - 2f(y_k)} \\
&= 1 + c_2 e_k + (3c_4 - 2c_2 c_3) e_k^2 + (3c_4 - 2c_2 c_3) e_k^3 + (2c_2^4 - 3c_2^2 c_3 - 2c_2 c_4 + 4c_5) e_k^4 \\
&+ (14c_2^3 c_3 - 4c_2^5 - 5c_2^2 c_4 + 2c_3 c_4 - c_2 (9c_3^2 + 2c_5) + 5c_6) e_k^5 \\
&+ (4c_2^6 - 24c_2^4 c_3 - 6c_3^3 + 17c_2^3 c_4 + 3c_4 + 3c_4^2) e_k^6 \\
&+ (c_2^2 (32c_3^2 - 7c_5) + 4c_3 c_5 - 2c_2 (13c_3 c_4 + c_5) + 6c_7) e_k^6 \\
&+ (16c_2^5 c_3 - 24c_2^4 c_4 - 23c_2^3 c_4 + 10c_4 c_5 + c_2^2 (21c_5 - 48c_3^2)) e_k^7 \\
&+ (c_2^2 (71c_3 c_4 - 9c_6) + 6c_3 c_6 + 2c_2 (14c_3^3 - 9c_4^2 - 17c_3 c_5 - c_7) + 7c_8) e_k^7 \\
&+ O[e_k^8].
\end{aligned}$$

Así, el polinomio de Taylor asociado al segundo paso z_k será:

$$\begin{aligned}
z_k - \xi &= (c_2^3 - c_2 c_3) e_k^4 \\
&- 2(2c_2^4 - 4c_2^2 c_3 + c_2 c_4) e_k^5 \\
&+ (10c_2^5 - 30c_2^3 c_3 + 12c_2^2 c_4 - 7c_3 c_4 - 3c_2 (c_5 - 6c_3^2)) e_k^6 \\
&- 2(10c_2^6 - 40c_2^4 c_3 - 6c_3^3 + 20c_2^3 c_4 + 3c_4^2 + 8c_2^2 (5c_3^2 - c_5) + 5c_3 c_5 + c_2 (2c_6 - 26c_3 c_4)) e_k^7 \\
&+ O[e_k^8].
\end{aligned}$$

De nuevo, expandiendo $f(z_k)$ alrededor de ξ , obtenemos:

$$\begin{aligned}
f(z_k) &= (c_2^3 - c_2 c_3) e_k^4 \\
&- 2(2c_2^4 - 4c_2^2 c_3 + c_2 c_4) e_k^5 \\
&+ (10c_2^5 - 30c_2^3 c_3 + 12c_2^2 c_4 - 7c_3 c_4 - 3c_2 (c_5 - 6c_3^2)) e_k^6 \\
&- 2(10c_2^6 - 40c_2^4 c_3 - 6c_3^3 + 20c_2^3 c_4 + 3c_4^2 + 8c_2^2 (5c_3^2 - c_5) + 5c_3 c_5 + c_2 (2c_6 - 26c_3 c_4)) e_k^7 + O[e_k^8],
\end{aligned}$$

y

$$\begin{aligned}
f(z_k)/f'(x_k) &= (c_2^3 - c_2 c_3) e_k^4 \\
&- 2(3c_2^4 - 5c_2^2 c_3 + c_3^2 + c_2 c_4) e_k^5 \\
&+ (22c_2^5 - 53c_2^3 c_3 + 16c_2^2 c_4 - 7c_3 c_4 + c_2 (25c_3^2 - 3c_5)) e_k^6 \\
&+ 2(32c_2^6 - 102c_2^4 c_3 - 9c_3^3 + 38c_2^3 c_4 + 3c_4^2 + c_2^2 (80c_3^2 - 11c_5) + 5c_3 c_5 + c_2 (2c_6 - 38c_3 c_4)) e_k^7 \\
&+ O[e_k^8].
\end{aligned}$$

Calculamos ahora los diferentes factores que aparecen en el último paso:

$$\begin{aligned}
H_2 &= \frac{f(z_k)}{a_1 f(y_k) + a_2 f(z_k)} \\
&= \frac{1}{a_1} (c_2^2 - c_3) e_k^2 - \frac{2}{a_1} (c_2^3 - 2c_2 c_3 + c_4) e_k^3 \\
&\quad + \frac{1}{a_1^2} [(a_1 - a_2) c_2^4 + 2(a_2 - 3a_1) c_2^2 c_3 + (3a_1 - a_2) c_3^2 + 5a_1 c_2 c_4 - 3a_1 c_5] e_k^4 + O[e_k^5], \\
H &= m_1 + m_2 H_1 + m_3 H_2 \\
&= m_1 + m_2 c_2 e_n + \frac{1}{a_1} [(m_3 - a_1 m_2) c_2^2 + (2a_1 m_2 - m_3) c_3] e_k^2 \\
&\quad + \frac{1}{a_1} [(4m_3 - 2a_1 m_2) c_2 c_3 - 2m_3 c_2^3 + (3a_1 m_2 - 2m_3) c_2] e_k^3 \\
&\quad + \frac{1}{a_1^2} [(2a_1^2 m_2 + a_1 m_3 - a_2 m_3) c_2^4 + (2a_2 m_3 - 3a_1^2 m_2 - 6a_1 m_3) c_2^2 c_3 + (3a_1 - a_2) m_3 c_3^2] e_k^4 \\
&\quad + \frac{1}{a_1} [(5m_3 - 2a_1 m_2) c_2 c_4 + (4a_1 m_2 - 3m_3) c_5] e_k^4 + O[e_k^5].
\end{aligned}$$

Es sencillo comprobar que para que el último paso tenga orden al menos siete, n debe tener valor dos. En este caso la ecuación del error queda:

$$\begin{aligned}
e_{k+1} &= (m_1^2 - 1) c_2 (c_2^2 - c_3) e_k^4 + 2(3m_1^2 - m_1 m_2 - 2) c_2^4 e_k^5 \\
&\quad + 2[(4 - 5m_1^2 + m_1 m_2 + 4) c_2^2 c_3 + (m_1^2 - 1) c_3^2 + (m_1^2 - 1) c_2 c_3] e_k^5 + O[e_k^6]. \tag{4.18}
\end{aligned}$$

Para tener convergencia de orden cinco debe verificarse $m_1 = \pm 1$. Tomamos $m_1 = 1$ y recalculando (4.18) obtenemos :

$$e_{k+1} = 2[(1 - m_2) c_2^2 (c_2^2 - c_3)] e_k^5 + O[e_k^6]. \tag{4.19}$$

Notamos que el orden será seis si $m_2 = 1$. Recalculamos (4.19) y obtenemos:

$$\begin{aligned}
e_{k+1} &= (2m_3 - a_1) c_2 (c_2^2 - c_3) e_k^6 \\
&\quad - \frac{1}{a_1} [2((c_2^2 - c_3)((2a_1 - 7m_3) c_2^4 + (13m_3 - 5a_1) c_2^2 c_3 + (a_1 - 2m_3) c_3^2 + 2(a_1 - 2m_3) c_2 c_4)] e_k^7 \\
&\quad + O[e_k^8].
\end{aligned}$$

De nuevo, si $m_3 = \frac{a_1}{2}$ la ecuación de error queda:

$$e_{k+1} = 3(c_2^3 - c_2 c_3)^2 e_k^7 + O[e_k^8], \text{ para todo } a_1, a_2 \in \mathbb{R}, a_1 \neq 0. \tag{4.20}$$

Finalmente el algoritmo resulta:

$$\begin{aligned}
y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\
z_k &= x_k - \left[1 + \frac{f(y_k)}{f(x_k) - 2f(y_k)} \right] \frac{f(x_k)}{f'(x_k)}, \\
x_{k+1} &= z_k - \left[1 + \frac{f(y_k)}{f(x_k) - 2f(y_k)} + \frac{a_1}{2} \frac{f(z_k)}{a_1 f(y_k) + a_2 f(z_k)} \right]^2 \frac{f(z_k)}{f'(x_k)},
\end{aligned} \tag{4.21}$$

y queda probado su orden de convergencia siete. □

Denotamos por M7 a los métodos de esta familia.

Observamos que si se toma $m_1 = -1$ el último paso tiene la expresión:

$$x_{k+1} = z_k - \left[-1 - \frac{f(y_k)}{f(x_k) - 2f(y_k)} - \frac{a_1}{2} \frac{f(z_k)}{(a_1 f(y_k) + a_2 f(z_k))} \right]^2 \frac{f(z_k)}{f'(x_k)}.$$

Casos particulares:

1. Si fijamos $a_2 = 0$ y $a_1 = 1$, entonces

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\ z_k &= x_k - \left[1 + \frac{f(y_k)}{f(x_k) - 2f(y_k)} \right] \frac{f(x_k)}{f'(x_k)}, \\ x_{k+1} &= z_k - \left[1 + \frac{f(y_k)}{f(x_k) - 2f(y_k)} + \frac{1}{2} \frac{f(z_k)}{f(y_k)} \right]^2 \frac{f(z_k)}{f'(x_k)}. \end{aligned} \quad (4.22)$$

2. Si fijamos $a_2 = -1$ y $a_1 = 1$, entonces

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\ z_k &= x_k - \left[1 + \frac{f(y_k)}{f(x_k) - 2f(y_k)} \right] \frac{f(x_k)}{f'(x_k)}, \\ x_{k+1} &= z_k - \left[1 + \frac{f(y_k)}{f(x_k) - 2f(y_k)} + \frac{1}{2} \frac{f(z_k)}{f(y_k) - f(z_k)} \right]^2 \frac{f(z_k)}{f'(x_k)}. \end{aligned} \quad (4.23)$$

Llamaremos M7A a este último caso particular de la familia y lo usaremos en las pruebas numéricas más adelante.

Tomando el método de Newton como predictor buscamos una combinación del método de Ostrowski cambiando el último paso del método M7 (véase (4.21)) con fin de obtener un método óptimo, en la forma:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\ z_k &= x_k - (b_1 + b_2 H_1) \frac{f(x_k)}{f'(x_k)}, \\ x_{k+1} &= z_k - (p + n_1 H_1 + n_2 H_2)^2 L \frac{f(z_k)}{f'(x_k)}, \end{aligned} \quad (4.24)$$

donde $H_1 = \frac{f(y_k)}{gf(x_k) - af(y_k)}$, $H_2 = \frac{f(z_k)}{mf(x_k) - nf(z_k)}$ y $L = \frac{k_1 f(x_k) + k_2 f(y_k) + k_3 f(z_k)}{r_1 f(x_k) + r_2 f(y_k) + r_3 f(z_k)}$.

Este método ha sido aceptado en *Applied Mathematics Letters* [24].

Teorema 4.2.5 Sea $f : I \subseteq \mathbb{R} \rightarrow \mathbb{R}$ suficientemente diferenciable en cada punto del intervalo abierto tal que $\xi \in I$ es solución de la ecuación no lineal $f(x) = 0$. Supongamos que $f'(x)$ es continua y es diferente de cero en ξ . Entonces, la sucesión $\{x_k\}_{k \geq 0}$ obtenida usando las expresiones (4.24) converge a ξ con convergencia de orden ocho si $a = 2g$, $b_1 = 1$, $b_2 = g$, $p = 1$, $n_1 = g$, $n_2 = \frac{m}{2}$, $L \neq 1$ con $r_1 = k_1$, $r_2 = k_2$, $r_3 = k_3 - 3k_1$ y para todo $g, m, n, k_1, k_2, k_3 \in \mathbb{R}$ con m y k_1 no nulos.

Demostración: Sea ξ un cero simple de f . Como f es una función suficientemente diferenciable, mediante la expansión de Taylor de $f(x_k)$ y $f'(x_k)$ alrededor de ξ y teniendo en cuenta que $f(\xi) = 0$, obtenemos:

$$\begin{aligned} f(x_k) &= f'(\xi) [e_k + c_2 e_k^2 + c_3 e_k^3 + c_4 e_k^4 + c_5 e_k^5 + c_6 e_k^6 + c_7 e_k^7 + c_8 e_k^8] + O[e_k^9], \\ f'(x_k) &= f'(\xi) [1 + 2c_2 e_k + 3c_3 e_k^2 + 4c_4 e_k^3 + 5c_5 e_k^4 + 6c_6 e_k^5 + 7c_7 e_k^6 + 8c_8 e_k^7] + O[e_k^8], \end{aligned}$$

donde $c_j = \frac{1}{k!} \frac{f^{(j)}(\xi)}{f'(\xi)}$ con $j = 2, 3, \dots$. Como los términos entre corchetes son polinomios en términos de e_k , la

división directa nos da:

$$\begin{aligned} \frac{f(x_k)}{f'(x_k)} &= e_k - c_2 e_k^2 + 2(c_2^2 - c_3) e_k^3 + (7c_2 c_3 - 4c_2^3 - 3c_4) e_k^4 + (8c_2^4 - 20c_2^2 c_3 + 6c_3^2 + 10c_2 c_4 - 4c_5) e_k^5 \\ &+ (-16c_2^5 + 52c_2^3 c_3 - 28c_2^2 c_4 + 17c_3 c_4 - 33c_2 c_3^2 + 13c_2 c_5 - 5c_6) e_k^6 \\ &+ 2(16c_2^6 - 64c_2^4 c_3 - 9c_3^3 + 36c_2^3 c_4 - 6c_4^2 + 63c_2^2 c_3^2 - 18c_2^2 c_5 + 11c_3 c_5 - 46c_2 c_3 c_4 + 8c_2 c_6 - 3c_7) e_k^7 \\ &+ (-64c_2^7 + 304c_2^5 c_3 - 176c_2^4 c_4 - 75c_3^2 c_4 + 31c_4 c_5 - 408c_2^3 c_3^2 + 92c_2^3 c_5) e_k^8 \\ &+ (4c_2^2(87c_3 c_4 - 11c_6 + 27c_3 c_6) + 135c_2 c_3^3 - 64c_2 c_4^2 - 108c_2 c_3 c_5 + 19c_2 c_7 - 7c_8) e_k^8 + O[e_k^9], \end{aligned}$$

y por lo tanto, la expresión para $y_k = x_k - \frac{f(x_k)}{f'(x_k)}$ es:

$$\begin{aligned} y_k &= \xi + c_2 e_k^2 + 2(c_3 - c_2^2) e_k^3 + (4c_2^3 + 3c_4 - 7c_2 c_3) e_k^4 + (-8c_2^4 + 20c_2^2 c_3 - 6c_3^2 - 10c_2 c_4 + 4c_5) e_k^5 \\ &+ (16c_2^5 - 52c_2^3 c_3 + 28c_2^2 c_4 - 17c_3 c_4 + 33c_2 c_3^2 - 13c_2 c_5 + 5c_6) e_k^6 \\ &- 2(16c_2^6 - 64c_2^4 c_3 - 9c_3^3 + 36c_2^3 c_4 - 6c_4^2 + 63c_2^2 c_3^2 - 18c_2^2 c_5 + 11c_3 c_5 - 46c_2 c_3 c_4 + 8c_2 c_6 - 3c_7) e_k^7 \\ &- (-64c_2^7 + 304c_2^5 c_3 - 176c_2^4 c_4 - 75c_3^2 c_4 + 31c_4 c_5 - 408c_2^3 c_3^2 + 92c_2^3 c_5 + 4c_2^2(87c_3 c_4 - 11c_6 + 27c_3 c_6)) e_k^8 \\ &- (135c_2 c_3^3 - 64c_2 c_4^2 - 108c_2 c_3 c_5 + 19c_2 c_7 - 7c_8) e_k^8 + O[e_k^9]. \end{aligned}$$

De nuevo, expandiendo $f(y_k)$ alrededor de ξ , obtenemos:

$$\begin{aligned} f(y_k) &= f'(\xi) [c_2 e_k^2 + 2(c_3 - c_2^2) e_k^3 + (5c_2^3 - 7c_2 c_3 + 3c_4) e_k^4 + (6c_2^4 - 12c_2^2 c_3 + 3c_3^2 + 5c_2 c_4 - 2c_5) e_k^5] \\ &+ f''(\alpha) (28c_2^5 - 73c_2^3 c_3 + 34c_2^2 c_4 - 17c_3 c_4 + 37c_2 c_3^2 - 13c_2 c_5 + 5c_6) e_k^6 \\ &- 2f'''(\alpha) (32c_2^6 - 103c_2^4 c_3 - 9c_3^3 + 52c_2^3 c_4 + 6c_4^2 + 80c_2^2 c_3^2 - 22c_2^2 c_5) e_k^7 \\ &- 2f^{(4)}(\alpha) (11c_3 c_5 - 52c_2 c_3 c_4 + 8c_2 c_6 - 3c_7) e_k^7 \\ &+ f^{(5)}(\alpha) (144c_2^7 - 552c_2^5 c_3 + 297c_2^4 c_4 + 75c_3^2 c_4 + 582c_2^3 c_3^2 - 134c_2^3 c_5 - 27c_3 c_6) e_k^8 \\ &+ f^{(6)}(\alpha) (c_2^2(54c_6 - 455c_3 c_4) - 135c_2 c_3^3 + 64c_2 c_4^2 + 108c_2 c_3 c_5 - 19c_2 c_7 - 31c_4 c_5 + 7c_8) e_k^8 \\ &+ O[e_k^9]. \end{aligned}$$

Ahora calculamos el valor de H_1 :

$$\begin{aligned} H_1 &= \frac{f(y_k)}{gf(x_k) - af(y_k)} \\ &= \frac{1}{g} c_2 e_k + \frac{1}{g^2} [(a - 3g) c_2^2 + 2gc_3] e_k^2 + \frac{1}{g^3} [(a^2 - 6ag + 8g^2) c_2^3 + 2(2a - 5g)gc_2 c_3 + 3g^2 c_4] e_k^3 \\ &+ \frac{1}{g^4} [(a^3 - 9a^2 g + 25ag^2 - 20g^3) c_2^4 + g(6a^2 - 32ag + 37g^2) c_2^2 c_3] e_k^4 \\ &+ \frac{1}{g^4} [2g^2(3a - 7g) c_2 c_4 + 4g^2((a - 2g) c_3^2 + gc_5)] e_k^4 + O[e_k^5]. \end{aligned}$$

Con esto podemos calcular $z_k = x_k - (b_1 + b_2 H_1) \frac{f(x_k)}{f'(x_k)}$ de donde se obtiene la ecuación del error para z_k :

$$\begin{aligned} e_{z_k} &= (1 - b) e_k - \frac{1}{g} (b_1 - bg) c_2 e_k^2 - \frac{1}{g^2} [(ab_1 + 2g(bg - 2b_1)) c_2^2 + 2(b_1 - bg)gc_3] e_k^3 \\ &- \frac{1}{g^3} [(a^2 b_1 - 7ab_1 g + (13b_1 - 4bg)) g^2 c_2^3 + g(4ab_1 + 7g(g - 2b_1)) c_2 c_3 + 3(b_1 - bg) g^2 c_4] e_k^4 + O[e_k^5]. \end{aligned}$$

Los valores de los parámetros a , b , b_1 y g necesarios para obtener convergencia de orden al menos cuatro se obtienen resolviendo las ecuaciones $1 - b_1 = 0$, $b_2 - b_1 g = 0$ y $ab_2 + 2g(b_1 g - 2b_2) = 0$ simultáneamente. Como resultado obtenemos los valores de a , b_1 y b_2 en función de g : $a = 2g$, $b_1 = 1$ y $b_2 = g$. Recalculando obtenemos

la ecuación de error para z_k en la forma:

$$\begin{aligned} e_{z_k} &= (c_2c_3 - c_2^3)e_k^4 + (c_2^2c_3 - 4c_2^4 - 2c_2c_4 - 2c_3^2)e_k^5 \\ &+ (10c_2^5 + 12c_2^2c_4 - 30c_2^3c_3 - 7c_3c_4 + 18c_2c_3^2 - 3c_2c_5)e_k^6 \\ &+ 2(10c_2^5 - 40c_2^4c_3 - 6c_3^3 + 20c_2^3c_4 + 3c_4^2 + 40c_2^2c_3^2 - 8c_2^2c_5 + 5c_3c_5 - 26c_2c_3c_4 + 2c_2c_6)e_k^7 \\ &+ O[e_k^8]. \end{aligned}$$

Mediante la expansión de Taylor de $f(z_k)$ alrededor de ξ obtenemos:

$$\begin{aligned} f(z_k) &= f'(\xi)[(c_2c_3 - c_2^3)e_k^4 + (8c_2^2c_3 - 4c_2^4 - 2c_2c_4 - 2c_3^2)e_k^5] \\ &+ f'(\xi)(10c_2^5 + 12c_2^2c_4 - 30c_2^3c_3 - 7c_3c_4 + 18c_2c_3^2 - 3c_2c_5)e_k^6 \\ &+ 2f'(\xi)(10c_2^6 - 40c_2^4c_3 - 6c_3^3 + 20c_2^3c_4 + 3c_4^2 + 40c_2^2c_3^2 - 8c_2^2c_5 + 5c_3c_5 - 26c_2c_3c_4 + 2c_2c_6)e_k^7 \\ &+ O[e_k^8]. \end{aligned}$$

Ahora calculamos los cocientes:

$$\frac{f(z_k)}{f'(x_k)} = (c_2c_3 - c_2^3)e_k^4 - (6c_2^4 - 10c_2^2c_3 + 2c_2c_4 + 2c_3^2)e_k^5 + O[e_k^6]$$

y

$$\begin{aligned} H_2 &= \frac{f(z_k)}{mf(y_k) - nf(z_k)} = \frac{1}{m}(c_2^2 - c_2c_3)e_k^2 - \frac{2}{m}(c_2^3 - 2c_2c_3 + c_4)e_k^3 \\ &+ \frac{1}{m^2}[(m+n)c_2^4 + (3m+n)(c_2^2 - 2c_2^2c_3) + 5mc_2c_4 - 3mc_5]e_k^4 \\ &+ \frac{1}{m^2}[4(m-n)c_2^5 - 4(m-3n)c_2^3c_3 - 4(m+n)c_2^2c_4 + 2(3m+2n)c_3c_4]e_k^5 \\ &+ \frac{1}{m^2}[6mc_5c_2 - 4(m+2n)c_2c_3^2 - 4mc_6]e_k^5 + O[e_k^6]. \end{aligned}$$

De manera que, mediante H_1 y H_2 calculamos $H = p + n_1H_1 + n_2H_2$:

$$\begin{aligned} H &= p + \frac{1}{g}n_1c_2e_k + \frac{1}{gm}[gn_2 - mn_1c_2^3 + (2mn_1 - gn_2)c_3]e_k^2 \\ &+ \frac{1}{gm}[(3mn - 2gn_2)c_4 - 2gn_2c_2^3 - 2(mn_1 - 2gn_2)c_2c_3]e_k^3 \\ &+ \frac{1}{gm^2}[(2m^2n_1 + gmn_2 + gnn_2)c_2^4 - (3m^2n_1 + 6gmn_2 + 2gnn_2)c_2^2c_3]e_k^4 \\ &+ \frac{1}{gm^2}[g(3m+n)n_2c_3^2 + m(5gn_2 - 2mn_1)c_2c_4 + n(4mn_1 - 3gn_2)c_5]e_k^4 \\ &+ \frac{1}{gm^2}[2(7m^2n_1 - 2gmn_2 + 6gnn_2)c_2^3c_3 - 4(m^2n_1 - gmn_2 + gnn_2)c_2^5]e_k^5 \\ &+ \frac{1}{gm^2}[2(m^2n_1 + 3gmn_2 + 2gnn_2)c_3c_4 - (5m^2n_1 + 4gmn_2 + 4gnn_2)c_2^2c_4]e_k^5 \\ &+ \frac{1}{gm^2}[-c_2((9m^2n_1 + 4gmn_2 + 8gnn_2)c_3^4 + 2m(mn_1 - gn_2)c_5) + m(5mn_1 - 4gn_2)c_6]e_k^5 \\ &+ O[e_k^6]. \end{aligned}$$

Dado que el último paso se obtiene de la expresión iterativa $x_{k+1} = z_k - H^2L \frac{f(z_k)}{f'(x_k)}$, la ecuación del error

para x_{k+1} será:

$$\begin{aligned}
e_{k+1} &= \frac{r_1 - k_1}{r_1} c_2 (c_2^2 - c_3) e_k^4 \\
&+ \frac{1}{r_1^2} [(-r_1(n_1(k_2 + 4r_1) + k_1(4r_1 + r_2)))c_2^4 - 2r_1(r_1 - k_1)c_2c_4] e_k^5 \\
&+ \frac{1}{r_1^2} [(r_1(k_2 + 8r_1) - k_1(8r_1 + r_2))c_2^2c_3 - 2r_1(r_1 - k_1)c_3^2] e_k^5 \\
&+ \frac{1}{r_1^3} [(r_1(7k_2r_1 + 10r_1^2 + k_2r_2) - k_1(10r_1^2 + 7r_1r_2 + r_2^2))c_2^5 - r_1(13k_2r_1 + 30r_1^2 + k_2r_2)c_2^3c_5] e_k^6 \\
&+ \frac{1}{r_1^3} [k_1(30r_1^2 + 13r_1r_2 + r_2^2)c_2^3c_5 + (2r_1(r_1(k_2 + 6r_1) + k_1(6r_1 + r_2)))c_2^2c_4 + 7(k_1 - r_1)r_1^2c_3c_4] e_k^6 \\
&+ \frac{1}{r_1^3} [(r_1c_2(-2(9k_1r_1 - 2k_2r_1 - 9r_1^2 + 2k_1r_2))c_3^2 + 3(k_1 - r_1)r_1c_5)] e_k^6 + c_7e_k^7 + O[e_k^8].
\end{aligned}$$

Para garantizar orden de convergencia al menos siete resolvemos las ecuaciones:

$$\begin{aligned}
k_1 - r_1 &= 0 \\
r_1(k_1 + 4r_1) + k_1(4r_1 + r_2) &= 0 \\
r_1(k_2 + 8r_1) - k_1(8r_1 + r_2) &= 0
\end{aligned}$$

simultaneamente. Los valores que satisfacen las ecuaciones son: $k_1 = r_1$ y $k_2 = r_2$. Los sustituimos en L y recalculamos e_{k+1} :

$$\begin{aligned}
e_{k+1} &= \frac{(c_2^3 - c_2c_3)^2}{k_1} [(3k_1 - k_3 + r_3)] e_k^7 \\
&+ \frac{(c_2^3 - c_2c_3)}{4k_1^2m} [k_1^2(89m + 4n) - 36k_1m(k_3 - r_3) + 4k_2m(r_3 - k_3)]c_2^4 e_k^8 \\
&+ \frac{(c_2^3 - c_2c_3)}{4k_1^2m} [-2(k_1^2(89m + 4n) - 34k_1m(k_3 - r_3) + 2k_2m(r_3 - k_3))c_2^2c_3] e_k^8 \\
&+ \frac{(c_2^3 - c_2c_3)}{4k_1^2m} [k_1(k_1(45m + 4n) - 16m(k_3 - r_3))c_3^2 + 4k_1m(11k_1 - 4k_3 + 4r_3)c_2c_4] e_k^8 \\
&+ O[e_k^9],
\end{aligned}$$

de donde se ve que si $r_3 = k_3 - 3k_1$ el orden de la convergencia será ocho. Tomamos este valor y recalculamos la ecuación del error para e_{k+1} :

$$\begin{aligned}
e_{k+1} &= \frac{(c_2^3 - c_2c_3)}{4k_1m} [(19k_1m + 12k_2m - 4k_1n)c_2^4 - 2(13k_1m + 6k_2m - 4k_1n)c_2^2c_3] e_k^8 \\
&+ \frac{(c_2^3 - c_2c_3)}{4k_1m} [k_1(3m - 4n)c_2^3 + 4k_1mc_2c_4] e_k^8 + O[e_k^9].
\end{aligned}$$

□

Finalmente, el algoritmo de esta familia de métodos óptimos queda:

$$\begin{aligned}
y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\
z_k &= x_k - (1 + H_1) \frac{f(x_k)}{f'(x_k)}, \\
x_{k+1} &= z_k - (1 + H_1 + \frac{m}{2}H_2)^2 L \frac{f(z_k)}{f'(x_k)},
\end{aligned} \tag{4.25}$$

donde $H_1 = \frac{f(y_k)}{f(x_k) - 2f(y_k)}$, $H_2 = \frac{f(z_k)}{mf(y_k) - nf(z_k)}$ y $L = \frac{k_1f(x_k) + k_2f(y_k) + k_3f(z_k)}{k_1f(x_k) + k_2f(y_k) + (k_3 - 3k_1)f(z_k)}$, para todo m, n, k_1, k_2 y $k_3 \in \mathbb{R}$ con m y k_1 no nulos.

Casos particulares:

El operador H depende de dos parámetros: $m \neq 0$ y n . Escogemos dos casos

CASO A: $m = 1$ y $n = 0$,

$$H = 1 + \frac{f(y_k)}{f(x_k) - 2f(y_k)} + \frac{1}{2} \frac{f(z_k)}{f(y_k)}. \quad (4.26)$$

CASO B: $m = 1$ y $n = 2$,

$$H = 1 + \frac{f(y_k)}{f(x_k) - 2f(y_k)} + \frac{1}{2} \frac{f(z_k)}{f(y_k) - 2f(z_k)}. \quad (4.27)$$

El operador L depende de tres parámetros: k_1, k_2 y k_3 . Desarrollamos dos casos para los cuales tomamos $k_2 = 0$:

1. Si $k_3 = 3k_1$, para todo $k_1 \in \mathbb{R}$ y $k_1 \neq 0$, $L = 1 + 3 \frac{f(z_k)}{f(x_k)}$;
2. Si $k_3 = 0$, para todo $k_1 \in \mathbb{R}$ y $k_1 \neq 0$, $L = \frac{f(x_k)}{f(x_k) - 3f(z_k)}$.

Para las pruebas numéricas escogemos de los casos particulares la combinación B-1 con la siguiente fórmula iterativa:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\ z_k &= x_k - \left[1 + \frac{f(y_k)}{f(x_k) - 2f(y_k)} \right] \frac{f(x_k)}{f'(x_k)}, \\ x_{k+1} &= z_k - \left[\left(1 + \frac{f(y_k)}{f(x_k) - 2f(y_k)} + \frac{1}{2} \frac{f(z_k)}{f(y_k) - 2f(z_k)} \right)^2 \left[1 + 3 \frac{f(z_k)}{f(x_k)} \right] \frac{f(z_k)}{f'(x_k)} \right]. \end{aligned} \quad (4.28)$$

Denotamos a este método por M8B.

A continuación presentamos un nuevo método óptimo de orden 8 generado a partir del método de orden siete M7A, al que sumamos un último paso en el que no se añade ninguna evaluación funcional.

Teorema 4.2.6 *Sea $f : I \subseteq \mathbb{R} \rightarrow \mathbb{R}$ suficientemente diferenciable en cada punto del intervalo abierto I . Sea $\xi \in I$ solución de la ecuación no lineal $f(x) = 0$. Supongamos que $f'(x)$ es continua y es diferente de cero en ξ . Entonces, la sucesión $\{x_k\}_{k \geq 0}$ obtenida usando el algoritmo:*

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\ z_k &= x_k - \left[1 + \frac{f(y_k)}{f(x_k) - 2f(y_k)} \right] \frac{f(x_k)}{f'(x_k)} \\ u_k &= z_k - \left[1 + \frac{f(y_k)}{f(x_k) - 2f(y_k)} + \frac{1}{2} \frac{f(z_k)}{f(y_k) - f(z_k)} \right]^2 \frac{f(z_k)}{f'(x_k)} \\ x_{k+1} &= u_k - m_1 \left[\frac{f(z_k)}{f(y_k) - f(z_k)} \right] \left[\frac{m_2 f(z_k) + m_3 f(y_k)}{m_4 f(x_k) + m_5 f(z_k)} \right] \frac{f(z_k)}{f'(x_k)} \end{aligned} \quad (4.29)$$

converge a ξ con convergencia de orden ocho si $m_3 = \frac{3m_4}{m_1}$ para todo $m_4, m_5, m_2 \in \mathbb{R}$ con $m_4 \neq 0$ y $m_1 \neq 0$.

Demostración: Sean ξ un cero simple de f que es una función suficientemente diferenciable y que $f(\xi) = 0$. Partimos de la expresión de la ecuación de error (4.20) tomando el caso particular 2 (véase 4.23) que determina la expresión para u_k

$$u_k = \xi + 3(c_2^3 - c_2c_3)^2 e_k^7 + \frac{1}{4} [c_2(c_2^2 - c_3)(93c_2^4 - 186c_2^2c_3 + 49c_3^2 + 44c_2c_4)] e_k^8 + O[e_k^9]. \quad (4.30)$$

Calculamos:

$$\begin{aligned} L &= \frac{m_2 f(z_k) + m_3 f(y_k)}{m_4 f(x_k) + m_5 f(z_k)} \\ &= \frac{m_3}{m_4} c_2 e_k + \frac{m_3}{m_4} (2c_3 - 3c_2^2) e_k^2 + \frac{1}{m_4} ((m_2 + 8m_3)c_2^3 - (m_2 + 10m_3)c_2c_3 + 3m_3c_4) e_k^3 \\ &+ \frac{1}{m_4^2} ((9m_2m_4 + m_3(37m_4 + m_5))c_2^2c_3 - (5m_2m_4 + m_3(20m_4 + m_5))c_2^4) e_k^4 \\ &- \frac{1}{m_4^2} (2(m_2 + 7m_3)m_4c_2c_4 - 2m_4((m_2 + 4m_3)c_3^2 - 2m_3c_5)) e_k^4 + O[e_k^5]. \end{aligned}$$

La expresión iterativa de x_{k+1} es de la forma:

$$x_{k+1} = u_k - m_1 \left[\frac{f(z_k)}{f(y_k) - f(z_k)} \right] L \frac{f(z_k)}{f'(x_k)}.$$

Sustituyendo L y los cocientes en términos de polinomios en e_k , obtenemos la ecuación del error

$$\begin{aligned} e_{k+1} &= \frac{m_1m_3 - 3m_4}{m_4} (c_2^3 - c_2c_3)^2 e_k^7 \\ &+ \frac{c_2(c_2^2 - c_3)}{4m_4^2} ((44m_1m_3 - 93m_4)c_2^4 + 2(-38m_1m_3 + 93m_4)c_2^2c_3) \\ &+ \frac{c_2(c_2^2 - c_3)}{4m_4^2} ((16m_1m_3 - 49m_4)c_3^2 + 4(4m_1m_3 - 11m_4)c_2c_4) e_k^8 + O[e_k^9]. \end{aligned}$$

Notamos que si tomamos $m_3 = \frac{3m_4}{m_1}$ obtendremos orden 8. En este caso la ecuación de error quedará:

$$e_{k+1} = \frac{1}{4} c_2(c_2^2 - c_3)(39c_2^4 - 42c_2^2c_3 - c_3^2 + 4c_2c_4) e_k^8 + O[e_k^9]$$

para todo $m_4, m_5, m_2 \in \mathbb{R}$ con $m_4 \neq 0$ y $m_1 \neq 0$. □

Tomando como caso particular $m_1 = 1$ y $m_2 = m_5 = 0$, resulta $x_{k+1} = u_k - \frac{3f(z_k)f(y_k)}{f(x_k)[f(y_k) - f(z_k)]} \frac{f(z_k)}{f'(x_k)}$.

Denotamos este método por M8C y lo usaremos en la sección siguiente para las pruebas numéricas.

A continuación desarrollamos otro método iterativo óptimo de orden ocho basado también en M7A añadiendo un paso más que no requiere nuevas evaluaciones funcionales. Este método fue publicado en *Journal of Computational and Applied Mathematics* [23].

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\ z_k &= x_k - \left[1 + \frac{f(y_k)}{f(x_k) - 2f(y_k)} \right] \frac{f(x_k)}{f'(x_k)} \\ u_k &= z_k - \left[1 + \frac{f(y_k)}{f(x_k) - 2f(y_k)} + \frac{1}{2} \frac{f(z_k)}{f(y_k) - 2f(z_k)} \right]^2 \frac{f(z_k)}{f'(x_k)} \\ x_{k+1} &= u_k - \frac{a_1(u_k - z_k) + a_2(y_k - x_k) + a_3(z_k - x_k)}{b_1(u_k - z_k) + b_2(y_k - x_k) + b_3(z_k - x_k)} \frac{f(z_k)}{f'(x_k)}. \end{aligned} \quad (4.31)$$

Teorema 4.2.7 Sea $\xi \in I$ un cero simple de una función suficientemente diferenciable $f : I \subseteq \mathbb{R} \rightarrow \mathbb{R}$ en el intervalo abierto I . Si x_0 está lo suficientemente cerca de ξ , entonces el método iterativo descrito por (4.31) alcanza orden de convergencia ocho, óptimo, para $a_2 = a_3 = 0$, $a_1 = 3(b_2 + b_3)$ y $b_2 + b_3 \neq 0$.

Demostración: Partimos de la expresión de la ecuación de error (4.20) tomando como caso particular B (véase (4.27)) determinando la expresión para u_k

$$u_k = \xi + 3(c_2^3 - c_2c_3)^2 e_k^7 + \frac{1}{4} [c_2(c_2^2 - c_3)(97c_2^4 - 194c_2^2c_3 + 53c_3^2 + 44c_2c_4)] e_k^8 + O[e_k^9]. \quad (4.32)$$

La expresión del error en el último paso es

$$\begin{aligned} e_{k+1} &= u_k - \frac{a_1(u_k - z_k) + a_2(y_k - x_k) + a_3(z_k - x_k)}{b_1(u_k - z_k) + b_2(y_k - x_k) + b_3(z_k - x_k)} \frac{f(z_k)}{f'(x_k)} \\ &= -\frac{a_2 + a_3}{b_2 + b_3} (c_2^3 - c_2c_3) e_k^4 \\ &\quad + \frac{1}{(b_2 + b_3)^2} [-(a_3b_2 - a_2b_3)c_2(c_2^3 - c_2c_3) + 2(a_2 + a_3)(b_2 + b_3)(3c_2^4 - 5c_2^2c_3 + c_3^2 + c_2c_4)] e_k^5 \\ &\quad + \frac{1}{(b_2 + b_3)^3} [(a_3b_2 - a_2b_3)(b_2 + 2b_3)c_2^2 - 2(b_2 + b_3)(c_2^3c_3 - c_2c_3^2)] e_k^6 \\ &\quad + \frac{1}{(b_2 + b_3)^3} [2(b_2 + b_3)(a_3b_2 - a_2b_3)c_2(3c_2^4 - 5c_2^2c_3 + c_3^2 + c_2c_4)] e_k^6 \\ &\quad - \frac{1}{(b_2 + b_3)^3} (a_2 + a_3)(b_2 + b_3)^2 (22c_2^5 - 53c_2^3c_3 + 16c_2^2c_4 - 7c_3c_4 + c_2(25c_3^2 - 3c_5)) e_k^6 + O[e_k^7] \end{aligned}$$

y, con el fin de obtener orden de convergencia siete, es necesario fijar el valor de algunos parámetros, específicamente $a_2 = a_3 = 0$ y $b_2 + b_3 \neq 0$. Entonces, la ecuación de error se convierte en

$$\begin{aligned} e_{k+1} &= -\frac{1}{b_2 + b_3} (a_1 - 3(b_2 + b_3))(c_2^3 - c_2c_3)^2 e_k^7 \\ &\quad + \frac{1}{4(b_2 + b_3)^2} c_2(c_2^2 - c_3)[(-97(b_2 + b_3)^2 + 4a_1(9b_2 + 10b_3))c_4^2 \\ &\quad + 2(97(b_2 + b_3)^2 - 2b_1(17b_2 + 18b_3))c_2^2c_3 + (b_2 + b_3)(16a_1 - 53(b_2 + b_3)c_3^2) \\ &\quad + 4(b_2 + b_3)(4a_1 - 11(b_2 + b_3))c_2c_4] e_k^8 + O[e_k^9]. \end{aligned}$$

Por último, si $a_1 = 3(b_2 + b_3)$ el orden de convergencia de cualquier método de la familia (4.31) llega a ocho, y la ecuación de error es

$$\begin{aligned} e_{k+1} &= \frac{1}{4(b_2 + b_3)} c_2(c_2^2 - c_3)[(11b_2 + 23b_3)c_4^2 - 2(5b_2 + 11b_3)c_2^2c_3 - 5(b_2 + b_3)c_3^2 + 4(b_2 + b_3)c_2c_4] e_k^8 \\ &\quad + O[e_k^9]. \end{aligned}$$

Con esto queda demostrado que el orden de convergencia es ocho. No hay ninguna restricción sobre el valor de b_1 y los valores de b_2 y b_3 deberán cumplir la restricción $b_2 + b_3 \neq 0$. \square

Caso particular: Si tomamos $b_1 = b_3 = 0$, entonces $b_2 = 1$, $a_1 = 3$ y el último paso de la fórmula iterativa será:

$$x_{k+1} = u_k - 3 \left[\frac{u_k - z_k}{y_k - x_k} \right] \frac{f(z_k)}{f'(x_k)}.$$

Este caso particular lo usaremos en la sección siguiente para las pruebas numéricas y lo denotaremos por M8D.

4.2.3. Pruebas numéricas

La Tabla 4.1 muestra el orden de convergencia, el número de evaluaciones funcionales y el número de operaciones en cada iteración, y los índices: I , IO y IC de los métodos diseñados: M7A, M8A, M8B, M8C y M8D. Para su comparación utilizamos de nuevo el método clásico de Newton (NC), el método de Ostrowski (OS) y el método óptimo de octavo orden, LW8, con $\alpha = 1$ y $G(t) = 4t$ (véase (3.49)) y el método iterativo BRW8 desarrollado por Bi et al. en [7], basándose en los métodos de King y la familia de métodos iterativos de sexto orden presentados en [82] con la siguiente expresión iterativa:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\ z_k &= y_k - \left[\frac{2f(x_k) - f(y_k)}{2f(x_k) - 5f(y_k)} \right] \frac{f(y_k)}{f'(x_k)}, \\ x_{k+1} &= z_k - H(\mu_k) \frac{f(z_k)}{f[z_k, y_k] + f[z_k, x_k, x_k](z_k - y_k)}, \end{aligned}$$

donde $\mu_k = f(z_k)/f(x_k)$ y $H(\mu_k)$ representa una función real y las diferencias divididas se denotan por $f[.,.]$. De esta familia de métodos consideramos el caso particular con $H(t) = 1 + \frac{2t}{1+\alpha t}$ y $\alpha = 1$.

Tabla 4.1: Comparación de los índices de los métodos iterativos

Métodos	p	d	op	I	IO	IC
M7A	7	4	6	1.6268	1.3831	1.2148
M8A	8	5	6	1.5157	1.4242	1.2081
M8B	8	4	9	1.6818	1.2599	1.1735
M8C	8	4	10	1.6818	1.2311	1.1601
M8D	8	4	8	1.6818	1.2968	1.1892
NC	2	2	1	1.4142	2.0000	1.2599
OS	4	3	3	1.5874	1.5864	1.2599
LW8	8	4	7	1.6818	1.3459	1.2081
BRW8	8	4	7	1.6818	1.3439	1.2081

Comparando los métodos vemos que los mayores índices de eficiencia los tienen todos los métodos óptimos de orden ocho. En caso de los índices operacional y computacional los mayores índices los tienen el método de Newton y de Ostrowski y entre los restantes métodos los mayores índices operacional y computacional los tienen los métodos M8A y M7A.

A continuación, en la Tabla 4.2 presentamos los resultados obtenidos al utilizar los métodos descritos en esta sección para estimar los ceros de las funciones de (1) a (18) (los mismos ejemplos anteriores).

1. $f_1(x) = \sin x - x^2 + 1$, $\xi \approx 1.404492$.
2. $f_2(x) = x^2 - \exp x - 3x + 2$, $\xi \approx 0.257530$.
3. $f_3(x) = \cos x - x$, $\xi \approx 0.739085$.
4. $f_4(x) = (x - 1)^3 - 1$, $\xi = 2$.
5. $f_5(x) = x^3 - 10$, $\xi \approx 2.154435$.
6. $f_6(x) = \cos x - x \exp x + x^2$, $\xi \approx 0.639154$.
7. $f_7(x) = \exp x - 1.5 - \arctan x$, $\xi \approx 0.767653$.
8. $f_8(x) = x^3 + 4x^2 - 10$, $\xi \approx 1.365230$.
9. $f_9(x) = 8x - \cos x - 2x^2$, $\xi \approx 0.128077$.

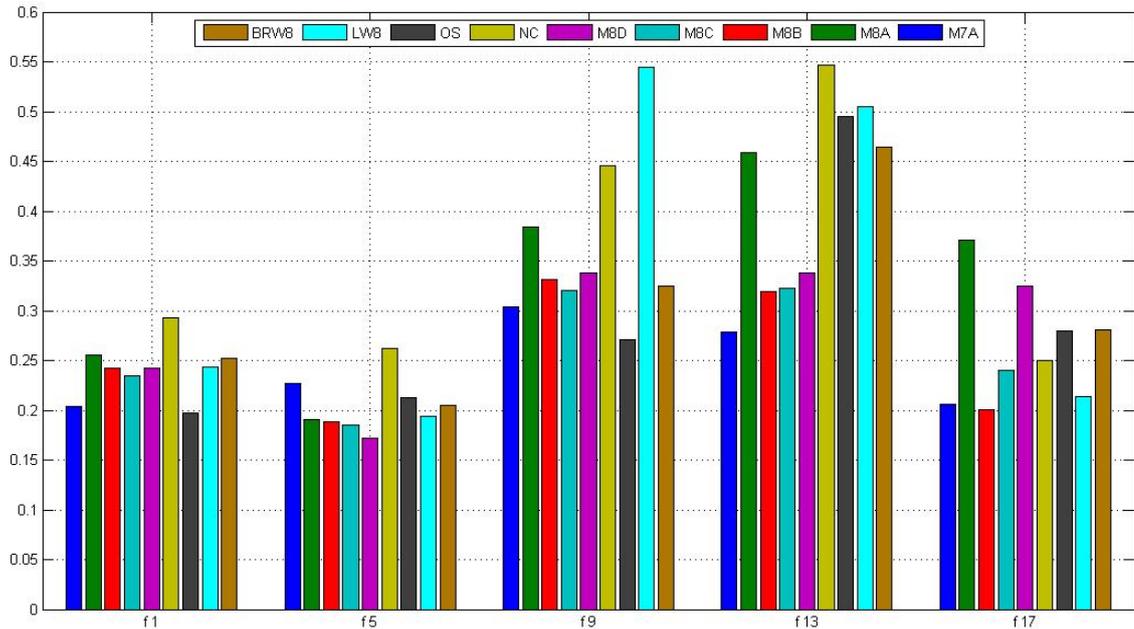


Figura 4.1: Comparación de los tiempos de ejecución

10. $f_{10}(x) = \arctan x$, $\xi = 0$.
11. $f_{11}(x) = \exp x - 4x^2$, $\xi \approx 0.714806$.
12. $f_{12}(x) = \left(\sin x - \frac{x}{2}\right)^2$, $\xi = 0$, doble.
13. $f_{13}(x) = x \exp x^2 - \sin x^2 + 3 \cos x + 5$, $\xi \approx -1.207648$.
14. $f_{14}(x) = \sqrt{x^2 + 2x + 5} - 2 \sin x - x^2 + 3$, $\xi \approx 2.331968$.
15. $f_{15}(x) = x^4 + \sin \frac{\pi}{x^2} - 5$, $\xi = \sqrt{2}$.
16. $f_{16}(x) = 10x \exp(-x^2) - 1$, $\xi \approx 1.679631$.
17. $f_{17}(x) = \exp(-x) + \cos x$, $\xi \approx 1.746140$.
18. $f_{18}(x) = \sqrt{x^4 + 8} \sin \frac{\pi}{x^2 + 2} + \frac{x^3}{x^4 + 1} - \sqrt{6} + \frac{8}{17}$, $\xi \approx -2$.

Todos los cálculos numéricos se han realizado bajo las mismas condiciones como los cálculos que permitieron obtener los resultados presentados en la Tablas 3.4 y 3.1.

Los datos presentados en la Tabla 4.2 muestran que los resultados numéricos y los teóricos están en concordancia. De nuevo $f_{10}(x) = 0$ y $f_{12}(x) = 0$ muestran sus particularidades. En el caso de $f_{10}(x) = 0$ ($f''(\xi) = 0$) obtenemos, en todos los casos, mayor orden de convergencia, excepto el método BRW8 que no converge. Notamos que en el caso de $f_{12}(x) = 0$ (doble raíz) todos los métodos tienen orden de convergencia uno y elevado número de iteraciones: el mayor número corresponde al método de Newton (536) y el menor al método M8B (187); excepto los métodos LW8 y BRW8 que no convergen.

Comparando los tiempos de ejecución ϵ , en general, vemos que los métodos diseñados en esta Sección necesitan menos tiempo para alcanzar los resultados mostrados en la Tabla 4.2. Esto también se puede comprobar en la Figura 4.1 que presenta las gráficas de los tiempos obtenidos con todos los métodos presentados en la Tabla

4.2 para las mismas funciones utilizadas en las secciones anteriores, f_1 , f_5 , f_9 , f_{13} y f_{17} . En el caso del método de Newton que tiene mayor índice computacional, igual que el método de Ostrowski, tenemos mayor tiempo computacional para alcanzar los resultados presentados en la Tabla 4.2 por la mayor cantidad de iteraciones que estos métodos necesitan.

Tabla 4.2: Ejemplos numéricos

	Parámetros	ξ	$ x_{k+1} - x_k $	$ f(x_{k+1}) $	Iter	ρ	$e - time$
f_1 , $x_0 = 1$	M7A	1.409624	$6.4e-155$	0	4	7.0000	0.2036
	M8A	1.409624	$3.4e-175$	0	4	7.9997	0.2560
	M8B	1.409624	$1.2e-226$	0	4	8.0000	0.2425
	M8C	1.409624	$1.1e-182$	0	4	7.9998	0.2341
	M8D	1.409624	$2.2e-220$	0	4	8.0000	0.2418
	NC	1.409624	$1.9e-273$	0	10	2.0000	0.2933
	OS	1.409624	$7.3e-139$	0	5	4.0000	0.1977
	LW8	1.409624	0	0	4	8.1069	0.2435
	BRW8	1.409624	$8.7e-239$	0	4	7.9997	0.2524
f_2 , $x_0 = 1$	M7A	0.257530	$2.2e-320$	0	4	7.0000	0.2798
	M8A	0.257530	$8.6e-058$	0	3	8.0948	0.2526
	M8B	0.257530	$2.7e-058$	0	3	7.9464	0.2353
	M8C	0.257530	$2.1e-058$	0	3	7.8333	0.2367
	M8D	0.257530	$2.5e-059$	0	3	7.8536	0.2460
	NC	0.257530	$4.5e-190$	0	8	2.0000	0.2721
	OS	0.257530	$1.8e-139$	0	5	4.0000	0.2659
	LW8	0.257530	$1.1e-201$	0	3	8.1108	0.2580
	BRW8	0.257530	$1.5e-060$	0	3	7.8561	0.2567
f_3 , $x_0 = 1$	M7A	0.739085	$2.6e-053$	0	3	6.9490	0.1711
	M8A	0.739085	$4.0e-067$	0	3	7.9431	0.2102
	M8B	0.739085	$5.3e-082$	0	3	7.8774	0.2031
	M8C	0.739085	$1.1e-067$	0	3	7.9369	0.1785
	M8D	0.739085	$2.1e-071$	0	3	7.9457	0.2067
	NC	0.739085	$7.1e-177$	0	8	2.0000	0.2405
	OS	0.739085	$4.2e-296$	0	5	4.0000	0.2295
	LW8	0.739085	$7.7e-286$	0	3	8.0668	0.1797
	BRW8	0.739085	$3.3e-083$	0	3	7.8957	0.1923
f_4 , $x_0 = 2.5$	M7A	2.000000	$1.8e-143$	0	4	6.9999	0.2264
	M8A	2.000000	$3.1e-184$	0	4	7.9998	0.2782
	M8B	2.000000	$4.2e-219$	0	4	8.0000	0.2593
	M8C	2.000000	$1.1e-193$	0	4	7.9999	0.2530
	M8D	2.000000	$1.9e-212$	0	4	8.0000	0.2689
	NC	2.000000	$7.9e-224$	0	10	2.0000	0.3104
	OS	2.000000	$4.1e-123$	0	5	4.0000	0.2288
	LW8	2.000000	0	0	5	7.8971	0.2892
	BRW8	2.000000	$2.5e-300$	0	4	8.0000	0.3052

Tabla 4.2 Ejemplos numéricos, continuación...

	Parámetros	ξ	$ x_{k+1} - x_k $	$ f(x_{k+1}) $	Iter	ρ	$e - time$
$f_5,$ $x_0 = 2$	M7A	2.154435	$7.7e-054$	0	3	7.0215	0.2264
	M8A	2.154435	$1.3e-064$	0	3	8.0327	0.1902
	M8B	2.154435	$1.7e-071$	0	3	8.0238	0.1890
	M8C	2.154435	$2.4e-066$	0	3	8.0302	0.1849
	M8D	2.154435	$1.2e-070$	0	3	8.0236	0.1721
	NC	2.154435	$4.5e-288$	0	9	2.0000	0.2616
	OS	2.154435	$1.1e-303$	0	5	4.0000	0.2125
	LW8	2.154435	$7.4e-288$	0	3	8.0065	0.1942
BRW8	2.154435	$2.3e-070$	0	3	8.0409	0.2054	
$f_6,$ $x_0 = 1$	M7A	0.639154	$8.7e-225$	0	4	7.0000	0.2507
	M8A	0.639154	$7.5e-280$	0	4	8.0000	0.3516
	M8B	0.639154	$9.2e-043$	0	3	7.9821	0.2244
	M8C	0.639154	$1.7e-305$	0	4	8.0000	0.2870
	M8D	0.639154	$3.1e-042$	0	3	7.9828	0.2325
	NC	0.639154	$2.3e-303$	0	10	2.0000	0.3829
	OS	0.639154	$3.9e-187$	0	5	4.0000	0.3654
	LW8	0.639154	$6.7e-172$	0	3	7.9880	0.3358
BRW8	0.639154	$1.8e-271$	0	3	8.0000	0.3003	
$f_7,$ $x_0 = 1$	M7A	0.767653	$6.4e-244$	0	4	7.0000	0.2536
	M8A	0.767653	$1.2e-041$	0	3	7.8446	0.2410
	M8B	0.767653	$9.7e-047$	0	3	7.8868	0.1994
	M8C	0.767653	$7.3e-043$	0	3	7.8517	0.2008
	M8D	0.767653	$7.4e-046$	0	3	7.8831	0.2189
	NC	0.767653	$1.4e-190$	0	9	2.0000	0.3170
	OS	0.767653	$1.6e-200$	0	5	4.0000	0.2922
	LW8	0.767653	$3.1e-191$	0	3	7.9730	0.3020
BRW8	0.767653	$1.1e-053$	0	3	7.8135	0.2456	
$f_8,$ $x_0 = 1$	M7A	1.365230	$5.4e-222$	0	4	7.0000	0.2899
	M8A	1.365230	$7.6e-275$	0	4	8.0000	0.3388
	M8B	1.365230	$9.8e-042$	0	3	8.1362	0.2597
	M8C	1.365230	$4.4e-287$	0	4	8.0000	0.3242
	M8D	1.365230	$4.9e-324$	0	4	7.9922	0.3224
	NC	1.365230	$4.0e-176$	0	9	2.0000	0.4000
	OS	1.365230	$1.5e-187$	0	5	4.0000	0.3112
	LW8	1.365230	$4.1e-169$	0	3	8.0336	0.3468
BRW8	1.365230	$9.9e-324$	0	4	7.9971	0.3611	
$f_9,$ $x_0 = 1$	M7A	0.128077	$6.7e-187$	0	4	7.0000	0.3073
	M8A	0.128077	$5.7e-231$	0	4	8.0000	0.3836
	M8B	0.128077	$2.0e-277$	0	4	8.0000	0.3316
	M8C	0.128077	$2.1e-234$	0	4	8.0000	0.3202
	M8D	0.128077	$3.6e-269$	0	4	8.0000	0.3380
	NC	0.128077	$9.0e-169$	0	9	2.0000	0.4455
	OS	0.128077	$8.3e-163$	0	5	4.0000	0.2704
	LW8	0.128077	0	0	4	8.0575	0.5445
BRW8	0.128077	$4.3e-041$	0	3	8.4428	0.3250	

Tabla 4.2 Ejemplos numéricos, continuación....

	Parámetros	ξ	$ x_{k+1} - x_k $	$ f(x_{k+1}) $	$Iter$	ρ	$e - time$
$f_{10},$ $x_0 = 0.5$	M7A	0.000000	$2.3e-307$	0	4	9.0000	0.1957
	M8A	0.000000	$1.9e-040$	0	3	9.0588	0.1791
	M8B	0.000000	$1.2e-042$	0	3	8.9356	0.1612
	M8C	0.000000	$2.6e-040$	0	3	9.1985	0.1560
	M8D	0.000000	$1.8e-049$	0	3	8.9484	0.1647
	NC	0.000000	$3.3e-289$	0	7	3.0000	0.2206
	OS	0.000000	$4.1e-302$	0	5	5.0000	0.2567
	LW8	0.000000	$4.8e-207$	0	3	8.9388	0.2302
	BRW8	<i>n.c.</i>	<i>n.c.</i>	<i>n.c.</i>	<i>n.c.</i>	<i>n.c.</i>	<i>n.c.</i>
$f_{11},$ $x_0 = 1$	M7A	0.714806	$2.2e-223$	0	4	7.0000	0.3143
	M8A	0.714806	$1.5e-306$	0	4	8.0000	0.3359
	M8B	0.714806	$9.1e-044$	0	3	7.8115	0.1966
	M8C	0.714806	$2.5e-311$	0	4	8.0000	0.3671
	M8D	0.714806	$2.8e-042$	0	3	7.8073	0.2226
	NC	0.714806	$5.3e-190$	0	9	2.0000	0.3829
	OS	0.714806	$3.6e-184$	0	5	4.0000	0.3654
	LW8	0.714806	$3.5e-191$	0	3	7.9076	0.3358
	BRW8	0.714806	$1.5e-051$	0	3	7.8565	0.2832
$f_{12},$ $x_0 = 0.5$	M7A	0.000000	$8.8e-162$	0	205	1.0000	16.7385
	M8A	0.000000	$4.1e-162$	0	204	1.0000	23.0236
	M8B	0.000000	$3.2e-162$	0	187	1.0000	17.1699
	M8C	0.000000	$4.4e-162$	0	200	1.0000	19.5139
	M8D	0.000000	$6.4e-162$	0	194	1.0000	20.3596
	NC	0.000000	$1.7e-162$	0	536	1.0000	21.0063
	OS	0.000000	$6.0e-162$	0	266	1.0000	18.4757
	LW8	<i>n.c.</i>	<i>n.c.</i>	<i>n.c.</i>	<i>n.c.</i>	<i>n.c.</i>	<i>n.c.</i>
	BRW8	<i>n.c.</i>	<i>n.c.</i>	<i>n.c.</i>	<i>n.c.</i>	<i>n.c.</i>	
$f_{13},$ $x_0 = -1$	M7A	-1.20158	$1.2e-047$	0	3	6.9142	0.2781
	M8A	-1.20158	$1.3e-282$	0	4	8.0000	0.4593
	M8B	-1.20158	$2.7e-061$	0	3	7.9404	0.3194
	M8C	-1.20158	$5.1e-054$	0	3	7.9348	0.3225
	M8D	-1.20158	$3.2e-061$	0	3	7.9386	0.3374
	NC	-1.20158	$9.6e-278$	0	10	2.0000	0.5465
	OS	-1.20158	$2.3e-266$	0	5	4.0000	0.4954
	LW8	-1.20158	$1.1e-259$	0	3	7.9547	0.5051
	BRW8	-1.20158	$1.3e-240$	0	4	8.0000	0.4639
$f_{14},$ $x_0 = 2$	M7A	2.331968	$7.7e-059$	0	3	6.8916	0.4778
	M8A	2.331968	$5.8e-082$	0	3	7.8912	0.6528
	M8B	2.331968	$8.1e-072$	0	3	7.9282	0.5530
	M8C	2.331968	$9.0e-076$	0	3	7.8873	0.5547
	M8D	2.331968	$7.3e-078$	0	3	7.9075	0.5363
	NC	2.331968	$3.7e-211$	0	8	2.0000	0.7165
	OS	2.331968	$3.5e-323$	0	5	3.9998	0.7312
	LW8	2.331968	$1.1e-280$	0	5	8.0286	0.8091
	BRW8	2.331968	$4.6e-083$	0	3	8.0028	0.5697

Tabla 4.2 Ejemplos numéricos, continuación...

	Parámetros	ξ	$ x_{k+1} - x_k $	$ f(x_{k+1}) $	<i>Iter</i>	ρ	<i>e - time</i>
$f_{15},$ $x_0 = 1$	M7A	1.41424	$1.4e-175$	0	4	7.0002	0.4778
	M8A	1.41424	$2.5e-203$	0	4	8.0000	0.6528
	M8B	1.41424	$4.1e-236$	0	4	8.0001	0.5530
	M8C	1.41424	$1.0e-270$	0	4	8.0000	0.5547
	M8D	1.41424	$2.2e-042$	0	3	7.6465	0.5363
	NC	1.41424	$2.9e-215$	0	9	2.0000	0.4989
	OS	1.41424	$7.0e-156$	0	5	4.0000	0.6759
	LW8	1.41424	0	0	4	8.0235	0.7447
	BRW8	1.41424	$7.5e-256$	0	3	8.0000	0.5697
$f_{16},$ $x_0 = 2$	M7A	1.679631	$8.5e-127$	0	4	7.0003	0.5139
	M8A	1.679631	$2.4e-124$	0	4	7.9978	0.5542
	M8B	1.679631	$7.1e-183$	0	4	7.9998	0.4920
	M8C	1.679631	$2.1e-137$	0	4	7.9988	0.4507
	M8D	1.679631	$3.3e-178$	0	3	7.9998	0.4766
	NC	1.679631	$4.7e-219$	0	10	2.0000	0.6829
	OS	1.679631	$9.6e-118$	0	5	4.0000	0.4352
	LW8	1.679631	0	0	5	8.1613	0.7079
	BRW8	1.679631	$5.3e-164$	0	4	7.9961	0.5696
$f_{17},$ $x_0 = 2$	M7A	1.7461395	$1.8e-049$	0	3	7.0836	0.2059
	M8A	1.7461395	$5.5e-062$	0	3	8.0834	0.3706
	M8B	1.7461395	$5.3e-078$	0	3	7.9182	0.2006
	M8C	1.7461395	$3.9e-062$	0	3	8.1048	0.2400
	M8D	1.7461395	$1.9e-066$	0	3	8.0930	0.3245
	NC	1.7461395	$9.6e-170$	0	8	2.0000	0.2503
	OS	1.7461395	$9.0e-280$	0	5	4.0000	0.2799
	LW8	1.7461395	$5.9e-273$	0	3	7.9370	0.2134
	BRW8	1.7461395	$2.7e-080$	0	3	7.9460	0.2804
$f_{18},$ $x_0 = -1.8$	M7A	≈ -2	$3.4e-219$	0	4	7.0000	0.9358
	M8A	≈ -2	$8.4e-300$	0	4	8.0000	1.0007
	M8B	≈ -2	$1.0e-295$	0	3	8.6091	0.6777
	M8C	≈ -2	$1.1e-041$	0	4	8.0000	0.9579
	M8D	≈ -2	$1.9e-066$	0	3	8.3082	0.7885
	NC	≈ -2	$1.1e-239$	0	9	2.0000	1.5642
	OS	≈ -2	$3.1e-188$	0	5	4.0000	1.7762
	LW8	≈ -2	$1.5e-179$	0	3	7.9840	1.7753
	BRW8	≈ -2	$7.3e-051$	0	3	8.1341	0.8311

MÉTODOS ITERATIVOS PARA SISTEMAS NO LINEALES

La búsqueda de soluciones de los sistemas de ecuaciones no lineales es un problema antiguo y difícil con amplias aplicaciones en matemáticas e ingeniería. Los métodos de resolución más comúnmente utilizados son iterativos: a partir de una o varias aproximaciones iniciales, se construye una secuencia, de tal manera que converge a una solución de las ecuaciones (véase [4]). Un estudio detallado de las técnicas de construcción de funciones de iteración punto a punto se puede encontrar en el texto de Traub [81]. En resumen la idea del desarrollo de ésta técnica iterativa para la aproximación del cero ξ de una función $F : D \in \mathbb{R}^n \rightarrow \mathbb{R}^n$ consiste en la aproximación de la función inversa de F , que llamamos G . Aproximando la función inversa G de distintas maneras se obtienen diferentes métodos iterativos. Por ejemplo, si utilizamos el polinomio de Taylor de segundo orden, se obtiene el método Tschebyshev-Euler [6, 43]:

$$\begin{aligned} y^{(k)} &= x^{(k)} - J_F(x^{(k)})^{-1}F(x^{(k)}), \\ x^{(k+1)} &= y^{(k)} - \frac{1}{2}J_F(x^{(k)})H_k(x^{(k)})(y^{(k)} - x^{(k)})^2, \end{aligned}$$

donde $H_F(x^{(k)})$ es la matriz Hessiana evaluada en el iterado $x^{(k)}$.

En [16, 17], Cordero y Torregrosa aplicando las fórmulas de cuadratura cerradas y abiertas, desarrollan familias de variantes del método de Newton con el fin de mejorar el orden de convergencia. Estas familias de orden tres incluyen las familias de métodos definidos por Frontini et al. en [38]. En [18], los mismos autores utilizando la fórmula genérica de cuadratura interpolatoria para obtener una familia de métodos, variantes de Newton para sistemas no lineales muestran que el orden de convergencia bajo ciertas condiciones es $2d + 1$, donde d es el orden hasta el cual de las derivadas parciales de cada función coordinada, evaluadas en la solución, se anulan.

Por otra parte, existe abundante literatura acerca de la convergencia local y semilocal en espacios de Banach de diferentes métodos, variantes de los esquemas de Chebyshev y Halley (véanse, por ejemplo, [32]-[33] y [43]).

Además de métodos multipaso basados en cuadratura interpolatoria se han desarrollado otros utilizando el método de descomposición de Adomian ([2], por ejemplo), como el método propuesto por Darvishi y Barati en [28] y [29] con convergencia super cúbica y los métodos propuestos por Cordero et al. en [19] con orden de convergencia 4 y 5. Otra técnica de desarrollo de métodos iterativos para sistemas no lineales es la sustitución de la segunda derivada por alguna aproximación. En su libro Traub presentó una familia de métodos de dos puntos utilizando como base el método de Tschebyshev y aproximando la segunda derivada que aparece en su fórmula iterativa por una diferencia finita entre las evaluaciones de la primera derivada en dos puntos distintos (véase también [40], [49]).

Otra técnica conocida para acelerar la convergencia de los métodos iterativos consiste en la composición de dos métodos iterativos de los órdenes p_1 y p_2 , respectivamente, para obtener un método de orden p_1p_2 ([81] y [22]). Por lo general, los procedimientos de aceleración de los métodos iterativos para sistemas de ecuaciones no lineales requieren nuevas evaluaciones de la matriz Jacobiana y/o de la función no lineal. Ultimamente, en muchas aplicaciones numéricas se usa alta precisión de los cálculos. Los resultados de los experimentos numéricos muestran que los métodos de orden superior asociados con la precisión de la aritmética de coma flotante es muy útil, ya que se reduce el número de iteraciones necesarias.

Esta parte de la memoria esta dedicada al desarrollo de métodos iterativos para sistemas de ecuaciones no lineales. En primer lugar, en la Sección 5.1, se describen los métodos iterativos provenientes de la cuadratura gaussiana con distintos predictores con el fin de acelerar la convergencia. Además se desarrollan nuevos métodos buscando elevado orden de convergencia y menor número de evaluaciones de la matriz Jacobiana y/o de la función no lineal.

5.1. Diseño de métodos iterativos empleando la cuadratura de Gauss

En esta sección desarrollamos en primer lugar el conjunto de familias de métodos provenientes de la cuadratura de Gauss que emplean como predictor el método de Newton, demostrando que la convergencia de las variantes es de orden tres, con condiciones específicas impuestas a los polinomios ortogonales que definen la cuadratura gaussiana correspondiente y de orden cinco dependiendo del comportamiento de las derivadas parciales segundas de las funciones que definen el sistema de ecuaciones no lineales. Además, sustituyendo al método de Newton como predictor por el método de Traub demostramos que la convergencia aumenta a orden cinco bajo las mismas condiciones impuestas sobre los polinomios ortogonales que definen la cuadratura gaussiana correspondiente. Dependiendo del comportamiento de las derivadas parciales segundas de las funciones que definen el sistema de ecuaciones no lineales, el orden aumenta hasta nueve ó hasta once si imponemos condiciones adicionales a los polinomios ortogonales que definen cada cuadratura gaussiana.

Sea $F : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ una función suficientemente diferenciable en el conjunto convexo $\Omega \in \mathbb{R}^n$ y sea ξ un cero del sistema de ecuaciones no lineales $F(x) = 0$. Para cualesquiera $x, y \in \Omega$, el teorema fundamental del cálculo integral nos asegura:

$$F(y) = F(x) + F'(x)(y-x) + \frac{1}{2!}F''(x)(y-x)^2 + \dots + \frac{1}{(r-1)!}F^{(r-1)}(x)(y-x)^{r-1} + \int_0^1 \frac{(1-t)^{r-1}}{r!}F^{(r)}(x-t(y-x))(y-x)^r dt.$$

Entonces, para $r = 1$ tenemos:

$$F(y) = F(x) + \int_0^1 F'(x-t(y-x))(y-x) dt,$$

de donde se obtiene para la k -ésima iteración $x^{(k)}$:

$$F(y) = F(x^{(k)}) + \int_0^1 F'(x^{(k)} + t(y-x^{(k)}))(y-x^{(k)}) dt, \quad (5.1)$$

donde $F'(x^{(k)} + t(y-x^{(k)})) = J_F(x^{(k)} + t(y-x^{(k)}))$ es la matriz Jacobiana asociada a F . En adelante denotaremos por $J_F(x^{(k)})$ a la matriz Jacobiana evaluada en el iterado $x^{(k)}$. Si estimamos $J_F(x^{(k)} + t(y-x^{(k)}))$ en el intervalo $[0, 1]$ por su valor en $t = 0$, es decir, por $J_F(x^{(k)})$ y tomamos $y = \xi$, entonces: $0 \approx F(x^{(k)}) + J_F(x^{(k)})(\xi - x^{(k)})$, y se puede dar una nueva aproximación de la solución mediante $x^{(k+1)} = x^{(k)} - [J_F(x^{(k)})]^{-1} F(x^{(k)})$, que es el método clásico de Newton (NC) para $k = 0, 1, \dots$

Frontini y Sormani en [38] aproximaron la integral en (5.1) por la fórmula general de la cuadratura interpolatoria de orden superior a cero, es decir,

$$\int_0^1 J_F(x-t(y-x))(y-x) dt \approx \sum_{i=0}^m \omega_i J_F(\eta_i^{(k)})(y-x),$$

con $\eta_i^{(k)} = x^{(k)} - \tau_i(z^{(k)} - x^{(k)})$, $\tau_i \in [0, 1]$, donde para evitar el problema implícito se usa el método de Newton clásico como predictor, $z^{(k)} = x^{(k)} - [J_F(x^{(k)})]^{-1} F(x^{(k)})$, y obtuvieron los siguientes métodos iterativos de orden tres:

$$x^{(k+1)} = x^{(k)} - \left[\sum_{i=0}^m \omega_i J_F(x^{(k)} - \tau_i(z^{(k)} - x^{(k)})) \right]^{-1} F(x^{(k)}). \quad (5.2)$$

Otros trabajos relacionados con esta técnica, de diferentes autores, son [20], [27], [30]. Basándonos en este método y en los propuestos en [16] y [17] desarrollamos el conjunto de familias de métodos que usan la cuadratura de Gauss para aproximar la integral (5.1).

Tal como hicimos en Sección 3.2, para el caso unidimensional, aproximamos la integral (5.1) mediante la cuadratura de Gauss y tomando $y = \xi$, obtenemos:

$$0 \approx F(x^{(k)}) + \frac{\xi - x^{(k)}}{2} \sum_{i=1}^m \omega_i J_F \left(\frac{(\xi - x^{(k)}) \tau_i + (\xi + x^{(k)})}{2} \right)$$

donde ω_i y τ_i son los pesos y los nodos respectivamente del polinomio ortogonal de grado m que define la cuadratura gaussiana correspondiente, y la nueva aproximación $x^{(k+1)}$ a ξ será:

$$x^{(k+1)} = x^{(k)} - 2 \left[\sum_{i=1}^m \omega_i J_F \left(\frac{(1 + \tau_i) x^{(k+1)} + (1 - \tau_i) x^{(k)}}{2} \right) \right]^{-1} F(x^{(k)}).$$

Con el fin de evitar el problema implícito que esta ecuación conlleva, se debe usar la k -ésima iteración $z^{(k)}$ obtenida por algún método predictor. Finalmente, la fórmula iterativa genérica:

$$x^{(k+1)} = x^{(k)} - 2 \left[\sum_{i=1}^m \omega_i J_F \left(\eta_i^{(k)} \right) \right]^{-1} F(x^{(k)}), \quad (5.3)$$

con $\eta_i^{(k)} = \frac{(1 + \tau_i) z^{(k)} + (1 - \tau_i) x^{(k)}}{2}$ y con la iteración $z^{(k)}$ obtenida por algún método predictor. Trabajaremos con diferentes predictores con el fin de tratar de establecer alguna secuencia en el comportamiento de los métodos resultantes, tal como ocurría en el caso de una variable (véase Capítulo 3).

5.1.1. Métodos iterativos provenientes de la cuadratura de Gauss con predictor el método de Newton

En este caso usamos para predecir $x^{(k+1)}$ la k -ésima iteración obtenida por el método clásico de Newton

$$\begin{aligned} z^{(k)} &= x^{(k)} - \left[J_F(x^{(k)}) \right]^{-1} F(x^{(k)}), \\ x^{(k+1)} &= x^{(k)} - 2 \left[\sum_{i=1}^m \omega_i J_F \left(\eta_i^{(k)} \right) \right]^{-1} F(x^{(k)}). \end{aligned} \quad (5.4)$$

Este resultado representa la fórmula genérica del conjunto de familias de variantes Newton-Gauss para sistema de ecuaciones no lineales.

Usando diferentes cuadraturas (Gauss-Tschebyshev, Gauss-Legendre, Gauss-Lobatto y Gauss-Radau) y teniendo en cuenta el predictor elegido obtenemos cuatro subfamilias de las variantes del método de Newton.

Caso I: Cuadratura de Gauss-Tschebyshev. Usando un solo nodo obtenemos la fórmula de iteración del método Newton-Gauss-Tschebyshev para sistemas de ecuaciones no lineales con un nodo (NeG-TS1):

$$x^{(k+1)} = x^{(k)} - \frac{2}{\pi} \left[J_F \left(\frac{z^{(k)} + x^{(k)}}{2} \right) \right]^{-1} F(x^{(k)}). \quad (5.5)$$

Si usamos dos nodos obtenemos (NeG-TS2)

$$x^{(k+1)} = x^{(k)} - \frac{4\sqrt{2}}{\pi} \left[J_F \left(\frac{2 + \sqrt{2}}{4} z^{(k)} + \frac{2 - \sqrt{2}}{4} x^{(k)} \right) + J_F \left(\frac{2 - \sqrt{2}}{4} z^{(k)} + \frac{2 + \sqrt{2}}{4} x^{(k)} \right) \right]^{-1} F(x^{(k)}) \quad (5.6)$$

y en el caso de tres nodos tendremos (NeG-TS3)

$$x^{(k+1)} = x^{(k)} - \frac{12}{\pi} \left[\sqrt{3} J_F \left(\frac{3z^{(k)} + x^{(k)}}{4} \right) + 2J_F(z^{(k)}) + \sqrt{3} J_F \left(\frac{z^{(k)} + 3x^{(k)}}{4} \right) \right]^{-1} F(x^{(k)}). \quad (5.7)$$

Caso II: Cuadratura de Gauss-Legendre. El proceso de obtención de la fórmula de iteración es el mismo que en el caso anterior. Usando un sólo nodo obtenemos la fórmula iterativa NeG-LDS1:

$$x^{(k+1)} = x^{(k)} - \left[J_F \left(\frac{z^{(k)} + x^{(k)}}{2} \right) \right]^{-1} F(x^{(k)}) \quad (5.8)$$

que coincide con el método de punto medio (véase [16]). En caso de dos nodos (NeG-LDS2) tenemos:

$$x^{(k+1)} = x^{(k)} - 2 \left[J_F \left(\frac{3 + \sqrt{3}}{6} x^{(k)} + \frac{3 - \sqrt{3}}{6} z^{(k)} \right) + J_F \left(\frac{3 - \sqrt{3}}{6} x^{(k)} + \frac{3 + \sqrt{3}}{6} z^{(k)} \right) \right]^{-1} F(x^{(k)}). \quad (5.9)$$

La fórmula iterativa del método que llamamos Newton-Gauss-Legendre para sistemas de tres nodos (NeG-LDS3) tendrá la forma:

$$\begin{aligned} x^{(k+1)} = x^{(k)} & - 18 \left[5J_F \left(\frac{5 - \sqrt{15}}{10} z^{(k)} + \frac{5 + \sqrt{15}}{10} x^{(k)} \right) \right. \\ & \left. - 8J_F \left(\frac{x^{(k)} + z^{(k)}}{2} \right) + 5J_F \left(\frac{5 + \sqrt{15}}{2} z^{(k)} + \frac{5 - \sqrt{15}}{2} x^{(k)} \right) \right]^{-1} F(x^{(k)}). \end{aligned} \quad (5.10)$$

Caso III: Cuadratura de Gauss-Lobatto. De la misma manera obtenemos la fórmula de iteración para un sólo nodo NeG-LTS1

$$x^{(k+1)} = x^{(k)} - \left[J_F \left(\frac{z^{(k)} + x^{(k)}}{2} \right) \right]^{-1} F(x^{(k)}), \quad (5.11)$$

que coincide de nuevo con el método de punto medio. En el caso de dos nodos obtenemos la variante NeG-LTS2:

$$x^{(k+1)} = x^{(k)} - \left[J_F(x^{(k)}) + J_F(z^{(k)}) \right]^{-1} F(x^{(k)}), \quad (5.12)$$

que coincide con el método de trapecios (véase [16]). Para tres nodos obtenemos la variante NeG-LTS3:

$$x^{(k+1)} = x^{(k)} - 6 \left[J_F(x^{(k)}) + 4J_F \left(\frac{z^{(k)} + x^{(k)}}{2} \right) + J_F(z^{(k)}) \right]^{-1} F(x^{(k)}), \quad (5.13)$$

que coincide con el método de Newton-Simpson (véase [17]).

Caso IV: Cuadratura de Gauss-Radau. Las fórmulas de iteración de las variantes del método de Newton que llamaremos Newton-Gauss-Radau para sistemas (NeG-RS) para los casos con uno, dos y tres nodos serán el método clásico de Newton para sistemas de ecuaciones no lineales (NC), NeG-RS2 y NeG-RS3, respectivamente:

$$x^{(k+1)} = x^{(k)} - \left[J_F(x^{(k)}) \right]^{-1} F(x^{(k)}), \quad (5.14)$$

$$x^{(k+1)} = x^{(k)} - 4 \left[J_F(x^{(k)}) + 3J_F \left(\frac{2z^{(k)} + x^{(k)}}{3} \right) \right]^{-1} F(x^{(k)}), \quad (5.15)$$

$$\begin{aligned} x^{(k+1)} = x^{(k)} & - 18 \left[(6 + \sqrt{16}) J_F \left(\frac{1 - \sqrt{6}}{10} z^{(k)} + \frac{1 + \sqrt{6}}{10} x^{(k)} \right) \right. \\ & \left. + 4J_F(x^{(k)}) + (6 - \sqrt{16}) J_F \left(\frac{1 + \sqrt{6}}{10} z^{(k)} + \frac{1 - \sqrt{6}}{10} x^{(k)} \right) \right]^{-1} F(x^{(k)}). \end{aligned} \quad (5.16)$$

Para estudiar el orden de convergencia y las condiciones que lo definen consideramos $x \in \mathbb{R}^n$, $n > 1$ y denotamos por $J_{ij}(x)$ la (i, j) -ésima entrada de la matriz Jacobiana, por H_{ij} la (i, j) -ésima entrada de su inversa y por $f_j(x)$, $j = 1, 2, \dots, n$ las funciones coordenadas de $F(x)$ (técnica definida en la Sección 2.1.2). Entonces,

$$\sum_{i=1}^n H_{ji}(x) J_{ik}(x) = \delta_{jk}. \quad (5.17)$$

Mediante derivación directa de (5.17) obtenemos las siguientes igualdades:

$$\begin{aligned} \sum_{i=1}^n \frac{\partial H_{ji}(x)}{\partial x_l} \frac{\partial f_i(x)}{\partial x_r} &= - \sum_{i=1}^n H_{ji}(x) \frac{\partial^2 f_i(x)}{\partial x_r \partial x_l}, \\ \sum_{i=1}^n \frac{\partial^2 H_{ji}(x)}{\partial x_s \partial x_l} \frac{\partial f_i(x)}{\partial x_r} &= - \sum_{i=1}^n \frac{\partial H_{ji}(x)}{\partial x_l} \frac{\partial^2 f_i(x)}{\partial x_s \partial x_r} - \sum_{i=1}^n \frac{\partial H_{ji}(x)}{\partial x_s} \frac{\partial^2 f_i(x)}{\partial x_r \partial x_l} \\ &\quad - \sum_{i=1}^n H_{ji}(x) \frac{\partial^3 f_i(x)}{\partial x_s \partial x_r \partial x_l}, \\ \sum_{i=1}^n \frac{\partial^3 H_{ji}(x)}{\partial x_u \partial x_s \partial x_l} \frac{\partial f_i(x)}{\partial x_r} &= - \sum_{i=1}^n \frac{\partial^2 H_{ji}(x)}{\partial x_s \partial x_l} \frac{\partial^2 f_i(x)}{\partial x_u \partial x_r} - \sum_{i=1}^n \frac{\partial^2 H_{ji}(x)}{\partial x_u \partial x_l} \frac{\partial^2 f_i(x)}{\partial x_s \partial x_r} \\ &\quad - \sum_{i=1}^n \frac{\partial^2 H_{ji}(x)}{\partial x_u \partial x_s} \frac{\partial^2 f_i(x)}{\partial x_r \partial x_l} - \sum_{i=1}^n \frac{\partial H_{ji}(x)}{\partial x_l} \frac{\partial^3 f_i(x)}{\partial x_u \partial x_s \partial x_r} \\ &\quad - \sum_{i=1}^n \frac{\partial H_{ji}(x)}{\partial x_s} \frac{\partial^3 f_i(x)}{\partial x_u \partial x_r \partial x_l} - \sum_{i=1}^n \frac{\partial H_{ji}(x)}{\partial x_u} \frac{\partial^3 f_i(x)}{\partial x_s \partial x_r \partial x_l} \\ &\quad - \sum_{i=1}^n \frac{\partial H_{ji}(x)}{\partial x_l} \frac{\partial^4 f_i(x)}{\partial x_u \partial x_s \partial x_r \partial x_l}. \end{aligned}$$

con las que se pueden demostrar fácilmente los lemas siguientes:

Lema 5.1.1 *Sea $\lambda_j(x)$ la j -ésima componente de la función de iteración del método clásico de Newton*

$$\lambda_j(x) = x_j - \sum_{i=1}^n H_{ji}(x) f_i(x)$$

para $j = 1, \dots, n$ y sea ξ solución de $F(x) = (f_1(x), \dots, f_n(x))^T$. Entonces,

$$\frac{\partial \lambda_j(\xi)}{\partial x_l} = 0, \quad (5.18)$$

$$\frac{\partial^2 \lambda_j(\xi)}{\partial x_r \partial x_l} = \sum_{i=1}^n H_{ji}(\xi) \frac{\partial^2 f_i(\xi)}{\partial x_r \partial x_l}, \quad (5.19)$$

$$\begin{aligned} \frac{\partial^3 \lambda_j(\xi)}{\partial x_s \partial x_r \partial x_l} &= \sum_{i=1}^n \frac{\partial H_{ji}(\xi)}{\partial x_r} \frac{\partial^2 f_i(\xi)}{\partial x_s \partial x_l} + \frac{\partial H_{ji}(\xi)}{\partial x_s} \frac{\partial^2 f_i(\xi)}{\partial x_r \partial x_l} + \frac{\partial H_{ji}(\xi)}{\partial x_l} \frac{\partial^2 f_i(\xi)}{\partial x_s \partial x_r} \\ &\quad + 2 \sum_{i=1}^n H_{ji}(\xi) \frac{\partial^3 f_i(\xi)}{\partial x_s \partial x_r \partial x_l}, \end{aligned} \quad (5.20)$$

$$\begin{aligned}
\frac{\partial^4 \lambda_j(\xi)}{\partial x_u \partial x_s \partial x_r \partial x_l} &= \sum_{i=1}^n \frac{\partial H_{ji}^2(\xi)}{\partial x_s \partial x_l} \frac{\partial^2 f_i(\xi)}{\partial x_u \partial x_r} + \sum_{i=1}^n \frac{\partial H_{ji}^2(\xi)}{\partial x_r \partial x_l} \frac{\partial^2 f_i(\xi)}{\partial x_u \partial x_s} + \sum_{i=1}^n \frac{\partial H_{ji}^2(\xi)}{\partial x_s \partial x_r} \frac{\partial^2 f_i(\xi)}{\partial x_u \partial x_l} \\
&+ \sum_{i=1}^n \frac{\partial H_{ji}^2(\xi)}{\partial x_u \partial x_r} \frac{\partial^2 f_i(\xi)}{\partial x_s \partial x_l} + \sum_{i=1}^n \frac{\partial H_{ji}^2(\xi)}{\partial x_u \partial x_l} \frac{\partial^2 f_i(\xi)}{\partial x_s \partial x_r} + \sum_{i=1}^n \frac{\partial H_{ji}^2(\xi)}{\partial x_u \partial x_s} \frac{\partial^2 f_i(\xi)}{\partial x_r \partial x_l} \\
&+ 2 \sum_{i=1}^n \frac{\partial H_{ji}(\xi)}{\partial x_r} \frac{\partial^3 f_i(\xi)}{\partial x_u \partial x_s \partial x_l} + 2 \sum_{i=1}^n \frac{\partial H_{ji}(\xi)}{\partial x_l} \frac{\partial^3 f_i(\xi)}{\partial x_u \partial x_r \partial x_s} \\
&+ 2 \sum_{i=1}^n \frac{\partial H_{ji}(\xi)}{\partial x_u} \frac{\partial^3 f_i(\xi)}{\partial x_s \partial x_r \partial x_l} + 2 \sum_{i=1}^n \frac{\partial H_{ji}(\xi)}{\partial x_s} \frac{\partial^3 f_i(\xi)}{\partial x_u \partial x_r \partial x_l} \\
&+ 3 \sum_{i=1}^n H_{ji}(\xi) \frac{\partial^4 f_i(\xi)}{\partial x_u \partial x_s \partial x_r \partial x_l},
\end{aligned} \tag{5.21}$$

para $i, j, l, r, s, u \in \{1, 2, \dots, n\}$.

Lema 5.1.2 Sea $\eta_k(x)$ la función de iteración

$$\eta_k(x) = x - \frac{1}{2}(1 - \tau_k)[J_F(x)]^{-1}F(x),$$

para $k = 1, \dots, m$, donde τ_k son los nodos de la fórmula de cuadratura gaussiana asociada a una familia de polinomios ortogonales tales que $\tau_k \in [0, 1]$, $k = 1, \dots, m$. Sea, además, ξ solución de $F(x) = 0$. Entonces,

$$\left. \frac{\partial(\eta_k(x))_q}{\partial x_l} \right|_{x=\xi} = \frac{1}{2}(1 - \tau_k)\delta_{ql}, \tag{5.22}$$

$$\left. \frac{\partial^2(\eta_k(x))_q}{\partial x_r \partial x_l} \right|_{x=\xi} = \frac{1}{2}(1 + \tau_k) \sum_{i=1}^n H_{qi}(\alpha) \frac{\partial^2 f_i(\xi)}{\partial x_r \partial x_l}, \tag{5.23}$$

$$\begin{aligned}
\left. \frac{\partial^3 \eta_k(x)_q}{\partial x_s \partial x_r \partial x_l} \right|_{x=\xi} &= \frac{1}{2}(1 + \tau_k) \sum_{i=1}^n \left[\frac{\partial H_{qi}(\xi)}{\partial x_l} \frac{\partial^2 f_i(\xi)}{\partial x_r \partial x_s} + \frac{\partial H_{qi}(\xi)}{\partial x_r} \frac{\partial^2 f_i(\xi)}{\partial x_l \partial x_s} \right] \\
&+ \frac{1}{2}(1 + \tau_k) \sum_{i=1}^n \left[\frac{\partial H_{qi}(\xi)}{\partial x_s} \frac{\partial^2 f_i(\xi)}{\partial x_r \partial x_l} + H_{qi}(\xi) \frac{\partial^3 f_i(\xi)}{\partial x_s \partial x_r \partial x_l} \right],
\end{aligned} \tag{5.24}$$

para $i, l, r, s \in \{1, 2, \dots, n\}$.

Lema 5.1.3 Sea $\eta_k(x)$ la función de iteración $\eta_k(x) = x - \frac{1}{2}(1 - \tau_k)[J_F(x)]^{-1}F(x)$ para $k = 1, \dots, m$ donde τ_k son las raíces correspondientes del polinomio ortogonal tales que pertenecen al intervalo $[0, 1]$. Entonces,

$$\left. \frac{\partial J_{ij}(\eta_k(x))}{\partial x_l} \right|_{x=\xi} = \frac{1}{2}(1 - \tau_k) \frac{\partial^2 f_i(\xi)}{\partial x_l \partial x_j}, \tag{5.25}$$

$$\left. \frac{\partial^2 J_{ij}(\eta_k(x))}{\partial x_r \partial x_l} \right|_{x=\xi} = \frac{1}{4}(1 - \tau_k)^2 \frac{\partial^3 f_i(\xi)}{\partial x_r \partial x_l \partial x_j} + \sum_{q=1}^n \left[\frac{1}{2}(1 + \tau_k) \right] \frac{\partial^2 f_i(\xi)}{\partial x_q \partial x_j} \sum_{p=1}^n H_{qp}(\xi) \frac{\partial^2 f_p(\xi)}{\partial x_r \partial x_l}, \tag{5.26}$$

$$\begin{aligned}
\left. \frac{\partial^3 J_{ij}(\eta_k(x))}{\partial x_s \partial x_r \partial x_l} \right|_{x=\xi} &= \frac{1}{8}(1-\tau_k)^3 \frac{\partial^4 f_i(\xi)}{\partial x_s \partial x_r \partial x_l \partial x_j} + \frac{1}{4} \sum_{q=1}^n \frac{\partial^3 f_i(\xi)}{\partial x_q \partial x_l \partial x_j} \sum_{p=1}^n H_{qp}(\xi) (1-\tau_k^2) \frac{\partial^2 f_p(\xi)}{\partial x_s \partial x_r} \\
&+ \frac{1}{4} \sum_{q=1}^n \frac{\partial^3 f_i(\xi)}{\partial x_q \partial x_r \partial x_j} \sum_{p=1}^n H_{qp}(\xi) (1-\tau_k^2) \frac{\partial^2 f_p(\xi)}{\partial x_s \partial x_l} \\
&+ \frac{1}{4} \sum_{q=1}^n \frac{\partial^3 f_i(\xi)}{\partial x_q \partial x_s \partial x_j} \sum_{p=1}^n H_{qp}(\xi) (1-\tau_k^2) \frac{\partial^2 f_p(\xi)}{\partial x_r \partial x_l} \\
&+ \frac{1}{2} \sum_{q=1}^n (1+\tau_k) \frac{\partial^2 f_p(\xi)}{\partial x_q \partial x_l} \sum_{p=1}^n \left[\frac{\partial H_{qp}(\xi)}{\partial x_r} \frac{\partial^2 f_i(\xi)}{\partial x_s \partial x_l} \frac{\partial H_{qp}(\xi)}{\partial x_s} \frac{\partial^2 f_i(\xi)}{\partial x_r \partial x_l} \right] \\
&+ \frac{1}{2} \sum_{q=1}^n (1+\tau_k) \frac{\partial^2 f_p(\xi)}{\partial x_q \partial x_j} \sum_{p=1}^n \left[\frac{\partial H_{qp}(\xi)}{\partial x_l} \frac{\partial^2 f_i(\xi)}{\partial x_s \partial x_r} + 2H_{qp}(\xi) \frac{\partial^3 f_i(\xi)}{\partial x_s \partial x_r \partial x_l} \right],
\end{aligned} \tag{5.27}$$

para $i, j, l, r, s \in \{1, 2, \dots, n\}$.

Usando los resultados anteriores, analizamos la convergencia de las familias Newton-Gauss descritas por (5.4).

Teorema 5.1.4 *Sea $F : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ suficientemente diferenciable en Ω y $\xi \in \Omega$ solución del sistema de ecuaciones no lineales $F(x) = 0$. Supongamos que $J_F(x)$ es continua y no singular en ξ . Entonces, la sucesión $\{x_k\}_{k \geq 0}$ obtenida usando la expresión (5.4) converge a ξ con convergencia de orden tres si los pesos ω_i y los nodos τ_i de la fórmula de cuadratura verifican*

$$\sum_{i=1}^m \omega_i = 2, \tag{5.28}$$

$$\sum_{i=1}^m \omega_i (1-\tau_i) = 2. \tag{5.29}$$

Si además $\frac{\partial^2 f_i(\xi)}{\partial a_1 \partial a_2} = 0$ para todo $a_1, a_2 \in \{1, \dots, n\}$ y

$$\sum_{i=1}^m \omega_i (1-\tau_i)^2 = \frac{8}{3}, \tag{5.30}$$

el orden de la convergencia es cuatro, mientras que el orden de convergencia será cinco si también se cumple

$$\sum_{i=1}^m \omega_i (1-\tau_i)^3 = 4. \tag{5.31}$$

Demostración: Consideramos que la solución $\xi \in \Omega$ del sistema de ecuaciones no lineales $F(x) = 0$ es el punto fijo de la función de iteración $G : \mathbb{R}^n \rightarrow \mathbb{R}^n$ descrita por (5.4). Denotamos por $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, n$, a las funciones coordenadas de G . Desarrollando $g_i(x)$ en serie de Taylor alrededor de ξ obtenemos:

$$g_i(x) = g_i(\xi) + \sum_{k=1}^n \frac{\partial g_i(\xi)}{\partial x_{a_k}} e_{a_k} + \sum_{a_1=1}^n \sum_{a_2=1}^n \frac{\partial^2 g_i(\xi)}{\partial x_{a_1} \partial x_{a_2}} e_{a_1} e_{a_2} + \frac{1}{6} \sum_{a_1=1}^n \sum_{a_2=1}^n \sum_{a_3=1}^n \frac{\partial^3 g_i(\xi)}{\partial x_{a_1} \partial x_{a_2} \partial x_{a_3}} e_{a_1} e_{a_2} e_{a_3},$$

donde $e_{a_k} = x_{a_k} - \xi_{a_k}$, $a_1, \dots, a_n \in \{1, \dots, n\}$. Denotamos por $L_{ij}(x)$ la (i, j) -ésima entrada de la matriz $L(x) = \sum_{k=1}^m \omega_k J_F(\eta_k(x))$, por $H_{ij}(x)$ la entrada (i, j) de $[J_F(x)]^{-1}$ y por $M_{i,j}(x)$ la entrada (i, j) de $[L(x)]^{-1}$.

Por lo tanto, la j -ésima componente de la función iterativa es

$$g_j(x) = \lambda_j(x) + \sum_{i=1}^n H_{ji}(x) f_i(x) - 2 \sum_{i=1}^n M_{ji}(x) f_i(x). \tag{5.32}$$

Por ser $M_{ji}(x)$ y $L_{ij}(x)$ elementos de matrices inversas, (5.32) puede ser reescrita como

$$\sum_{j=1}^n L_{ij}(x) \left[g_j(x) - \lambda_j(x) - \sum_{p=1}^n H_{jp}(x) f_p(x) \right] + 2f_i(x) = 0. \quad (5.33)$$

De la derivación directa de (5.33) respecto a x_l , siendo i y l arbitrarios y fijos se tiene

$$\begin{aligned} & \sum_{j=1}^n \frac{\partial L_{ij}(x)}{\partial x_l} [g_j(x) - \lambda_j(x) - \sum_{p=1}^n H_{jp}(x) f_p(x)] \\ & + \sum_{j=1}^n L_{ij}(x) \left[\frac{\partial g_j(x)}{\partial x_l} - \frac{\partial \lambda_j(x)}{\partial x_l} - \sum_{p=1}^n \frac{\partial H_{jp}(x)}{\partial x_l} f_p(x) - \delta_{jl} \right] + 2 \frac{\partial f_i(x)}{\partial x_l} = 0. \end{aligned} \quad (5.34)$$

Cuando $x = \xi$, aplicando el Lema 5.1.1, expresión (5.18), teniendo en cuenta que $g_j(\xi) = \xi_j$, $\lambda_j(\xi) = \xi_j$ y $f_i(\xi) = 0$ se obtiene:

$$\sum_{j=1}^n L_{ij}(\xi) \frac{\partial g_j(\xi)}{\partial x_j} - L_{il}(\xi) + 2 \frac{\partial f_i(\xi)}{\partial x_l} = 0.$$

Entonces,

$$\sum_{j=1}^n \left(\sum_{k=1}^m \omega_k J_{ij}(\xi) \right) \frac{\partial g_j(\xi)}{\partial x_l} + \left(2 - \sum_{k=1}^m \omega_k \right) J_{il}(\xi) = 0.$$

Por lo tanto, dado que asumimos que la matriz Jacobiana no es singular en ξ y siendo i y l arbitrarios y fijos, entonces si

$$\sum_{k=1}^m \omega_k = 2,$$

se tiene

$$\frac{\partial g_j(\xi)}{\partial x_l} = 0$$

y el orden del método será, al menos, dos. A continuación, derivamos (5.34) respecto a x_r , con r arbitrario y fijo:

$$\begin{aligned} & \sum_{j=1}^n \sum_{k=1}^m \omega_k \frac{\partial^2 J_{ij}(x)}{\partial x_r \partial x_l} \left[g_j(x) - \lambda_j(x) - \sum_{p=1}^n H_{jp}(x) f_p(x) \right] \\ & + \sum_{j=1}^n \sum_{k=1}^m \omega_k \frac{\partial J_{ij}(\eta_k(x))}{\partial x_l} \left[\frac{\partial g_j(x)}{\partial x_r} - \frac{\partial \lambda_j(x)}{\partial x_r} - \sum_{p=1}^n \frac{\partial H_{jp}(x)}{\partial x_r} f_p(x) - \delta_{jr} \right] \\ & + \sum_{j=1}^n \sum_{k=1}^m \omega_k \frac{\partial J_{ij}(\eta_k(x))}{\partial x_r} \left[\frac{\partial g_j(x)}{\partial x_l} - \frac{\partial \lambda_j(x)}{\partial x_l} - \sum_{p=1}^n \frac{\partial H_{jp}(x)}{\partial x_l} f_p(x) - \delta_{jl} \right] \\ & + \sum_{j=1}^n \sum_{k=1}^m \omega_k J_{ij}(\eta_k(x)) \left[\frac{\partial^2 g_j(x)}{\partial x_r \partial x_l} - \frac{\partial^2 \lambda_j(x)}{\partial x_r \partial x_l} - \sum_{p=1}^n \frac{\partial^2 H_{jp}(x)}{\partial x_r \partial x_l} f_p(x) \right] \\ & - \sum_{j=1}^n \sum_{k=1}^m \omega_k J_{ij}(\eta_k(x)) \left[\sum_{p=1}^n H_{jp}(x) \frac{\partial^2 f_p(x)}{\partial x_r \partial x_l} \right] + 2 \frac{\partial^2 f_i(x)}{\partial x_r \partial x_l} = 0 \end{aligned} \quad (5.35)$$

de donde, tras reemplazar x por ξ y aplicar las igualdades (5.25) del Lema 5.1.3, (5.18) y (5.20) del Lema 5.1.1, se obtiene:

$$\sum_{j=1}^n \left(\sum_{k=1}^m \omega_k J_{ij}(\xi) \right) \frac{\partial^2 g_j(\xi)}{\partial x_r \partial x_l} + \left(2 - \sum_{k=1}^m \omega_k (1 - \tau_k) \right) \frac{\partial^2 f_i(\xi)}{\partial x_r \partial x_l} = 0.$$

Entonces, si

$$\sum_{k=1}^m \omega_k (1 - \tau_k) = 2,$$

se puede concluir que

$$\frac{\partial^2 g_j(\xi)}{\partial x_r \partial x_l} = 0 \quad (5.36)$$

y el orden de la convergencia será tres.

Analizando las condiciones que garantizan convergencia de orden mayor que tres, es necesario derivar la expresión (5.35) respecto a x_s , siendo s arbitrario y fijo, y evaluar el resultado en $x = \xi$. La expresión siguiente se obtiene usando (5.25) y (5.26) del Lema 5.1.3 y desde (5.18) hasta (5.21) del Lema 5.1.1, además de exigir que se verifiquen las condiciones (5.28) y (5.29)

$$\begin{aligned} & 2 \sum_{j=1}^n \sum_{k=1}^m \omega_k J_{ij}(\xi) \frac{\partial^3 g_j(\xi)}{\partial x_s \partial x_r \partial x_l} + \left(2 - \frac{3}{4} \sum_{k=1}^m \omega_k (1 - \tau_k)^2 \right) \frac{\partial^3 f_i(\xi)}{\partial x_s \partial x_r \partial x_l} \\ & - \frac{1}{2} \sum_{k=1}^m \omega_k (1 + \tau_k) \sum_{j=1}^n \sum_{q=1}^m \frac{\partial^2 f_i(\xi)}{\partial x_q \partial x_s} \sum_{p=1}^n H_{qp}(\xi) \frac{\partial^2 f_p(\xi)}{\partial x_r \partial x_l} \\ & - \frac{1}{2} \sum_{k=1}^m \omega_k (1 + \tau_k) \sum_{j=1}^n \sum_{q=1}^m \frac{\partial^2 f_i(\xi)}{\partial x_q \partial x_{sl}} \sum_{p=1}^n H_{qp}(\xi) \frac{\partial^2 f_p(\xi)}{\partial x_s \partial x_l} \\ & - \frac{1}{2} \sum_{k=1}^m \omega_k (1 + \tau_k) \sum_{j=1}^n \sum_{q=1}^m \frac{\partial^2 f_i(\xi)}{\partial x_q \partial x_{sl}} \sum_{p=1}^n H_{qp}(\xi) \frac{\partial^2 f_p(\xi)}{\partial x_s \partial x_r} = 0. \end{aligned}$$

En el caso de que se satisfagan las condiciones

$$\frac{\partial^2 f_i(\xi)}{\partial x_{a_1} \partial x_{a_2}} = 0, \text{ para todo, } a_1, a_2 \in \{1, \dots, n\} \quad (5.37)$$

y

$$\sum_{k=1}^m \omega_k (1 - \tau_k)^2 = \frac{8}{3},$$

se concluye que

$$\frac{\partial^3 g_j(\xi)}{\partial x_s \partial x_r \partial x_l} = 0,$$

por lo que tenemos orden de convergencia, al menos, cuatro. Derivando de nuevo respecto de x_u , para u arbitrario y fijo y exigiendo que se verifiquen las condiciones (5.28), (5.29), y (5.30), aplicando el Lema 5.1.1 (las expresiones (5.18)-(5.21)) y el Lema 5.1.3 (expresiones (5.25)-(5.27)) se puede demostrar que:

$$\frac{\partial^4 f_i(\xi)}{\partial x_u \partial x_s \partial x_r \partial x_l} \left(2 - \frac{1}{2} \sum_{k=1}^m \omega_k (1 - \tau_k)^3 \right) + 2 \sum_{j=1}^n J_{ij}(\xi) \frac{\partial^4 g_j(\xi)}{\partial x_u \partial x_s \partial x_r \partial x_l} = 0.$$

Por lo que si se satisface la condición

$$\sum_{k=1}^m \omega_k (1 - \tau_k)^3 = 4,$$

se concluye que

$$\frac{\partial^4 g_j(\xi)}{\partial x_u \partial x_s \partial x_r \partial x_l} = 0$$

y así se obtiene el orden cinco. □

Recordando la notación introducida en (3.18) (Sección 3.2.2) es fácil comprobar que las condiciones (5.28) a (5.31) del Teorema 5.1.4 coinciden con las condiciones del Teorema 3.2.1 de la Sección 3.2.2, es decir,

1. $\sum_{i=1}^m \omega_i = \sigma = 2,$

2. $\sum_{i=1}^m \omega_i(1 - \tau_i) = \sigma(1 - \sigma_1) = 2$ de donde se concluye que $\sigma_1 = 0$;
3. $\sum_{i=1}^m \omega_i(1 - \tau_i)^2 = \sigma(1 - 2\sigma_1 + \sigma_2) = \frac{8}{3}$ de donde se concluye que $\sigma_2 = \frac{1}{3}$;
4. $\sum_{i=1}^m \omega_i(1 - \tau_i)^3 = \sigma(1 - 3\sigma_1 + 3\sigma_2 - \sigma_3) = 4$ de donde se concluye que $\sigma_3 = 0$;

de donde se deduce que las condiciones bajo las cuales se demuestra la convergencia de los métodos Newton-Gauss son las mismas, en lo que en los polinomios ortogonales empleados se refiere, tanto si diseñamos métodos de resolución de ecuaciones no lineales como de sistemas.

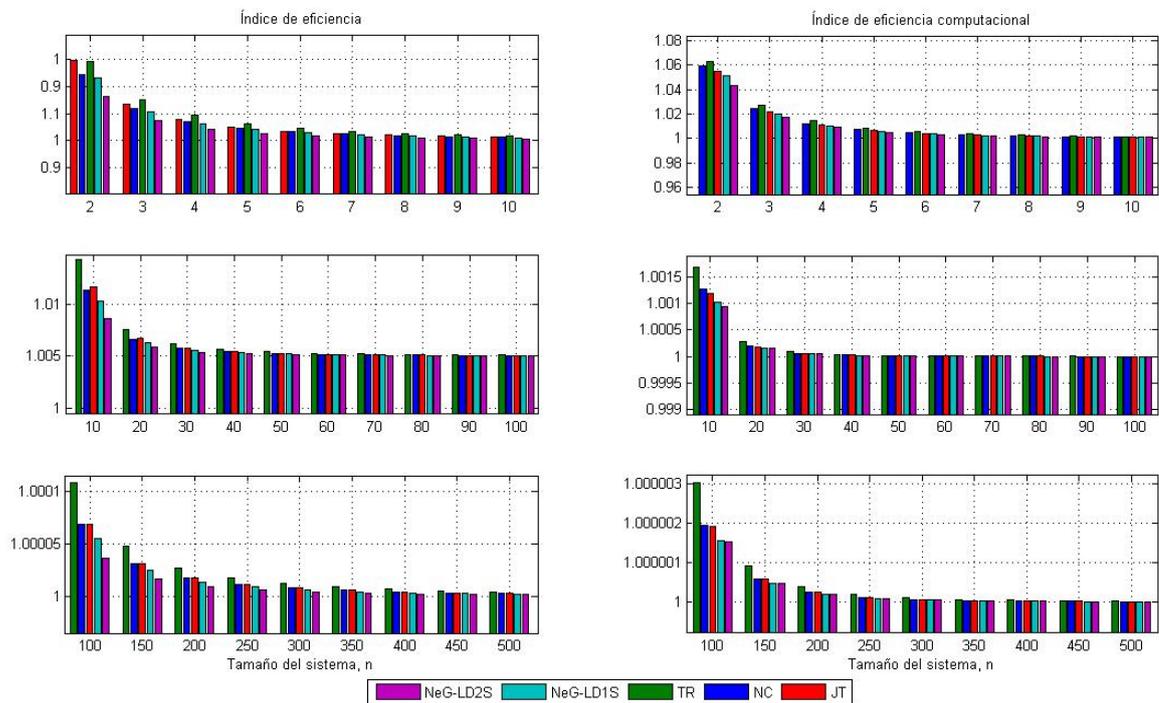


Figura 5.1: Índice de eficiencia, I e Índice de eficiencia computacional, IC

Desde punto de vista de la eficiencia las gráficas presentadas en la Figura 5.1 representan los índices de eficiencia (I) y de eficiencia computacional (IC) de los métodos desarrollados y también de los que usamos para comparar: los métodos de Newton (NC), Traub (TR) y Jarratt (JT), en función del tamaño del sistema de ecuaciones: de 2 hasta 500 ecuaciones. De los métodos desarrollados de la familia NeG se muestran NeG-LD1S que representa los métodos NeG-LT1S, NeG-LT2S y NeG-R2S (tienen mismos índices que NeG-LD1) y NeG-LD2S. El peor índice de eficiencia de la familia NeG corresponde a NeG-LD2, mientras que el mejor en términos absolutos es el método de Traub, TR.

5.1.2. Métodos iterativos con la cuadratura de Gauss y predictor el método de Traub

En la sección anterior se obtuvieron métodos iterativos usando el método de Newton como predictor y corrigiendo el resultado obtenido con la cuadratura gaussiana. Al igual que en la Sección 3.4 para el caso unidimensional en esta sección usaremos como predictor el método de Traub. La fórmula genérica del conjunto de familias de

variantes Traub-Gauss para sistemas de ecuaciones no lineales tiene la forma:

$$\begin{aligned} y^{(k)} &= x^{(k)} - \left[J_F(x^{(k)}) \right]^{-1} F(x^{(k)}) \\ z^{(k)} &= x^{(k)} - \left[J_F(x^{(k)}) \right]^{-1} \left(F(x^{(k)}) + F(y^{(k)}) \right) \\ x^{(k+1)} &= y^{(k)} - 2 \left[\sum_{i=1}^m \omega_i J_F(\eta_i^{(k)}) \right]^{-1} F(y^{(k)}). \end{aligned} \quad (5.38)$$

donde de nuevo denotamos $\eta_i^{(k)} = \frac{(1 + \tau_i) z^{(k)} + (1 - \tau_i) x^{(k)}}{2}$. Utilizamos de nuevo las notaciones (3.18) de la Sección 3.2.2 que recordamos

$$\begin{aligned} \sum_{i=1}^m \omega_i &= \sigma, \\ \sum_{i=1}^m \frac{\omega_i \tau_i^l}{\sigma} &= \sigma_l \quad \text{con } l = 1, 2, \dots \end{aligned}$$

En la demostración del siguiente teorema, así como en el resto del capítulo, usamos la técnica de desarrollo de Taylor introducida en la Sección 2.1.2.

Teorema 5.1.5 *Sea $F : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ suficientemente diferenciable en Ω y $\xi \in \Omega$ es solución del sistema de ecuaciones no lineales $F(x) = 0$. Supongamos que $J_F(x)$ es continua y no singular en ξ . Entonces, la sucesión $\{x^{(k)}\}_{k \geq 0}$ obtenida usando la expresión (5.38) converge a ξ con convergencia de orden cinco si los pesos ω_i y los nodos τ_i de la fórmula de cuadratura verifican: $\sigma = 2$ y $\sigma_1 = 0$.*

Demostración: Desarrollando en serie de Taylor las funciones $F(x^{(k)})$ y $J_F(x^{(k)})$ obtenemos:

$$F(x^{(k)}) = J_F(\xi) [e_k + C_2 e_k^2 + C_3 e_k^3 + C_4 e_k^4 + C_5 e_k^5] + O[e_k^6], \quad (5.39)$$

$$J_F(x^{(k)}) = J_F(\xi) [I + 2C_2 e_k + 3C_3 e_k^2 + 4C_4 e_k^3 + 5C_5 e_k^4] + O[e_k^5]. \quad (5.40)$$

Buscamos

$$\left[J_F(x^{(k)}) \right]^{-1} = [I + X_2 e_k + X_3 e_k^2 + X_4 e_k^3 + X_5 e_k^4] [F'(\xi)]^{-1} + O[e_k^5], \quad (5.41)$$

sabiendo que $[J_F(x^{(k)})]^{-1} J_F(x^{(k)}) = I$. Obtenemos las siguientes expresiones para los X_i :

$$\begin{aligned} X_2 &= -2C_2, \\ X_3 &= 4C_2^2 - 3C_3, \\ X_4 &= 6C_3 C_2 - 8C_2^3 + 6C_2 C_3 - 4C_4, \\ X_5 &= 16C_2^4 - 12C_3 C_2^2 - 12C_2 C_3 C_2 + 8C_4 C_2 + 9C_3^2 - 12C_2^2 C_3 + 8C_2 C_4 - 5C_5. \end{aligned}$$

Así,

$$\begin{aligned} y^{(k)} &= x^{(k)} - \left[J_F(x^{(k)}) \right]^{-1} F(x^{(k)}) \\ &= \xi + C_2 e_k^2 + 2(C_3 - C_2^2) e_k^3 + (3C_4 - 4C_2 C_3 - 3C_3 C_2 + 4C_2^3) e_k^4 \\ &\quad + (4C_5 - 6C_2 C_4 - 6C_3^2 + 8C_2^2 C_3 + 8C_2 C_3 C_2 + 5C_4 C_2 - 12C_2^4) e_k^5 + O[e_k^6]. \end{aligned} \quad (5.42)$$

Introducimos la notación: $y^{(k)} = \xi + e_{y^{(k)}}$, donde

$$e_{y^{(k)}} = C_2 e_k^2 + 2(C_3 - C_2^2) e_k^3 + (3C_4 - 4C_2 C_3 - 3C_3 C_2 + 4C_2^3) e_k^4 + O[e_k^5].$$

De nuevo desarrollamos en serie de Taylor:

$$\begin{aligned} F(y^{(k)}) &= J_F(\xi) [C_2 e_k^2 + 2(C_3 - C_2^2) e_k^3 + (5C_3^2 + 3C_4 - 4C_2 C_3 - 3C_3 C_2) e_k^4] \\ &\quad + J_F(\xi) [(4C_5 - 6C_2 C_4 - 6C_3^2 + 10C_2^2 C_3 + 10C_2 C_3 C_2 + 5C_4 C_2 - 16C_2^4) e_k^5] + O[e_k^6]. \end{aligned}$$

Calculamos la suma de $F(x^{(k)})$ y $F(y^{(k)})$:

$$\begin{aligned} F(x^{(k)}) + F(y^{(k)}) &= J_F(\xi) [e_k + 2C_2e_k^2 + (3C_3 - 2C_2^2)e_k^3 + (5C_2^3 + 4C_4 - 4C_2C_3 - 3C_3C_2)e_k^4] \\ &+ J_F(\xi) [(5C_5 - 6C_2C_4 - 6C_3^2 + 10C_2^2C_3 + 10C_2C_3C_2 + 5C_4C_2 - 16C_2^4)e_k^5] + O[e_k^6]. \end{aligned}$$

De esta manera obtenemos $z^{(k)}$:

$$\begin{aligned} z^{(k)} &= x^{(k)} - [J_F(x^{(k)})]^{-1} (F(x^{(k)}) + F(y^{(k)})) \\ &= \xi + 2C_2^2e_k^3 + (3C_3C_2 + 4C_2C_3 - 9C_2^3)e_k^4 \\ &+ (6C_4C_2 + 6C_3^2 - 18C_2^2C_3 - 16C_2C_3C_2 - 5C_4C_2 + 18C_2^4 - 6C_3C_2^2 + 16C_2^4)e_k^5 + O[e_k^6]. \end{aligned}$$

Introducimos la notación $z^{(k)} = \xi + e_{z^{(k)}}$ donde

$$e_{z^{(k)}} = (3C_3 - 2C_2^2)e_k^3 + (3C_3C_2 + 4C_2C_3 - 9C_2^3)e_k^4 + O[e_k^5]$$

y buscamos $J_F(\eta_i^{(k)})$ con

$$\eta_i^{(k)} = \frac{(1 + \tau_i)z^{(k)} + (1 - \tau_i)y^{(k)}}{2} = \xi + \frac{1}{2} [(e_{z^{(k)}} + e_{y^{(k)}}) + \tau_i(e_{z^{(k)}} - e_{y^{(k)}})],$$

y teniendo

$$\eta_i^{(k)} - \xi = \frac{1}{2} [(e_{z^{(k)}} + e_{y^{(k)}}) + \tau_i(e_{z^{(k)}} - e_{y^{(k)}})],$$

con

$$\begin{aligned} e_{z^{(k)}} + e_{y^{(k)}} &= C_2e_k^2 + 2C_3e_k^3 + (3C_4 - 5C_2^3)e_k^4 + O[e_k^5], \\ e_{z^{(k)}} - e_{y^{(k)}} &= -C_2e_k^2 + (4C_2^2 - 2C_3)e_k^3 + (-13C_2^3 + 8C_2C_3 + 6C_3C_2 - 3C_4)e_k^4 + O[e_k^5]. \end{aligned}$$

De nuevo desarrollando en serie de Taylor, obtenemos:

$$\begin{aligned} J_F(\eta_i^{(k)}) &= J_F(\xi) [I + 2C_2(\eta_i^{(k)} - \xi) + 3C_3(\eta_i^{(k)} - \xi)^2] + O[e_k^5] \\ &= J_F(\xi) [B + C\tau_i + D\tau_i^2] + O[e_k^5], \end{aligned}$$

con

$$\begin{aligned} B &= I + C_2^2e_k^2 + 2C_2C_3e_k^3 + \left(3C_2C_4 - 5C_2^4 + \frac{3}{4}C_3C_2^2\right)e_k^4, \\ C &= -C_2^2e_k^2 + (4C_2^3 - 2C_2C_3)e_k^3 + \left(-13C_2^4 + 8C_2^2C_3 + 6C_2C_3C_2 - 3C_2C_4 - \frac{3}{2}C_3C_2^2\right)e_k^4, \\ D &= \frac{3}{4}C_3C_2^2e_k^4. \end{aligned}$$

Usando la notación (3.18) y por un procedimiento análogo al que nos permitió obtener la expresión de K (3.23) obtenemos

$$K = \sigma J_F(\xi) (I + H_2e_k^2 + H_3e_k^3 + H_4e_k^4) + O[e_k^5],$$

donde

$$\begin{aligned} H_2 &= (1 - \sigma_1)C_2^2, \\ H_3 &= 4\sigma_1C_2^3 + 2(1 - \sigma_1)C_2C_3, \\ H_4 &= 3(1 - \sigma_1)C_2C_4 - (5 + 13\sigma_1)C_2^4 + 8\sigma_1C_2^2C_3 + 6\sigma_1C_2C_3C_2 + \frac{3}{4}(1 + 2\sigma_1 + 3\sigma_2)C_3C_2^2. \end{aligned}$$

Buscamos $K^{-1} = \frac{1}{\sigma} (I + Y_2e_k + Y_3e_k^2 + Y_4e_k^3 + Y_5e_k^4) [J_F(\xi)]^{-1} + O[e_k^5]$. Sabiendo que $K^{-1}K = I$ obtenemos las siguientes expresiones $Y_2 = 0$, $Y_3 = -(1 - \sigma_1)C_2^2$, $Y_4 = -4\sigma_1C_2^3 - 2(1 - \sigma_1)C_2C_3$ e $Y_5 = -H_4 + (1 - \sigma_1)^2C_2^4$

que nos permiten calcular $x^{(k+1)}$:

$$\begin{aligned}
x^{(k+1)} &= y^{(k)} - 2K^{-1}F(y^{(k)}) \\
&= \xi + C_2e_k^2 + 2(C_3 - C_2^2)e_k^3 + (3C_4 - 4C_2C_3 - 3C_3C_2 + 4C_2^3)e_k^4 \\
&+ (4C_5 - 6C_2C_4 - 6C_3^2 + 8C_2^2C_3 + 8C_2C_3C_2 + 5C_4C_2 - 12C_2^4)e_k^5 \\
&- \frac{2}{\sigma} [C_2e_k^2 + 2(C_3 - C_2^2)e_k^3 + (3C_4 - 4C_2C_3 - 3C_3C_2 + 5C_2^3 + Y_3C_2)e_k^4] \\
&- \frac{2}{\sigma} [(4C_5 - 6C_2C_4 - 6C_3^2 + 10C_2^2C_3 + 10C_2C_3C_2 + 5C_4C_2 - 16C_2^4 + 2Y_3(C_3 - 2C_2^2) + Y_4C_2)e_k^5] \\
&+ O[e_k^6],
\end{aligned}$$

de donde la ecuación de error es:

$$\begin{aligned}
e_{k+1} &= C_2e_k^2 + 2(C_3 - C_2^2)e_k^3 + (3C_4 - 4C_2C_3 - 3C_3C_2 + 4C_2^3)e_k^4 \\
&+ (4C_5 - 6C_2C_4 - 6C_3^2 + 8C_2^2C_3 + 8C_2C_3C_2 + 5C_4C_2 - 12C_2^4)e_k^5 \\
&- \frac{2}{\sigma} [C_2e_k^2 + 2(C_3 - C_2^2)e_k^3 + (3C_4 - 4C_2C_3 - 3C_3C_2 + 5C_2^3 + Y_3C_2)e_k^4] \\
&- \frac{2}{\sigma} [(4C_5 - 6C_2C_4 - 6C_3^2 + 10C_2^2C_3 + 10C_2C_3C_2 + 5C_4C_2 - 16C_2^4 + 2Y_3(C_3 - 2C_2^2) + Y_4C_2)e_k^5] \\
&+ O[e_k^6].
\end{aligned}$$

Notamos que el orden de convergencia será al menos cuatro si $\sigma = 2$. Entonces la ecuación del error queda:

$$\begin{aligned}
e_{k+1} &= (-C_2^3 - Y_3C_2)e_k^4 + [-2C_2^2C_3 - 2C_2C_3C_2 - 2Y_3(C_3 - 2C_2^2) - Y_4C_2]e_k^5 + O[e_k^6] \\
&= -\sigma_1C_2^3e_k^4 + [2C_2^4 + 2\sigma_1(3C_2^4 - C_2^2C_3 - C_2C_3C_2)]e_k^5 + O[e_k^6]
\end{aligned}$$

Para tener orden de convergencia cinco se debe verificar que $-C_2^3 - Y_3C_2 = 0$. Sustituyendo en esta igualdad la expresión que corresponde a $Y_3 = -(1 - \sigma_1)C_2^2$ obtenemos $-C_2^3 + (1 - \sigma_1)C_2^3 = 0$ de donde concluimos que si $\sigma_1 = 0$ el orden de convergencia será cinco y la ecuación de error queda:

$$e_{k+1} = 2C_2^4e_k^5 + O[e_k^6].$$

Por lo tanto queda demostrado que el método es de orden cinco. □

En las gráficas de la Figura 5.2 se presentan de nuevo el Índice de eficiencia (I) y de eficiencia computacional (IC) de los métodos desarrollados, y los usados en la comparación: método de Newton (NC), método de Traub (TR) y el método de Jarratt (JT), en función del tamaño del sistema de ecuaciones: de 2 hasta 500 ecuaciones. De los métodos desarrollados de la familia TrG se muestran TrG-LD1S que representa los métodos TrG-LT1S, TrG-LT2S y TrG-R2S (tienen mismos índices que el de TrG-LD1S) y TrG-LD2S. El menor índice de eficiencia (I) lo tiene de nuevo el método de la familia NeG con cuadratura Gauss-Legendre de 2 nodos (TrG-LD2S), pero su índice de eficiencia computacional (IC) para sistemas de 10 ecuaciones es igual al de Newton y para sistemas de mayor tamaño supera al de Newton y Jarratt. Como en el caso de la Figura 5.1, el método TR tiene los mejores índices de eficiencia en general excepto en el caso de sistemas de dos ecuaciones donde el I del método de Jarratt es mejor.

5.1.3. Pseudocomposición para sistemas

Al igual que en el caso unidimensional, las técnicas introducidas en las Secciones 5.1.1 y 5.1.2 se pueden generalizar a métodos predictor-corrector en los que partimos de un método cualquiera de orden p como predictor, corrigiéndolo después con un paso de cuadratura gaussiana.

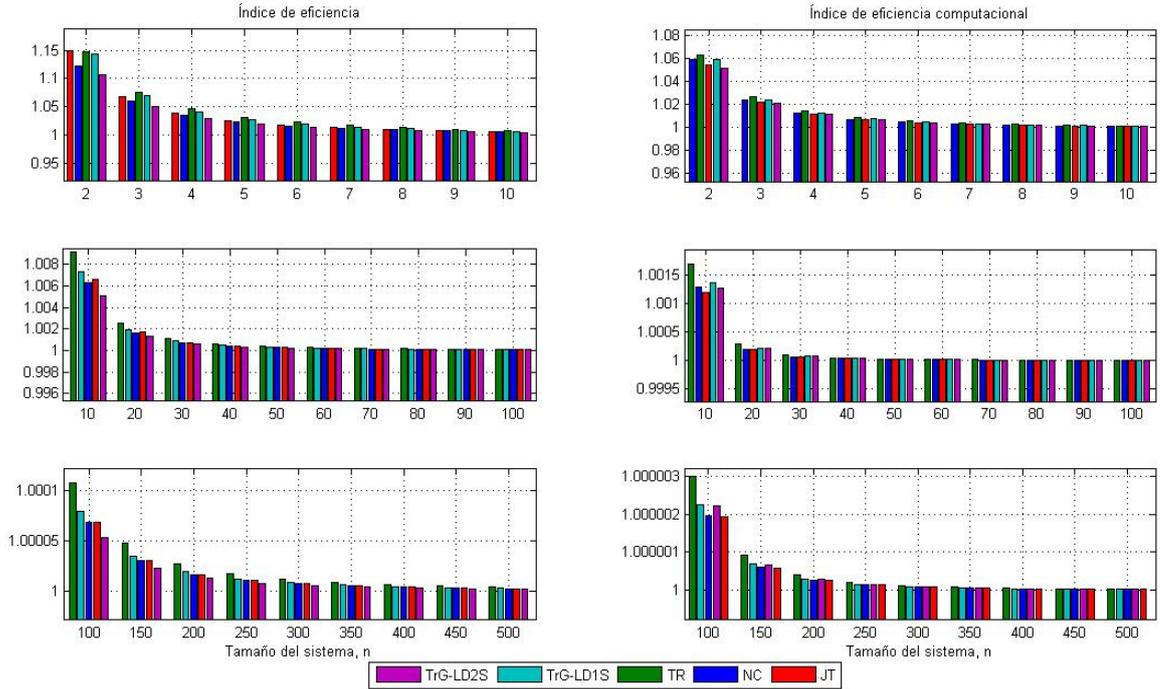


Figura 5.2: Índice de eficiencia, I e índice de eficiencia computacional, IC

Sean $y^{(k)}$ el penúltimo y $z^{(k)}$ el último puntos obtenidos a través de algún método iterativo de orden p :

$$\begin{aligned}
 y^{(k)} &= \xi + M_q e_k^q + M_{q+1} e_k^{q+1} + \dots + M_{2q} e_k^{2q} + M_{2q+1} e_k^{2q+1} + \dots + O[e_k^{5q}], \\
 &= \xi + \sum_{j=q}^{5q-1} M_j e_k^j + O[e_k^{5q}], \\
 z^{(k)} &= \xi + N_p e_k^p + N_{p+1} e_k^{p+1} + \dots + N_{2p} e_k^{2p} + N_{2p+1} e_k^{2p+1} + \dots + O[e_k^{5p}], \\
 &= \xi + \sum_{j=p}^{5p-1} M_j e_k^j + O[e_k^{5p}].
 \end{aligned}$$

Utilizando este método iterativo como predictor introducimos como corrector algunas variaciones sobre el paso de la cuadratura de Gauss. Se pueden obtener 4 casos con las siguientes fórmulas iterativas:

$$\begin{aligned}
 \text{Caso A: } x^{(k+1)} &= y^{(k)} - 2 \left[\sum_{i=1}^m \omega_i J_F(\eta_i^{(k)}) \right]^{-1} F(y^{(k)}), \\
 \text{Caso B: } x^{(k+1)} &= z^{(k)} - 2 \left[\sum_{i=1}^m \omega_i J_F(\eta_i^{(k)}) \right]^{-1} F(z^{(k)}), \\
 \text{Caso C: } x^{(k+1)} &= y^{(k)} - 2 \left[\sum_{i=1}^m \omega_i J_F(\eta_i^{(k)}) \right]^{-1} F(z^{(k)}), \\
 \text{Caso D: } x^{(k+1)} &= z^{(k)} - 2 \left[\sum_{i=1}^m \omega_i J_F(\eta_i^{(k)}) \right]^{-1} F(y^{(k)}),
 \end{aligned}$$

donde, para todos los casos,

$$\eta_i^{(k)} = \frac{(1 + \tau_i)z^{(k)} + (1 - \tau_i)y^{(k)}}{2},$$

está calculada utilizando los puntos obtenidos en los últimos dos pasos del predictor. Introducimos la misma notación que en la Sección 3.4 con el fin de simplificar, adaptándola para el caso multidimensional: $\sum_{j=q}^{5q-1} M_j e_k^j =$

$A_{1(q)}$ y $\sum_{j=q}^{5p-1} N_j e_k^j = A_{2(p)}$ donde la expresión entre paréntesis del subíndice denota el valor de la potencia más pequeña que toma j en la suma. Utilizando esta notación, buscando $\eta_i^{(k)}$, obtenemos

$$\begin{aligned}\eta_i^{(k)} &= \frac{1}{2}[(1 + \tau_i)z^{(k)} + (1 - \tau_i)y^{(k)}] \\ &= \xi + \frac{1}{2}(A_{2(p)} + A_{1(q)}) + \frac{1}{2}\tau_i(A_{2(p)} - A_{1(q)}) \\ &= \xi + \frac{1}{2}(R + \tau_i S)_{(q)},\end{aligned}$$

de donde $R_{(q)} = \frac{1}{2}(A_{2(p)} + A_{1(q)})$ y $S_{(q)} = \frac{1}{2}\tau_i(A_{2(p)} - A_{1(q)})$. Mediante expansión de $F(y^{(k)})$, $F(z^{(k)})$ y $J_F(\eta_i^{(k)})$ en serie de Taylor alrededor de ξ , obtenemos

$$\begin{aligned}F(y^{(k)}) &= J_F(\xi) \left[A_{1(q)} + C_2 A_{1(2q)}^2 + C_3 A_{1(3q)}^3 + C_4 A_{1(4q)}^4 \right] + O[e_k^{5q}], \\ F(z^{(k)}) &= J_F(\xi) \left[A_{1(p)} + C_2 A_{1(2p)}^2 + C_3 A_{1(3p)}^3 + C_4 A_{1(4p)}^4 \right] + O[e_k^{5p}], \\ J_F(\eta_i^{(k)}) &= J_F(\xi) \left[I + 2C_2(\eta_i^{(k)} - \xi)_{(q)} + 3C_3(\eta_i^{(k)} - \xi)_{(2q)}^2 + 4C_4(\eta_i^{(k)} - \xi)_{(3q)}^3 \right] + O[e_k^{4q}] \\ &= J_F(\xi) \left[1 + (B + C\tau_i + D\tau_i^2 + E\tau_i^3)_{(q)} \right] + O[e_k^{4q}],\end{aligned}$$

donde

$$\begin{aligned}B &= C_2 R + \frac{3}{4}C_3 R^2 + \frac{1}{2}C_4 R^3, \\ C &= C_2 S + \frac{3}{4}C_3 R S + \frac{1}{3}C_3 S R + \frac{1}{2}C_4 R^2 S + \frac{1}{2}C_4 R S R + \frac{1}{2}C_4 S R^2, \\ D &= \frac{3}{4}C_3 S^2 + \frac{1}{2}C_4 S R S + \frac{1}{2}C_4 S^2 R, \\ E &= \frac{1}{2}C_4 S^3.\end{aligned}$$

Ahora, desarrollamos la expresión $K = \sum_{i=1}^m \omega_i J_F(\eta_i^{(k)})$

$$\begin{aligned}K &= \sum_{i=1}^m \omega_i J_F(\eta_i^{(k)}) = \sum_{i=1}^m \omega_i J_F(\xi) \left[I + (B + C\tau_i + D\tau_i^2 + E\tau_i^3)_{(q)} \right] + O[e_k^{4q}] \\ &= J_F(\xi) \left[\sum_{i=1}^m \omega_i I + \left(B \sum_{i=1}^m \omega_i + C \sum_{i=1}^m \omega_i \tau_i + D \sum_{i=1}^m \omega_i \tau_i^2 + E \sum_{i=1}^m \omega_i \tau_i^3 \right)_{(q)} \right] + O[e_k^{4q}].\end{aligned}$$

Multiplicando y dividiendo por $\sum_{i=1}^m \omega_i$, e introduciendo las notaciones (3.18) calculamos:

$$\begin{aligned}K &= \sigma J_F(\xi) [I + (B + C\sigma_1 + D\sigma_2 + E\sigma_3)] + O[e_k^{4q}] \\ &= \sigma J_F(\xi) \left[I + C_2 (R + \sigma_1 S)_{(q)} + \frac{3}{4}C_3 (R^2 + \sigma_1 (RS + SR) + \sigma_2 S^2) \right]_{(2q)} \\ &\quad + \sigma J_F(\xi) \left[\frac{1}{2}C_4 (R^3 + \sigma_1 (R^2 S + R S R + S R^2) + \sigma_2 (S R S + S^2 R) + \sigma_3 S^3) \right]_{(3q)}.\end{aligned}$$

Recordando que $K^{-1}K = I$, obtenemos K^{-1}

$$K^{-1} = \frac{1}{\sigma} [I + X_{1(q)} + X_{2(2q)} + \dots] [J_F(\xi)]^{-1},$$

donde

$$\begin{aligned} X_{1(q)} &= -C_2(R + \sigma_1 S)_{(q)}, \\ X_{2(2q)} &= \left[-\frac{3}{4}C_3 [R^2 + \sigma_1(RS + SR) + \sigma_2 S^2] - [C_2(R + \sigma_1 S)C_2(R + \sigma_1 S)] \right]_{(2q)}. \end{aligned}$$

Por lo tanto, considerando el caso A, obtenemos para $L = 2K^{-1}F(y^{(k)})$ la siguiente expresión:

$$\begin{aligned} L &= 2K^{-1}F(y^{(k)}) \\ &= \frac{2}{\sigma}A_{1(q)} + \frac{2}{\sigma}[(C_2A_1 + X_1)A_1]_{(2q)} + \frac{2}{\sigma}[(C_3A_1^2 + X_1C_2A_1 + X_2)A_1]_{(3q)} + \dots \end{aligned}$$

Como $x_{k+1} = y_k - L$, calculamos e_{k+1}

$$e_{k+1} = A_{1(q)} - \frac{2}{\sigma}A_{1(q)} - \frac{2}{\sigma}[(C_2A_1 + X_1)A_1]_{(2q)} - \frac{2}{\sigma}[(C_3A_1^2 + X_1C_2A_1 + X_2)A_1]_{(3q)} + \dots$$

Notamos que si $\sigma = 2$ obtenemos orden de convergencia al menos $2q$. Entonces, la expresión de la ecuación de error e_{k+1} queda:

$$e_{k+1} = -[(C_2A_1 + X_1)A_1]_{(2q)} - C_3[(C_3A_1^2 + X_1C_2A_1 + X_2)A_1]_{(3q)} + \dots \quad (5.43)$$

De la expresión $(C_2A_1 + X_1)A_1$ depende la posibilidad de tener orden de convergencia mayor que $2q$. La desarrollamos y obtenemos:

$$(C_2A_1 + X_1)A_1 = \sigma_1 C_2(A_1^2)_{(2q)} - (1 + \sigma_1)C_2(A_2A_1)_{(p+q)}.$$

Luego, si $\sigma_1 = 0$ la expresión del error queda:

$$e_{k+1} = -C_2(A_2A_1)_{(p+q)} - [C_3A_1^3 + C_2RC_2A_1^2 + C_2RC_2RA_1 - \frac{3}{4}C_3(R^2 + \sigma_2 S^2)A_1]_{(3q)} + \dots$$

Puede ocurrir uno de los siguientes dos casos:

- (a) Si $p + q \geq 3q$ el orden de convergencia sera $3q$.
- (b) Si $p + q < 3q$ el orden de convergencia sera $p + q$.

Esto demuestra que el mayor orden de convergencia que se puede obtener en el caso A será $\min\{p + q, 3q\}$.

Para estudiar el caso B calculamos de nuevo L esta vez en la forma $L = 2K^{-1}F(z^{(k)})$ obteniendo:

$$\begin{aligned} L &= 2K^{-1}F(z^{(k)}) \\ &= \frac{2}{\sigma}A_{2(p)} + \frac{2}{\sigma}[(C_2A_2 + X_1)A_2]_{(p+q)} + \frac{2}{\sigma}[(C_3A_2^3)_{(3p)} + (X_1C_2A_2^2)_{(q+2p)} + (X_2A_2)_{(2q+p)}] + \dots \end{aligned}$$

Como en este caso $x_{k+1} = z_k - L$, calculamos e_{k+1}

$$e_{k+1} = A_{2(p)} - \frac{2}{\sigma}A_{2(p)} - \frac{2}{\sigma}[(C_2A_2 + X_1)A_2]_{(p+q)} - \frac{2}{\sigma}[(C_3A_2^3)_{(3p)} + (X_1C_2A_2^2)_{(q+2p)} + (X_2A_2)_{(2q+p)}] + \dots$$

y el orden de convergencia del método resultante sera p . Luego, si $\sigma = 2$ la expresión del error queda

$$e_{k+1} = -[(C_2A_2 + X_1)A_2]_{(p+q)} - [(C_3A_2^3)_{(3p)} + (X_1C_2A_2^2)_{(q+2p)} + (X_2A_2)_{(2q+p)}] + \dots$$

Desarrollamos la expresión $(C_2A_2 + X_1)A_2$ obteniendo:

$$(C_2A_2 + X_1)A_2 = -\frac{1}{2}(1 - \sigma_1)C_2[A_{2(2p)}^2 + (A_1A_2)_{(p+q)}].$$

Luego, el orden de convergencia sera $p + q$ al menos que $\sigma_1 = 1$ ó $C_2 = 0$, en cuyo caso el orden de convergencia sera al menos $2q + p$.

Por último, fácilmente se puede mostrar que en los casos C y D el orden de convergencia será q que es menor que el orden del predictor. Debemos destacar que en el caso B se necesitan dos evaluaciones funcionales nuevas y la resolución de un sistema de ecuaciones lineales nuevo lo que, desde punto de vista de eficiencia computacional, hace que el método que se obtiene no sea efectivo. Observamos también que la obtención de orden de convergencia mayor que $p + q$ dependerá de las expresiones A_1 y A_2 que representan los errores del penúltimo y del último pasos del método predictor. Los comentarios anteriores nos permiten enunciar el siguiente resultado:

Teorema 5.1.6 *Sea $F : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ suficientemente diferenciable en Ω y $\xi \in \Omega$ solución del sistema de ecuaciones no lineales $F(x) = 0$. Supongamos que $J_F(x)$ es continua y no singular en ξ . Sean $y^{(k)}$ y $z^{(k)}$ los penúltimo y último pasos, de órdenes p y q , respectivamente de un cierto método iterativo. Tomando este método como predictor obtenemos una nueva aproximación $x^{(k+1)}$ de ξ dada por*

$$x^{(k+1)} = y^{(k)} - 2 \left[\sum_{i=1}^m \omega_i J_F(\eta_i^{(k)}) \right]^{-1} F(y^{(k)}),$$

donde $\eta_i^{(k)} = \frac{1}{2} [(1 + \tau_i)z^{(k)} + (1 - \tau_i)y^{(k)}]$ y τ_i y ω_i son los nodos y los pesos del polinomio ortogonal correspondiente a la cuadratura gaussiana utilizada. Entonces,

1. el conjunto de familias obtenidas tendrá orden de convergencia q ;
2. si se cumple la condición $\sigma = 2$ el orden de convergencia será $2q$;
3. si, además, $\sigma_1 = 0$ el orden de la convergencia será $\min\{p + q, 3q\}$. □

El orden de convergencia que tendrá el método en el caso de que se cumpla la condición $C_2 = 0$ dependerá de las expresiones de A_1 y A_2 , de su suma y diferencia y de los valores de σ_2 y σ_3 en algunos casos.

Retomando lo visto en las Secciones 5.1.1 y 5.1.2 a la vista del anterior resultado, concluimos que:

1. En las familias NeG tenemos $q = 1$ y $p = 2$, por lo tanto, el orden de la familia de métodos resultante, según el Teorema 5.1.6 será al menos dos si $\sigma = 2$. En el caso de que $\sigma_1 = 0$ obtenemos orden tres. lo que concuerda con lo demostrado en el Teorema 5.1.4 de la Sección 5.1.1.
2. En las familias TrG tenemos $q = 2$ y $p = 3$, por lo tanto, el orden de la familia de métodos según el Teorema 5.1.6 será al menos cuatro si $\sigma = 2$. En el caso de que $\sigma_1 = 0$ obtenemos orden cinco. Esto fue demostrado con el Teorema 5.1.5 de la Sección 5.1.2.

5.2. Otros métodos iterativos

En esta sección vamos a diseñar métodos iterativos de orden alto utilizando la técnica de pseudocomposición y partiendo de diferentes predictores. Compararemos los distintos esquemas empleando las herramientas habituales: los índices de eficiencia y eficiencia computacional.

En primer lugar vamos a utilizar la familia de métodos cuyo esquema iterativo es:

$$\begin{aligned} y^{(k)} &= x^{(k)} - \alpha \left[J_F(x^{(k)}) \right]^{-1} F(x^{(k)}), \\ x^{(k+1)} &= y^{(k)} - [a_1 J_F(x^{(k)}) + a_2 J_F(y^{(k)})]^{-1} [b_1 J_F(x^{(k)}) + b_2 J_F(y^{(k)})] \left[J_F(x^{(k)}) \right]^{-1} F(y^{(k)}). \end{aligned} \tag{5.44}$$

Este método se ha obtenido a partir del método de Jarratt, mejorando el orden de convergencia de éste, sin añadir más evaluaciones funcionales. Ha sido publicado en el trabajo: *Artificial Satellites Preliminary Orbit Determination by modified high-order Gauss methods*, International Journal of Computer Mathematics, (véase [5]).

Teorema 5.2.1 Sea $F : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ suficientemente diferenciable en Ω y $\xi \in \Omega$ solución del sistema de ecuaciones no lineales $F(x) = 0$. Supongamos que $J_F(x)$ es continua y no singular en ξ . Entonces, la sucesión $\{x^{(k)}\}_{k \geq 0}$ obtenida usando la expresión (5.44) converge a ξ con convergencia de orden 5 si los parámetros toman los siguientes valores: $\alpha = 1$, $a_2 = -5a_1$, $b_1 = -3a_1$ y $b_2 = -a_1$, con $a_1 \neq 0$.

Demostración: Desarrollamos en serie de Taylor las funciones $F(x^{(k)})$ y $J_F(x^{(k)})$ y obtenemos de nuevo las expresiones (5.39) y (5.40). Sabiendo que $[J_F(x^{(k)})]^{-1} J_F(x^{(k)}) = I$, buscamos $[J_F(x^{(k)})]^{-1}$ que tiene la forma (5.41) recordando que $X_1 = I$ y $X_m = -\sum_{j=2}^m j X_{m-j+1} C_j$ $m = 2, 3, \dots$. Así, calculamos el producto

$$[J_F(x^{(k)})]^{-1} F(x^{(k)}):$$

$$\left[J_F(x^{(k)}) \right]^{-1} F(x^{(k)}) = e_k + [M_2 e_k^2 + M_3 e_k^3 + M_4 e_k^4 + M_5 e_k^5 + M_6 e_k^6 + M_7 e_k^7 + M_8 e_k^8] + O[e_k^9]$$

donde $M_2 = -C_2$ y

$$M_s = C_s + \sum_{j=3}^s X_{s-j+2} C_{j-1} + X_s, \quad s = 3, 4, \dots$$

Por lo tanto, $y^{(k)}$ tiene la forma:

$$y^{(k)} = x^{(k)} - \alpha [J_F(x^{(k)})]^{-1} F(x^{(k)}) = \xi + (1 - \alpha)e_k - \alpha M,$$

donde

$$M = M_2 e_k^2 + M_3 e_k^3 + M_4 e_k^4 + M_5 e_k^5 + M_6 e_k^6 + M_7 e_k^7 + M_8 e_k^8 + O[e_k^9].$$

De nuevo desarrollamos en serie de Taylor $F(y^{(k)})$ y $J_F(y^{(k)})$:

$$\begin{aligned} F(y^{(k)}) &= J_F(\xi) [(1 - \alpha)e_k + Q_2 e_k^2 + Q_3 e_k^3 + Q_4 e_k^4 + Q_5 e_k^5] + O[e_k^6], \\ J_F(y^{(k)}) &= J_F(\xi) [I + 2(1 - \alpha)C_2 e_k + T_2 e_k^2 + T_3 e_k^3 + T_4 e_k^4] + O[e_k^5], \end{aligned}$$

donde

$$\begin{aligned} Q_2 &= (\alpha + (1 - \alpha)^2)C_2, \\ Q_3 &= -\alpha M_3 + 2\alpha(1 - \alpha)C_2^2 + (1 - \alpha)^3 C_3, \\ Q_4 &= -\alpha M_4 + \alpha^2 C_3^2 - 2\alpha(1 - \alpha)C_2 M_3 + 3\alpha(1 - \alpha)^2 C_3 C_2 + (1 - \alpha)^4 C_4, \\ Q_5 &= -\alpha M_5 - 2\alpha(1 - \alpha)C_2 M_4 + \alpha^2 (C_2^2 M_3 + C_2 M_3 C_2) + 3\alpha^2 (1 - \alpha)C_3 C_2^2 - 3\alpha(1 - \alpha)^2 C_3 M_3 \\ &\quad + 4\alpha(1 - \alpha)^3 C_4 C_2 + (1 - \alpha)^5 C_5, \\ T_2 &= 2\alpha C_2^2 + 3(1 - \alpha)^2 C_3, \\ T_3 &= -2\alpha C_2 M_3 + 6\alpha(1 - \alpha)C_3 C_2 + 4(1 - \alpha)^5 C_4, \\ T_4 &= -2\alpha C_2 M_4 + 3\alpha^2 C_3 C_2^2 + 12\alpha(1 - \alpha)C_3 M_3 - 3\alpha(1 - \alpha)^2 C_4 C_2 + 5(1 - \alpha)^4 C_5. \end{aligned}$$

Calculamos las siguientes expresiones:

$$\begin{aligned} a_1 J_F(x^{(k)}) + a_2 J_F(y^{(k)}) &= J_F(\xi) [(a_1 + a_2)I + L_1 e_k + L_2 e_k^2 + L_3 e_k^3 + L_4 e_k^4 + L_5 e_k^5] + O[e_k^6], \\ b_1 J_F(x^{(k)}) + b_2 J_F(y^{(k)}) &= J_F(\xi) [(b_1 + b_2)I + P_1 e_k + P_2 e_k^2 + P_3 e_k^3 + P_4 e_k^4 + P_5 e_k^5] + O[e_k^6], \end{aligned}$$

donde

$$\begin{aligned} L_1 &= 2(a_1 + a_2(1 - \alpha))C_2, \\ L_s &= (s + 1)a_1 C_{s+1} + a_2 T_s, \quad s = 2, 3, \dots, \\ P_1 &= 2(b_1 + b_2(1 - \alpha))C_2, \\ P_s &= (s + 1)b_1 C_{s+1} + b_2 T_s, \quad s = 2, 3, \dots \end{aligned}$$

Recordando que

$$\left[a_1 J_F(x^{(k)}) + a_2 J_F(y^{(k)}) \right]^{-1} \left[a_1 J_F(x^{(k)}) + a_2 J_F(y^{(k)}) \right] = I,$$

obtenemos $[a_1 J_F(x^{(k)}) + a_2 J_F(y^{(k)})]^{-1}$ en la forma:

$$\left[a_1 J_F(x^{(k)}) + a_2 J_F(y^{(k)}) \right]^{-1} = \left[\frac{1}{a_1 + a_2} I + Z_2 e_k + Z_3 e_k^2 + Z_4 e_k^3 + Z_5 e_k^4 + Z_6 e_k^5 \right] [J_F(\xi)]^{-1} + O[e_k^6]$$

donde $Z_2 = -\frac{L_1}{(a_1 + a_2)^2}$ y con $a_1 + a_2 \neq 0$ y

$$Z_s = -\frac{1}{(a_1 + a_2)^2} L_{s-1} - \frac{1}{a_1 + a_2} \sum_{j=3}^s Z_{j-1} L_{s-j+1}, \quad s = 3, 4, \dots$$

Buscamos los productos:

$$\begin{aligned} \left[a_1 J_F(x^{(k)}) + a_2 J_F(y^{(k)}) \right]^{-1} [b_1 J_F(x^{(k)}) + b_2 J_F(y^{(k)})] &= \beta I + S_1 e_k + S_2 e_k^2 + S_3 e_k^3 + S_4 e_k^4 + S_5 e_k^5 + O[e_k^6], \\ \left[J_F(x^{(k)}) \right]^{-1} F(y^{(k)}) &= (1 - \alpha) e_k + R_2 e_k^2 + R_3 e_k^3 + R_4 e_k^4 + R_5 e_k^5 + O[e_k^6]. \end{aligned}$$

donde

$$\begin{aligned} \beta &= \frac{b_1 + b_2}{a_1 + a_2}, \\ S_1 &= \frac{P_1}{a_1 + a_2} + (b_1 + b_2) Z_2, \\ S_s &= \frac{1}{a_1 + a_2} P_s + \sum_{j=2}^s Z_j L_{s-j+1} + (b_1 + b_2) Z_{s+1}, \quad s = 2, 3, \dots, \\ R_2 &= Q_2 + X_2(1 - \alpha), \\ R_s &= Q_s + \sum_{j=2}^{s-1} X_j Q_{s-j+1} + (1 - \alpha) X_s, \quad s = 3, 4, \dots \end{aligned}$$

Finalmente obtenemos $x^{(k+1)}$:

$$\begin{aligned} x^{(k+1)} &= \xi + [(1 - \alpha) - \beta(1 - \alpha)] e_k + (\alpha C_2 - \beta R_2 - (1 - \alpha) S_1) e_k^2 \\ &+ (-\alpha M_3 - \beta R_3 - S_1 R_2 - (1 - \alpha) S_2) e_k^3 \\ &+ (-\alpha M_4 - \beta R_4 - S_1 R_3 - S_2 R_2 - (1 - \alpha) S_3) e_k^4 \\ &+ (-\alpha M_5 - \beta R_5 - S_1 R_4 - S_2 R_3 - S_3 R_2 - (1 - \alpha) S_4) e_k^5 + O[e_k^6]. \end{aligned} \quad (5.45)$$

Las condiciones para tener orden de convergencia cinco se encuentran resolviendo simultáneamente las ecuaciones:

$$\begin{aligned} (1 - \alpha) - \beta(1 - \alpha) &= 0 \\ \alpha C_2 - \beta R_2 - (1 - \alpha) S_1 &= 0 \\ -\alpha M_3 - \beta R_3 - S_1 R_2 - (1 - \alpha) S_2 &= 0 \\ -\alpha M_4 - \beta R_4 - S_1 R_3 - S_2 R_2 - (1 - \alpha) S_3 &= 0 \end{aligned}$$

Desde las dos primeras ecuaciones se determina que $\alpha = 1$ y $\beta = 1$. En este caso el sistema se transforma en:

$$M_3 + R_3 + S_1 R_2 = 0 \quad (5.46)$$

$$M_4 + R_4 + S_1 R_3 + S_2 R_2 = 0. \quad (5.47)$$

Del resultado $\beta = 1$ sigue que

$$a_1 + a_2 = b_1 + b_2, \quad (5.48)$$

con $a_1 + a_2 \neq 0$. Por lo tanto, de la ecuación (5.46) se obtiene $b_1 = 2a_1 + a_2$. De la ecuación (5.48) obtenemos $b_2 = -a_1$, $b_1 = -3a_1$ y $a_2 = -5a_1$. Recalculando (5.45) obtenemos que la expresión de la ecuación del error finalmente será:

$$e_{k+1} = \frac{3}{2} C_2^4 - \frac{1}{2} C_2 C_3 C_2 + \frac{3}{2} C_3 C_2^2 + O[e_k^6].$$

Por lo tanto, el teorema queda demostrado. □

Finalmente, la fórmula de iteración del método, tomando $a_1 = 1$ queda:

$$y^{(k)} = x^{(k)} - [J_F(x^{(k)})]^{-1} F(x^{(k)}),$$

$$x^{(k+1)} = y^{(k)} + [J_F(x^{(k)}) - 5J_F(y^{(k)})]^{-1} [3J_F(x^{(k)}) + J_F(y^{(k)})] [J_F(x^{(k)})]^{-1} F(y^{(k)}).$$

Este método, que denotaremos por M5S, será utilizado en las pruebas numéricas.

Si aplicamos la pseudocomposición utilizando como predictor el métodos M5S y como corrector el método de punto medio, el estudio del caso B de la pseudocomposición nos permite afirmar que el método resultante, que denotaremos por M5SG, tiene orden 7 y su fórmula iterativa es:

$$y^{(k)} = x^{(k)} - [J_F(x^{(k)})]^{-1} F(x^{(k)}),$$

$$z^{(k)} = y^{(k)} + [J_F(x^{(k)}) - 5J_F(y^{(k)})]^{-1} [3F'(x^{(k)}) + J_F(y^{(k)})] [F(x^{(k)})]^{-1} F(y^{(k)}),$$

$$x^{(k+1)} = y^{(k)} - \left[J_F \left(\frac{y^{(k)} + z^{(k)}}{2} \right) \right]^{-1} F(y^{(k)}).$$

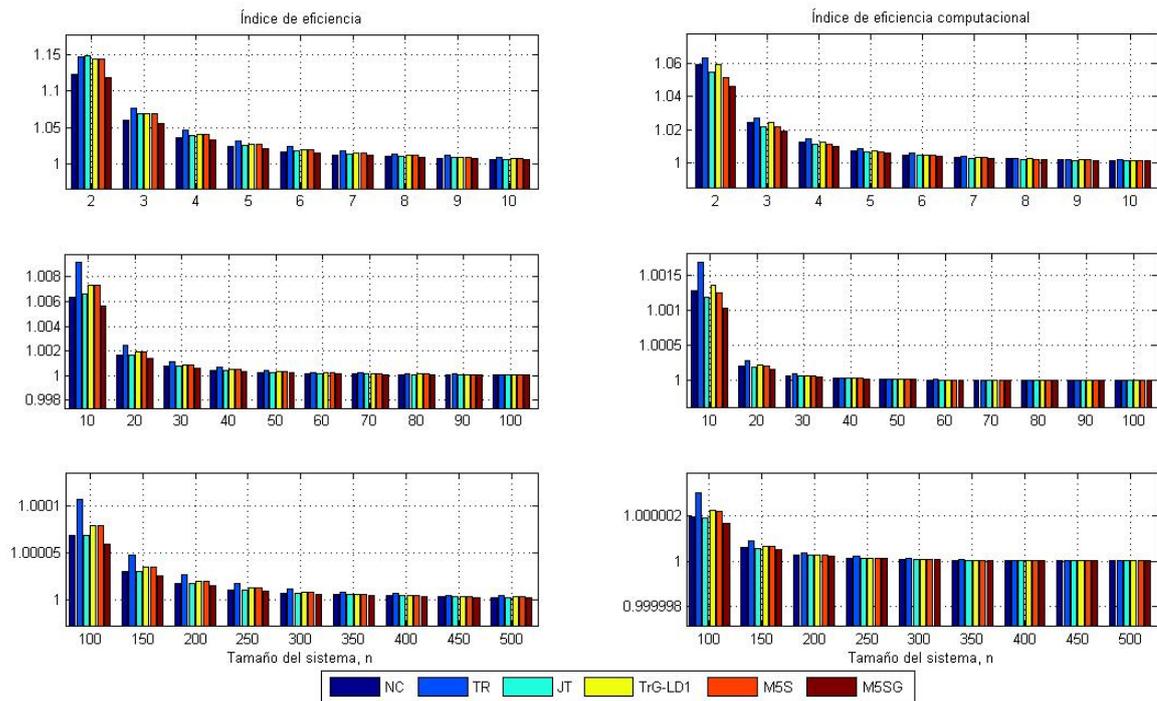


Figura 5.3: Índice de eficiencia, I e índice de eficiencia computacional, IC

Para comparar la eficiencia de los métodos desarrollados, en la Figura 5.3 se muestran las gráficas de los distintos índices de eficiencia en función del tamaño del sistema. Vemos que el método de Traub tiene los mejores índices para todo tamaño de sistemas con la excepción del caso de sistema con $n = 2$ cuando el método de Jarratt tiene mejor I. En cuanto al índice clásico, observamos que los métodos M5S y TrG-LD1S tienen el mismo índice. Este es mejor que el índice del resto de métodos para $n \geq 3$, excepto el método de Traub. Además, el índice de M5SG sólo supera al de Newton para $n \leq 4$. Comparando los IC de los métodos TrG-LD1S, M5S y Jarratt vemos que para $n = 2$, $IC_{TrG-LD1S} > IC_{JT} > IC_{M5S}$ pero aumentando el tamaño del sistema se pueden ordenar de manera siguiente: $IC_{TrG-LD1S} > IC_{M5S} > IC_{JT}$. Desde punto de vista de la eficiencia, no está justificada la pseudocomposición sobre M5S por la necesidad de una evaluación funcional

más y resolución de un sistema lineal nuevo, mientras que aplicada sobre el método de Traub (TrG-LD1S) es muy eficiente.

A continuación diseñamos algunos métodos de orden alto para sistemas no lineales. Algunos de ellos son adaptaciones de métodos de resolución de ecuaciones no lineales, mientras que otros (como el que exponemos en continuación) han sido diseñados específicamente para sistemas.

Mostramos a continuación una familia de métodos iterativos de tres pasos, sobre la que se ha tratado de establecer los valores de los parámetros que optimizan el orden de convergencia.

$$\begin{aligned} y^{(k)} &= x^{(k)} + \alpha \left[J_F(x^{(k)}) \right]^{-1} F(x^{(k)}), \\ z^{(k)} &= x^{(k)} + \left[\beta_3 J_F(x^{(k)}) + \beta_4 J_F(y^{(k)}) \right]^{-1} \left[\beta_1 J_F(x^{(k)}) + \beta_2 J_F(y^{(k)}) \right], \\ x^{(k+1)} &= z^{(k)} + \left[\gamma_1 J_F(x^{(k)}) + \gamma_2 J_F(y^{(k)}) \right]^{-1} F(z^{(k)}) \end{aligned}$$

En este proceso se ha demostrado que si los parámetros introducidos tienen valores $\alpha = -\frac{1}{2}$, $\beta_1 = -\frac{3\beta_4}{2}$, $\beta_2 = 2\beta_4$, $\beta_3 = -\frac{\beta_4}{2}$, $\gamma_1 = 1$ y $\gamma_2 = -2$, con $\beta \neq 0$ el método iterativo tiene orden de convergencia 4 utilizando sólo los primeros dos pasos y seis, si se introduce el tercer paso. Su esquema iterativo para $\beta_4 = \gamma_2 = -2$ queda:

$$\begin{aligned} y^{(k)} &= x^{(k)} - \frac{1}{2} \left[J_F(x^{(k)}) \right]^{-1} F(x^{(k)}), \\ z^{(k)} &= x^{(k)} + \left[J_F(x^{(k)}) - 2J_F(y^{(k)}) \right]^{-1} \left[3J_F(x^{(k)}) - 4J_F(y^{(k)}) \right], \\ x^{(k+1)} &= z^{(k)} + \left[J_F(x^{(k)}) - 2J_F(y^{(k)}) \right]^{-1} F(z^{(k)}) \end{aligned}$$

Si, además, lo tomamos como predictor con $p = 4$ y $q = 6$, y aplicamos la cuadratura de Gauss como corrector, el método resultante tendrá orden de convergencia 10, según el Teorema 5.1.6, (si $\sigma = 2$ y $\sigma_1 = 0$ para los polinomios ortogonales correspondientes a la cuadratura utilizada). La fórmula iterativa de la familia de métodos obtenida es:

$$\begin{aligned} y^{(k)} &= x^{(k)} - \frac{1}{2} \left[J_F(x^{(k)}) \right]^{-1} F(x^{(k)}), \\ z^{(k)} &= x^{(k)} + \left[J_F(x^{(k)}) - 2J_F(y^{(k)}) \right]^{-1} \left[3F(x^{(k)}) - 4J_F(y^{(k)}) \right], \\ u^{(k)} &= z^{(k)} + \left[J_F(x^{(k)}) - 2J_F(y^{(k)}) \right]^{-1} F(z^{(k)}), \\ x^{(k+1)} &= z^{(k)} - 2 \left[\sum_{i=1}^m \omega_i J_F \left(\frac{(1 + \tau_i)u^{(k)} + (1 - \tau_i)z^{(k)}}{2} \right) \right]^{-1} F(z^{(k)}). \end{aligned} \tag{5.49}$$

Aunque, una vez demostrado el orden seis del predictor, bastaría con aplicar el Teorema 5.1.6 para concluir el orden diez de la familia (5.49) en el siguiente resultado llevamos a cabo la demostración completa hasta el paso de la pseudocomposición. En el resto de métodos diseñados en este capítulo, omitiremos estos cálculos haciendo referencia simplemente al teorema general de la pseudocomposición.

Teorema 5.2.2 *Sea $F : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ suficientemente diferenciable en Ω y $\xi \in \Omega$ es solución del sistema de ecuaciones no lineales $F(x) = 0$. Supongamos que $J_F(x)$ es continua y no singular en ξ . Entonces, la sucesión $\{x^{(k)}\}_{k \geq 0}$ obtenida usando la expresión (5.49) converge a ξ con convergencia de orden 4 usando sus primeros dos pasos y orden 6 usando los primeros tres pasos. Si aplicamos los cuatro pasos de (5.49) y si se cumple las condiciones $\sigma = \sum_{i=1}^m \omega_i = 2$ y $\sigma_1 = 0$ para los polinomios ortogonales de la cuadratura gaussiana el orden de convergencia es diez.*

Demostración: Desarrollamos en serie de Taylor las funciones $F(x^{(k)})$ y $J_F(x^{(k)})$

$$\begin{aligned} F(x^{(k)}) &= J_F(\xi) [e_k + C_2 e_k^2 + C_3 e_k^3 + C_4 e_k^4 + C_5 e_k^5 + C_6 e_k^6 + C_7 e_k^7] \\ &+ J_F(\xi) [C_8 e_k^8 + C_9 e_k^9 + C_{10} e_k^{10}] + O[e_k^{11}], \end{aligned} \quad (5.50)$$

$$\begin{aligned} J_F(x^{(k)}) &= J_F(\xi) [I + 2C_2 e_k + 3C_3 e_k^2 + 4C_4 e_k^3 + 5C_5 e_k^4 + 6C_6 e_k^5 + 7C_7 e_k^6] \\ &+ J_F(\xi) [8C_8 e_k^7 + 9C_9 e_k^8 + 10C_{10} e_k^9] + O[e_k^{10}]. \end{aligned} \quad (5.51)$$

Sabiendo que $[J_F(x^{(k)})]^{-1} J_F(x^{(k)}) = I$, buscamos $[J_F(x^{(k)})]^{-1}$ en la forma:

$$\begin{aligned} [J_F(x^{(k)})]^{-1} &= [I + X_2 e_k + X_3 e_k^2 + X_4 e_k^3 + X_5 e_k^4 + X_6 e_k^5 + X_7 e_k^6 + X_8 e_k^7 + X_9 e_k^8 + X_{10} e_k^9] [J_F(\xi)]^{-1} \\ &+ O[e_k^{10}], \end{aligned}$$

donde $X_1 = I$ y $X_s = -\sum_{j=2}^s X_{s-j+1} C_j$ y $s = 2, 3, \dots$. Así obtenemos:

$$\begin{aligned} y^{(k)} &= x^{(k)} - \frac{1}{2} [J_F(x^{(k)})]^{-1} F(x^{(k)}) \\ &= \xi + \frac{1}{2} (e_k + C_2 e_k^2 - M_1 e_k^3 - M_2 e_k^4 - M_3 e_k^5 - M_4 e_k^6 - M_5 e_k^7 - M_6 e_k^8 - M_7 e_k^9 - M_8 e_k^{10}) + O[e_k^{11}], \end{aligned}$$

donde $M_s = C_{s+2} + \sum_{j=1}^s X_{j+1} C_{s-j+2} + X_{s+2}$ y $s = 1, 2, \dots$. De nuevo desarrollamos en serie de Taylor para obtener $F(y^{(k)})$:

$$F(y^{(k)}) = J_F(\xi) \left[\frac{1}{2} e_k + Q_2 e_k^2 + Q_3 e_k^3 + Q_4 e_k^4 + Q_5 e_k^5 + Q_6 e_k^6 + Q_7 e_k^7 + Q_8 e_k^8 + Q_9 e_k^9 + Q_{10} e_k^{10} \right] + O[e_k^{11}],$$

donde

$$\begin{aligned} Q_2 &= \frac{3}{4} C_2, \\ Q_3 &= -M_1 + \frac{1}{2} C_2^2 + \frac{1}{8} C_3, \\ Q_4 &= -M_2 + \frac{1}{4} C_2 \alpha_1 + \frac{3}{8} C_3 C_2 + \frac{1}{16} C_4, \\ Q_5 &= -M_3 - \frac{1}{4} C_2 \alpha_2 + \frac{3}{8} C_3 \beta_1 + \frac{1}{4} C_4 C_2 + \frac{1}{36} C_5, \\ Q_6 &= -\frac{1}{2} M_4 + \frac{1}{4} C_2 \alpha_3 - \frac{1}{8} C_3 \beta_2 + \frac{1}{16} C_4 \gamma_1 + \frac{5}{32} C_5 C_2 + \frac{1}{64} C_6, \\ Q_7 &= -\frac{1}{2} M_5 + \frac{1}{4} C_2 \alpha_4 + \frac{1}{8} C_3 \beta_3 + \frac{1}{16} C_4 \gamma_2 + \frac{1}{32} C_5 \delta_1 + \frac{3}{32} C_6 C_2 + \frac{1}{128} C_7, \\ Q_8 &= -\frac{1}{2} M_6 + \frac{1}{4} C_2 \alpha_5 + \frac{1}{8} C_3 \beta_4 + \frac{1}{16} C_4 \gamma_3 + \frac{1}{36} C_5 \delta_2 + \frac{1}{64} C_6 (\delta_1 + 5C_2^2 - M_1) + \frac{7}{128} C_7 C_2 + \frac{9}{512} C_8, \\ Q_9 &= -\frac{1}{2} M_7 + \frac{1}{4} C_2 \alpha_6 + \frac{1}{8} C_3 \beta_5 + \frac{1}{16} C_4 \gamma_4 + \frac{1}{32} C_5 \delta_3 + \frac{1}{64} C_6 (\delta_2 + C_2 \delta_1 - 5M_1 C_2 - C_2) \\ &+ \frac{1}{32} C_7 (\delta_1 + 8C_2^2 - 2M_1) + \frac{1}{256} C_8 C_2 + \frac{1}{512} C_9, \\ Q_{10} &= -\frac{1}{2} M_8 + \frac{1}{4} C_2 \alpha_7 + \frac{1}{8} C_3 \beta_6 + \frac{1}{16} C_4 \gamma_5 + \frac{1}{32} C_5 \delta_4 + \frac{1}{64} (\delta_3 + C_2 \delta_2 - M_1 \delta_1 - 5M_2 C_2 - M_3) \\ &+ \frac{1}{128} C_7 (\delta_2 + 2C_2 \delta_1 - 11M_1 C_2 - 2M_2 + 5C_2^3 - C_2 M_1) + \frac{1}{32} C_8 (\delta_1 + 15C_2^2 - 3M_1) \\ &+ \frac{1}{512} C_9 C_2 + \frac{1}{1024} C_{10}, \end{aligned}$$

con

$$\begin{aligned}
\alpha_1 &= -2M_1 + C_2^2, \\
\alpha_2 &= -2M_2 - C_2M_1 - M_1C_2, \\
\alpha_3 &= -2M_3 - C_2M_2 + M_1^2 - M_2C_2, \\
\alpha_4 &= -2M_4 - C_2M_3 + M_1M_2 + M_2M_1 - M_3C_2, \\
\alpha_5 &= -2M_5 - C_2M_4 + M_1M_3 + M_2^2 + M_3M_1 - M_4C_2, \\
\alpha_6 &= -2M_6 - C_2M_5 + M_1M_4 + M_2M_3 + M_3M_2 + M_4M_1 - M_5C_2, \\
\alpha_7 &= -2M_7 - C_2M_6 + M_1M_5 + M_2M_4 + M_3^2 + M_4M_2 + M_5M_1 - M_6C_2, \\
\beta_1 &= \alpha_1 + 2C_2^2 - M_1, \\
\beta_2 &= \alpha_2 + C_2\alpha_1 - 2M_1C_2 - M_2, \\
\beta_3 &= \alpha_3 + C_2\alpha_2 - M_1\alpha_1 - 2M_2C_2 - M_3, \\
\beta_4 &= \alpha_4 + C_2\alpha_3 - M_1\alpha_2 - M_2\alpha_1 - 2M_3C_2 - M_4, \\
\beta_5 &= \alpha_5 + C_2\alpha_4 - M_1\alpha_3 - M_2\alpha_2 - M_3\alpha_1 - 2M_4C_2 - M_5, \\
\beta_6 &= \alpha_6 + C_2\alpha_5 - M_1\alpha_4 - M_2\alpha_3 - M_3\alpha_2 - M_4\alpha_1 - 2M_5C_2 - M_6, \\
\gamma_1 &= \beta_1 + 3C_2^2 - M_1, \\
\gamma_2 &= \beta_2 + C_2\beta_1 - 3M_1C_2 - M_2, \\
\gamma_3 &= \beta_3 + C_2\beta_2 - M_1\beta_1 - 3M_2C_2 - M_3, \\
\gamma_4 &= \beta_4 + C_2\beta_3 - M_1\beta_2 - M_2\beta_1 - 3M_3C_2 - M_4, \\
\gamma_5 &= \beta_5 + C_2\beta_4 - M_1\beta_3 - M_2\beta_2 - M_3\beta_1 - 3M_4C_2 - M_5, \\
\delta_1 &= \gamma_1 + 4C_2^2 - M_1, \\
\delta_2 &= \gamma_2 + C_2\gamma_1 - 4M_1C_2 - M_2, \\
\delta_3 &= \gamma_3 + C_2\gamma_2 - M_1\gamma_1 - 4M_2C_2 - M_3, \\
\delta_4 &= \gamma_4 + C_2\gamma_3 - M_1\gamma_2 - M_2\gamma_1 - 4M_3C_2 - M_4.
\end{aligned}$$

De la misma manera, desarrollando en serie de Taylor, calculamos $J_F(y^{(k)})$:

$$J_F(y^{(k)}) = J_F(\xi) [I + C_2e_k + T_2e_k^2 + T_3e_k^3 + T_4e_k^4 + T_5e_k^5 + T_6e_k^6 + T_7e_k^7 + T_8e_k^8 + T_9e_k^9] + O[e_k^{10}],$$

donde

$$\begin{aligned}
T_2 &= C_2^2 + \frac{3}{4}C_3, \\
T_3 &= -C_2M_1 + \frac{3}{2}C_3C_2 + \frac{1}{2}C_4, \\
T_4 &= -C_2M_2 + \frac{3}{4}C_3\alpha_1 - \frac{3}{2}C_4C_2 + \frac{5}{16}C_5, \\
T_5 &= -C_2M_2 - \frac{3}{4}C_3\alpha_2 + \frac{1}{2}C_4\beta_1 + \frac{5}{4}C_5C_2 + \frac{3}{16}C_6, \\
T_6 &= -C_2M_3 - \frac{3}{4}C_3\alpha_3 + \frac{1}{2}C_4\beta_2 + \frac{5}{16}C_5\gamma_1 + \frac{15}{16}C_6C_2 + \frac{7}{64}C_7, \\
T_7 &= -C_2M_4 - \frac{3}{4}C_3\alpha_4 + \frac{1}{2}C_4\beta_3 + \frac{5}{16}C_5\gamma_2 + \frac{3}{16}C_6\delta_1 + \frac{21}{32}C_7C_2 + \frac{1}{16}C_8, \\
T_8 &= -C_2M_5 - \frac{3}{4}C_3\alpha_5 + \frac{1}{2}C_4\beta_4 + \frac{5}{16}C_5\gamma_3 + \frac{3}{16}C_6\delta_2 + \frac{7}{64}C_7(\delta_1 + 5C_2^2 - M_1) + \frac{7}{16}C_8C_2 + \frac{1}{256}C_9, \\
T_9 &= -C_2M_6 - \frac{3}{4}C_3\alpha_6 + \frac{1}{2}C_4\beta_5 + \frac{5}{16}C_5\gamma_4 + \frac{3}{16}C_6\delta_3 + \frac{7}{64}C_7(\delta_2 + C_2\delta_1 - 5M_1C_2 - M_2) \\
&\quad + \frac{1}{16}C_8C(\delta_2 + 2C_2\delta_1 - 11M_1C_2 - C_2M_1 + 5C_2^3 - 2M_2) + \frac{9}{32}C_9C_2 + \frac{5}{256}C_{10}.
\end{aligned}$$

Calculamos las siguientes expresiones:

$$\begin{aligned}
3J_F(x^{(k)}) - 4J_F(y^{(k)}) &= J_F(\xi) [e_k + L_3e_k^3 + L_4e_k^4 + L_5e_k^5 + L_6e_k^6 + L_7e_k^7 + L_8e_k^8 + L_9e_k^9 + L_{10}e_k^{10}] + O[e_k^{11}], \\
J_F(x^{(k)}) - 2J_F(y^{(k)}) &= J_F(\xi) [I + N_2e_k^2 + N_3e_k^3 + N_4e_k^4 + N_5e_k^5 + N_6e_k^6 + N_7e_k^7 + N_8e_k^8 + N_9e_k^9] + O[e_k^{10}],
\end{aligned}$$

con $L_s = 3C_s - 4Q_s$, $s = 3, 4, \dots$ y $N_s = -(s+1)C_{s+1} + 2T_s$, $s = 2, 3, \dots$

Sabiendo que $[J_F(x^{(k)}) - 2J_F(y^{(k)})]^{-1}[J_F(x^{(k)}) - 2J_F(y^{(k)})] = I$, calculamos

$$[J_F(x^{(k)}) - 2J_F(y^{(k)})]^{-1} = [I + Y_3e_k^2 + Y_4e_k^3 + Y_5e_k^4 + Y_6e_k^5 + Y_7e_k^6 + Y_8e_k^7 + Y_9e_k^8 + Y_{10}e_k^9] [J_F(\xi)]^{-1} + O[e_k^{10}],$$

donde $Y_3 = -N_2$, $Y_4 = -N_3$ y $Y_s = -N_{s-1} - \sum_{j=2}^{s-3} Y_{s-j}N_j$ para $s = 5, 6, \dots$. Así, podemos obtener $z^{(k)}$:

$$\begin{aligned} z^{(k)} &= x^{(k)} + [J_F(x^{(k)}) - 2J_F(y^{(k)})]^{-1}[3J_F(x^{(k)}) - 4J_F(y^{(k)})] \\ &= \xi - J_4e_k^4 - J_5e_k^5 - J_6e_k^6 - J_7e_k^7 - J_8e_k^8 - J_9e_k^9 - J_{10}e_k^{10} + O[e_k^{11}], \end{aligned}$$

donde $J_4 = L_4 + Y_4$, $J_s = L_s + \sum_{j=3}^{s-2} Y_{s-j+1}L_j + Y_s$, para $s = 5, 6, \dots$, con lo que se demuestra que los dos primeros pasos del método son, en sí mismos, un método de orden 4. Denotamos el error en este paso como $e_{z^{(k)}}$:

$$e_{z^{(k)}} = -J_4e_k^4 - J_5e_k^5 - J_6e_k^6 - J_7e_k^7 - J_8e_k^8 - J_9e_k^9 - J_{10}e_k^{10} + O[e_k^{11}].$$

De nuevo, desarrollamos en serie de Taylor $F(z^{(k)})$

$$\begin{aligned} F(z^{(k)}) &= J_F(\xi)[-J_4e_k^4 - J_5e_k^5 - J_6e_k^6 - J_7e_k^7 + (C_2J_4^2 - J_8)e_k^8 \\ &\quad + [C_2J_4J_5 + J_5J_4] - J_9]e_k^9 + [C_2(J_4J_6 + J_5^2 + J_6J_4) - J_{10}]e_k^{10} + O[e_k^{11}], \end{aligned}$$

y obtenemos $u^{(k)}$

$$\begin{aligned} u^{(k)} &= z^{(k)} + [J_F(x^{(k)}) - 2J_F(y^{(k)})]^{-1}F(z^{(k)}) \\ &= \xi + S_6e_k^6 + S_7e_k^7 + S_8e_k^8 + S_9e_k^9 + S_{10}e_k^{10} + O[e_k^{11}], \end{aligned}$$

donde

$$\begin{aligned} S_6 &= Y_3J_4, \\ S_7 &= Y_3J_5 + Y_4J_4, \\ S_8 &= Y_3J_6 + Y_5J_4 + J_4J_5 - C_2J_4, \\ S_9 &= Y_3J_7 + Y_4J_6 + J_5J_5 + Y_6J_4 - C_2(J_4J_5 + J_5J_4), \\ S_{10} &= Y_3(J_8 - C_2J_4) + Y_4J_7 + Y_5J_6 + Y_6J_5 + Y_7J_4 - C_2(J_4J_6 + J_5^2 + J_6J_4). \end{aligned}$$

Denotamos el error en este paso con $e_{u^{(k)}}$

$$e_{u^{(k)}} = S_6e_k^6 + S_7e_k^7 + S_8e_k^8 + S_9e_k^9 + S_{10}e_k^{10} + O[e_k^{11}].$$

Pretendemos desarrollar $J_F(\eta_i^{(k)})$ con $\eta_i^{(k)} = \frac{(1+\tau_i)u^{(k)} + (1-\tau_i)z^{(k)}}{2} = \xi + \frac{1}{2}[(e_{u^{(k)}}) + e_{z^{(k)}}] + \tau_i(e_{u^{(k)}} - e_{z^{(k)}})$, teniendo

$$(\eta_i^{(k)}) - \xi = \frac{1}{2}[(e_{u^{(k)}}) + e_{z^{(k)}}] + \tau_i(e_{u^{(k)}} - e_{z^{(k)}}),$$

con

$$\begin{aligned} e_{u^{(k)}} + e_{z^{(k)}} &= -J_4e_k^4 - J_5e_k^5 + (S_6 - J_6)e_k^6 + (S_7 - J_7)e_k^7 \\ &\quad + (S_8 - J_8)e_k^8 + (S_9 - J_9)e_k^9 + (S_{10} - J_{10})e_k^{10} + O[e_k^{11}], \\ e_{u^{(k)}} - e_{z^{(k)}} &= J_4e_k^4 + J_5e_k^5 + (S_6 + J_6)e_k^6 + (S_7 + J_7)e_k^7 \\ &\quad + (S_8 + J_8)e_k^8 + (S_9 + J_9)e_k^9 + (S_{10} + J_{10})e_k^{10} + O[e_k^{11}]. \end{aligned}$$

De nuevo desarrollamos en serie de Taylor:

$$J_F(\eta_i^{(k)}) = J_F(\xi) \left[I + 2C_2(\eta_i^{(k)} - \xi) + 3C_3(\eta_i^{(k)} - \xi)^2 \right] + O[e_k^{11}].$$

Calculamos $K = \sum_{i=1}^m \omega_i J_F(\eta_i^{(k)})$ usando la notación (3.18) y el procedimiento establecido en la Sección 3.2.2:

$$K = \sigma J_F(\xi) (I + H_4 e_k^4 + H_5 e_k^5 + H_6 e_k^6 + H_7 e_k^7 + H_8 e_k^8 + H_9 e_k^9) + O[e_k^{10}],$$

donde

$$\begin{aligned} H_4 &= (\sigma_1 - 1)C_2 J_4, \\ H_5 &= (\sigma_1 - 1)C_2 J_5, \\ H_6 &= C_2(S_6 - J_6) + \sigma_1 C_2(S_6 + J_6), \\ H_7 &= C_2(S_7 - J_7) + \sigma_1 C_2(S_7 + J_7), \\ H_8 &= C_2(S_8 - J_8) + \sigma_1 C_2(S_8 + J_8) + \frac{3}{4}(1 - 2\sigma_1 + \sigma_2)C_3 J_4^2, \\ H_9 &= C_2(S_9 - J_9) + \sigma_1 C_2(S_9 + J_9) + \frac{3}{4}(1 - 2\sigma_1 + \sigma_2)C_3(J_4 J_5 + J_5 J_4). \end{aligned}$$

Así,

$$\begin{aligned} K^{-1} &= \frac{1}{\sigma} (I - H_4 e_k^4 - H_5 e_k^5 - H_6 e_k^6 - H_7 e_k^7 - (H_8 - H_4^2) e_k^8 - (H_9 - H_4 H_5 - H_5 H_4) e_k^9) [J_F(\xi)]^{-1} \\ &+ O[e_k^{10}]. \end{aligned}$$

Calculamos

$$\begin{aligned} x^{(k+1)} &= z^{(k)} - 2K^{-1}F(z^{(k)}) \\ &= \xi - (J_4 - \frac{2}{\sigma}J_4)e_k^4 - (J_5 - \frac{2}{\sigma}J_5)e_k^5 - (J_6 - \frac{2}{\sigma}J_6)e_k^6 - (J_7 - \frac{2}{\sigma}J_7)e_k^7 \\ &- [J_8 - \frac{2}{\sigma}(C_2 J_4^2 - J_8 - Z_4 J_4)]e_k^8 - [J_9 - (C_2(J_4 J_5 + J_5 J_4) - J_9 - Z_4 J_5 - Z_5 J_4)]e_k^9 \\ &- [J_{10} - \frac{2}{\sigma}(C_2(J_4 J_6 + J_5^2 + J_6 J_4 + J_6 J_4 + S_6 J_4 - J_4 S_6) - J_{10} - Z_4 Z_6 - Z_5 J_5 - Z_6 Z_4)]e_k^{10} \\ &+ O[e_k^{11}] \end{aligned}$$

Notamos que para tener orden ocho σ debe ser igual a 2. En este caso la ecuación del error será:

$$\begin{aligned} e_{k+1} &= [(C_2 J_4^2 - Z_4 J_4)e_k^8 + (C_2(J_4 J_5 + J_5 J_4) - Z_4 J_5 - Z_5 J_4)] e_k^9 \\ &+ [(C_2(J_4 J_6 + J_5^2 + J_6 J_4 + J_6 J_4 + S_6 J_4 - J_4 S_6) - Z_4 Z_6 - Z_5 J_5 - Z_6 Z_4)] e_k^{10} + O[e_k^{11}]. \end{aligned}$$

Simplificamos la ecuación del error:

$$\begin{aligned} e_{k+1} &= -\sigma_1 C_2 J_4^2 e_k^8 - \sigma_1 C_2 (J_4 J_5 + J_5 J_4) e_k^9 \\ &- [C_2 S_6 J_4 - \sigma_1 C_2 (J_4 J_6 + J_5^2 + S_6 J_4) + J_6 J_4 + J_6 J_4 + S_6 J_4 - J_4 S_6] e_k^{10} + O[e_k^{11}], \end{aligned}$$

de donde concluimos que si $\sigma_1 = 0$ el orden de convergencia será diez:

$$e_{k+1} = -C_2 S_6 J_4 e_k^{10} + O[e_k^{11}]. \quad (5.52)$$

Sustituyendo los valores de S_6 y J_4 se demuestra fácilmente que la ecuación del error tiene la siguiente expresión:

$$e_{k+1} = -\frac{1}{64} C_2 (8C_2^2 - 9C_3) (47C_4 - 64C_2 C_3 - 66C_3 C_2 + 92C_2^3)^2 e_k^{10} + O[e_k^{11}].$$

□

El caso particular de esta familia de métodos iterativos en el que la cuadratura gaussiana empleada es la regla de punto medio, se denota por M10S y será utilizado más adelante en las pruebas numéricas.

En la Figura 5.4 de nuevo comparamos los I e IC del método desarrollado con los métodos de Newton, Traub, Jarratt y los métodos TrG-LD1S y M5S. El método M10S tiene mayor I que el de Newton para todo tamaño

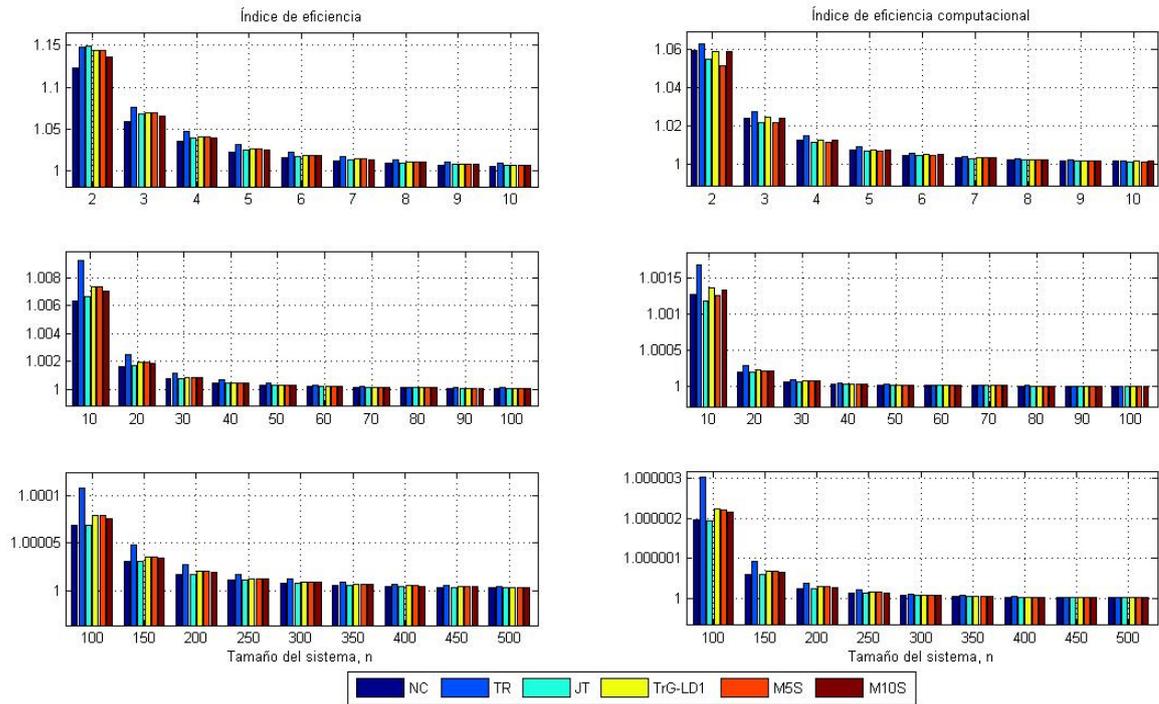


Figura 5.4: Índice de eficiencia, I e índice de eficiencia computacional, IC

del sistema y mayor que el de Jarratt para sistemas con $n > 3$. Su índice de eficiencia computacional es mayor que el de Newton y Jarratt para todo tamaño de sistema y mayor que el de método M5S para sistemas con $n < 100$. Vemos de nuevo que el método de Traub tiene los mejores índices para todo tamaño de sistemas con la excepción del caso de sistema con $n = 2$ cuando el método de Jarratt tiene mejor I.

Por otra parte, en el Capítulo 4 hemos desarrollado una familia de métodos iterativos óptimos de orden 4 que extendimos hasta orden 8 con cinco evaluaciones funcionales. Este método puede ser adaptado fácilmente a sistemas. Consideramos el elemento de esta familia descrito por la expresión (4.9) que adaptada a sistemas resulta:

$$\begin{aligned}
 y^{(k)} &= x^{(k)} - \frac{2}{3} \left[J_F \left(x^{(k)} \right) \right]^{-1} F \left(x^{(k)} \right), \\
 z^{(k)} &= y^{(k)} + \frac{1}{6} \left[J_F \left(x^{(k)} \right) \right]^{-1} F \left(x^{(k)} \right), \\
 u^{(k)} &= z^{(k)} + \left[J_F \left(x^{(k)} \right) - 3J_F \left(y^{(k)} \right) \right]^{-1} F \left(x^{(k)} \right), \\
 v^{(k)} &= z^{(k)} + \left[J_F \left(x^{(k)} \right) - 3J_F \left(y^{(k)} \right) \right]^{-1} \left[F \left(x^{(k)} \right) + 2F \left(u^{(k)} \right) \right], \\
 x^{(k+1)} &= v^{(k)} - \frac{1}{2} \left[J_F \left(x^{(k)} \right) \right]^{-1} \left[5J_F \left(x^{(k)} \right) - 3J_F \left(y^{(k)} \right) \right] \left[J_F \left(x^{(k)} \right) \right]^{-1} F \left(v^{(k)} \right).
 \end{aligned}
 \tag{5.53}$$

Denotaremos en lo sucesivo a éste método por M8A.

Teorema 5.2.3 Sea $F : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ suficientemente diferenciable en Ω y $\xi \in \Omega$ solución del sistema de ecuaciones no lineales $F(x) = 0$. Supongamos que $J_F(x)$ es continua y no singular en ξ . Entonces, la sucesión $\{x^{(k)}\}_{k \geq 0}$ obtenida usando la expresión (5.53) converge a ξ con convergencia de orden 8.

Demostración: Utilizando los desarrollos en serie de Taylor de las funciones $F(x^{(k)})$ y $J_F(x^{(k)})$ obtenemos

$$\begin{aligned} [J_F(x^{(k)})]^{-1} F(x^{(k)}) &= e_k + M_2 e_k^2 + M_3 e_k^3 + M_4 e_k^4 + M_5 e_k^5 + M_6 e_k^6 + M_7 e_k^7 + M_8 e_k^8 \\ &+ O[e_k^9], \end{aligned} \quad (5.54)$$

donde $M_2 = C_2 + X_4$ y

$$M_s = C_s + \sum_{j=3}^s X_{s-j+2} C_{j-1} + X_{s+2}, \quad s = 3, 4, \dots$$

Por lo tanto, las expresiones de $y^{(k)}$ y $z^{(k)}$ serán:

$$\begin{aligned} y^{(k)} &= x^{(k)} - \frac{2}{3} [J_F(x^{(k)})]^{-1} F(x^{(k)}) = \xi + \frac{1}{3} e_k - \frac{2}{3} M, \\ z^{(k)} &= y^{(k)} + \frac{1}{6} [J_F(x^{(k)})]^{-1} F(x^{(k)}) = \xi + \frac{1}{2} e_k - \frac{1}{2} M, \end{aligned}$$

donde $M = M_2 e_k^2 + M_3 e_k^3 + M_4 e_k^4 + M_5 e_k^5 + M_6 e_k^6 + M_7 e_k^7 + M_8 e_k^8 + O[e_k^9]$. De nuevo desarrollamos en serie de Taylor para obtener $J_F(y^{(k)})$.

$$J_F(y^{(k)}) = J_F(\xi) [I + Q_1 e_k + Q_2 e_k^2 + Q_3 e_k^3 + Q_4 e_k^4 + Q_5 e_k^5 + Q_6 e_k^6 + Q_7 e_k^7 + Q_8 e_k^8] + O[e_k^9],$$

donde

$$\begin{aligned} Q_1 &= \frac{2}{3} C_2, \\ Q_2 &= \frac{1}{3} C_3 - \frac{4}{3} C_2 M_2, \\ Q_3 &= \frac{4}{27} C_4 - \frac{4}{3} C_3 M_2 - \frac{4}{3} C_2 M_3, \\ Q_4 &= \frac{5}{81} C_5 - \frac{8}{9} C_4 M_2 + \frac{4}{3} C_3 (M_2^2 - 2M_3) - \frac{4}{3} C_2 M_4, \\ Q_5 &= \frac{2}{81} C_6 - \frac{40}{81} C_5 M_2 + \frac{4}{9} C_4 (M_2^2 - M_3) + \frac{4}{3} C_3 (M_2 M_3 + M_3 M_2 - M_4) - \frac{4}{3} C_2 M_5, \\ Q_6 &= \frac{7}{729} C_7 - \frac{20}{81} C_6 M_2 + \frac{40}{81} C_5 (3M_2^2 - M_3) + \frac{8}{27} C_4 (6M_2 M_3 + 6M_3 M_2 - 3M_4 - 4M_2^3) - C_2 M_6, \\ Q_7 &= -\frac{4}{3} C_2 M_7 + \frac{4}{3} C_3 (M_2 M_5 + M_3 M_4 + M_4 M_3 + M_5 M_2 + M_6 M_2 - M_7) \\ &+ \frac{8}{27} C_4 (6M_2 M_4 + 6M_3^2 + 6M_4 M_2 - 3M_5 - 4M_2^2 M_3 - 4M_2 M_3 M_2 - 4M_3 M_2^2) \\ &+ \frac{40}{81} C_5 (3M_2 M_3 + 3M_3 M_2 - M_4 - 4M_2^3) + \frac{20}{81} C_6 (4M_2^2 - M_3) - \frac{28}{243} C_7 M_2 + \frac{8}{2187} C_8, \\ Q_8 &= -\frac{4}{3} C_2 M_8 + \frac{4}{3} C_3 (M_2 M_6 + M_3 M_5 + M_4^2 + M_5 M_3 + M_6 M_2 - M_7) \\ &+ \frac{16}{9} C_4 (M_2 M_5 + M_3 M_4 + M_4 M_3 + M_5 M_2 - M_6) \\ &+ \frac{32}{27} C_4 (M_2^2 M_4 + M_2 M_3^2 M_4 M_2 + M_3 M_2 M_3 + M_3^2 M_2 + M_4 M_2^2) \\ &+ \frac{40}{81} C_5 (-M_5 - 4M_2^2 M_3 - 4M_2 M_3 M_2 - 4M_3 M_2^2 + 3M_2 M_4 + 3M_3^2 + 3M_4 M_2 + 2M_4) \\ &+ \frac{20}{41} C_6 (4M_2 M_3 + 4M_3 M_2 - M_4 - 8M_2^3) \\ &+ \frac{28}{243} C_7 (5M_2^2 - M_3) - \frac{112}{2187} C_8 M_2 + \frac{1}{729} C_9. \end{aligned}$$

Entonces, obtenemos la expresión de $J_F(x^{(k)}) - 3J_F(y^{(k)})$ en la forma:

$$J_F(x^{(k)}) - 3J_F(y^{(k)}) = J_F(\xi) [-2I + A_1 e_k + A_2 e_k^2 + A_3 e_k^3 + A_4 e_k^4 + A_5 e_k^5 + A_6 e_k^6 + A_7 e_k^7 + A_8 e_k^8] + O[e_k^9],$$

donde $A_s = (s+1)C_{s+1} - 3Q_s$, $s = 1, 2, \dots$. Calculamos $[J_F(x^{(k)}) - 3J_F(y^{(k)})]^{-1}$ obteniendo:

$$\left[J_F(x^{(k)}) - 3J_F(y^{(k)}) \right]^{-1} = \left[-\frac{1}{2}I + Y_2e_k + Y_3e_k^2 + Y_4e_k^3 + Y_5e_k^4 + Y_6e_k^5 + Y_7e_k^6 + Y_8e_k^7 \right] [J_F(\xi)]^{-1} + O[e_k^8],$$

donde $Y_2 = 0$ y $Y_s = \frac{1}{2} \sum_{j=3}^s Y_{s-j+2} A_{j-2} - \frac{1}{4} A_{s-1}$, $s = 3, 4, \dots$

A continuación, calculamos $[J_F(x^{(k)}) - 3J_F(y^{(k)})]^{-1} F(x^{(k)})$ y obtenemos

$$\left[J_F(x^{(k)}) - 3J_F(y^{(k)}) \right]^{-1} F(x^{(k)}) = -\frac{1}{2}e_k + R_2e_k^2 + R_3e_k^3 + R_4e_k^4 + R_5e_k^5 + R_6e_k^6 + R_7e_k^7 + R_8e_k^8 + O[e_k^9],$$

donde $R_2 = Y_2 - \frac{1}{2}C_2$ y $R_s = Y_s + \sum_{j=3}^s Y_{s-j+2}C_{j-1} - \frac{1}{2}C_s$, $s = 3, 4, \dots$

Calculamos $u^{(k)} = z^{(k)} + [J_F(x^{(k)}) - 3J_F(y^{(k)})]^{-1} F(x^{(k)})$ y escribimos la ecuación del error en este paso como:

$$\begin{aligned} e_{u^{(k)}} &= \frac{1}{2}e_k - \frac{1}{2}M + \left[J_F(x^{(k)}) - 3J_F(y^{(k)}) \right]^{-1} F(x^{(k)}) \\ &= \frac{1}{2}e_k - \frac{1}{2} \left[M_2e_k^2 + M_3e_k^3 + M_4e_k^4 + M_5e_k^5 + M_6e_k^6 + M_7e_k^7 + M_8e_k^8 \right] \\ &\quad - \frac{1}{2}e_k + R_2e_k^2 + R_3e_k^3 + R_4e_k^4 + R_5e_k^5 + R_6e_k^6 + R_7e_k^7 + R_8e_k^8 + O[e_k^9] \\ &= P_2e_k^2 + P_3e_k^3 + P_4e_k^4 + P_5e_k^5 + P_6e_k^6 + P_7e_k^7 + P_8e_k^8 + O[e_k^9], \end{aligned} \quad (5.55)$$

donde $P_s = -\frac{1}{2}M_s + R_s$ y $s = 2, 3, \dots$, de donde concluimos que para tener convergencia de orden 4, los dos primeros elementos deben ser nulos. Esto se muestra fácilmente calculando P_2 y P_3 retomando los valores de los elementos que los definen. Así, la expresión (5.55) queda:

$$e_{u^{(k)}} = P_4e_k^4 + P_5e_k^5 + P_6e_k^6 + P_7e_k^7 + P_8e_k^8 + O[e_k^9]. \quad (5.56)$$

Utilizando la expresión (5.56) obtenemos $F(u^{(k)})$ desarrollando en serie de Taylor en torno a ξ . Entonces,

$$F(u^{(k)}) = J_F(\xi) [P_4e_k^4 + P_5e_k^5 + P_6e_k^6 + P_7e_k^7 + (P_8 + P_4^2)e_k^8] + O[e_k^9].$$

Ahora, calculamos:

$$\begin{aligned} F(x^{(k)}) + 2F(u^{(k)}) &= J_F(\xi) [e_k + C_2e_k^2 + C_3e_k^3 + (C_4 + 2P_4)e_k^4 + (C_5 + 2P_5)e_k^5] \\ &\quad + J_F(\xi) [(C_6 + 2P_6)e_k^6 + (C_7 + 2P_7)e_k^7 + (C_8 + 2P_8)e_k^8] + O[e_k^9], \end{aligned}$$

y expresamos el producto $[J_F(x^{(k)}) - 3J_F(y^{(k)})]^{-1} [F(x^{(k)}) + 2F(u^{(k)})]$ como

$$\begin{aligned} \left[J_F(x^{(k)}) - 3J_F(y^{(k)}) \right]^{-1} [F(x^{(k)}) + 2F(u^{(k)})] &= -\frac{1}{2}e_k + L_2e_k^2 + L_3e_k^3 + L_4e_k^4 + L_5e_k^5 + L_6e_k^6 \\ &\quad + L_7e_k^7 + L_8e_k^8 + O[e_k^9], \end{aligned}$$

donde $L_2 = Y_2 - \frac{1}{2}(C_2 + P_2)$ y $L_s = Y_s + \sum_{j=3}^s Y_{s-j+2}C_{j-1} - \frac{1}{2}(C_s + P_s)$, $s = 3, 4, \dots$. Por tanto, obtenemos

para $v^{(k)}$ la expresión:

$$\begin{aligned} v^{(k)} &= z_k + \left[J_F(x^{(k)}) - 3J_F(y^{(k)}) \right]^{-1} [F(x^{(k)}) + 2F(u^{(k)})] \\ &= \xi + N_2e_k^2 + N_3e_k^3 + N_4e_k^4 + N_5e_k^5 + N_6e_k^6 + N_7e_k^7 + N_8e_k^8 + O[e_k^9], \end{aligned}$$

donde $N_s = L_s - \frac{1}{2}M_s$ y $s = 2, 3, \dots$. Fácilmente se demuestra, que N_2, N_3, N_4 y N_5 se anulan, retomando los valores de los elementos que los definen. Así obtenemos

$$v^{(k)} = \xi + N_6e_k^6 + N_7e_k^7 + N_8e_k^8 + O[e_k^9],$$

y esto prueba que los primeros cuatro pasos de (5.53) son en si mismos un método de orden seis. Continuamos la demostración con el siguiente paso aproximando $F(v^{(k)})$

$$F(v^{(k)}) = J_F(\xi) [N_6 e_k^6 + N_7 e_k^7 + N_8 e_k^8] + O[e_k^9].$$

Calculamos

$$5J_F(x^{(k)}) - 3J_F(y^{(k)}) = J_F(\xi) [2I + B_1 e_k + B_2 e_k^2 + B_3 e_k^3 + B_4 e_k^4 + B_5 e_k^5 + B_6 e_k^6 + B_7 e_k^7 + B_8 e_k^8] + O[e_k^9],$$

donde $B_s = 5(s+1)C_{s+1} - 3Q_s$ y $s = 1, \dots$. Ahora desarrollamos los productos:

$$\begin{aligned} \left[J_F(x^{(k)}) \right]^{-1} [5J_F(x^{(k)}) - 3J_F(y^{(k)})] &= 2I + S_1 e_k + S_2 e_k^2 + S_3 e_k^3 + S_4 e_k^4 + S_5 e_k^5 \\ &+ S_6 e_k^6 + S_7 e_k^7 + S_8 e_k^8 + O[e_k^9], \end{aligned} \quad (5.57)$$

$$\begin{aligned} \left[J_F(x^{(k)}) \right]^{-1} F(v^{(k)}) &= 2N_6 e_k^6 + (N_7 + X_2 N_6) e_k^7 \\ &+ (N_8 + X_2 N_7 + X_3 N_6) e_k^8 + O[e_k^9], \end{aligned} \quad (5.58)$$

donde $S_1 = B_1 + 2X_2$ y $S_s = B_s + \sum_{j=2}^s X_{s-j+2} B_{j-1} + 2X_{s+1}$ y $s = 2, 3, \dots$. Ahora, multiplicando las expresiones (5.57) y (5.58) obtenemos

$$\left[J_F(x^{(k)}) \right]^{-1} [5J_F(x^{(k)}) - 3J_F(y^{(k)})] \left[J_F(x^{(k)}) \right]^{-1} F(v^{(k)}) = 2N_6 e_k^6 + K_7 e_k^7 + K_8 e_k^8 + O[e_k^9],$$

donde $K_7 = 2(N_7 + X_2 N_6) + S_1 N_6$ y $K_8 = 2(N_8 + X_2 N_7 + X_3 N_6) + S_1(N_7 + X_2 N_6) + S_2 N_6$. Finalmente, podemos obtener $x^{(k+1)}$:

$$\begin{aligned} x^{(k+1)} &= v^{(k)} - \left[J_F(x^{(k)}) \right]^{-1} [5J_F(x^{(k)}) - 3J_F(y^{(k)})] \left[J_F(x^{(k)}) \right]^{-1} F(v^{(k)}) \\ &= \xi + \left[L_6 - \frac{1}{2} M_6 - N_6 \right] e_k^6 + \left[L_7 - \frac{1}{2} M_7 - \frac{1}{2} (2(N_7 + X_2 N_6) + S_1 N_6) \right] e_k^7 \\ &+ \left[L_8 - \frac{1}{2} M_8 - \frac{1}{2} (2(N_8 + X_2 N_7 + X_3 N_6) + S_1(N_7 + X_2 N_6) + S_2 N_6) \right] e_k^8. \end{aligned}$$

Entonces, la ecuación de error es:

$$\begin{aligned} e_{k+1} &= \left[L_6 - \frac{1}{2} M_6 - N_6 \right] e_k^6 + \left[L_7 - \frac{1}{2} M_7 - \frac{1}{2} (2(N_7 + X_2 N_6) + S_1 N_6) \right] e_k^7 \\ &+ \left[L_8 - \frac{1}{2} M_8 - \frac{1}{2} (2(N_8 + X_2 N_7 + X_3 N_6) + S_1(N_7 + X_2 N_6) + S_2 N_6) \right] e_k^8 + O[e_k^9]. \end{aligned} \quad (5.59)$$

Fácilmente se demuestra que los dos primeros elementos de la expresión (5.59) se anulan y el orden de convergencia es 8. Finalmente, la ecuación de error (5.59) queda:

$$e_{k+1} = \left(C_2^2 - \frac{1}{2} C_3 \right) \left(2C_2^3 + 2C_3 C_2 - 2C_2 C_3 - \frac{20}{9} C_4 \right) e_k^8 + O[e_k^9]$$

lo que pone en manifiesto que el método M8A es efectivamente de orden ocho. \square

Es conocido que, aplicando la pseudocomposición, podemos diseñar métodos con alto orden de convergencia. Esta pseudocomposición puede ser aplicada directamente sobre el método M8A, pero también sobre sus tres primeros pasos (a los que denotamos M4A) ó sus cuatro primeros pasos (M6A). De este modo podemos obtener tres métodos más con ordenes 14, 5 y 10, respectivamente. Utilizando M4A como predictor introducimos la cuadratura de Gauss como corrector:

$$x^{(k+1)} = z^{(k)} - 2 \left[\sum_{i=1}^m \omega_i J_F \left(\frac{(1 + \tau_i) u^{(k)} + (1 - \tau_i) z^{(k)}}{2} \right) \right]^{-1} F(z^{(k)}),$$

obtenemos un método de orden 5 al que denotamos por M5AG, si $\sigma = 2$ y $\sigma_1 = 0$. Si tomamos como predictor el método M6A podemos obtener con la pseudocomposición el método de orden 10, M10AG.

$$x^{(k+1)} = u^{(k)} - 2 \left[\sum_{i=1}^m \omega_i J_F \left(\frac{(1 + \tau_i)v^{(k)} + (1 - \tau_i)u^{(k)}}{2} \right) \right]^{-1} F(u^{(k)})$$

(si $\sigma = 2$ y $\sigma_1 = 0$). Finalmente, tomamos (5.53) como predictor y aplicamos como corrector la cuadratura de Gauss,

$$x^{(k+1)} = v^{(k)} - 2 \left[\sum_{i=1}^m \omega_i J_F \left(\frac{(1 + \tau_i)w^{(k)} + (1 - \tau_i)v^{(k)}}{2} \right) \right]^{-1} F(v^{(k)}),$$

donde $v^{(k)}$ es el último paso de (5.53). Obtenemos un método de orden 12 si $\sigma = 2$ y, de orden 14, si $\sigma_1 = 0$. Denotamos este método por M14AG. Estudiamos sus índices de eficiencia como lo hemos hecho en la sección anterior, empleando en todos los casos la fórmula de cuadratura de punto medio.

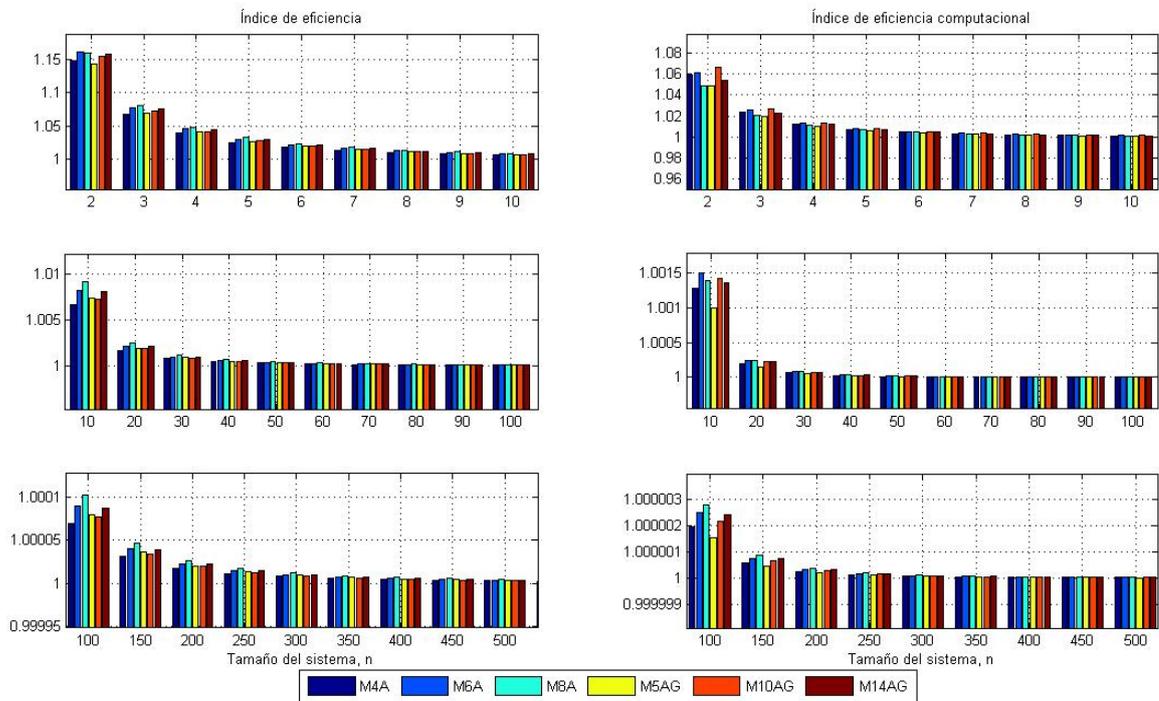


Figura 5.5: Índice de eficiencia, I e índice de eficiencia computacional, IC

La Figura 5.5 presenta el índice de eficiencia y el de eficiencia computacional de los seis métodos desarrollados a partir de M8A en función del tamaño del sistema. Analizando el índice I observamos que para $n = 2$ el mejor método es M6A, siendo M8A el mejor para $n \geq 3$. El método con peor índice I es M5AG para sistemas de tamaño pequeño, $n < 10$, mientras que para sistemas más grandes M4A es el peor. Los métodos obtenidos a partir de la pseudocomposición, M10AG y M14AG, no mejoran los métodos de partida excepto los métodos M5AG y M4A para $n > 2$.

En cuanto al índice de eficiencia computacional, IC, para sistemas pequeños, $n < 10$, el mejor método es M10AG. Si el tamaño del sistema es $10 \leq n < 100$ el mejor índice corresponde a M6A, mientras que para sistemas grandes, $n \geq 100$, el mejor es el M8A. En este caso el peor índice corresponde a M5AG, lo que se justifica porque el orden aumenta en solo una unidad, mientras que se utilizan dos evaluaciones funcionales más y se resuelve un nuevo sistema lineal. Sin embargo, otros métodos pseudocompuestos tienen mejor comportamiento. Por ejemplo, M10AG y M14AG mejoran el índice de sus predictores correspondientes para $n \leq 10$, si bien la tendencia se invierte a medida que el tamaño del sistema crece.

A continuación presentamos otro método de orden ocho que es aplicable para ecuaciones no lineales, pero no es óptimo según la conjetura de Kung-Traub razón por la cual no fue introducido en la sección de métodos óptimos para ecuaciones no lineales. El esquema iterativo de este método es:

$$\begin{aligned}
y^{(k)} &= x^{(k)} - \frac{1}{2}[J_F(x^{(k)})]^{-1}F(x^{(k)}), \\
z^{(k)} &= \frac{1}{3}(4y^{(k)} - x^{(k)}), \\
u^{(k)} &= y^{(k)} + [J_F(x^{(k)}) - 3J_F(z^{(k)})]^{-1}F(x^{(k)}), \\
v^{(k)} &= u^{(k)} + 2[J_F(x^{(k)}) - 3J_F(z^{(k)})]^{-1}F(u^{(k)}), \\
x^{(k+1)} &= v^{(k)} + 2[J_F(x^{(k)}) - 3J_F(z^{(k)})]^{-1}F(v^{(k)}).
\end{aligned} \tag{5.60}$$

al que denotaremos por M8B. Este método, al igual que el método M8A, se puede extender hasta orden 14 aplicando la pseudocomposición.

Teorema 5.2.4 *Sea $F : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ suficientemente diferenciable en Ω y $\xi \in \Omega$ solución del sistema de ecuaciones no lineales $F(x) = 0$. Supongamos que $J_F(x)$ es continua y no singular en ξ . Entonces, la sucesión $\{x^{(k)}\}_{k \geq 0}$ obtenida usando la expresión (5.60) converge a ξ con convergencia de orden 8.*

Demostración: A partir de los desarrollos de Taylor de $F(x^{(k)})$ y $J_F(x^{(k)})$ obtenemos:

$$\begin{aligned}
[J_F(x^{(k)})]^{-1} &= [I + X_2e_k + X_3e_k^2 + X_4e_k^3 + X_5e_k^4 + X_6e_k^5 + X_7e_k^6 + X_8e_k^7 + X_9e_k^8 + X_{10}e_k^9][J_F(\xi)]^{-1} \\
&+ O[e_k^{10}],
\end{aligned}$$

donde $X_1 = I$ y $X_s = -\sum_{j=2}^s X_{s-j+1}C_j$. Así obtenemos:

$$\begin{aligned}
y^{(k)} &= x^{(k)} - \frac{1}{2}[J_F(x^{(k)})]^{-1}F(x^{(k)}) \\
&= \xi + \frac{1}{2}(e_k + C_2e_k^2 - M_1e_k^3 - M_2e_k^4 - M_3e_k^5 - M_4e_k^6 - M_5e_k^7 - M_6e_k^8) + O[e_k^9],
\end{aligned}$$

donde $M_s = C_{s+2} + \sum_{j=1}^s X_{j+1}C_{s-j+2} + X_{s+2}$ y $s = 1, 2, \dots$. Ahora, utilizando los puntos $x^{(k)}$ y $y^{(k)}$ obtenemos:

$$\begin{aligned}
z^{(k)} &= \frac{1}{3}(4y^{(k)} - x^{(k)}) \\
&= \xi + \frac{1}{3}e_k + \frac{1}{2}(C_2e_k^2 - M_1e_k^3 - M_2e_k^4 - M_3e_k^5 - M_4e_k^6 - M_5e_k^7) + O[e_k^8].
\end{aligned}$$

Desarrollando en serie de Taylor la función $J_F(z^{(k)})$ alrededor de ξ obtenemos:

$$J_F(z^{(k)}) = J_F(\xi)[I + \frac{2}{3}C_2e_k + Q_2e_k^2 + Q_3e_k^3 + Q_4e_k^4 + Q_5e_k^5 + Q_6e_k^6 + Q_7e_k^7] + O[e_k^8],$$

donde

$$\begin{aligned}
Q_2 &= \frac{3}{4}C_2^2 + \frac{1}{3}C_3, \\
Q_3 &= -\frac{4}{3}C_2M_1 + \frac{4}{3}C_3C_2 + \frac{4}{27}C_4, \\
Q_4 &= -\frac{4}{3}C_2M_2 + \frac{4}{3}C_3\alpha_1 + \frac{8}{9}C_4C_2 + \frac{5}{81}C_5, \\
Q_5 &= -\frac{4}{3}C_2M_3 - \frac{4}{3}C_3\alpha_2 - 4C_4\beta_1 + \frac{40}{81}C_5C_2 + \frac{2}{81}C_6, \\
Q_6 &= -\frac{4}{3}C_2M_4 + \frac{4}{3}C_3\alpha_3 - 4C_4\beta_2 - 5C_5\gamma_1 + \frac{20}{81}C_6C_2 + \frac{7}{243}C_7, \\
Q_7 &= -\frac{4}{3}C_2M_5 + \frac{4}{3}C_3\alpha_4 - 4C_4\beta_3 - 5C_5\gamma_2 - 6C_6\delta_1 + \frac{28}{81}C_7C_2 + \frac{8}{729}C_8,
\end{aligned}$$

con

$$\begin{aligned}\alpha_1 &= -\frac{4}{9}(M_1 + C_2^2), \\ \alpha_2 &= -\frac{4}{9}(M_2 - C_2M_1 - M_1C_2), \\ \alpha_3 &= -\frac{4}{9}(M_3 - C_2M_2 + M_1^2 - M_2C_2), \\ \alpha_4 &= -\frac{4}{9}(M_4 - C_2M_3 + M_1M_2 + M_2M_1 - M_3C_2), \\ \beta_1 &= -\frac{8}{27}(2\alpha_1 + C_2^2 - 4M_1), \\ \beta_2 &= -\frac{8}{27}(2\alpha_2 + C_2\alpha_1 - M_1C_2 - 4M_2), \\ \beta_3 &= -\frac{8}{27}(2\alpha_3 + C_2\alpha_2 - M_1\alpha_1 - M_2C_2 - 4M_3),\end{aligned}$$

$$\begin{aligned}\gamma_1 &= -\frac{1}{3}\beta_1 + \frac{4}{27}C_2^2 - \frac{2}{81}M_1, \\ \gamma_2 &= -\frac{1}{3}\beta_2 - \frac{2}{3}C_2\beta_1 - \frac{4}{27}3M_1C_2 - \frac{2}{81}M_2,\end{aligned}$$

y

$$\delta_1 = -\frac{1}{3}\gamma_1 + \frac{16}{243}C_2^2 - \frac{2}{243}M_1.$$

Esto nos permite desarrollar $J_F(x^{(k)}) - 3J_F(z^{(k)})$ en la forma:

$$J_F(x^{(k)}) - 3J_F(z^{(k)}) = J_F(\xi)[-2I + T_2e_k^2 + T_3e_k^3 + T_4e_k^4 + T_5e_k^5 + T_6e_k^6 + T_7e_k^7] + O[e_k^8],$$

donde $T_s = (s+1)C_{s+1} - 3Q_s$, $s = 2, 3, \dots$. Así, obtenemos

$$[J_F(x^{(k)}) - 3J_F(z^{(k)})]^{-1} = [-\frac{1}{2}I + Y_3e_k^2 + Y_4e_k^3 + Y_5e_k^4 + Y_6e_k^5 + Y_7e_k^6 + Y_8e_k^7] + O[e_k^8],$$

donde

$$\begin{aligned}Y_3 &= -\frac{1}{4}T_2, \\ Y_4 &= -\frac{1}{4}T_3, \\ Y_5 &= -\frac{1}{8}(2T_4 + T_2^2), \\ Y_6 &= -\frac{1}{16}(4T_5 + 2T_2T_3 + 2T_4T_2 + T_2^3), \\ Y_7 &= -\frac{1}{16}(4T_6 + 2T_2T_4 + 2T_3^2 + 2T_4T_2 + T_2^3), \\ Y_8 &= -\frac{1}{32}(8T_7 + 4T_2T_5 + 4T_3T_4 + 4T_4T_3 + 2T_2^2T_3 + 4T_5T_2 + 2T_2T_3T_2 + 2T_4T_2^2 + T_2^4).\end{aligned}$$

Para calcular $u^{(k)}$ necesitamos obtener primero $[J_F(x^{(k)}) - 3J_F(z^{(k)})]^{-1}F(x^{(k)})$

$$[J_F(x^{(k)}) - 3J_F(z^{(k)})]^{-1}F(x^{(k)}) = -\frac{1}{2}e_k - \frac{1}{2}C_2e_k^2 + N_3e_k^3 + N_4e_k^4 + N_5e_k^5 + N_6e_k^6 + N_7e_k^7 + N_8e_k^8 + O[e_k^9],$$

donde $N_3 = -\frac{1}{2}C_3 + Y_3$, y $N_s = -\frac{1}{2}C_s + \sum_{j=4}^s Y_{j-1}C_{s-j+2} + Y_s$, $s = 4, 5, \dots$. Por lo tanto, obtenemos $u^{(k)}$ en la forma:

$$\begin{aligned}u^{(k)} &= y^{(k)} + 2[J_F(x^{(k)}) - 3J_F(z^{(k)})]^{-1}F(x^{(k)}) \\ &= \xi + L_3e_k^3 + L_4e_k^4 + L_5e_k^5 + L_6e_k^6 + L_7e_k^7 + L_8e_k^8 + O[e_k^9],\end{aligned}$$

donde $L_s = N_s - \frac{1}{2}M_{s-2}$, $s = 3, 4, \dots$. Fácilmente se demuestra que L_3 se anula, con lo que:

$$u^{(k)} = \xi + L_4 e_k^4 + L_5 e_k^5 + L_6 e_k^6 + L_7 e_k^7 + L_8 e_k^8 + O[e_k^9].$$

De nuevo, desarrollamos en serie de Taylor la función $F(u^{(k)})$ alrededor de ξ :

$$F(u^{(k)}) = L_4 e_k^4 + L_5 e_k^5 + L_6 e_k^6 + L_7 e_k^7 + (L_8 + C_2 L_4) e_k^8 + O[e_k^9].$$

Para obtener $v^{(k)}$ buscamos primero $[J_F(x^{(k)}) - 3J_F(z^{(k)})]^{-1}F(u^{(k)})$ que viene dado por:

$$[J_F(x^{(k)}) - 3J_F(z^{(k)})]^{-1}F(u^{(k)}) = -\frac{1}{2}L_4 e_k^4 - \frac{1}{2}L_5 e_k^5 + R_6 e_k^6 + R_7 e_k^7 + R_8 e_k^8 + O[e_k^9],$$

donde

$$\begin{aligned} R_6 &= -\frac{1}{2}L_6 + Y_3 L_4, \\ R_7 &= -\frac{1}{2}L_7 + Y_3 L_5 + Y_4 L_4, \\ R_8 &= -\frac{1}{2}L_8 - C_2 L_4 + Y_3 L_6 + Y_4 L_5 + Y_5 L_4. \end{aligned}$$

Por tanto,

$$v^{(k)} = \xi + 2Y_3 L_4 e_k^6 + 2(Y_3 L_5 + Y_4 L_4) e_k^7 + 2(-C_2 L_4 + Y_3 L_6 + Y_4 L_5 + Y_5 L_4) e_k^8 + O[e_k^9].$$

Ahora, desarrollando en serie de Taylor la función $F(v^{(k)})$ alrededor de ξ , obtenemos:

$$F(v^{(k)}) = 2Y_3 L_4 e_k^6 + 2(Y_3 L_5 + Y_4 L_4) e_k^7 + 2(-C_2 L_4 + Y_3 L_6 + Y_4 L_5 + Y_5 L_4) e_k^8 + O[e_k^9].$$

Calculando $[J_F(x^{(k)}) - 3J_F(z^{(k)})]^{-1}F(v^{(k)})$, obtenemos el siguiente resultado:

$$[J_F(x^{(k)}) - 3J_F(z^{(k)})]^{-1}F(v^{(k)}) = -Y_3 L_4 e_k^6 - (Y_3 L_5 + Y_4 L_4) e_k^7 + [C_2 L_4 - Y_4 L_5 - Y_5 L_4 + Y_3 R_6] e_k^8 + O[e_k^9].$$

Esto nos permite obtener $x^{(k+1)}$:

$$\begin{aligned} x^{(k+1)} &= v^{(k)} + 2[J_F(x^{(k)}) - 3J_F(z^{(k)})]^{-1}F(v^{(k)}) \\ &= \xi - \frac{1}{2}T_2(L_6 + 2R_6)e_k^8 + O[e_k^9]. \end{aligned}$$

Desarrollando la expresión $T_2(L_6 + 2R_6)$ finalmente obtenemos:

$$e_{k+1} = \frac{1}{9}(C_3 - C_2^2)(C_4 - 9C_3 C_2 + 9C_2^3)e_k^8 + O[e_k^9].$$

Con lo que se demuestra que el método propuesto con el esquema iterativo (5.60) tiene orden de convergencia ocho. \square

Como se ha demostrado en el Teorema 5.2.4 los primeros tres pasos del esquema iterativo (5.60) proporcionan una familia de métodos de orden 4 que denotamos por M4B y si tomamos los primeros 4 pasos, una otra familia de métodos de orden 6, que denotamos por M6B. Aplicando la pseudocomposición directamente sobre M4B, M6B y M8B, se obtienen otras familias de métodos. Si tomamos M4B como predictor e introducimos la cuadratura de Gauss como corrector

$$x^{(k+1)} = u^{(k)} - 2 \left[\sum_{i=1}^m \omega_i J_F \left(\frac{(1 + \tau_i)u^{(k)} + (1 - \tau_i)z^{(k)}}{2} \right) \right]^{-1} F(u^{(k)}),$$

el resultado es un método con orden de convergencia 5 si $\sigma = 2$ y $\sigma_1 = 0$, que denotamos por M5BG.

Luego, tomamos el método M6B como predictor y aplicando la cuadratura de Gauss como corrector

$$x^{(k+1)} = u^{(k)} - 2 \left[\sum_{i=1}^m \omega_i J_F \left(\frac{(1 + \tau_i)v^{(k)} + (1 - \tau_i)u^{(k)}}{2} \right) \right]^{-1} F(u^{(k)}),$$

obtenemos la familia de métodos con orden de convergencia 10 si $\sigma = 2$ y $\sigma_1 = 0$.

Si tomamos el esquema iterativo (5.60) como predictor y aplicamos como corrector la cuadratura de Gauss,

$$x^{(k+1)} = v^{(k)} - 2 \left[\sum_{i=1}^m \omega_i J_F \left(\frac{(1 + \tau_i)w^{(k)} + (1 - \tau_i)v^{(k)}}{2} \right) \right]^{-1} F(v^{(k)}),$$

obtenemos un método de orden 12 si $\sigma = 2$ y, si $\sigma_1 = 0$, obtenemos un método con orden de convergencia 14. Denotamos este método por M14BG. Estudiamos sus índices de eficiencia como lo hemos hecho en el caso anterior, empleando en todos los casos la fórmula de cuadratura de punto medio.

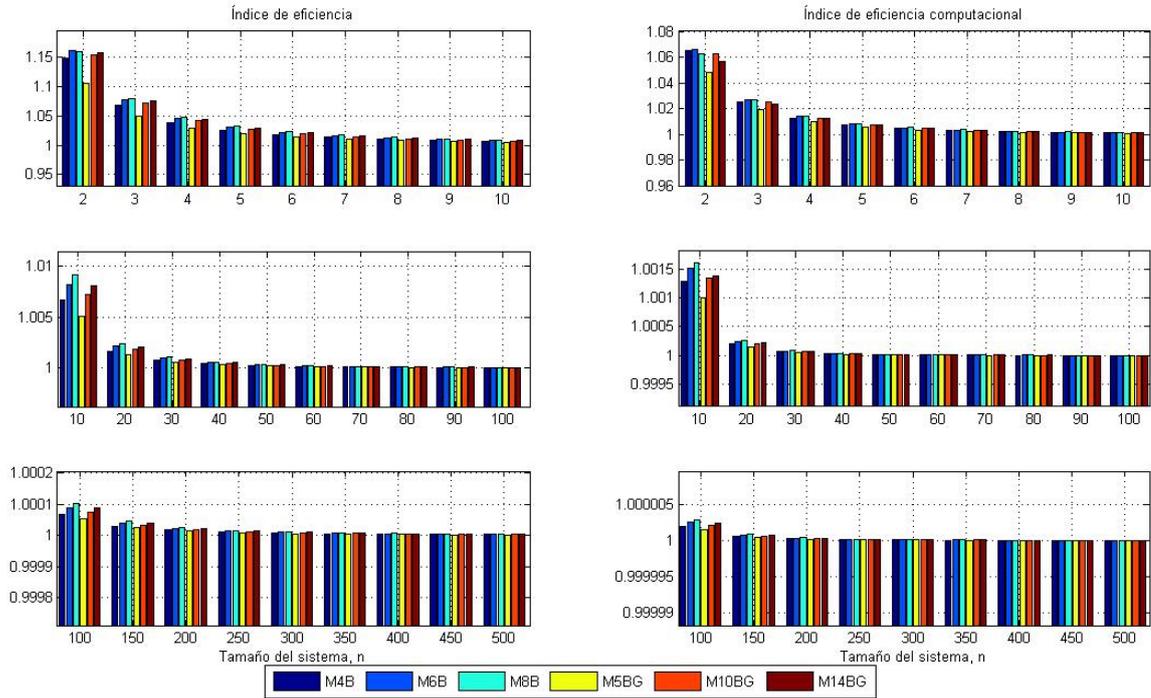


Figura 5.6: Índice de eficiencia, I e índice de eficiencia computacional, IC

Las gráficas mostradas en la Figura 5.6 representan el índice de eficiencia y el de eficiencia computacional de los seis métodos en función del tamaño del sistema de ecuaciones. Analizando el índice I observamos que para sistemas con $n = 2$ el mayor I lo tiene el método M6B y para sistemas con $n > 3$, el método M8B. El menor I para todos los tamaños de sistemas lo tiene M5BG. En todos casos los métodos M6B y M14BG tienen casi iguales I y los sigue el método M6B. Si hacemos comparación entre los IC de los distintos métodos vemos que para pequeños sistemas ($n < 5$) el mayor índice de eficiencia computacional lo tiene M6B y para sistemas con $n > 5$ el mayor IC lo tiene M8B. El menor IC para cualquier tamaño del sistema lo tiene el método M5BG, lo cual se justifica porque el orden aumenta en sólo una unidad, mientras que se utilizan dos evaluaciones funcionales más y se resuelve un nuevo sistema de ecuaciones lineales. Entre los métodos con corrector Gauss vemos que para $n = 2$, $IC_{M6B} > IC_{M4B} > IC_{M8B}$ y para $n > 10$ tenemos $IC_{M8B} > IC_{M6B} = IC_{M14BG}$.

En la Figura 5.7 se comparan los índices de los dos conjuntos de métodos excluyendo M4A y M4B. Para sistemas con $n = 2$ los mayores índices de eficiencia corresponden a los métodos M6A y M6B y para $n > 2$, M8A y M8B tienen los mejores índices de eficiencia. Los mayores índices de eficiencia computacional (para $n = 2$) los tienen M10AG y M6B. Para sistemas con $10 < n < 100$ el método con mayor índice de eficiencia computacional es M8B y para grandes sistemas ($n > 100$), M8B y M8A.

En la Figura 5.8 se muestran las gráficas de los índice de eficiencia y de eficiencia computacional en función del tamaño del sistema de los métodos con mejores índices: Traub, TrG-LD1S, M6A, M8A, M10AG, M6B, M8B y M10BG. Los I de los métodos M6A y M6B son mayores para sistemas con $n = 2$. Luego, para sistemas con

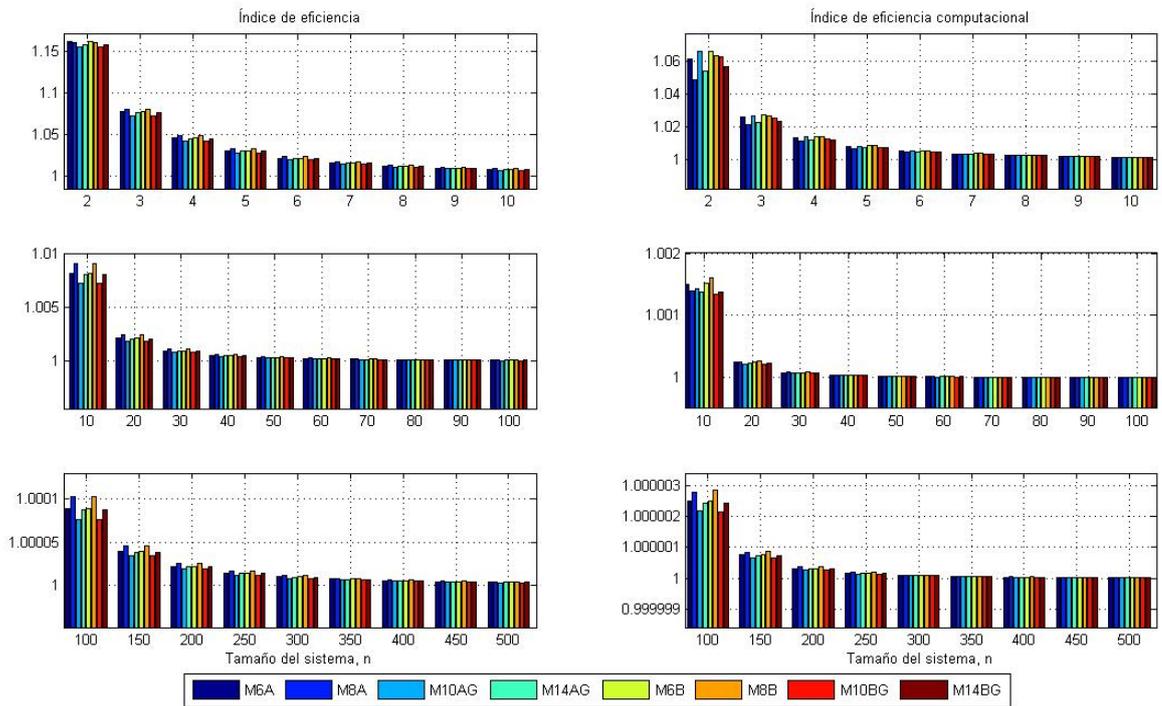


Figura 5.7: Índice de eficiencia, I e índice de eficiencia computacional, IC

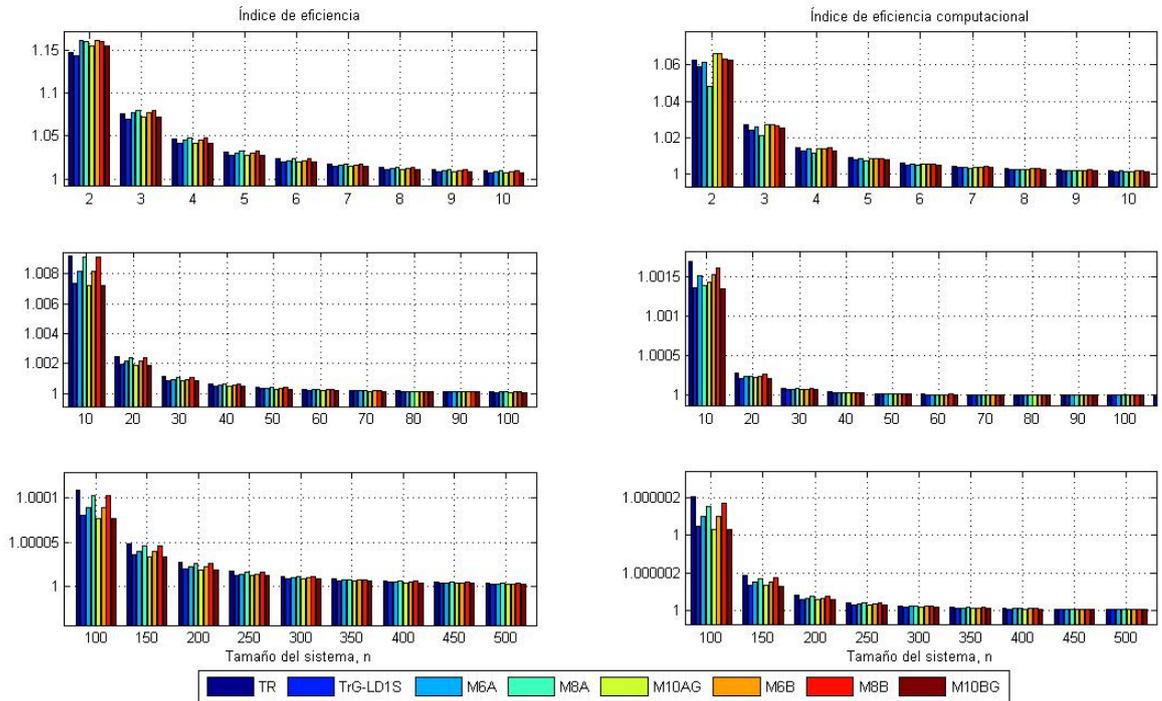


Figura 5.8: Índice de eficiencia, I e índice de eficiencia computacional, IC

$2 < n < 100$ el mayor índice de eficiencia lo tienen los métodos M8A y M8B y para sistemas con $n > 100$ el método de Traub tiene mayor I. Si comparamos los índices de eficiencia computacional vemos que para sistemas de $n = 2$ y $n = 3$ los mejores índices corresponden a los métodos M10AG y M6B. Para sistemas de mayor tamaño el índice de eficiencia computacional del método de Traub es de nuevo mayor. Por ejemplo, para

sistemas de $n = 7$ podemos ordenar los métodos según sus IC de la siguiente manera:

$$IC_{TR} > IC_{M8B} > IC_{M6B} > IC_{M6A} > IC_{M10AG} > IC_{TrG-LD1S} > IC_{M10BG} > IC_{M8A},$$

y para sistemas de tamaño $7 < n < 100$ como sigue:

$$IC_{TR} > IC_{M8B} > IC_{M6B} > IC_{M6A} > IC_{M8A} > IC_{M10AG} > IC_{M10BG} > IC_{TrG-LD1S}.$$

En el caso de que el sistema tenga tamaño mayor de 100 los índices de los métodos verifican la desigualdad

$$IC_{TR} > IC_{M8B} > IC_{M8A} > IC_{M6B} > IC_{M6A} > IC_{TrG-LD1S} > IC_{M10BG} > IC_{M10AG}.$$

5.3. Pruebas numéricas

En esta sección ponemos a prueba los métodos desarrollados para ilustrar su eficacia comparándolos con otros métodos. En primer lugar trabajaremos con una lista de sistemas no lineales con los que comprobaremos nuestros métodos, para luego aplicarlos sobre algunos problemas de interés práctico.

1. $F_1(x_1, x_2) = (\exp(x_1)\exp(x_2) + x_1\cos(x_2), x_1 + x_2 - 1)$, $\xi \approx (3.47063096, -2.47063096)^T$.
2. $F_2(f_1(x), f_2(x), f_3(x), f_4(x))$ donde $x = (x_1, x_2, x_3, x_4)^T$ y $f_i : \mathbb{R}^4 \rightarrow \mathbb{R}, i = 1, 2, 3, 4$, de tal manera que

$$\begin{aligned} f_1(x) &= x_2x_3 + x_4(x_2 + x_3), \\ f_2(x) &= x_1x_3 + x_4(x_1 + x_3), \\ f_3(x) &= x_1x_2 + x_4(x_1 + x_2), \\ f_4(x) &= x_1x_2 + x_1x_3 + x_2x_3 - 1 \end{aligned}$$

$$\text{y } \xi = \left(\pm \frac{1}{\sqrt{3}}, \pm \frac{1}{\sqrt{3}}, \pm \frac{1}{\sqrt{3}}, \pm \frac{1}{\sqrt{3}} \right)^T.$$

3. $F_3(x_1, x_2) = (x_1^2 + x_2^2 - 1, x_1^2 - x_2^2 - \frac{1}{2})$, $\xi = \left(\frac{1}{2}, \frac{\sqrt{3}}{2} \right)^T$.

4. $F_4(f_1(x), f_2(x), \dots, f_n(x))$ donde $x = (x_1, x_2, \dots, x_n)^T$ y $f_i : \mathbb{R}^n \rightarrow \mathbb{R}, i = 1, 2, \dots, n$, de tal manera que

$$\begin{aligned} f_i(x) &= x_i x_{i+1} - 1, \quad i = 1, 2, \dots, n-1 \\ f_n(x) &= x_n x_1 - 1. \end{aligned}$$

Cuando n es impar, los ceros exactos de $F(x)$ son $\xi_1 = (1, 1, \dots, 1)^T$ y $\xi_2 = (-1, -1, \dots, -1)^T$.

5. $F_5(x_1, x_2) = (x_1^2 - x_1 - x_2^2 - 1, -\sin(x_1) + x_2)$, $\xi \approx (0.725950, 0.502945)^T$.
6. $F_6(x_1, x_2) = (x_1^2 + x_2^2 - 4, -\exp(x_1) + x_2 - 1)$, $\xi \approx (1.004168, -1.729637)^T$.
7. $F_7(x_1, x_2, x_3) = (x_1^2 + x_2^2 + x_3^2 - 9, x_1x_2x_3 - 1, x_1 + x_2 - x_3^2)$, $\xi \approx (2.14025, -2.09029, -0.223525)^T$.
8. $F_8(x_1, x_2, x_3) = (\sin(x_1) + x_2^2 + \log(x_3) - 7, 3x_1 + 2x_2^2 - x_3^{-3} + 1, x_1 + x_2 - x_3 - 5)$, $\xi \approx (-2.215371, 2.499693, 4.715678)^T$.
9. $F_9(x_1, x_2) = ((x_1 - 1)^6 - x_2, x_2 - 1)$, $\xi = (0, 1)^T$.
10. $F_{10}(x_1, x_2) = (\ln(x_1^2) - 2\ln(\cos(x_2)), x_1 \tan\left(\frac{x_1}{\sqrt{2}} + x_2\right) - \sqrt{2})$, $\xi \approx (0.9548041416, 6.5849814845)^T$.

11. $F_{11}(x_1, x_2) = (x_1 + \exp(x_2) - \cos(x_2), 3x_1 - x_2 - \sin(x_2)), \xi = (0, 0)^T$.
12. $F_{12}(x_1, x_2, x_3) = (\cos(x_2) - \sin(x_1), x_3^{x_1} - \frac{1}{x_2}, \exp(x_1) - x_3^2),$
 $\xi \approx (0.90956949, 0.66122683, 1.57583414)^T$.

Como en el caso unidimensional, para la comparación de los métodos desarrollados usaremos el método clásico de Newton (NC), el método de Traub (TR) con convergencia de orden 3 y el método de Jarratt (JT) que tiene orden de convergencia 4. Los cálculos numéricos se han realizado en MATLAB, utilizando aritmética de precisión variable, que usa representación en coma flotante de 2000 dígitos decimales de mantisa. Cada iteración se obtiene de la anterior por medio de un proceso iterativo expresión $x^{(k+1)} = x^{(k)} - A^{-1}b$, donde $x^{(k)} \in \mathbb{R}^n$, A es una matriz real de $n \times n$ y $b \in \mathbb{R}^n$. La matriz A y el vector b son diferentes según el método utilizado, pero en cualquier caso, utilizamos para el cálculo inverso $-A^{-1}b$ la solución del sistema lineal $Ay = b$, con la eliminación de Gauss con pivoteo parcial. Así, la nueva estimación se obtiene fácilmente mediante la adición de la solución del sistema lineal y el anterior iteración: $x^{(k+1)} = x^{(k)} + y^{(k)}$.

El criterio de parada utilizado es $\|x^{(k+1)} - x^{(k)}\| + \|F(x^{(k)})\| < 10^{-500}$. Por tanto, estamos garantizando que los iterados convergen a un límite y que ese límite es una solución del sistema no lineal. Presentamos los resultados en dos tablas. En la Tabla 5.1 presentamos los resultados obtenidos al utilizar los métodos de las familias NeG y TrG para estimar los ceros de las funciones de (1) a (12) y en Tabla 5.2 los resultados obtenidos para estimar los ceros de las mismas funciones aplicando los métodos desarrollados en la Sección 5.2. Para cada ejemplo mostramos los siguientes datos: la aproximación inicial $x^{(0)}$ y, para cada método, el número de iteraciones necesarios para obtener $\|x^{(k+1)} - x^{(k)}\| + \|F(x^{(k)})\| < 10^{-500}$, la distancia entre las dos últimas iteraciones, el valor de la función F en la última iteración, el tiempo medio (en segundos) transcurrido (ϵ) después de 100 actuaciones del programa, calculado mediante el comando "tic, ..., toc" de MATLAB, y el orden de convergencia ρ (véase [17]) calculado aproximadamente por la fórmula:

$$\rho \approx \frac{\ln(\|x^{(k+1)} - x^{(k)}\|) / (\|x^{(k)} - x^{(k-1)}\|)}{\ln(\|x^{(k)} - x^{(k-1)}\|) / (\|x^{(k-1)} - x^{(k-2)}\|)}. \quad (5.61)$$

El valor de ρ que aparece en las Tablas 5.1 y 5.2 es la última coordenada del vector ρ cuando la variación entre sus coordenadas es pequeña.

En todos los casos se confirma el orden de convergencia. En el caso del sistema $F_8(x) = 0$ obtenemos convergencia lineal para todos los métodos, que se justifica por el hecho de que la matriz Jacobiana asociada al sistema está mal condicionada.

Todos los sistemas de la lista son de tamaño pequeño: 2 y 3 ecuaciones y un caso de 4 ecuaciones. La excepción es el problema 4 en que se puede variar el tamaño. Aprovecharemos este hecho para estudiar el comportamiento de los métodos en caso de sistemas grandes.

En general, los cuatro métodos de las familias de NeG estudiados necesitan menor tiempo computacional para obtener los resultados mostrados en la Tabla 5.1, que los de las familias de TrG, excepto los problemas $F_8(x) = 0$ y $F_{11}(x) = 0$.

Para visualizar mejor los resultados de la Tabla 5.1 mostramos en la Figura 5.9 el tiempo computacional empleado para las funciones F_1, F_3, F_5, F_6, F_9 y F_{11} (que definen sistemas de 2 ecuaciones) utilizando los métodos estudiados de las familias NeG. En todos los casos, el método NeG-LD2S necesita mayor tiempo para obtener las raíces de las funciones excepto el caso de F_9 donde el método de Traub necesita más tiempo. El menor tiempo computacional se obtiene con el método NeG-LT2S resolviendo los problemas $F_3(x) = 0$, $F_6(x) = 0$ y $F_9(x) = 0$ y con NeG-R2S resolviendo $F_5(x) = 0$. En el caso del problema $F_3(x) = 0$, los menores tiempos se obtienen con los métodos NeG-LD1S, NeG-LT2S y NeG-R2S. Con los métodos de comparación NC, TR y JT obtenemos resultados con menor tiempo computacional en la solución de los problemas $F_1(x) = 0$ y $F_{11}(x) = 0$.

En la Figura 5.10 se presentan los resultados de los tiempos computacionales que se obtienen al resolver mediante los métodos de la familia TrG los mismos problemas utilizados en la Figura 5.9. Notamos que, en general, los métodos estudiados de las familias de TrG necesitan mayores tiempos para obtener los resultados

Tabla 5.1: Ejemplos numéricos para las familias NeG y TrG

N°	Método	ξ	$\ x^{(k+1)} - x^{(k)}\ $	$\ F(x^{(k+1)})\ $	Iter	ρ	ϵ, s
$F_1,$ $x_0 = (3, -2)$	NeG-LD1S	(3.4706310, -2.4706310)	0	0	7	3.0000	1.1168
	NeG-LD2S	(3.4706310, -2.4706310)	$7.7e-118$	0	5	3.0000	1.1496
	NeG-LT2S	(3.4706310, -2.4706310)	$1.0e-203$	0	6	3.0000	1.0539
	NeG-R2S	(3.4706310, -2.4706310)	$1.5e-112$	0	5	3.0000	0.9521
	TrG-LD1S	(3.4706310, -2.4706310)	0	0	5	5.0000	1.1661
	TrG-LD2S	(3.4706310, -2.4706310)	$7.7e-118$	0	5	5.0000	1.5981
	TrG-LT2S	(3.4706310, -2.4706310)	$1.0e-203$	0	5	5.0000	1.2907
	TrG-R2S	(3.4706310, -2.4706310)	$1.5e-112$	0	5	5.0000	1.3912
	NC	(3.4706310, -2.4706310)	0	0	9	2.0000	0.8656
	TR	(3.4706310, -2.4706310)	0	0	6	3.0000	0.9630
JT	(3.4706310, -2.4706310)	0	0	6	4.0000	1.0133	
$F_2,$ $x_0 = (1, 1, 1, -0.5)$	NeG-LD1S	($\sqrt{3}/3, \sqrt{3}/3,$ $\sqrt{3}/3, -\sqrt{3}/6$)	0	0	7	3.0000	2.6012
	NeG-LD2S	($\sqrt{3}/3, \sqrt{3}/3,$ $\sqrt{3}/3, -\sqrt{3}/6$)	$2.2e-139$	0	6	3.0000	3.3279
	NeG-LT2S	($\sqrt{3}/3, \sqrt{3}/3,$ $\sqrt{3}/3, -\sqrt{3}/6$)	$2.2e-139$	0	6	3.0000	2.4186
	NeG-R2S	($\sqrt{3}/3, \sqrt{3}/3,$ $\sqrt{3}/3, -\sqrt{3}/6$)	$2.2e-139$	0	6	3.0000	2.5663
	TrG-LD1S	($\sqrt{3}/3, \sqrt{3}/3,$ $\sqrt{3}/3, -\sqrt{3}/6$)	0	0	5	4.9994	2.3505
	TrG-LD2S	($\sqrt{3}/3, \sqrt{3}/3,$ $\sqrt{3}/3, -\sqrt{3}/6$)	0	0	5	4.9994	3.6293
	TrG-LT2S	($\sqrt{3}/3, \sqrt{3}/3,$ $\sqrt{3}/3, -\sqrt{3}/6$)	0	0	5	4.9994	2.6906
	TrG-R2S	($\sqrt{3}/3, \sqrt{3}/3,$ $\sqrt{3}/3, -\sqrt{3}/6$)	0	0	5	4.9994	2.8574
	NC	($\sqrt{3}/3, \sqrt{3}/3,$ $\sqrt{3}/3, -\sqrt{3}/6$)	0	0	11	2.0000	2.6006
	TR	($\sqrt{3}/3, \sqrt{3}/3,$ $\sqrt{3}/3, -\sqrt{3}/6$)	0	0	7	3.0000	2.9968
JT	($\sqrt{3}/3, \sqrt{3}/3,$ $\sqrt{3}/3, -\sqrt{3}/6$)	0	0	6	4.0000	2.8504	
$F_3,$ $x_0 = (1, 1)$	NeG-LD1S	($\sqrt{3}/2, 0.5$)	0	0	7	3.0000	0.9310
	NeG-LD2S	($\sqrt{3}/2, 0.5$)	$1.2e-116$	0	6	3.0000	1.1133
	NeG-LT2S	($\sqrt{3}/2, 0.5$)	$1.2e-116$	0	6	3.0000	0.8581
	NeG-R2S	($\sqrt{3}/2, 0.5$)	$1.2e-116$	0	6	3.0000	0.9482
	TrG-LD1S	($\sqrt{3}/2, 0.5$)	0	0	6	5.0000	1.1910
	TrG-LD2S	($\sqrt{3}/2, 0.5$)	0	0	6	5.0000	1.5713

Tabla 5.1 Ejemplos numéricos para las familias NeG y TrG, continuación...

N^o	Método	ξ	$\ x^{(k+1)} - x^{(k)}\ $	$\ F(x^{(k+1)})\ $	Iter	ρ	ϵ, s
	TrG-LT2S	$(\sqrt{3}/2, 0.5)$	0	0	6	5.0000	1.2516
	TrG-R2S	$(\sqrt{3}/2, 0.5)$	0	0	6	5.0000	1.3566
	NC	$(\sqrt{3}/2, 0.5)$	0	0	11	2.0000	1.1475
	TR	$(\sqrt{3}/2, 0.5)$	0	0	8	3.0000	1.4459
	JT	$(\sqrt{3}/2, 0.5)$	0	0	6	4.0000	1.0883
$F_4,$ $x_0 = (0.8, \dots, 0.8)$	NeG-LD1S	$(1, \dots, 1)$	0	0	7	3.0000	26.0924
	NeG-LD2S	$(1, \dots, 1)$	$2.6e-231$	0	6	3.0000	29.8702
	NeG-LT2S	$(1, \dots, 1)$	$2.6e-231$	0	6	3.0000	24.8158
	NeG-R2S	$(1, \dots, 1)$	$2.6e-231$	0	6	3.0000	17.6566
	TrG-LD1S	$(1, \dots, 1)$	0	0	5	5.0000	24.0408
	TrG-LD2S	$(1, \dots, 1)$	0	0	5	5.0000	30.1988
	TrG-LT2S	$(1, \dots, 1)$	0	0	5	5.0000	25.0347
	TrG-R2S	$(1, \dots, 1)$	0	0	5	5.0000	25.2705
	NC	$(1, \dots, 1)$	0	0	10	2.0000	23.6270
	TR	$(1, \dots, 1)$	0	0	7	3.0000	26.3586
JT	$(1, \dots, 1)$	0	0	6	4.0000	90.1726	
$F_5,$ $x_0 = (-0.5, -0.5)$	NeG-LD1S	$(-0.8452567,$ $-0.74814149)$	0	0	7	3.0147	1.1466
	NeG-LD2S	$(-0.8452567,$ $-0.74814149)$	$2.6e-175$	0	6	3.0247	1.3864
	NeG-LT2S	$(-0.8452567,$ $-0.74814149)$	$5.3e-167$	0	6	3.0399	1.0101
	NeG-R2S	$(-0.8452567,$ $-0.74814149)$	$4.7e-175$	0	6	3.0246	0.9867
	TrG-LD1S	$(-0.8452567,$ $-0.74814149)$	0	0	6	4.9980	1.1607
	TrG-LD2S	$(-0.8452567,$ $-0.74814149)$	0	0	6	4.9975	1.4847
	TrG-LT2S	$(-0.8452567,$ $-0.74814149)$	0	0	6	4.9966	1.2574
	TrG-R2S	$(-0.8452567,$ $-0.74814149)$	0	0	6	4.9976	1.2989
	NC	$(-0.8452567,$ $-0.74814149)$	0	0	11	2.0136	1.0904
	TR	$(-0.8452567,$ $-0.74814149)$	0	0	8	3.0030	1.2538
JT	$(-0.8452567,$ $-0.74814149)$	0	0	7	3.9825	1.2363	
$F_6,$ $x_0 = (0.5, -1.5)$	NeG-LD1S	$(1.0041687,$ $-1.7296373)$	0	0	7	3.0000	1.0884
	NeG-LD2S	$(1.0041687,$ $-1.7296373)$	$1.7e-130$	0	6	3.0000	1.2938
	NeG-LT2S	$(1.0041687,$ $-1.7296373)$	$2.6e-117$	0	6	3.0000	0.9000
	NeG-R2S	$(1.0041687,$ $-1.7296373)$	$7.1e-131$	0	6	3.0000	1.0413
	TrG-LD1S	$(1.0041687,$ $-1.72963729)$	0	0	6	5.0000	1.2354
	TrG-LD2S	$(1.00416874$ $-1.72963729)$	0	0	6	5.0000	1.6876
	TrG-LT2S	$(1.00416874,$ $-1.72963729)$	0	0	6	5.0000	1.2860

Tabla 5.1 Ejemplos numéricos para las familias NeG y TrG, continuación...

N°	Método	ξ	$\ x^{(k+1)} - x^{(k)}\ $	$\ F(x^{(k+1)})\ $	Iter	ρ	ϵ, s
	TrG-R2S	(1.00416874, -1.72963729)	0	0	6	5.0000	1.4081
	NC	(1.00416874, -1.72963729)	0	0	11	2.0000	1.0967
	TR	(1.00416874, -1.72963729)	0	0	8	3.0000	1.2501
	JT	(1.00416874, -1.72963729)	0	0	6	4.0001	1.0302
$F_7,$ $x_0 = (2 - 1.5, -0.5)$	NeG-LD1S	(2.1402581, -2.0902946, -0.2235251)	0	0	7	3.0007	2.0007
	NeG-LD2S	(2.1402581, -2.0902946, -0.2235251)	$4.3e-112$	0	6	3.0006	1.6005
	NeG-LT2S	(2.1402581, -2.0902946, -0.2235251)	$3.1e-111$	0	6	3.0004	1.1838
	NeG-R2S	(2.1402581, -2.0902946, -0.2235251)	$4.3e-112$	0	6	3.0006	1.8658
	TrG-LD1S	(2.1402581, -2.0902946, -0.2235251)	0	0	6	5.0004	2.6573
	TrG-LD2S	(2.1402581, -2.0902946, -0.2235251)	0	0	6	5.0005	3.2478
	TrG-LT2S	(2.1402581, -2.0902946, -0.2235251)	0	0	6	5.0008	2.5248
	TrG-R2S	(2.1402581, -2.0902946, -0.2235251)	0	0	6	5.0005	2.5887
	NC	(2.1402581, -2.0902946, -0.2235251)	0	0	11	2.0002	1.7010
	TR	(2.1402581, -2.0902946, -0.2235251)	0	0	9	3.0000	2.5486
	JT	(2.1402581, -2.0902946, -0.2235251)	0	0	6	4.0009	1.8729
	$F_8,$ $x_0 = (-2, 2, 4)$	NeG-LD1S	(-2.412897, 2.643372, 4.769524)	0	$4.7e-045$	22	0.9980
NeG-LD2S		(-2.412897, 2.643372, 4.769524)	0	$4.7e-045$	22	0.9908	8.0264
NeG-LT2S		(-2.412897, 2.643372, 4.769524)	0	$4.7e-045$	22	0.9775	5.5103
NeG-R2S		(-2.412897, 2.643372, 4.769524)	0	$4.7e-045$	22	0.9908	5.2301

Tabla 5.1 Ejemplos numéricos para las familias NeG y TrG, continuación...

N^o	Método	ξ	$\ x^{(k+1)} - x^{(k)}\ $	$\ F(x^{(k+1)})\ $	Iter	ρ	ϵ, s
	TrG-LD1S	(-2.412897, 2.643372, 4.769524)	0	$4.7e-045$	13	1.0140	4.5379
	TrG-LD2S	(-2.412897, 2.643372, 4.769524)	0	$4.7e-045$	13	1.0128	6.1402
	TrG-LT2S	(-2.412897, 2.643372, 4.769524)	0	$4.7e-045$	13	1.0198	5.2506
	TrG-R2S	(-2.412897, 2.643372, 4.769524)	0	$4.7e-045$	13	1.0128	4.5547
	NC	(-2.412897, 2.643372, 4.769524)	0	$4.7e-045$	24	0.9916	4.4071
	TR	(-2.412897, 2.643372, 4.769524)	0	$4.7e-045$	14	0.9801	3.4662
	JT	(-2.412897, 2.643372, 4.769524)	0	$4.7e-045$	22	0.9451	5.5333
$F_9,$ $x_0 = (0.25, 0.75)$	NeG-LD1S	(0, 1)	0	0	9	2.9999	1.1740
	NeG-LD2S	(0, 1)	$4.4e-237$	0	8	3.0000	1.3493
	NeG-LT2S	(0, 1)	$1.0e-139$	0	8	3.0000	1.0527
	NeG-R2S	(0, 1)	$1.5e-246$	0	8	3.0000	1.1563
	TrG-LD1S	(0, 1)	0	0	23	5.0000	1.6190
	TrG-LD2S	(0, 1)	0	0	8	5.0000	2.3024
	TrG-LT2S	(0, 1)	0	0	7	5.0000	1.5122
	TrG-R2S	(0, 1)	0	0	13	4.9999	2.8960
	NC	(0, 1)	0	0	13	2.0000	1.0765
	TR	(2, 1)	0	0	11	3.0000	1.5048
JT	(0, 1)	0	0	7	4.0000	1.0794	
$F_{10},$ $x_0 = (1, 6)$	NeG-LD1S	(0.9548041 6.5849815)	0	$6.7e-044$	8	3.0000	3.6092
	NeG-LD2S	(0.9548041 6.5849815)	0	$6.7e-044$	8	3.0000	4.6373
	NeG-LT2S	(0.9548041 6.5849815)	0	$6.7e-044$	9	3.0000	4.5858
	NeG-R2S	(0.9548041 6.5849815)	0	$6.7e-044$	8	3.0000	4.2432
	TrG-LD1S	(0.9548041 6.5849815)	0	6.7^{-044}	8	4.9926	5.4096
	TrG-LD2S	(0.9548041 6.5849815)	0	6.7^{-044}	7	4.9950	6.6451
	TrG-LT2S	(0.9548041 6.5849815)	0	6.7^{-044}	6	4.9919	5.0704
	TrG-R2S	(0.9548041 6.5849815)	0	6.7^{-044}	7	4.9976	5.3897
	NC	(0.9548041 6.5849815)	0	$6.7e-044$	12	2.0164	3.1129
	TR	n.c.	n.c.	n.c.	n.c.	n.c.	n.c.
	JT	(0.9548041 6.5849815)	0	$6.7e-044$	6	4.1118	4.5939

Tabla 5.1 Ejemplos numéricos para las familias NeG y TrG, continuación...

N°	Método	ξ	$\ x^{(k+1)} - x^{(k)}\ $	$\ F(x^{(k+1)})\ $	Iter	ρ	ϵ, s
$F_{11},$ $x_0 = (0.5, 0.5)$	NeG-LD1S	(0, 0)	0	0	9	2.0000	1.5490
	NeG-LD2S	(0, 0)	$1.4e-267$	0	8	2.0000	1.8759
	NeG-LT2S	(0, 0)	$5.2e-260$	0	8	2.0000	1.4167
	NeG-R2S	(0, 0)	$1.8e-267$	0	8	2.0000	1.4517
	TrG-LD1S	(0, 0)	0	0	6	2.0000	1.2712
	TrG-LD2S	(0, 0)	0	0	6	4.9950	1.6207
	TrG-LT2S	(0, 0)	0	0	6	4.9919	1.3169
	TrG-R2S	(0, 0)	0	0	6	4.9976	1.3501
	NC	(0, 0)	0	0	11	2.0000	1.2119
	TR	(0, 0)	0	0	7	4.5166	1.2191
JT	(0, 0)	0	0	8	2.0000	1.4298	
$F_{12},$ $x_0 = (0.8, 0.5, 1.4)$	NeG-LD1S	(0.909570, 0.661227, 1.575834)	0	0	7	3.0004	1.8443
	NeG-LD2S	(0.909570, 0.661227, 1.575834)	$6.6e-319$	0	7	3.0000	2.7876
	NeG-LT2S	(0.909570, 0.661227, 1.575834)	$1.2e-272$	0	7	3.0000	2.0925
	NeG-R2S	(0.909570, 0.661227, 1.575834)	$1.4e-317$	0	7	3.0000	2.2260
	TrG-LD1S	(0.909570, 0.661227, 1.575834)	0	0	6	5.0000	2.3532
	TrG-LD2S	(0.909570, 0.661227, 1.575834)	0	0	6	5.0000	3.2537
	TrG-LT2S	(0.909570, 0.661227, 1.575834)	0	0	6	5.0000	2.6423
	TrG-R2S	(0.909570, 0.661227, 1.575834)	0	0	6	5.0000	2.7230
	NC	(0.909570, 0.661227, 1.575834)	0	0	11	2.0001	2.0123
	TR	(0.909570, 0.661227, 1.575834)	0	0	8	3.0000	2.5194
JT	(0.909570, 0.661227, 1.575834)	0	0	6	3.9870	2.0186	

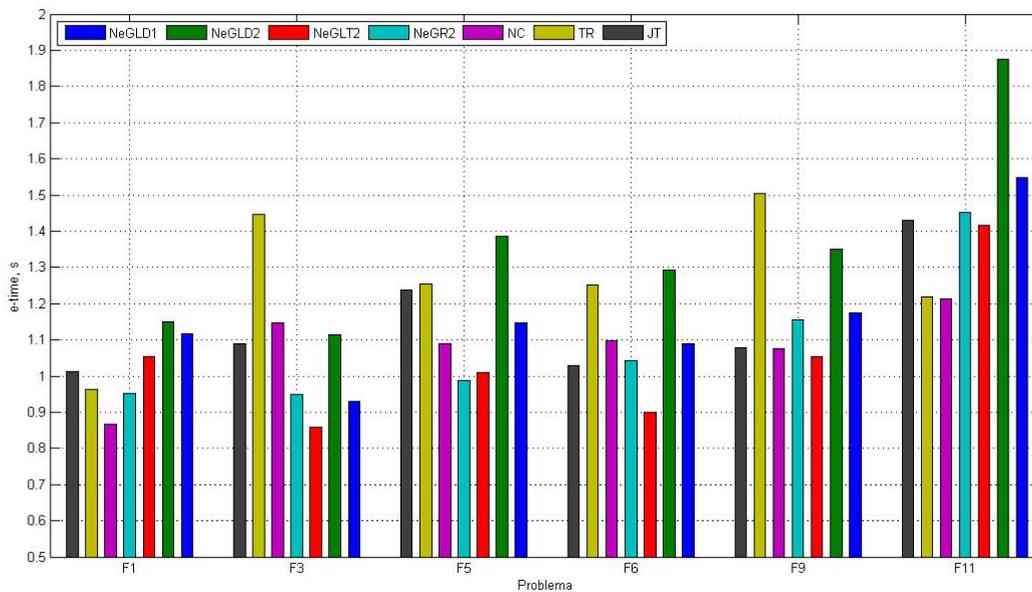


Figura 5.9: Gráfica de ϵ de la familia NeG para las funciones F_1 , F_3 , F_5 , F_6 , F_9 y F_{11}

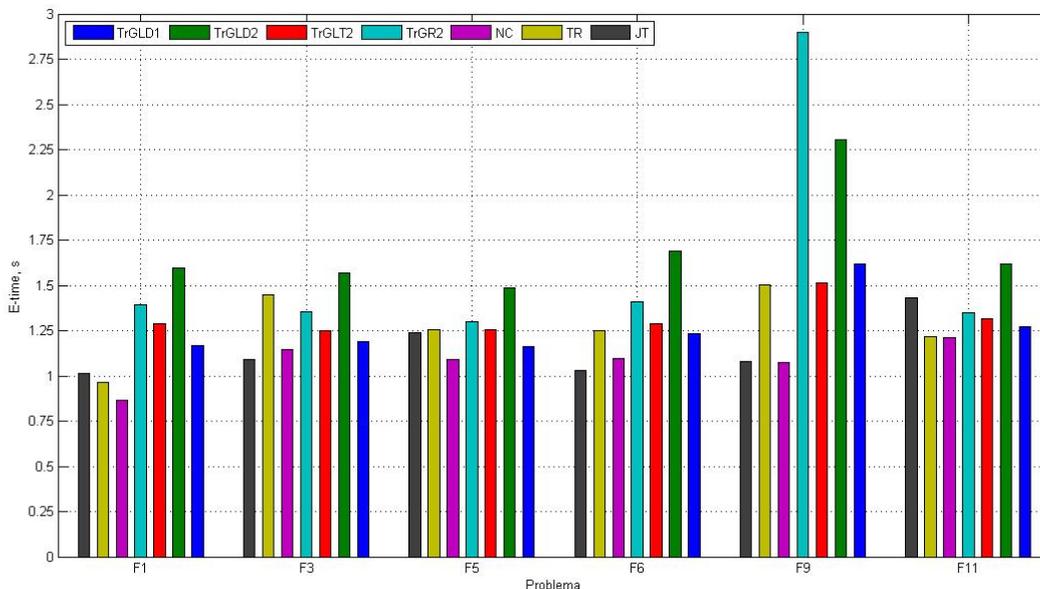


Figura 5.10: Gráfica de ϵ de la familia TrG para las funciones F_1 , F_3 , F_5 , F_6 , F_9 y F_{11}

presentados en la Tabla 5.1. Entre los métodos de TrG los resultados obtenidos con los métodos TrG-LD2S y TrG-R2S necesitan mayores tiempos que los otros dos métodos (TrG-LD1S y TrG-LT2S). En todos los problemas al método TrG-LD2S le corresponden los mayores tiempos computacionales, excepto en el caso del problema $F_9(x) = 0$ donde el método que más tiempo necesita es TrG-R2. La solución de los problemas $F_1(x) = 0$, $F_6(x) = 0$ y $F_9(x) = 0$ se obtiene con menores tiempos computacionales utilizando los métodos NC, TR y JT. En el caso $F_3(x) = 0$, el método de TR utiliza mayor tiempo computacional que todos los restantes

métodos y en el caso $F_{11}(x) = 0$, JT obtiene el resultado con más tiempo que los otros métodos.

Los problemas $F_7(x) = 0$, $F_8(x) = 0$ y $F_{12}(x) = 0$ son casos de sistemas de tres ecuaciones. El sistema $F_8(x) = 0$ debido a la convergencia de orden uno por el mal condicionamiento de la matriz Jacobiana, necesita un elevado número de iteraciones. La solución del problema se obtiene con precisión $\|F(x^{(k+1)})\| < 1e-045$ y $\|x^{(k+1)} - x^{(k)}\| = 0$ con todos los métodos utilizados. Notamos también, que entre los métodos de las familias de NeG y TrG, los de TrG requieren menores tiempos de ejecución. En el caso de los sistemas $F_7(x) = 0$ y $F_8(x) = 0$, los tiempos computacionales de NeG son menores que las de TrG. Comparando los tiempos de los métodos de la familia NeG con los métodos NC, TR y JT podemos ordenarlos de la siguiente manera:

1. para F_7 : $NeG - LT2S < NC < NeG - LD2S < NeG - R2S < JT < NeG - LD1S < TR$,
2. para F_{12} : $NeG - LD1S < NC < JT < NeG - LT2S < NeG - R2S < TR < NeG - LD2S$.

Haciendo lo mismo para los métodos estudiados de las familias de TrG, obtenemos:

1. para F_7 : $NC < JT < TrG - LT2S < TR < TrG - R2S < TrG - LD1S < TrG - LD2S$,
2. para F_{12} : $NC < JT < TrG - LD1S < TR < TrG - LT2S < TrG - R2S < TrG - LD2S$.

El problema $F_2(x) = 0$ es un sistema de 4 ecuaciones. Comparando los tiempos computacionales de NeG y TrG, los menores tiempos se obtienen con los métodos de la familia TrG, y comparandolos con los métodos NC, TR y JT, obtenemos los siguientes resultados:

1. $NeG - LT2 < NC < NeG - LD1 < NeG - R2 < JT < TR < NeG - LD2$,
2. $TrG - LD1 < NC < TrG - R2 < JT < TrGR - 2 < TR < TrG - LD2$.

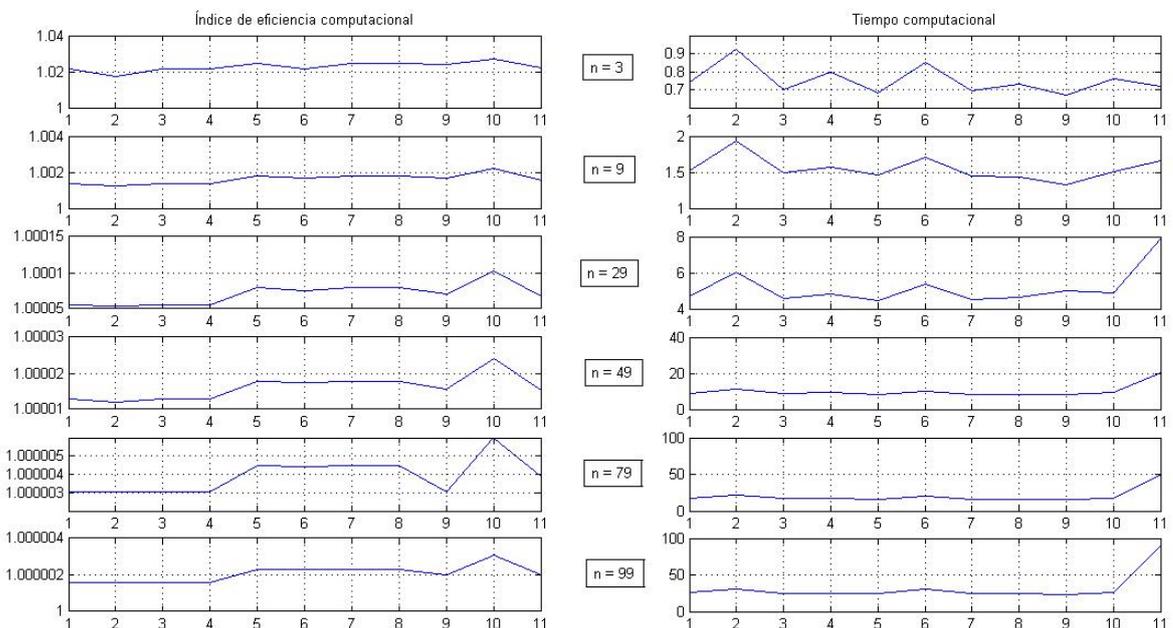


Figura 5.11: Gráficas de IC y $e - time$ de los métodos de la Tabla 5.1 para el problema $F_4(x) = 0$

En la Figura 5.11 se presentan los índices de eficiencia computacional y los tiempos de ejecución requerido para obtener la solución del problema $F_4(x) = 0$ para distintos tamaños del sistema. El eje de abcisas representa los métodos estudiados denotados de 1 a 11 en el siguiente orden:

1 - NeG-LD1, 2 - NeG-LD2, 3 - NeG-LT2, 4 - NeG-R2,
 5 - TrG-LD1, 6 - TrG-LD2, 7 - TrG-LT2, 8 - TrG-R2,
 9 - NC, 10 - TR, 11 - JT.

Notamos que, en general, a mayor índice computacional le corresponde menor tiempo de ejecución. Los métodos NeG-LD2 y TrG-LD2 tienen, para cualquier tamaño del sistema, el menor índice de su familia y necesitan mayor tiempo. Los índices computacionales de los métodos NeG-LD1, NeG-LT2 y NeG-R2 son iguales y sus tiempos computacionales no presentan grandes diferencias. Lo mismo es válido para los métodos TrG-LD1, TrG-LT2 y TrG-R2.

En la Tabla 5.2 se muestran los resultados obtenidos al resolver los problemas numéricos aplicando los métodos desarrollados en la Sección 5.2. Notamos que el orden de convergencia teórico coincide con el obtenido numericamente excepto de nuevo en el problema $F_8(x) = 0$ que presenta convergencia lineal y mayor número de iteraciones debido a que la matriz Jacobiana del sistema está mal condicionada.

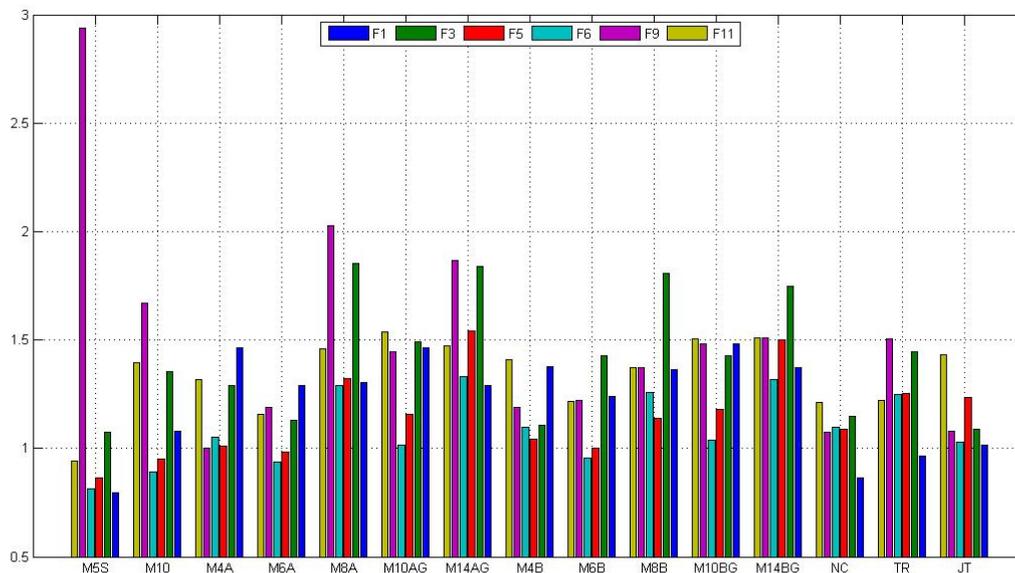


Figura 5.12: Gráfica de $e - time$ de los nuevos métodos para F_1 , F_3 , F_5 , F_6 , F_9 y F_{11}

En la Figura 5.12 presentamos los tiempos de ejecución que necesitan los métodos para obtener los resultados de la Tabla 5.2. Los sistemas presentados son de 2 ecuaciones. El método M5S es el que menos tiempo necesita para obtener la solución de los problemas, excepto en el problema $F_9(x) = 0$, en el cual el menor tiempo corresponde al método M4A. En el caso de los sistemas con $n = 3$, $F_7(x) = 0$ y $F_{12}(x) = 0$, sus respectivas soluciones se obtienen con menores tiempos de ejecución utilizando los métodos M10S y M4A, respectivamente. El método M5S es el más rápido en los sistemas $F_2(x) = 0$ ($n=4$) y $F_8(x) = 0$ ($n=3$). En general, los mayores tiempos se obtienen utilizando métodos de más alto orden de convergencia para sistemas pequeños (2, 3 y 4 ecuaciones).

En la Figura 5.13 se muestran las gráficas del índice de eficiencia computacional y los tiempos de ejecución requerido para obtener la solución del problema $F_4(x) = 0$ para distintos tamaños del sistema. El eje de abscisas la Figura 5.13 representa los métodos estudiados, denotados del 1 a 15 en el siguiente orden:

1 - M5S, 2 - M10S, 3 - M4A, 4 - M6A, 5 - M8A,
 6 - M10AG, 7 - M14AG, 8 - M4B, 9 - M6B, 10 - M8B,
 11 - M10BG, 12 - M14BG, 13 - NC, 14 - TR, 15 - JT

Tabla 5.2: Ejemplos numéricos

N°	Método	ξ	$\ x^{(k+1)} - x^{(k)}\ $	$\ F(x^{(k+1)})\ $	$Iter$	ρ	ϵ, s
$F_1,$ $x_0 = (3, -2)$	M5S	(3.4706310, -2.4706310)	0	0	5	5.0000	0.7949
	M10	(3.4706310, -2.4706310)	0	0	4	9.7125	1.0782
	M4A	(3.4706310, -2.4706310)	0	0	6	4.0000	1.4640
	M6A	(3.4706310, -2.4706310)	0	0	5	6.0000	1.2921
	M8A	(3.4706310, -2.4706310)	0	0	4	7.6887	1.3034
	M10AG	(3.4706310, -2.4706310)	0	0	4	9.7050	1.4634
	M14AG	(3.4706310, -2.4706310)	0	0	4	13.7164	1.2877
	M4B	(3.4706310, -2.4706310)	0	0	6	4.0000	1.3775
	M6B	(3.4706310, -2.4706310)	0	0	5	6.0000	1.2377
	M8B	(3.4706310, -2.4706310)	0	0	4	7.6887	1.3611
	M10BG	(3.4706310, -2.4706310)	0	0	4	9.7050	1.4806
	M14BG	(3.4706310, -2.4706310)	0	0	4	13.7164	1.3700
	NC	(3.4706310, -2.4706310)	0	0	9	2.0000	0.8656
	TR	(3.4706310, -2.4706310)	0	0	6	3.0000	0.9630
	JT	(3.4706310, -2.4706310)	0	0	6	4.0000	1.0133
$F_2,$ $x_0 = (1, 1, 1, -0.5)$	M5S	$(\sqrt{3}/3, \sqrt{3}/3,$ $\sqrt{3}/3, -\sqrt{3}/6)$	0	0	5	4.9994	2.0337
	M10	$(\sqrt{3}/3, \sqrt{3}/3,$ $\sqrt{3}/3, -\sqrt{3}/6)$	0	0	4	9.7219	2.9992
	M4A	$(\sqrt{3}/3, \sqrt{3}/3,$ $\sqrt{3}/3, -\sqrt{3}/6)$	0	0	6	4.0000	2.8462
	M6A	$(\sqrt{3}/3, \sqrt{3}/3,$ $\sqrt{3}/3, -\sqrt{3}/6)$	0	0	5	5.9999	3.0522
	M8A	$(\sqrt{3}/3, \sqrt{3}/3,$ $\sqrt{3}/3, -\sqrt{3}/6)$	0	0	5	8.0000	4.2143
	M10AG	$(\sqrt{3}/3, \sqrt{3}/3,$ $\sqrt{3}/3, -\sqrt{3}/6)$	0	0	4	9.7219	3.5989
	M14AG	$(\sqrt{3}/3, \sqrt{3}/3,$ $\sqrt{3}/3, -\sqrt{3}/6)$	0	0	4	13.6557	4.3326
	M4B	$(\sqrt{3}/3, \sqrt{3}/3,$ $\sqrt{3}/3, -\sqrt{3}/6)$	0	0	6	4.0000	3.0908
	M6B	$(\sqrt{3}/3, \sqrt{3}/3,$ $\sqrt{3}/3, -\sqrt{3}/6)$	0	0	5	5.9999	3.0617
	M8A	$(\sqrt{3}/3, \sqrt{3}/3,$ $\sqrt{3}/3, -\sqrt{3}/6)$	0	0	5	8.0000	4.1511

Tabla 5.2 Ejemplos numéricos, continuación...

N°	Método	ξ	$\ x^{(k+1)} - x^{(k)}\ $	$\ F(x^{(k+1)})\ $	Iter	ρ	ϵ, s
	M10BG	$(\sqrt{3}/3, \sqrt{3}/3, \sqrt{3}/3, -\sqrt{3}/6)$	0	0	4	9.7219	3.4169
	M14BG	$(\sqrt{3}/3, \sqrt{3}/3, \sqrt{3}/3, -\sqrt{3}/6)$	0	0	4	13.6557	4.2221
	NC	$(\sqrt{3}/3, \sqrt{3}/3, \sqrt{3}/3, -\sqrt{3}/6)$	0	0	11	2.0000	2.6006
	TR	$(\sqrt{3}/3, \sqrt{3}/3, \sqrt{3}/3, -\sqrt{3}/6)$	0	0	7	3.0000	2.9968
	JT	$(\sqrt{3}/3, \sqrt{3}/3, \sqrt{3}/3, -\sqrt{3}/6)$	0	0	6	4.0000	2.8504
$F_3,$ $x_0 = (1, 1)$	M5S	$(\sqrt{3}/2, 0.5)$	0	0	6	5.0000	1.0759
	M10	$(\sqrt{3}/2, 0.5)$	0	0	4	9.5331	1.3520
	M4A	$(\sqrt{3}/2, 0.5)$	0	0	6	4.0000	1.2894
	M6A	$(\sqrt{3}/2, 0.5)$	0	0	5	5.9995	1.4301
	M8A	$(\sqrt{3}/2, 0.5)$	0	0	5	7.9998	1.8510
	M10AG	$(\sqrt{3}/2, 0.5)$	0	0	4	9.5351	1.4930
	M14AG	$(\sqrt{3}/2, 0.5)$	0	0	4	13.4272	1.8399
	M4B	$(\sqrt{3}/2, 0.5)$	0	0	6	4.0000	1.1089
	M6B	$(\sqrt{3}/2, 0.5)$	0	0	5	5.9995	1.4277
	M8B	$(\sqrt{3}/2, 0.5)$	0	0	5	7.9999	1.8080
	M10BG	$(\sqrt{3}/2, 0.5)$	0	0	4	9.5351	1.4294
	M14BG	$(\sqrt{3}/2, 0.5)$	0	0	4	13.4967	1.7485
	NC	$(\sqrt{3}/2, 0.5)$	0	0	11	2.0000	1.1475
	TR	$(\sqrt{3}/2, 0.5)$	0	0	8	3.0000	1.4459
JT	$(\sqrt{3}/2, 0.5)$	0	0	6	4.0000	1.0883	
$F_4,$ $x_0 = (0.8, \dots, 0.8)$	M5S	$(1, \dots, 1)$	0	0	5	5.0000	70.2377
	M10S	$(1, \dots, 1)$	0	0	4	10.0545	24.0777
	M4A	$(1, \dots, 1)$	0	0	6	4.0000	23.4114
	M6A	$(1, \dots, 1)$	0	0	5	6.0000	24.4916
	M8A	$(1, \dots, 1)$	0	0	4	8.0913	64.4585
	M10AG	$(1, \dots, 1)$	0	0	4	10.0545	23.4149
	M14AG	$(1, \dots, 1)$	0	0	4	14.0702	67.2670
	M4B	$(1, \dots, 1)$	0	0	6	4.0000	23.5516
	M6B	$(1, \dots, 1)$	0	0	5	6.0000	23.1829
	M8B	$(1, \dots, 1)$	0	0	4	8.0584	22.6821
	M10BG	$(1, \dots, 1)$	0	0	4	10.0545	23.0617
	M14BG	$(1, \dots, 1)$	0	0	4	14.0540	26.0373
	NC	$(1, \dots, 1)$	0	0	10	2.0000	23.6270
	TR	$(1, \dots, 1)$	0	0	7	3.0000	26.3586
JT	$(1, \dots, 1)$	0	0	6	4.0000	90.1726	
$F_5,$ $x_0 = (-0.5, -0.5)$	M5S	$(-0.8452567, -0.74814149)$	0	0	6	3.9670	0.8627
	M10S	$(-0.8452567, -0.74814149)$	0	0	4	9.9092	0.9529
	M4A	$(-0.8452567, -0.74814149)$	0	0	6	4.0279	1.0093
	M6A	$(-0.8452567, -0.74814149)$	0	0	5	5.9048	0.9841
	M8A	$(-0.8452567, -0.74814149)$	0	0	5	7.9706	1.3221

Tabla 5.2 Ejemplos numéricos, continuación...

N°	Método	ξ	$\ x^{(k+1)} - x^{(k)}\ $	$\ F(x^{(k+1)})\ $	Iter	ρ	ϵ, s
	M10AG	(-0.8452567 -0.74814149)	0	0	4	10.2609	1.1561
	M14AG	(-0.8452567 -0.74814149)	0	0	4	13.9829	1.5405
	M4B	(-0.8452567 -0.74814149)	0	0	6	4.0279	1.0403
	M6B	(-0.8452567 -0.74814149)	0	0	5	5.9048	1.0022
	M8B	(-0.8452567 -0.74814149)	0	0	4	7.7043	1.1407
	M10BG	(-0.8452567 -0.74814149)	0	0	4	10.2629	1.1777
	M14BG	(-0.8452567 -0.74814149)	0	0	4	13.8766	1.5027
	NC	(-0.8452567 -0.74814149)	0	0	11	2.0136	1.0904
	TR	(-0.8452567 -0.74814149)	0	0	8	3.0030	1.2538
	JT	(-0.8452567 -0.74814149)	0	0	7	3.9825	1.2363
$F_6,$ $x_0 = (0.5, -1.5)$	M5S	(1.0041687 -1.7296373)	0	0	6	3.9545	0.8146
	M10S	(1.0041687 -1.7296373)	0	0	4	10.1024	0.8893
	M4A	(1.0041687 -1.7296373)	0	0	6	4.0001	1.0507
	M6A	(1.0041687 -1.7296373)	0	0	5	5.9985	0.9377
	M8A	(1.0041687 -1.7296373)	0	0	5	7.9984	1.2901
	M10AG	(1.0041687 -1.7296373)	0	0	4	10.0477	1.0156
	M14AG	(1.0041687 -1.7296373)	0	0	4	14.1010	1.3327
	M4B	(1.0041687 -1.7296373)	0	0	6	4.0001	1.0988
	M6B	(1.0041687 -1.7296373)	0	0	5	5.9985	0.9566
	M8B	(1.0041687 -1.7296373)	0	0	5	7.9999	1.2576
	M10BG	(1.0041687 -1.7296373)	0	0	4	10.0477	1.0390
	M14BG	(1.0041687 -1.7296373)	0	0	4	14.0427	1.3177
	NC	(1.0041687 -1.7296373)	0	0	11	2.0000	1.0967
	TR	(1.0041687 -1.7296373)	0	0	8	3.0000	1.2501
	JT	(1.0041687 -1.7296373)	0	0	6	4.0001	1.0302

Tabla 5.2 Ejemplos numéricos, continuación...

N°	Método	ξ	$\ x^{(k+1)} - x^{(k)}\ $	$\ F(x^{(k+1)})\ $	Iter	ρ	ϵ, s
$F_7,$ $x_0 = (2 - 1.5, -0.5)$	M5S	(2.1402581, -2.0902946, -0.2235251)	0	0	6	3.9749	1.6827
	M10S	(2.1402581, -2.0902946, -0.22352512)	0	0	4	10.2161	1.6711
	M4A	(2.1402581, -2.0902946, -0.22352512)	0	0	8	4.0014	2.2441
	M6A	(2.1402581, -2.0902946, -0.22352512)	0	0	5	5.9956	1.8590
	M8A	(2.1402581, -2.0902946, -0.22352512)	0	0	5	8.0047	2.6465
	M10AG	(2.1402581, -2.0902946, -0.22352512)	0	0	4	10.2365	2.0874
	M14AG	(2.1402581, -2.0902946, -0.22352512)	0	0	4	14.4466	2.7984
	M4B	(2.1402581, -2.0902946, -0.22352512)	0	0	8	4.0014	2.4434
	M6B	(2.1402581, -2.0902946, -0.22352512)	0	0	5	5.9956	1.8656
	M8B	(2.1402581, -2.0902946, -0.22352512)	0	0	5	7.9957	2.5328
	M10BG	(2.1402581, -2.0902946, -0.22352512)	0	0	4	10.2365	2.1608
	M14BG	(2.1402581, -2.0902946, -0.22352512)	0	0	4	14.2296	2.4
	NC	(2.1402581, -2.0902946, -0.22352512)	0	0	11	2.0002	1.7010
	TR	(2.1402581, -2.0902946, -0.22352512)	0	0	9	3.0000	2.5486
JT	(2.1402581, -2.0902946, -0.22352512)	0	0	6	4.0009	1.8729	
$F_8,$ $x_0 = (-2, 2.4)$	M5S	(-2.4128971, 2.6433722, 4.7695249)	0	$4.7e-045$	13	1.0184	3.2757
	M10S	(-2.4128971, 2.6433722, 4.7695249)	0	$4.7e-045$	12	1.0042	5.4175

Tabla 5.2 Ejemplos numéricos, continuación...

N°	Método	ξ	$\ x^{(k+1)} - x^{(k)}\ $	$\ F(x^{(k+1)})\ $	Iter	ρ	ϵ, s
	M4A	(-2.4128971, 2.6433722, 4.7695249)	0	$4.7e-045$	22	1.0208	5.0676
	M6A	(-2.4128971, 2.6433722, 4.7695249)	0	$4.7e-045$	12	0.9740	4.5183
	M8A	(-2.4128971, 2.6433722, 4.7695249)	0	$4.7e-045$	9	1.0006	5.1310
	M10AG	(-2.4128971, 2.6433722, 4.7695249)	0	$4.7e-045$	12	0.9990	5.0473
	M14AG	(-2.4128971, 2.6433722, 4.7695249)	0	$4.7e-045$	9	1.0005	5.4104
	M4B	(-2.4128971, 2.6433722, 4.7695249)	0	$4.7e-045$	22	0.9451	5.0051
	M6B	(-2.4128971, 2.6433722, 4.7695249)	0	$4.7e-045$	12	0.9740	4.4103
	M8B	(-2.4128971, 2.6433722, 4.7695249)	0	$4.7e-045$	9	1.0006	3.9853
	M10BG	(-2.4128971, 2.6433722, 4.7695249)	0	$4.7e-045$	12	0.9990	5.0431
	M14BG	(-2.4128971, 2.6433722, 4.7695249)	0	$4.7e-045$	9	1.0005	5.8087
	NC	(-2.412897, 2.643372, 4.769524)	0	$4.7e-045$	24	0.9916	4.4071
	TR	(-2.412897, 2.643372, 4.769524)	0	$4.7e-045$	14	0.9801	3.4662
	JT	(-2.412897, 2.643372, 4.769524)	0	$4.7e-045$	22	0.9451	5.5333
	$F_9,$ $x_0 = (0.25, 0.75)$	M5S	(0, 1)	0	0	21	5.0000
M10S		(0, 1)	0	0	5	10.0000	1.6695
M4A		(0, 1)	0	0	7	4.0000	1.0012
M6A		(0, 1)	0	0	6	6.0000	1.1907
M8A		(0, 1)	0	0	7	7.9355	2.0286
M10AG		(0, 1)	0	0	5	9.9999	1.4439
M14AG		(0, 1)	0	0	5	13.9804	1.8678
M4B		(0, 1)	0	0	4	4.0000	1.1902
M6A		(0, 1)	0	0	6	6.0000	1.2215
M8A		(0, 1)	0	0	5	7.9924	1.3735
M10AG		(0, 1)	0	0	5	9.9989	1.4812
M14AG		(0, 1)	0	0	4	13.3566	1.5116

Tabla 5.2 Ejemplos numéricos, continuación...

N°	Método	ξ	$\ x^{(k+1)} - x^{(k)}\ $	$\ F(x^{(k+1)})\ $	Iter	ρ	ϵ, s
	NC	(0, 1)	0	0	13	2.0000	1.0765
	TR	(2, 1)	0	0	11	3.0000	1.5048
	JT	(0, 1)	0	0	7	4.0000	1.0794
$F_{10},$ $x_0 = (1, 6)$	M5S	(0.9548041 6.5849815)	0	$6.7e-044$	7	4.0370	6.7831
	M10S	(0.9548041 6.5849815)	0	$6.7e-044$	4	8.7191	4.7355
	M4A	(0.9548041 6.5849815)	0	$6.7e-044$	6	4.1118	3.7004
	M6A	(0.9548041 6.5849815)	0	$6.7e-044$	5	5.8439	7.4141
	M8A	(0.9548041, 6.5849815)	0	$6.7e-044$	5	7.9562	7.4593
	M10AG	(0.9548041, 6.5849815)	0	$6.7e-044$	5	10.4534	5.0725
	M14AG	(0.9548041, 6.5849815)	0	$6.7e-044$	4	13.9766	4.6319
	M4B	(0.9548041, 6.5849815)	0	$6.7e-044$	6	4.1118	3.2034
	M6B	(0.9548041, 6.5849815)	0	$6.7e-044$	5	5.8439	4.6319
	M8B	(0.9548041, 6.5849815)	0	$6.7e-044$	5	7.7268	6.6459
	M10BG	(0.9548041, 6.5849815)	0	$6.7e-044$	4	10.4534	5.0122
	M14BG	(0.9548041, 6.5849815)	0	$6.7e-044$	4	14.8388	6.5803
	NC	(0.9548041, 6.5849815)	0	$6.7e-044$	12	2.0164	3.1129
	TR	n.c.	n.c.	n.c.	n.c.	n.c.	n.c.
JT	(0.9548041, 6.5849815)	0	$6.7e-044$	6	4.1118	4.5939	
$F_{11},$ $x_0 = (0.5, 0.5)$	M5S	(0, 0)	0	0	6	6.5070	0.9432
	M10S	(0, 0)	0	0	5	3.4661	1.3963
	M4A	(0, 0)	0	0	8	2.0000	1.3184
	M6A	(0, 0)	0	0	5	3.6039	1.1569
	M8A	(0, 0)	0	0	5	8.0295	1.4589
	M10AG	(0, 0)	0	0	5	3.4734	1.5380
	M14AG	(0, 0)	0	0	5	11.7300	1.4745
	M4B	(0, 0)	0	0	4	2.0000	1.4101
	M6B	(0, 0)	0	0	5	3.6039	1.2160
	M8B	(0, 0)	0	0	5	8.0119	1.3735
	M10BG	(0, 0)	0	0	5	3.4734	1.5036
	M14BG	(0, 0)	0	0	4	11.6456	1.5116
	NC	(0, 0)	0	0	11	2.0000	1.2119
	TR	(0, 0)	0	0	7	3.0000	1.2191
JT	(0, 0)	0	0	8	2.0000	1.4298	
$F_{12},$ $x_0 = (0.8, 0.5, 1.4)$	M5S	(0.90956949, 0.66122683, 1.57583414)	0	0	6	4.1682	1.9089
	M10S	(0.90956949, 0.66122683, 1.57583414)	0	0	4	10.0105	2.0980

Tabla 5.2 Ejemplos numéricos, continuación....

N°	Método	ξ	$\ x^{(k+1)} - x^{(k)}\ $	$\ F(x^{(k+1)})\ $	Iter	ρ	ϵ, s
	M4A	(0.90956949, 0.66122683, 1.57583414)	0	0	6	3.9870	1.6417
	M6A	(0.90956949, 0.66122683, 1.57583414)	0	0	5	6.0333	2.2466
	M8A	(0.90956949, 0.66122683, 1.57583414)	0	0	5	7.9334	3.2194
	M10AG	(0.90956949, 0.66122683, 1.57583414)	0	0	4	9.7330	2.3229
	M14AG	(0.90956949, 0.66122683, 1.57583414)	0	0	4	13.7787	3.1679
	M4B	(0.90956949, 0.66122683, 1.57583414)	0	0	6	3.9870	2.3410
	M6B	(0.90956949, 0.66122683, 1.57583414)	0	0	5	6.0333	3.2194
	M8B	(0.90956949, 0.66122683, 1.57583414)	0	0	5	7.9874	3.1751
	M10BG	(0.90956949, 0.66122683, 1.57583414)	0	0	4	9.7330	2.2466
	M14BG	(0.90956949, 0.66122683, 1.57583414)	0	0	4	13.7787	2.8624
	NC	(0.90956949, 0.66122683, 1.57583414)	0	0	11	2.0001	2.0123
	TR	(0.90956949, 0.66122683, 1.57583414)	0	0	8	3.0000	2.5194
	JT	(0.90956949, 0.66122683, 1.57583414)	0	0	6	3.9870	2.0186

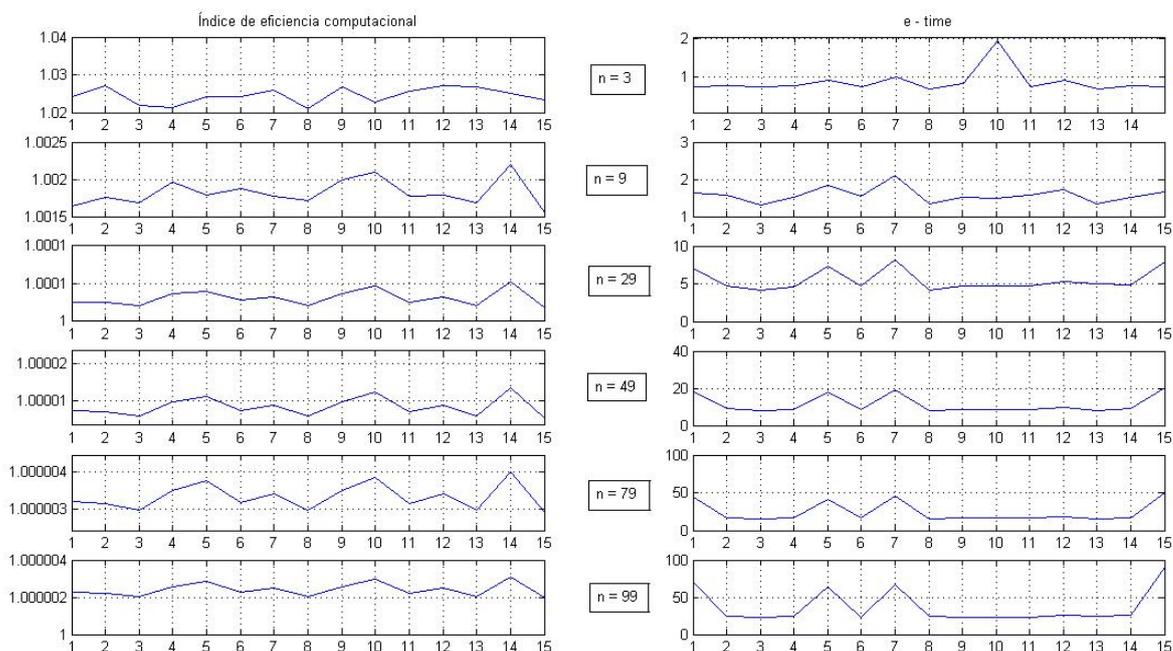


Figura 5.13: Gráficas de índice de eficiencia computacional y de $e - time$ de los métodos de la Tabla 5.2 para el sistema $F_4(x) = 0$

De nuevo notamos que, en general, a mayor índice computacional le corresponde menor tiempo de ejecución. Se puede concluir que el método M5S es mejor para sistemas pequeños. Aumentando el tamaño del sistema los métodos más rápidos son M4A y M4B. En caso de sistemas con $n = 99$ los mejores métodos son M8B y M10BG. Además, notamos que los métodos M14AG y M14BG en todos los casos necesitan más tiempo que el resto de los métodos diseñados. Comparando estos dos métodos con JT y TR vemos que desde $n = 29$ los métodos M14AG y M14BG son mejores.

5.3.1. Determinación de la órbita preliminar de un satélite artificial

Entre los numerosos problemas que resuelve la Mecánica Celeste está el de la determinación preliminar de las órbitas que describen los cuerpos celestes. En la actualidad los métodos de determinación de órbitas son un instrumento esencial que se aplica al cálculo de órbitas de satélites artificiales y constituyen un elemento fundamental para el control de la navegación, seguimiento y supervisión. Para resolver el problema de la determinación preliminar de las órbitas en primera aproximación, es decir, considerando el movimiento de los cuerpos basado únicamente en la ley de la gravitación universal, se hallan, en primer lugar, los seis elementos de la órbita preliminar (elementos orbitales o elementos Keplerianos). Para su determinación se define el sistema de coordenadas inercial geocéntrico ecuatorial con origen el geocentro de la Tierra (véase Figura 5.14).

Se define como plano fundamental el plano ecuatorial XY con el eje X apuntando a Aries y el eje fundamental Z que coincide con el eje de rotación de la Tierra. La posición de la órbita con respecto al sistema XYZ queda determinada por medio de tres de los elementos orbitales que determinan la orientación del plano de la órbita - los ángulos de Euler:

1. La ascensión recta del nodo escendente, Ω , que es el ángulo formado por el eje X y la línea de nodos en dirección del nodo ascendente.
2. La inclinación de la órbita, i , que es el ángulo formado por el plano de la órbita y el plano fundamental.

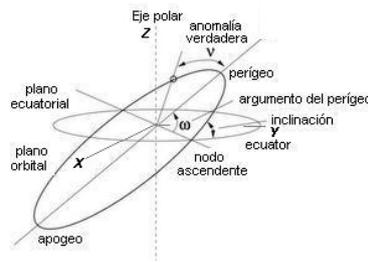


Figura 5.14: Elementos orbitales

3. El argumento del perigeo, ω , que es el ángulo que forman la línea de los nodos desde el nodo ascendente hasta el perigeo, medido en el plano orbital del objeto en sentido de su movimiento.

Otros dos de los elementos orbitales: el semieje mayor a y la excentricidad e sirven para determinar la magnitud y la forma de la órbita. Finalmente, es necesario obtener la época de paso por el perigeo, T .

Se han desarrollado diferentes métodos para determinación de los elementos orbitales (véase [25], [31] y [79]) casi todos derivados de los de Laplace (1789), Olbers (1797) y Gauss (1809). Mediante el uso de estos métodos, es posible determinar los elementos orbitales de la órbita preliminar conociendo las coordenadas de posición y velocidad en un determinado tiempo que puede ser refinada con observaciones posteriores desde las estaciones de la Tierra. Para conseguir la posición del objeto en la elipse se deben determinar algunos ángulos (anomalías) de la órbita: anomalía verdadera, ν , y la anomalía excéntrica, E (véase Figura 5.15). La anomalía verdadera es un parámetro que define la posición de un cuerpo que se mueve a lo largo de una órbita de Kepler, representada por el ángulo \widehat{AFP} que forma el radio vector del objeto trazado desde el foco y el semieje mayor. La anomalía excéntrica es el ángulo $\widehat{AOP'}$ que forma el radio desde el centro de la circunferencia auxiliar hasta el punto P' obtenido al trazar la perpendicular desde punto P hasta el semieje mayor.

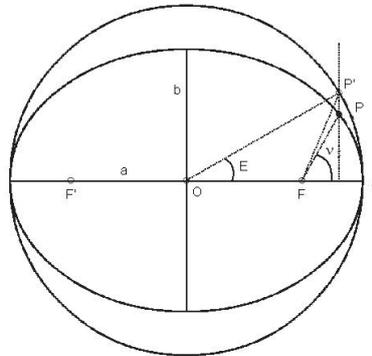


Figura 5.15: Anomalías verdadera, ν , y excéntrica, E

Utilizando la Tierra como centro del sistema de coordenadas establecemos como unidad de distancia el radio de la Tierra, $e.r.$ (aproximadamente 6,370 kilómetros), y unidad de tiempo el minuto (en ocasiones se utilizan días julianos, JD). Las constantes fundamentales expresadas en estas unidades son: constante gravitacional de la Tierra, $k = \sqrt{G \cdot m_{tierra}} = 0.07436574(e.r)^{3/2}/min$, deducidas a partir de la constante de gravitación universal, G , y la masa de la Tierra, m_{tierra} (véase [31]). En nuestro caso los objetos de estudio son los satélites artificiales cuya masa se puede despreciar por ser muy pequeña comparada con la de la Tierra. Por lo tanto, la relación entre las masas es unidad, $\mu = \frac{1}{m_{tierra}}(m_{tierra} + m_{objeto}) \approx 1$. Se modifica la variable tiempo para la simplificación de los cálculos y se presenta como:

$$\tau = k(t_2 - t_1), \tag{5.62}$$

donde t_1 es el tiempo arbitrario inicial y t_2 es el tiempo de observación.

Para la estimación de la velocidad se pueden usar las formas cerradas de las series:

$$f = 1 - \frac{a}{|r_1|} [1 - \cos(E_2 - E_1)], \quad (5.63)$$

$$g = \tau - \frac{\sqrt{a^3}}{\mu} [(E_2 - E_1) - \sin(E_2 - E_1)], \quad (5.64)$$

por lo que se puede expresar la diferencia entre dos vectores de posición como

$$\dot{\vec{r}} = \frac{\vec{r}_2 - f\vec{r}_1}{g}. \quad (5.65)$$

Por lo tanto, es evidente que, conocidos dos vectores de posición en sus respectivos instantes de tiempo, el objetivo principal de los diferentes métodos para determinar las órbitas preliminares es el cálculo de la semieje mayor a y la diferencia de las anomalías excéntricas, $E_2 - E_1$. Cuando se determinan, utilizando (5.65) se puede obtener el vector velocidad que corresponde a uno de los vectores posición conocido y, luego, obtener los elementos orbitales. La mayoría de estos métodos tienen algo en común: la necesidad de obtener la solución de una ecuación no lineal (o un sistema, como en el método de Gauss). Por lo general, los métodos clásicos utilizan el método de punto fijo o el método de la secante.

De los datos de entrada disponibles (dos vectores de posición y los tiempos de observación), se puede deducir la diferencia de las anomalías verdaderas, $\nu_2 - \nu_1$, mediante:

$$\cos(\nu_2 - \nu_1) = \frac{\vec{r}_1 \cdot \vec{r}_2}{|\vec{r}_1| \cdot |\vec{r}_2|}, \quad (5.66)$$

$$\sin(\nu_2 - \nu_1) = \pm \frac{x_1 y_2 - x_2 y_1}{|x_1 y_2 - x_2 y_1|} \sqrt{1 - \cos(\nu_2 - \nu_1)}, \quad (5.67)$$

donde el signo positivo es para órbitas directas, y el signo negativo para órbitas retrógradas. Luego se utiliza un método específico para la determinación de la órbita. En nuestro caso vamos a introducir el método de Gauss. Este método utiliza solamente dos observaciones (vectores de posición). Está basado en la relación entre las áreas del sector ABC y el triángulo ABC ilustrados en la Figura 5.16,

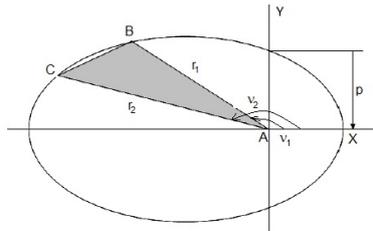


Figura 5.16: Relación de áreas

determinada por los dos vectores de posición \vec{r}_1 y \vec{r}_2 según la fórmula:

$$y = \frac{\sqrt{\mu p} \cdot \tau}{r_2 r_1 \sin(\nu_2 - \nu_1)} = \frac{\sqrt{\mu} \cdot \tau}{2\sqrt{a}\sqrt{r_2 r_1} \sin\left(\frac{E_2 - E_1}{2}\right) \cos\left(\frac{\nu_2 - \nu_1}{2}\right)}, \quad (5.68)$$

con $(\nu_2 - \nu_1) \neq \pi$. De la primera igualdad se obtiene la primera

$$y^2 = \frac{m}{l + x} \quad (5.69)$$

y la segunda

$$y^2(y - 1) = mX \quad (5.70)$$

ecuaciones de Gauss, donde las constantes del problema

$$l = \frac{r_2 + r_1}{4\sqrt{r_2 r_1} \cos\left(\frac{\nu_2 - \nu_1}{2}\right)} - \frac{1}{2}, \quad (5.71)$$

$$m = \frac{\mu\tau^2}{[2\sqrt{r_2 r_1} \cos\left(\frac{\nu_2 - \nu_1}{2}\right)]^2}, \quad (5.72)$$

se obtienen a partir de los datos y empleando (5.65), (5.66) y (5.67). Se deben determinar también los valores de

$$x = \frac{1}{2} \left[1 - \cos\left(\frac{E_2 - E_1}{2}\right) \right] = \sin^2\left(\frac{E_2 - E_1}{4}\right), \quad (5.73)$$

$$X = \frac{E_2 - E_1 - \sin(E_2 - E_1)}{\sin^3\left(\frac{E_2 - E_1}{2}\right)}. \quad (5.74)$$

Nos proponemos resolver el problema siguiendo diferentes esquemas: el método clásico y el método modificado. El método clásico reduce las dos ecuaciones de Gauss a una única ecuación no lineal que se obtiene sustituyendo la segunda ecuación de Gauss (5.70) en la primera ecuación (5.69):

$$y = 1 + X(l + x). \quad (5.75)$$

Utilizando el método de punto fijo para estimar la solución de (5.75), tomando la estimación inicial $y_0 = 1$ y utilizando la primera ecuación de Gauss se obtiene x_0

$$x_0 = \frac{m}{y_0^2} - l. \quad (5.76)$$

Utilizando la ecuación (5.73) se calculan

$$\begin{aligned} \cos\left(\frac{E_2 - E_1}{2}\right) &= 1 - 2x_0, \\ \sin\left(\frac{E_2 - E_1}{2}\right) &= +\sqrt{4x_0(1 - x_0)}, \end{aligned}$$

determinando la diferencia de las anomalías excéntricas. Luego, con la ecuación (5.74) se obtiene la estimación de X , X_0 , que permite, utilizando la ecuación reducida (5.75), estimar y_1

$$y_1 = 1 + X_0(l + x_0).$$

El proceso iterativo continua hasta que se cumple una condición de tolerancia especificada.

Aquí, proponemos un nuevo enfoque para el problema de la determinación de la órbita preliminar, que consiste en resolver directamente el sistema no lineal formado por ambas ecuaciones de Gauss, por medio de métodos conocidos tales como Newton, Traub, Jarratt y los métodos diseñados en la Sección 5.2. Llamamos a esta técnica Método modificado de Gauss y, utilizando la relación $u = y$ y la diferencia de las anomalías excéntrica, $v = E_2 - E_1$, como nuestras incógnitas, establecemos el sistema a resolver al sustituir (5.73) y (5.74) en la primera y segunda ecuación de Gauss, (5.69) y (5.70) obteniendo el sistema $F(u, v) = 0$ en la forma:

$$u^2 l + \frac{u^2}{2} \left(1 - \cos\frac{v}{2} \right) - m = 0, \quad (5.77)$$

$$u^3 + u^2 - m \frac{v - \sin v}{\sin^3 \frac{v}{2}} = 0 \quad (5.78)$$

donde l y m son las constantes descritas en (5.71) y (5.72).

Los cálculos numéricos se han llevado a cabo utilizando aritmética de precisión variable, con 500 dígitos, en MATLAB 7.1. El criterio de parada utilizado es $\|x^{(k+1)} - x^{(k)}\| + \|F(x^{(k)})\| < 10^{-250}$, por lo tanto comprobamos que la sucesión de iteraciones converge a la aproximación de la solución del sistema no lineal. Para cada método

contamos el número de iteraciones necesarias para alcanzar la tolerancia deseada y el tiempo transcurrido. La órbita de referencia o prueba que se ha usado es la siguiente:

$$\vec{r}_1 = [2.46080928705339, 2.04052290636432, 0.14381905768815]e.r.$$

$$\vec{r}_2 = [1.98804155574820, 2.50333354505224, 0.31455350605251]e.r.$$

$$t_1 = 0JD, \quad t_2 = 0.01044412000000JD,$$

$$\Omega = 30^\circ, \quad \omega = 10^\circ, \quad i = 15^\circ, \quad a = 4e.r., \quad e = 0.2, \quad T = 0 \text{ min}$$

y se puede encontrar en [31].

Tabla 5.3: Comparación de los diferentes métodos utilizados

	punto fijo	Newton	Traub	Jarratt	HMT
ρ	1.0022	1.9029	3.0000	4.0000	4.0000
i	$4.4259e - 199$				
ω	$1.6900e - 069$				
Ω	$9.9573e - 199$				
a	$3.3032e - 070$				
e	$6.6064e - 071$				
T	$2.0726e - 069$				
$\ x^{(k+1)} - x^{(k)}\ $	$6.5929e - 202$	$1.5335e - 264$	0	0	0
<i>Iter</i>	107	9	6	5	5
ϵ	6.8438	3.0781	3.0625	2.9281	4.3125
	M5S	M6SA	M6SB	M8SA	M8SB
ρ	4.3177	6.0000	6.0000	7.8892	7.9123
i	$4.4259e - 199$				
ω	$1.6900e - 069$				
Ω	$9.9573e - 199$				
a	$3.3032e - 070$				
e	$6.6064e - 071$				
T	$2.0726e - 069$				
$\ x^{(k+1)} - x^{(k)}\ $	0	$1.9853e - 208$	$1.9853e - 208$	0	0
<i>Iter</i>	4	4	4	4	4
ϵ	3.8594	3.0313	2.9844	3.9688	3.6719

En la Tabla 5.3 se muestran los resultados obtenidos incluyendo la aproximación del orden de convergencia ρ , el error absoluto cometido en la determinación de los elementos orbitales ($i, \omega, \Omega, a, e, T$), la diferencia entre los dos últimos iterados (en norma), el número de iteraciones llevadas a cabo por el método y el tiempo de ejecución, ϵ . Se pueden extraer varias conclusiones:

- En el caso del método clásico de Gauss (de punto fijo), el número de iteraciones y el tiempo de ejecución son mucho mayores de los que se obtienen aplicando el método modificado de Gauss.
- Los métodos de orden 5, 6 y 8 son los que requieren un menor número de iteraciones.
- El método de Jarratt y el método M6SB son los que necesitan menor tiempo de ejecución.
- Con los métodos de orden ocho no se disminuye el número de iteraciones frente a los métodos de órdenes 5 y 6 y el tiempo de ejecución es mayor.
- Los resultados obtenidos tras la aplicación de los métodos de órdenes 10 y 14 no están incluidos en la tabla porque no mejoran el número de iteraciones y aumentan el tiempo de ejecución debido al mal condicionamiento de la matriz Jacobiana $J_F(u, v)$ asociada al sistema. Por otra parte, como el tamaño del sistema es pequeño, el efecto del menor número de evaluaciones y operaciones realizadas con la matriz Jacobiana para los métodos de orden menor que ocho no es muy visible. En este caso, los métodos M6A y M6B muestran ser muy eficientes.

5.3.2. Aplicación a la resolución de problemas de frontera

Problema 1: Consideremos el siguiente problema de frontera

$$y'' = -(y')^2 - y + \ln x, \quad x \in [1, 2]$$

$$y(1) = 0, \quad y(2) = \ln 2$$

con solución exacta $y(x) = \ln x$. Al utilizar el método de diferencias finitas de segundo orden, los ceros de la siguiente función no lineal nos darán una estimación de la solución del problema de frontera:

$$F(y) = (f_1(y), f_2(y), \dots, f_{n-1}(y))^T,$$

donde $y = (y_1, y_2, \dots, y_{n-1})^T$ y $f_i: \mathbb{R}^{n-1} \rightarrow \mathbb{R}$, $i = 1, 2, \dots, n-1$, de tal manera que

$$\begin{aligned} f_1(y) &= 4y_2 + y_2^2 + 4y_1(h^2 - 2) - 4h^2 \ln x_2, \\ f_i(y) &= 4(y_{i+1} + y_{i-1}) + (y_{i+1} - y_{i-1})^2 + 4y_i(h^2 - 2) - 4h^2 \ln x_{i+1}, \quad i = 2, 3, \dots, n-2 \\ f_{n-1}(y) &= 4(\ln 2 + y_{n-2}) + (\ln 2 - y_{n-2})^2 + 4y_{n-1}(h^2 - 2) - 4h^2 \ln x_n \end{aligned}$$

En esta función, los nodos $x_i = 1 + ih$, $i = 0, 1, \dots, n$, se obtienen con el paso $h = \frac{1}{n}$ y se utiliza la aproximación de segundo orden para $y'(x_i)$ y $y''(x_i)$. Denotamos la estimación de la solución en cada nodo x_i $y(x_i)$ por y_i , $i = 1, \dots, n-1$.

Los cálculos numéricos se han llevado a cabo utilizando aritmética de precisión variable, con 2000 dígitos, en MATLAB 7.1. El criterio de parada utilizado es $\|x^{(k+1)} - x^{(k)}\| + \|F(x^{(k)})\| < 10^{-400}$, por lo tanto comprobamos que la sucesión de iteraciones converge a la aproximación de la solución del sistema no lineal. Para cada método contamos el número de iteraciones necesarias para alcanzar la tolerancia deseada, el orden de convergencia y el tiempo transcurrido. Los resultados obtenidos están mostrados en la Tabla 5.4. Notamos que:

1. el número de iteraciones que necesita cada método no depende del tamaño de sistema;
2. el orden de convergencia obtenido está en concordancia con el teórico;
3. para el sistema con $n = 10$ los mejores tiempos corresponden a los métodos M4A y M4B;
4. si el sistema es con $n = 25$ el método de NC obtiene los resultados con el menor tiempo y lo siguen M4A y M4B;
5. en el caso de sistema con $n = 50$ los métodos M4B y M4A obtienen los mejores tiempos;
6. y para el sistema con $n = 75$ los mejores tiempos corresponden a los métodos M4A y NC;

La solución que se obtiene aplicando los distintos métodos es la misma. No la presentamos en la tabla por la gran cantidad de dígitos que contiene. Para visualizar el comportamiento de los métodos cambiando n , calculamos el error absoluto en cada caso.

En la Figura 5.17 se muestran las gráficas del error absoluto para $n = 10, 25, 50$ y 75 . Analizando las gráficas mostradas observamos que el error absoluto disminuye aumentando el tamaño del sistema.

Problema 2: Consideremos el problema de frontera

$$y''(x) = -(1 + a^2(y')^2), \quad x \in [0, 1]$$

$$y(0) = 0, \quad y(1) = 0$$

(véase [3]). Este problema se plantea en el estudio de las deflexiones finitas de una cuerda elástica con una carga transversal y su solución exacta es

$$y(x) = \ln \left(\frac{\cos(a(x - 1/2))}{\cos a/2} \right) \frac{1}{a^2}.$$

Tabla 5.4: Comparación de los diferentes métodos utilizados (problema de frontera 1)

Parámetro	NC	TR	JT	M5S	M10S	M4A	M4B	M6A
Sistema $n = 10$								
<i>Iter</i>	11	8	6	6	4	6	6	5
p	1.9997	2.9992	4.0011	4.0414	10.4480	4.0011	4.0011	6.0168
ϵ, s	3.7393	3.7729	3.1612	3.2671	3.5745	2.8859	2.9072	3.5831
Sistema $n = 25$								
<i>Iter</i>	11	8	6	6	4	6	6	5
p	2.0001	2.9967	4.0056	4.0260	10.4480	4.0056	4.0056	6.0336
ϵ, s	8.2872	10.7743	11.8633	11.9640	9.7858	8.3026	8.3643	9.9132
Sistema $n = 50$								
<i>Iter</i>	11	8	6	6	4	6	6	5
p	2.0005	2.9996	4.0030	4.0214	10.6150	4.0030	4.0030	6.0251
ϵ, s	19.6071	23.7650	34.1808	34.2814	22.4888	18.5489	18.5449	21.3354
Sistema $n = 75$								
<i>Iter</i>	11	8	6	6	4	6	6	5
p	2.0007	3.0005	4.0027	4.0212	10.6220	4.0027	4.0027	6.0194
ϵ, s	32.3045	40.3444	66.1215	66.9715	37.4851	32.0736	32.5235	36.0291
Parámetro	M6B	M8A	M8B	M10AG	M10BG	M14AG	M14BG	NeGLD1S
Sistema $n = 10$								
<i>Iter</i>	5	5	5	4	4	4	4	7
p	6.0168	8.0067	10.4480	10.4480	10.4480	14.5870	14.4570	3.0000
ϵ, s	3.4913	4.9169	4.6663	3.7432	3.8063	4.9264	4.6396	3.7588
Sistema $n = 25$								
<i>Iter</i>	5	5	5	4	4	4	4	7
ρ	6.0336	8.0238	8.0193	10.5810	10.5810	14.8000	14.6250	3.0024
ϵ, s	10.0629	15.1171	12.2736	10.0622	10.1081	14.5888	12.2079	9.6792
Sistema $n = 50$								
<i>Iter</i>	5	5	5	4	4	4	4	7
ρ	6.0251	8.0305	8.0282	10.6150	10.6150	14.9080	14.6730	3.0025
ϵ, s	21.3505	40.6638	27.4005	21.9530	21.9432	37.2559	26.6334	20.7464
Sistema $n = 75$								
<i>Iter</i>	5	5	5	4	4	4	4	7
ρ	6.0194	8.0334	8.0255	10.6220	10.6220	14.9470	14.6800	3.0014
ϵ, s	36.1041	79.8391	51.5608	37.7581	38.0828	79.9892	50.5288	35.5994
Parámetro	NeGLD2S	NeGLT2S	NeGR2S	TrGLD1S	TrGLD2S	TrGLT2S	TrGR2S	DCG
Sistema $n = 10$								
<i>Iter</i>	7	7	7	6	6	6	6	5
ρ	3.0000	3.0000	3.0000	4.9982	4.9982	4.9982	4.9982	7.0214
ϵ, s	4.8023	3.2606	3.4025	4.1888	6.1013	4.1316	4.5614	4.7398
Sistema $n = 25$								
<i>Iter</i>	7	7	7	6	6	6	6	5
ρ	3.0024	3.0024	3.0024	5.0036	5.0036	5.0036	5.0036	7.0563
ϵ, s	13.4307	9.9436	9.9731	11.9956	15.3133	12.3277	11.9100	11.9474
Sistema $n = 50$								
<i>Iter</i>	7	7	7	6	6	6	6	5
ρ	3.0025	3.0025	3.0025	5.0011	5.0011	5.0011	5.0011	7.0647
ϵ, s	28.8947	21.3047	21.2611	24.8669	31.9843	25.0037	25.3531	26.6676
Sistema $n = 75$								
<i>Iter</i>	7	7	7	6	6	6	6	5
ρ	3.0014	3.0014	3.0014	5.0007	5.0007	5.0007	5.0007	7.0592
ϵ, s	53.5852	38.0168	37.9365	44.3291	60.0891	46.7360	47.0470	50.9537

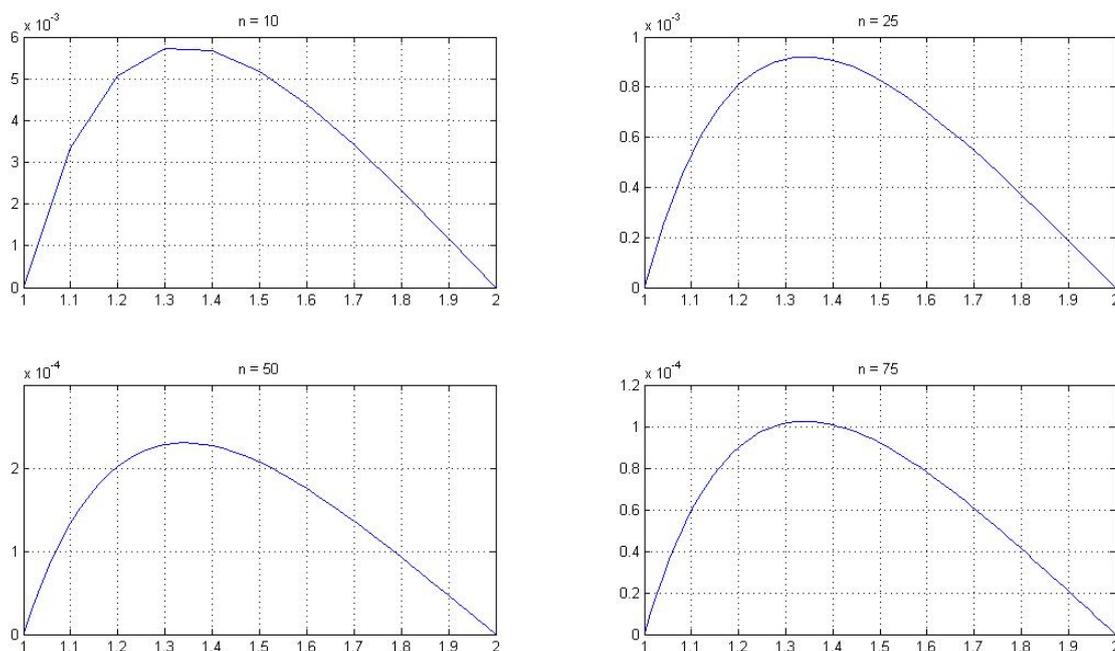


Figura 5.17: Error absoluto de las soluciones del problema 1 para diferentes valores de n

Con el fin de obtener una estimación de esta solución, utilizamos el método de diferencias finitas de segundo orden y, por tanto, el siguiente sistema no lineal nos proporciona una estimación de la solución del problema de contorno original:

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + 1 + a^2 \left(\frac{y_{i+1} - y_{i-1}}{2h} \right)^2 = 0, \quad i = 1, 2, 3, \dots, n, \quad (5.79)$$

donde y_i representa la aproximación de la solución y en el nodo $x_i = ih$, $i = 0, 1, \dots, n$, siendo $h = \frac{1}{n}$. Los resultados obtenidos mediante la aplicación de diferentes métodos en el sistema (5.79), con $a = 1/7$, de igual manera que en el problema 1 realizamos los cálculos numéricos utilizando aritmética de precisión variable, con 2000 dígitos, en MATLAB 7.1 con el mismo criterio de parada $\|x^{(k+1)} - x^{(k)}\| + \|F(x^{(k)})\| < 10^{-400}$, por lo tanto comprobamos que la sucesión de iteraciones converge a la aproximación de la solución del sistema no lineal. Contamos el número de iteraciones necesarias para alcanzar la tolerancia deseada, el orden de convergencia y el tiempo transcurrido para cada método. Los resultados obtenidos se muestran en la Tabla 5.5. Notamos que el orden de convergencia obtenido está en concordancia con el teórico y que el número de iteraciones que necesita cada método no depende del tamaño del sistema. Además, podemos ordenar los mejores cinco métodos de la siguiente manera según el tiempo transcurrido

1. con $n = 10$: $M4B < M4A < M6B < M6A < TR$;
2. para $n = 25$: $M4A < M4B < NC < M6B < M6A$;
3. cuando $n = 50$: $M4A < NC < M4B < M6B < NeG - R2S$;
4. y si $n = 75$: $M4A < M4B < NC < M6B < M6A$;

La solución que se obtiene aplicando los distintos métodos es la misma, aunque por la gran cantidad de dígitos que contiene no aparece en la Tabla 5.5.

Tabla 5.5: Comparación de los diferentes métodos utilizados (problema de frontera 2)

Parámetro	NC	TR	JT	M5S	M10S	M4A	M4B	M6A
Sistema $n = 10$								
<i>Iter</i>	9	6	5	5	4	5	5	4
ρ	2.0000	3.0000	4.0000	4.2263	10.0060	4.0000	4.0000	6.0067
ϵ, s	3.3095	3.0016	3.1320	3.1601	3.5835	2.5357	2.5105	2.8540
Sistema $n = 25$								
<i>Iter</i>	9	6	5	5	4	5	5	4
ρ	2.0000	3.0000	4.0000	4.1169	10.0140	4.0000	4.0000	6.0148
ϵ, s	6.7491	7.9697	9.4713	9.6287	9.4217	6.5889	6.6335	7.4781
Sistema $n = 50$								
<i>Iter</i>	9	6	5	5	4	5	5	4
ρ	2.0000	3.0000	4.0000	4.0697	10.0160	4.0000	4.0000	6.0180
ϵ, s	15.4806	17.9677	28.1644	28.3998	22.4888	21.8672	15.3352	17.2100
Sistema $n = 75$								
<i>Iter</i>	9	6	5	5	4	5	5	4
ρ	2.0000	3.0000	4.0000	4.0764	10.0170	4.0000	4.0000	6.0191
ϵ, s	26.9700	30.4810	55.0713	55.6658	36.8947	26.0111	26.1509	29.1288
Parámetro	M6B	M8A	M8B	M10AG	M10BG	M14AG	M14BG	NeGLD1S
Sistema $n = 10$								
<i>Iter</i>	4	4	4	4	4	3	3	6
ρ	6.0067	8.0067	8.0064	10.0060	10.0060	—	—	3.0000
ϵ, s	2.8345	4.0864	3.6657	3.5895	3.5977	3.6853	3.3825	3.0095
Sistema $n = 25$								
<i>Iter</i>	4	4	4	4	4	3	3	6
ρ	6.0148	8.0149	8.0142	10.0140	10.0140	—	—	3.0000
ϵ, s	7.4291	11.8566	9.6051	9.2990	9.3721	10.4251	8.9255	8.1550
Sistema $n = 50$								
<i>Iter</i>	4	4	4	4	4	3	3	6
ρ	6.0180	8.0181	8.0173	10.0160	10.0160	—	—	3.0000
ϵ, s	17.0623	32.5463	21.8187	21.5460	21.5062	27.7027	20.1881	17.3515
Sistema $n = 75$								
<i>Iter</i>	4	4	4	4	4	3	3	6
ρ	6.0191	8.0193	8.0184	10.0170	10.0170	—	—	3.0000
ϵ, s	28.8064	59.9139	36.7385	36.6763	36.3293	51.0188	33.6438	29.6059
Parámetro	NeGLD2S	NeGLT2S	NeGR2S	TrGLD1S	TrGLD2S	TrGLT2S	TrGR2S	DCG
Sistema $n = 10$								
<i>Iter</i>	6	6	6	5	5	5	5	4
ρ	3.0000	3.0000	3.0000	5.0000	5.0000	5.0000	5.0000	7.0066
ϵ, s	4.1753	3.0917	3.1182	3.4890	4.6537	3.4812	3.7650	3.8084
Sistema $n = 25$								
<i>Iter</i>	6	6	6	5	5	5	5	4
ρ	3.0000	3.0000	3.0000	5.0000	5.0000	5.0000	5.0000	7.0150
ϵ, s	11.1473	7.7777	8.3872	9.3816	12.1995	9.9093	9.7285	9.3349
Sistema $n = 50$								
<i>Iter</i>	6	6	6	5	5	5	5	4
ρ	3.0000	3.0000	3.0000	5.0000	5.0000	5.0000	5.0000	7.0185
ϵ, s	24.3572	17.9506	17.9231	20.4797	26.3218	20.8672	21.2243	21.7669
Sistema $n = 75$								
<i>Iter</i>	6	6	6	5	5	5	5	4
ρ	3.0000	3.0000	3.0000	5.0000	5.0000	5.0000	5.0000	7.0198
ϵ, s	41.6299	29.9881	30.5291	34.9632	47.1036	35.5464	35.6051	37.0040

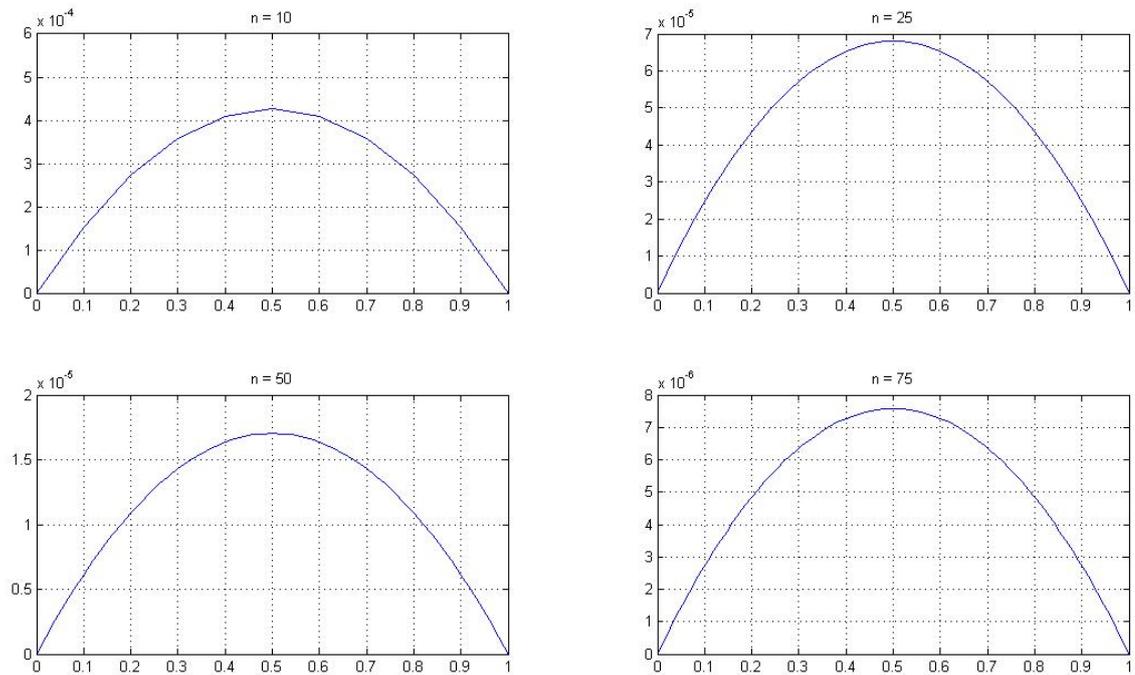


Figura 5.18: Error absoluto de las soluciones del problema 2 para diferentes valores de n

En la Figura 5.18 se muestran las gráficas del error absoluto para $n = 10, 25, 50$ y 75 , donde observamos que el error absoluto disminuye aumentando el tamaño del sistema.

En el caso particular de los problemas de frontera, observamos que los métodos más eficientes, independientemente del tamaño del sistema, son los de orden cuatro, M4A y M4B. Probablemente esto sea debido a la estructura poco densa (triangular) de las matrices Jacobianas implicadas en la resolución de los sistemas lineales, en cada paso del proceso iterativo.

En general, se ha observado que los métodos de alto orden son especialmente eficientes en sistemas de tamaño muy grande, mientras que para sistemas pequeños son más apropiados los métodos de bajo orden de convergencia. Esta relación la marca el índice de eficiencia computacional de los métodos, que depende claramente del tamaño del sistema a resolver (véase las Figuras 5.11 y 5.13).

PROBLEMAS DE VALOR INICIAL Y MÉTODOS ITERATIVOS

Las fórmulas de cuadratura en general, y las de Gauss en particular, nos permiten aproximar numéricamente, la solución tanto de ecuaciones o sistemas de ecuaciones no lineales, generando métodos iterativos como los descritos en esta memoria, como de problemas de valor inicial, lineales o no (véase Figura 6.1). De este modo, las fórmulas de cuadratura constituyen el nexo de unión entre ambos problemas.

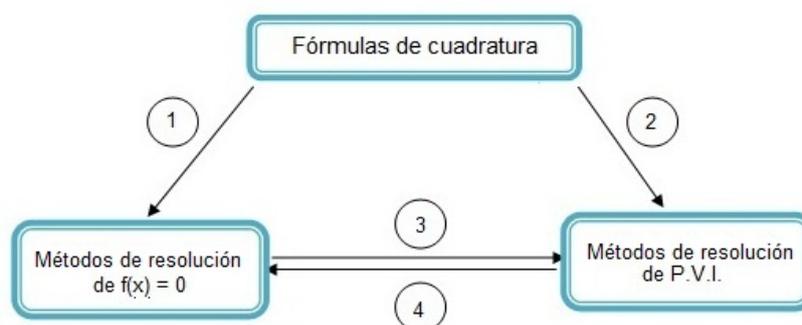


Figura 6.1: PVI vs $f(x) = 0$

Siguiendo el camino 1 de la Figura 6.1 para obtener nuevos métodos iterativos se han utilizado distintas fórmulas de integración numérica. Por ejemplo, Weerakoon y Fernando en [82] han aplicado la regla trapezoidal, Frontini y Sormani en [38] y Özban en [73] la regla de punto medio. En [16, 17], Cordero y Torregrosa han aplicado las fórmulas de cuadratura cerradas y abiertas para desarrollar familias de variantes del método de Newton que incluyen las familias de métodos definidos por Frontini et al. en [38]. En [18], los mismos autores utilizan la fórmula genérica de cuadratura interpolatoria para obtener otra familia de métodos, variantes de Newton para sistemas no lineales.

En los capítulos anteriores de esta memoria hemos desarrollado conjuntos de familias de métodos provenientes de la cuadratura de Gauss. Utilizando como predictor el método de Newton y la cuadratura de Gauss-Legendre de un nodo como corrector hemos obtenido el método iterativo de punto medio. En el caso de utilizar la cuadratura de Gauss-Lobatto con uno, dos y tres nodos se han obtenido los métodos iterativos conocidos como punto medio, trapecios y Simpson, respectivamente. Por último, utilizando la cuadratura de Gauss-Radau se obtienen los métodos de Newton y el de Noor, empleando uno y dos nodos, respectivamente. Se ha demostrado que aumentando el número de nodos no se puede aumentar el orden de convergencia propio de la familia de métodos. Además, los métodos generados utilizando mayor cantidad de nodos requieren más evaluaciones de derivadas para ecuaciones, y de matrices Jacobianas para sistemas, lo que disminuye la eficiencia computacional de los métodos, así obtenidos.

6.1. Métodos provenientes de la cuadratura gaussiana

Los conceptos previos acerca de problemas de valor inicial han sido introducidos en el Capítulo 2. Consideramos el problema de valor inicial en el que buscamos la solución aproximada de la ecuación diferencial

$$y' = f(x, y(x)), \quad (6.1)$$

sujeta a la condición inicial

$$y(x_0) = y_0. \quad (6.2)$$

Supongamos que $[x_0, b]$ es el intervalo en el que queremos hallar la solución del problema de valor inicial y que $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ satisface la condición de Lipschitz. Con el propósito de construir el conjunto finito de puntos (x_k, y_k) que aproximan la solución en $[x_0, b]$, $y(x_k) \approx y_k$, dividimos el intervalo $[x_0, b]$ en N subintervalos de mismo tamaño obteniendo $x_{i+1} = x_i + h$, $i = 0, \dots, N-1$, siendo $h = (b - x_0)/N$ el tamaño del paso. Nuestro punto de partida es la fórmula

$$y(x_0 + h) = y(x_0) + \int_{x_0}^{x_0+h} f(x, y(x)) dx, \quad (6.3)$$

donde $y(x)$ denota la solución de la ecuación diferencial. Utilizando la cuadratura gaussiana reescribimos (6.3):

$$y(x_{k+1}) = y(x_k) + \frac{x_{k+1} - x_k}{2} \sum_{i=1}^n \omega_i f(\eta_{i,k}, y(\eta_{i,k})), \quad (6.4)$$

donde

$$\begin{aligned} \eta_{i,k} &= \frac{1}{2}[(1 + \tau_i)x_{k+1} + (1 - \tau_i)x_k] \\ &= x_k + (1 + \tau_i) \frac{h}{2}. \end{aligned}$$

y $\omega_i, \tau_i, i = 1, \dots, n$ son los pesos y nodos respectivamente, de la cuadratura gaussiana empleada. Utilizando la Tabla 2.3 y sustituyendo la solución en los nodos por las sucesivas aproximaciones $y(x_i) = y_i$ describimos los diferentes métodos que se obtienen para la resolución del problema de valor inicial.

Caso I: *Cuadratura de Gauss-Tschebyshev.* Usando un solo nodo obtenemos la siguiente expresión:

$$\begin{aligned} \eta_{1,k} &= x_k + \frac{h}{2}, \\ y_{k+1} &= y_k + \frac{\pi h}{2} f(\eta_{1,k}, y_{\eta_{1,k}}), \end{aligned}$$

donde

$$y_{\eta_{1,k}} = \frac{1}{2}(y_k + y_{k+1}).$$

Para evitar el problema implícito que se presenta predecimos y_{k+1} utilizando el método de Euler: $y_{k+1} = y_k + hf(x_k, y_k)$. De esta manera, reescribiendo las expresiones anteriores obtenemos:

$$y_{k+1} = y_k + \frac{\pi h}{2} f\left[x_k + \frac{h}{2}, y_k + \frac{h}{2} f(x_k, y_k)\right]. \quad (6.5)$$

Utilizando el mismo procedimiento, con dos y tres nodos en la familia de cuadratura obtenemos, respectivamente, los métodos siguientes:

$$y_{k+1} = y_k + \frac{\pi h}{2} [f(\eta_{1,k}, y(\eta_{1,k})) + f(\eta_{2,k}, y(\eta_{2,k}))] \quad (6.6)$$

donde

$$\begin{aligned} \eta_{1,k} &= x_k + (2 - \sqrt{2}) \frac{h}{4}, & \eta_{2,k} &= x_k + (2 + \sqrt{2}) \frac{h}{4}, \\ y(\eta_{1,k}) &= y_k + \frac{2 - \sqrt{2}}{4} hf(x_k, y_k), & y(\eta_{2,k}) &= y_k + \frac{2 + \sqrt{2}}{4} hf(x_k, y_k), \end{aligned}$$

y

$$y_{k+1} = y_k + \frac{\pi h}{2} [f(\eta_{1,k}, y(\eta_{1,k})) + f(\eta_{2,k}, y(\eta_{2,k})) + f(\eta_{3,k}, y(\eta_{3,k}))], \quad (6.7)$$

donde

$$\begin{aligned} \eta_{1,k} &= x_k + \frac{1}{6}(3 - \sqrt{3})h, & \eta_{2,k} &= x_k + \frac{1}{2}h, & \eta_{3,k} &= x_k + \frac{1}{6}(3 + \sqrt{3})h, \\ y(\eta_{1,k}) &= y_k + \frac{3 - \sqrt{3}}{6}hf(x_k, y_k), & y(\eta_{2,k}) &= y_k + \frac{h}{2}f(x_k, y_k), & y(\eta_{3,k}) &= y_k + \frac{3 + \sqrt{3}}{4}hf(x_k, y_k). \end{aligned}$$

Caso II: *Cuadratura de Gauss-Legendre.* Usando un solo nodo obtenemos:

$$y_{k+1} = y_k + hf(\eta_{1,k}, y(\eta_{1,k})), \quad (6.8)$$

donde

$$\eta_{1,k} = x_k + \frac{h}{2}, \quad y(\eta_{1,k}) = y_k + \frac{h}{2}f(x_k, y_k),$$

que es el método Euler modificado o de punto medio. Podemos escribirlo también en la forma:

$$y_{k+1} = y_k + hk_2,$$

donde $k_2 = f(\eta_{1,k}, y_k + \frac{h}{2}k_1)$ y $k_1 = f(x_k, y_k)$. Este método pertenece a la familia de métodos de Runge-Kutta. Utilizando dos nodos se obtiene la expresión:

$$y_{k+1} = y_k + \frac{h}{2} [f(\eta_{1,k}, y(\eta_{1,k})) + f(\eta_{2,k}, y(\eta_{2,k}))], \quad (6.9)$$

donde

$$\begin{aligned} \eta_{1,k} &= x_k + \frac{1}{6}(3 - \sqrt{3})h, & \eta_{2,k} &= x_k + \frac{1}{6}(3 + \sqrt{3})h, \\ y(\eta_{1,k}) &= y_k + \frac{3 - \sqrt{3}}{6}hf(x_k, y_k), & y(\eta_{2,k}) &= y_k + \frac{3 + \sqrt{3}}{6}hf(x_k, y_k). \end{aligned}$$

Esta vez utilizamos tres nodos que generan la expresión:

$$y_{k+1} = y_k + \frac{h}{18} [5f(\eta_{1,k}, y(\eta_{1,k})) + 8f(\eta_{2,k}, y(\eta_{2,k})) + 5f(\eta_{3,k}, y(\eta_{3,k}))]. \quad (6.10)$$

donde

$$\begin{aligned} \eta_{1,k} &= x_k + \frac{1}{10}(5 - \sqrt{3})h, & \eta_{2,k} &= x_k + \frac{1}{2}h, & \eta_{3,k} &= x_k + \frac{1}{10}(5 + \sqrt{3})h, \\ y(\eta_{1,k}) &= y_k + \frac{5 - \sqrt{3}}{10}hf(x_k, y_k), & y(\eta_{2,k}) &= y_k + \frac{h}{2}f(x_k, y_k), & y(\eta_{3,k}) &= y_k + \frac{5 + \sqrt{3}}{10}hf(x_k, y_k). \end{aligned}$$

Caso III: *Cuadratura de Gauss-Lobatto.* Usando un solo nodo obtenemos el mismo método que en el caso de cuadratura de Gauss-Legendre con un solo nodo:

$$y_{k+1} = y_k + hf(\eta_{1,k}, y(\eta_{1,k})), \quad (6.11)$$

donde

$$\begin{aligned} \eta_{1,k} &= x_k + \frac{h}{2}, \\ y(\eta_{1,k}) &= y_k + \frac{h}{2}f(x_k, y_k). \end{aligned}$$

Ahora, utilizando dos nodos, obtenemos:

$$y_{k+1} = y_k + \frac{h}{2}[f(\eta_{1,k}, y(\eta_{1,k})) + f(\eta_{2,k}, y(\eta_{2,k}))], \quad (6.12)$$

que es el método de trapecios, donde

$$\begin{aligned} \eta_{1,k} &= x_k, & \eta_{2,k} &= x_k + h, \\ y(\eta_{1,k}) &= y_k, & y(\eta_{2,k}) &= y_k + hf(x_k, y_k). \end{aligned}$$

Como parte de la familia de métodos de Runge-Kutta podemos presentarlo en la forma:

$$\begin{aligned} k_1 &= f(x_k, y_k), \\ k_2 &= f(x_k + h, y_k + hk_1), \\ y_{k+1} &= y_k + \frac{h}{2}(k_1 + k_2). \end{aligned}$$

Si utilizamos tres nodos obtenemos la siguiente expresión:

$$y_{k+1} = y_k + \frac{h}{6}[f(\eta_{1,k}, y(\eta_{1,k})) + 4f(\eta_{2,k}, y(\eta_{2,k})) + f(\eta_{3,k}, y(\eta_{3,k}))], \quad (6.13)$$

que coincide con el método obtenido al aproximar la integral en (6.3) por la cuadratura de Simpson donde

$$\begin{aligned} \eta_{1,k} &= x_k, & \eta_{2,k} &= x_k + \frac{h}{2}, & \eta_{3,k} &= x_k + h, \\ y(\eta_{1,k}) &= y_k, & y(\eta_{2,k}) &= y_k + \frac{h}{2}f(x_k, y_k), & y(\eta_{3,k}) &= y_k + hf(x_k, y_k). \end{aligned}$$

Fácilmente se demuestra que este método también pertenece de la familia de métodos de Runge-Kutta y puede presentarse de la siguiente manera:

$$y_{k+1} = y_k + \frac{h}{6}(k_1 + 4k_2 + k_3),$$

donde

$$\begin{aligned} k_1 &= f(x_k, y_k), \\ k_2 &= f(x_k + \frac{h}{2}, y_k + \frac{h}{2}k_1), \\ k_3 &= f(x_k + h, y_k + hk_1). \end{aligned}$$

Caso IV: *Cuadratura de Gauss-Radau.* Usando un solo nodo obtenemos el método de Euler:

$$y_{k+1} = y_k + hf(\eta_{1,k}, y(\eta_{1,k})), \quad (6.14)$$

donde

$$\begin{aligned} \eta_{1,k} &= x_k, \\ y(\eta_{1,k}) &= y_k + hf(x_k, y_k). \end{aligned}$$

Utilizando dos nodos

$$y(x_{k+1}) = y(x_k) + \frac{h}{4}[f(\eta_{1,k}, y(\eta_{1,k})) + 3f(\eta_{2,k}, y(\eta_{2,k}))], \quad (6.15)$$

obtenemos el método de Heun, donde

$$\begin{aligned} \eta_{1,k} &= x_k, & \eta_{2,k} &= x_k + \frac{2}{3}h, \\ y(\eta_{1,k}) &= y_k + hf(x_k, y_k), & y(\eta_{2,k}) &= y_k + \frac{2h}{3}f(x_k, y_k), \end{aligned}$$

que se puede presentar como parte de la familia de Runge-Kutta en la forma siguiente:

$$y_{k+1} = y_k + \frac{h}{4}(k_1 + 3k_2),$$

donde

$$\begin{aligned} k_1 &= f(x_k, y_k), \\ k_2 &= f\left(x_k + \frac{2h}{3}, y_k + \frac{2h}{3}k_1\right). \end{aligned}$$

Si utilizamos tres nodos obtenemos el método:

$$y_{k+1} = y_k + \frac{h}{36}[4f(\eta_{1,k}, y(\eta_{1,k})) + (16 + \sqrt{6})f(\eta_{2,k}, y(\eta_{2,k})) + (16 - \sqrt{6})f(\eta_{3,k}, y(\eta_{3,k}))], \quad (6.16)$$

donde

$$\begin{aligned} \eta_{1,k} &= x_k, \quad \eta_{2,k} = x_k + \frac{1}{10}(6 - \sqrt{6})h, \quad \eta_{3,k} = x_k + \frac{1}{10}(6 + \sqrt{6})h, \\ y(\eta_{1,k}) &= y_k + hf(x_k, y_k), \quad y(\eta_{2,k}) = y_k + \frac{1}{10}(6 - \sqrt{6})hf(x_k, y_k), \quad y(\eta_{3,k}) = y_k + \frac{1}{10}(6 + \sqrt{6})hf(x_k, y_k). \end{aligned}$$

6.1.1. Orden de precisión y consistencia de los métodos obtenidos

Consideremos la familia de métodos Euler-Gauss de un sólo paso

$$\begin{aligned} y_{k+1} &= y_k + h \sum_{i=1}^m \frac{1}{2} \omega_i f(x_k + \beta_i h, y_k + \beta_i h f(x_k, y_k)) \\ &= y_k + h\Phi(x_k, y_k, h), \\ \beta_i &= \frac{1}{2}(1 + \tau_i), \end{aligned} \quad (6.17)$$

donde τ_i y ω_i son los nodos y pesos del polinomio ortogonal correspondiente a la cuadratura gaussiana. Según la Definición 2.3.2 el conjunto de familias de métodos Euler-Gauss es consistente. Desarrollando los dos lados de la expresión (6.17) en serie de Taylor y simplificando obtenemos:

$$\begin{aligned} y(x+h) &= y(x) + hf(x, y) + \frac{h^2}{2!} f'(x, y) + \frac{h^3}{3!} f''(x, y) + \frac{h^4}{4!} f^{(3)}(x, y) \\ &+ \frac{h^5}{5!} f^{(4)}(x, y) + \frac{h^6}{6!} f^{(5)}(x, y) + \dots \end{aligned} \quad (6.18)$$

$$\begin{aligned} y(x) + h\Phi(x, y, h) &= y(x) + \sum_{i=1}^m \frac{1}{2} \omega_i hf(x, y) + \sum_{i=1}^m \frac{1}{2} \omega_i \beta_i h^2 f'(x, y) \\ &+ \sum_{i=1}^m \frac{1}{2(2!)} \omega_i \beta_i^2 h^3 f''(x, y) + \sum_{i=1}^m \frac{1}{2(3!)} \omega_i \beta_i^3 h^4 f^{(3)}(x, y) \\ &+ \sum_{i=1}^m \frac{1}{2(4!)} \omega_i \beta_i^4 h^5 f^{(4)}(x, y) + \sum_{i=1}^m \frac{1}{2(5!)} \omega_i \beta_i^5 h^6 f^{(5)}(x, y) + \dots \end{aligned} \quad (6.19)$$

Restando las expresiones (6.18) y (6.19) tenemos:

$$\begin{aligned} &\left(1 - \sum_{i=1}^m \frac{1}{2} \omega_i\right) hf(x, y) + \left(1 - \sum_{i=1}^m \omega_i \beta_i\right) h^2 f'(x, y) + \left(\frac{1}{3!} - \sum_{i=1}^m \frac{1}{2(2!)} \omega_i \beta_i^2\right) h^3 f''(x, y) \\ &+ \left(\frac{1}{4!} - \sum_{i=1}^m \frac{1}{2(3!)} \omega_i \beta_i^3\right) h^4 f^{(3)}(x, y) + \left(\frac{1}{5!} - \sum_{i=1}^m \frac{1}{2(4!)} \omega_i \beta_i^4\right) h^5 f^{(4)}(x, y) \\ &+ \left(\frac{1}{6!} - \sum_{i=1}^m \frac{1}{2(5!)} \omega_i \beta_i^5\right) h^6 f^{(5)}(x, y) + \dots \end{aligned}$$

Vemos que tenemos orden de precisión uno (Definición 2.3.3) si

$$\sum_{i=1}^m \omega_i = 2.$$

Si, además,

$$\sum_{i=1}^m \omega_i \beta_i = 1 \tag{6.20}$$

obtenemos orden de precisión 2. Desarrollando la expresión (6.20) obtenemos

$$\sum_{i=1}^m \omega_i \tau_i = 0.$$

Si introducimos además la condición

$$\sum_{i=1}^m \frac{1}{2(2!)} \omega_i \beta_i^2 = \frac{1}{3!}, \tag{6.21}$$

obtenemos orden de precisión tres. Desarrollando la expresión (6.21) resulta

$$\sum_{i=1}^m \omega_i \tau_i^2 = \frac{2}{3}.$$

Notamos que podemos aumentar el orden de precisión con las condiciones

$$\sum_{i=1}^m \omega_i \beta_i^{j-1} = \frac{2}{j}, \quad \text{con } j = 1, 2, \dots \tag{6.22}$$

Esta condición, relacionándola con los pesos y nodos, queda:

$$\sum_{i=1}^m \omega_i (1 + \tau_i)^{j-1} = \frac{2}{j}.$$

Por lo tanto, podemos escribir las condiciones que definen el orden de precisión de la forma

$$\sum_{i=1}^m \omega_i (1 + \tau_i)^j = \frac{2^{j+1}}{j+1}, \quad \text{para tener orden } j+1 \text{ donde } j = 0, 1, \dots \tag{6.23}$$

O, equivalentemente,

$$\sum_{i=1}^m \omega_i \tau_i^j = \begin{cases} 0, & j = 2k+1, \quad k = 0, 1, \dots, \\ \frac{2}{j+1}, & j = 2k, \quad k = 1, 2, \dots \end{cases}$$

De esta manera se puede demostrar el siguiente resultado:

Teorema 6.1.1 *Sea $y(x)$ la solución del problema de valor inicial (6.1), (6.2) y sea $\{y_k\}_{k=0}^N$ la solución numérica generada por 6.17. Además, supongamos que*

- a) $\Phi(x, y, h)$ de (6.17) es una función continua en sus argumentos;
- b) $\Phi(x, y, h)$ es una función contractiva en y con constante de Lipschitz L ;
- c) el método (6.17) es consistente.

Entonces, el método numérico tiene orden de precisión q si cumple las condiciones presentadas en la expresión (6.23) para $j = 0, 1, \dots, q-1$.

Tabla 6.1: Precisión de los métodos de la familia Euler-Gauss

Método	$\sum_{i=1}^m \omega_i$	$\sum_{i=1}^m \omega_i \tau_i$	$\sum_{i=1}^m \omega_i \tau_i^2$	$\sum_{i=1}^m \omega_i \tau_i^3$	$\sum_{i=1}^m \omega_i \tau_i^4$	$\sum_{i=1}^m \omega_i \tau_i^5$	q
EuGLD1	2	0	0	0	0	0	2
EuGLD2	2	0	2/3	0	2/9	0	4
EuGLD3	2	0	2/15	0	2/125	0	2
EuGLT1	2	0	0	0	0	0	2
EuGLT2	2	0	2	0	2	0	2
EuGLT3	2	0	2/3	0	2/3	0	4
EuGR1	2	-2	2	-2	2	-2	1
EuGR2	2	0	2/3	-4/9	19/27	-40/81	3
EuGR3	2	0	2/3	0	2/5	-8/125	5

Volvemos a los métodos desarrollados anteriormente y usando la Tabla 2.3 obtenemos los resultados presentados en la Tabla 6.1.

En la tabla no hemos incluido la familia de Euler-Gauss-Tschebyshev por no cumplir con ninguna de las condiciones de precisión establecidas. Vemos que aumentando el número de nodos aumenta el orden de precisión de los métodos del conjunto de familias Euler-Gauss si se cumplen las condiciones establecidas en la expresión

(6.23). Por ejemplo, tomando el método Euler-Gauss-Lobatto con 4 nodos: $\tau_{1,2} = \pm \frac{\sqrt{5}}{5}$ y $\tau_{3,4} = \pm 1$ con pesos $\omega_{1,2} = \frac{5}{6}$ y $\omega_{3,4} = \frac{1}{6}$, respectivamente, obtenemos un método de orden de precisión 5 ya que:

$$\sum_{i=1}^m \omega_i = 2, \quad \sum_{i=1}^m \omega_i \tau_i = 0, \quad \sum_{i=1}^m \omega_i \tau_i^2 = \frac{2}{3}, \quad \sum_{i=1}^m \omega_i \tau_i^3 = 0, \quad \sum_{i=1}^m \omega_i \tau_i^4 = \frac{2}{5}, \quad \sum_{i=1}^m \omega_i \tau_i^5 = 0$$

$$\text{y } \sum_{i=1}^m \omega_i \tau_i^6 = 26/75 \neq \frac{2}{7}.$$

6.1.2. Cotas de error

En los métodos desarrollados en la Sección 6.1.1 la solución se aproxima en un conjunto finito de puntos. De (6.17) tenemos

$$y_{k+1} = y_k + h \sum_{i=1}^m \omega_i f(x_k + \beta_i h, y_k + \beta_i h f(x_k, y_k)) = y_k + h \Phi(x_k, y(x_k), h). \quad (6.24)$$

De (2.40) obtenemos

$$y(x_{k+1}) = y(x_k) + h \Phi(x_k, y(x_k), h) + \varepsilon_{k+1}. \quad (6.25)$$

Tomando en consideración las suposiciones del Teorema 6.1.1 enunciamos el siguiente resultado

Teorema 6.1.2 *Supongamos que existe una constante D , independiente de h y dependiente de y y $f(x, y)$ tal que*

$$\|\varepsilon_{k+1}\| \leq Dh^{p+1}, \quad (6.26)$$

para $x \in [a, b]$, $h \in [0, h_0]$. Entonces, tenemos la siguiente cota de error

$$\max_k \|y_k - y(x_k)\| \leq \frac{D}{L} \left(e^{Lh(b-a)} - 1 \right) h^p \quad (6.27)$$

para $0 \leq h \leq h_0$.

Demostración: Las expresiones (2.41), (6.24) y (6.25) implican

$$e_{k+1} = e_k + h [\Phi(x_k, y_k, h) - \Phi(x_k, y(x_k), h)] - \varepsilon_{k+1}.$$

La condición de que Φ es lipschitziana

$$\|\Phi(x_k, y_k, h) - \Phi(x_k, y(x_k), h)\| \leq L \|y_k - y(x_k)\|$$

y la expresión (6.26), a su vez implican

$$\|e_{k+1}\| \leq \|e_k\|(1 + hL) + Dh^{p+1}. \quad (6.28)$$

Al aplicar de forma recursiva (6.28), obtenemos

$$\begin{aligned} \|e_k\| &\leq (1 + hL)^k e_0 + Dh^{p+1}(1 + (1 + hL) + (1 + hL)^2 + \cdots + (1 + hL)^{k-1}) \\ &= (1 + hL)^k e_0 + \frac{((1 + hL)^k - 1)}{hL} Dh^{p+1}. \end{aligned} \quad (6.29)$$

Notamos que

$$1 + hL \leq e^{hL} = 1 + hL + \frac{(hL)^2}{2} + \cdots,$$

y ya que $e_0 = y_0 - y(0) = 0$, por lo tanto

$$\|e_{k+1}\| \leq \frac{e^{khL} - 1}{hL} Dh^{p+1} = \frac{D}{L} (e^{Lh(b-a)} - 1) h^p,$$

para $0 \leq k \leq N$. \square

Notamos que $\|\varepsilon_{k+1}\| = O(h^{p+1})$ y $\|e_k\| = O(h^p)$, es decir, el orden del error de discretización local del método es un orden superior a la cota de error (error de discretización global).

6.1.3. Estabilidad

Supongamos que en el k -ésimo paso, debido a los errores de redondeo, en realidad no se obtiene y_k sino z_k tal que $y_k - z_k \neq 0$ y que en el N -ésimo paso final tenemos $y_N - z_N \neq 0$. Para que los métodos desarrollados en la Sección 6.1 sean numéricamente estables debe existir una constante M independiente de h tal que $\|y_N - z_N\| \leq M \|y_k - z_k\|$ para toda $k \leq N$ donde $h = (b - a)/N$. Supongamos que se verifican las hipótesis necesaria para que los Teoremas 6.1.1 y 6.1.2 se satisfagan. Consideramos $j \geq k$, entonces

$$\begin{aligned} z_{j+1} &= z_j + h\Phi(x_j, z_j, h), \\ y_{j+1} &= y_j + h\Phi(x_j, y_j, h), \\ y_{j+1} - z_{j+1} &= y_j - z_j + h[\Phi(x_j, y_j, h) - \Phi(x_j, z_j, h)], \end{aligned} \quad (6.30)$$

para $k \leq j \leq N - 1$ y, por lo tanto,

$$\begin{aligned} \|y_N - z_N\| &\leq (1 + hL)^{N-k} \|y_k - z_k\| \\ &\leq (1 + hL)^N \|y_k - z_k\| \\ &\leq e^{hLN} \|y_k - z_k\| \\ &= e^{L(b-a)} \|y_k - z_k\| = M \|y_k - z_k\|. \end{aligned} \quad (6.31)$$

Definición 6.1.1 Diremos que un método numérico para resolver el problema de valor inicial definido por (6.1) y (6.2) es absolutamente estable para el paso $h > 0$ si al aplicarlo a la ecuación escalar $y' = \lambda y$, $\lambda < 0$, $y(0) = 1$, genera una sucesión de valores $\{y_k\}_{k \geq 0}$ con la propiedad que $y_k \rightarrow 0$ cuando $j \rightarrow \infty$. El conjunto de valores de λ y h para los que el método es absolutamente estable se llama conjunto de estabilidad absoluta.

El conjunto de valores de estabilidad absoluta de todos los métodos descritos en la Sección 6.1.1 es el intervalo $(-2, 0)$. El caso más simple corresponde al método de Euler-Gauss-Radau de un nodo (Euler). Aplicándolo con paso h al PVI obtenemos: $y_{k+1} = y_k + h\lambda y_k = y_k(1 + h\lambda)$, donde $y_k = y_0(1 + h\lambda)^{k+1}$. Asumiendo que λ es real, entonces $y_k \rightarrow 0$ cuando $j \rightarrow \infty$ si y sólo si $|1 + h\lambda| < 1$ que es lo mismo que $-2 < h\lambda < 0$. Por lo tanto la región de la estabilidad absoluta del método Euler-Gauss-Radau de un nodo es el intervalo $(-2, 0)$.

Aplicando el mismo procedimiento, obtenemos el siguiente resultado para, por ejemplo, el método de Euler-Gauss-Radau de 3 nodos:

$$\begin{aligned} y_{k+1} &= y_k + \frac{\lambda h}{36} \left(4y_k^* + (16 + \sqrt{6})y_k^{**} + (16 - \sqrt{6})y_k^{***} \right) \\ &= \frac{\lambda h}{36} (36y_k + 18\lambda h y_k) \\ &= y_k \left(1 + \lambda h + \frac{\lambda^2 h^2}{2} \right) \end{aligned}$$

donde $y_k^* = y_k$, $y_k^{**} = y_k + \frac{6 - \sqrt{6}}{10} \lambda h y_k$, $y_k^{***} = y_k + \frac{6 + \sqrt{6}}{10} \lambda h y_k$, y, además, aplicando el método iteradamente se obtiene $y_k = y_0 \left(1 + \lambda h + \frac{\lambda^2 h^2}{2} \right)^{k+1}$. Por lo tanto, $y_k \rightarrow 0$ cuando $j_k \rightarrow \infty$ si y sólo si $|1 + \lambda h + \frac{\lambda^2 h^2}{2}| < 1$ que es lo mismo que $-2 < h\lambda < 0$ para λ real. Para todos los restantes métodos, aplicando este procedimiento, se obtiene el mismo resultado: $|1 + \lambda h + \frac{\lambda^2 h^2}{2}| < 1$ de donde se concluye que les corresponde el intervalo $(-2, 0)$ de estabilidad absoluta.

6.2. Métodos de resolución de problemas de valor inicial no provinientes de fórmulas de cuadratura

En esta sección discutiremos los caminos 3 y 4 de la Figura 6.1. Miquel Grau-Sánchez et al. presentan en [40] y [41] una técnica para obtener métodos iterativos de resolución de ecuaciones no lineales basados en las técnicas de Adams y de Obreshkov de varios pasos para la solución numérica de PVI (camino 4 de la Figura 6.1). Su idea básicamente consiste en lo siguiente: sea $f : I \subseteq \mathbb{R} \rightarrow \mathbb{R}$ una función con una raíz simple, es decir, $f(\xi) = 0$ y $f'(\xi) \neq 0$ en un entorno I de ξ ; entonces, existe la función inversa $x = g(y)$ en $I_0 \subseteq f(I)$ donde $g(0) = \xi$ y derivando respecto a y , obtenemos:

$$\frac{dx}{dy} = g'(y) = \frac{1}{f'(x)}, \tag{6.32}$$

o, de manera equivalente,

$$D(x)(y) = x'(y) = F(x). \tag{6.33}$$

Además, si consideramos x_0 una primera aproximación de la raíz simple de f , y denotamos por $y_0 = f(x_0)$, definimos el problema de valor inicial

$$\begin{aligned} x'(y) &= F(x) \\ x(y_0) &= x_0. \end{aligned} \tag{6.34}$$

Notemos que la ecuación diferencial de (6.34) es autónoma. La integración numérica de este problema, desde y_0 a 0 con un solo paso, nos daría una aproximación de $x(0) = \xi$. Para ilustrar esta idea la aplicamos sobre el método de Euler en la resolución del P.V.I. (6.34). La aproximación de x en el siguiente instante sería

$$x_1 = x_0 + hF(x_0).$$

Esto se trasladaría al problema $f(x) = 0$ como:

$$x_1 = x_0 + h \frac{1}{f'(x_0)}$$

lo que nos exige que $h = -f(x_0)$ para que el método iterativo resultante sea Newton.

Como los métodos de Newton (para ecuaciones no lineales) y de Euler (para problema de valor inicial) se pueden considerar como métodos provenientes de cuadraturas, entonces, podemos "trasladar" el conjunto de familias de métodos iterativos definidos utilizando cuadratura gaussiana en la Sección 3.2 al problema de valor inicial obteniendo el conjunto de familias correspondientes definidas en la Sección 6.1 a los que habíamos llegado siguiendo el camino 2 de la Figura 6.1.

Sea $f : I \subseteq \mathbb{R} \rightarrow \mathbb{R}$, una función suficientemente diferenciable y $\xi \in I$ un cero real de la ecuación no lineal $f(x) = 0$. Entonces, la fórmula de iteración del conjunto de familias Newton-Gauss (*NeG*) dada en la Sección 3.2.1 es (3.9) que recordamos aquí:

$$x_{k+1} = x_k - \frac{2f(x_k)}{\sum_{i=1}^n \omega_i f'(\eta_{i,k}, y_{\eta_{i,k}})}$$

donde $\eta_{i,k} = \frac{(1 + \tau_i)z_k + (1 - \tau_i)x_k}{2}$ y $z_k = x_k - \frac{f(x_k)}{f'(x_k)}$. Este esquema iterativo es de la forma predictor-corrector, donde el predictor es el método clásico de Newton. Para transferir este esquema a problemas de valor inicial reescribimos (3.9)

$$\frac{x_{k+1} - x_k}{2} \sum_{i=1}^n \omega_i f'(\eta_{i,k}, y_{\eta_{i,k}}) + f(x_k) = 0,$$

teniendo en cuenta que $f(\xi) = 0$ y que, en este caso $x_{k+1} = \xi$, reordenamos la expresión anterior de la manera siguiente:

$$\frac{\xi - x_k}{2} \sum_{i=1}^n \omega_i f'(\eta_{i,k}, y_{\eta_{i,k}}) = f(\xi) - f(x_k).$$

El término de la izquierda de la igualdad anterior representa la aproximación de la integral $\int_{x_k}^{\xi} f'(t) dt$, es decir,

$$\int_{x_k}^{\xi} f'(t) dt \approx \frac{\xi - x_k}{2} \sum_{i=1}^n \omega_i f'(\eta_{i,k}, y_{\eta_{i,k}}).$$

Entonces, en este caso podemos definir el siguiente esquema para el problema de valor inicial:

$$y(x_{k+1}) = y(x_k) + \frac{h}{2} \sum_{i=1}^n \omega_i f'(\eta_{i,k}, y_{\eta_{i,k}})$$

donde $\eta_{i,k} = x_k + (1 + \tau_i) \frac{h}{2}$, es la expresión (6.4) obtenida en la Sección 6.1 del conjunto de familias de métodos iterativos para problema de valor inicial.

Observando la Figura 6.1, vemos que:

1. Partiendo de una cuadratura, tomando el camino 1, obtenemos métodos iterativos para ecuaciones y sistema de ecuaciones no lineales.
2. Partiendo de una cuadratura, tomando el camino 2, obtenemos métodos iterativos para el problema de valor inicial.
3. Una fórmula iterativa para ecuaciones no lineales provenientes de cuadratura puede transformarse en un método de resolución de problema de valor inicial y viceversa.
4. Partiendo de técnicas para la resolución numérica de problema de valor inicial, que no provienen de cuadratura, se obtienen métodos iterativos de solución de ecuaciones no lineales (camino 4).

Observando la similitud de las técnicas de resolución de ecuaciones no lineales y de problemas de valor inicial nos preguntamos: ¿se pueden transferir métodos iterativos de resolución de ecuaciones no lineales no provinientes de cuadratura a problemas de valor inicial? Para ello proponemos dos posibilidades: generar a partir de un método para ecuaciones no lineales una fórmula de cuadratura "ad hoc" y aplicarla posteriormente en el diseño de un método para resolver PVI (camino 2) ó usar la técnica descrita por Miquel Grau-Sánchez et al. pasando directamente por el camino 3. Las siguientes secciones estan dedicadas a dar respuesta a esta pregunta.

6.2.1. Método Euler-Traub

Consideramos el método iterativo de resolución de ecuaciones no lineales de Traub ó Potra-Ptak con orden de convergencia tres, que no proviene de ninguna fórmula de cuadratura:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\ x_{k+1} &= y_k - \frac{f(y_k)}{f'(x_k)} \\ &= x_k - \frac{f(x_k) + f(y_k)}{f'(x_k)}. \end{aligned}$$

Siguiendo la notación introducida en (6.32) a (6.34), transferimos este método a PVI, tal y como hemos visto con Euler y Newton. En este caso, la solución en el siguiente instante se estimaría por

$$x_1 = y_0 + h_2 F(x_0) = x_0 + (h_1 + h_2)F(x_0).$$

donde $F(x_0) = \frac{1}{f'(x_0)}$, $y_0 = x_0 + h_1 F(x_0)$ y tanto $h_1 = -f(x_0)$ como $h_2 = -f(y_0)$ son variables, es decir, dependen del paso en que estamos. Para analizar el orden de precisión del método obtenido para resolver el PVI, planteamos el operador asociado al método como:

$$\mathcal{L}(x(y); h_1) = x(y + h_1) - x(y) - (h_1 + h_2)D(x)(y), \tag{6.35}$$

donde, en general, $h_1 = -f(x)$ y $h_2 = -f(y)$. Desarrollamos $x(y + h_1)$ en torno a y :

$$x(y + h_1) = x(y) + h_1 D(x)(y) + \frac{h_1^2}{2} D^{(2)}(x)(y) + \frac{h_1^3}{6} D^{(3)}(x)(y) + O(h_1^4) \tag{6.36}$$

y lo sustituimos en (6.35)

$$\mathcal{L}(x(y); h_1) = -h_2 D(x)(y) + \frac{h_1^2}{2} D^{(2)}(x)(y) + \frac{h_1^3}{6} D^{(3)}(x)(y) + O(h_1^4). \tag{6.37}$$

Como $h_2 = -f(y)$ también depende de h_1 , lo obtenemos a partir del desarrollo de Taylor de $f(y)$ en el que $y - x = h_1 D(x)(y)$:

$$\begin{aligned} f(y) &= f(x) + f'(x)(y - x) + \frac{1}{2} f^{(2)}(x)(y - x)^2 + \frac{1}{6} f^{(3)}(x)(y - x)^3 + O(h_1^4) \\ &= f(x) + f'(x)h_1 D(x)(y) + \frac{1}{2} f^{(2)}(x)(h_1 D(x)(y))^2 + \frac{1}{6} f^{(3)}(x)(h_1 D(x)(y))^3 + O(h_1^4) \\ &= \frac{h_1^2}{2} f^{(2)}(x)(D(x)(y))^2 + \frac{h_1^3}{6} f^{(3)}(x)(D(x)(y))^3 + O(h_1^4) \end{aligned} \tag{6.38}$$

Por otra parte, derivando $f(x)$ sucesivas veces obtenemos:

$$\begin{aligned} f'(x) &= \frac{1}{F(x)} = \frac{1}{D(x)(y)}, \\ f^{(2)}(x) &= -F'(x)(f'(x))^2 = -\frac{F'(x)}{(D(x)(y))^2}, \\ f^{(3)}(x) &= -\frac{F''(x)D(x)(y) - 2[F'(x)]^2}{(D(x)(y))^3}, \end{aligned}$$

y sustituyendo estas expresiones en (6.38):

$$f(y) = -\frac{h_1^2}{2}F'(x) - \frac{h_1^3}{6}(F''(x)D(x)(y) - 2[F'(x)]^2) + O(h_1^4)$$

de donde obtenemos para h_2 la siguiente expresión:

$$h_2 = \frac{h_1^2}{2}F'(x) + \frac{h_1^3}{6}(F''(x)D(x)(y) - 2[F'(x)]^2) + O(h_1^4). \quad (6.39)$$

Sustituimos (6.39) en (6.37)

$$\mathcal{L}(x(y); h_1) = \frac{h_1^2}{2}[-F'(x)D(x)(y) + D^{(2)}(x)(y)] + \frac{h_1^3}{6}[-F''(x)(D(x)(y))^2 + 2F'(x)D(x)(y) + D^{(3)}(x)(y)] + O[h_1^4].$$

Para simplificar esta expresión, derivando obtenemos:

$$\begin{aligned} D(x)(y) &= \frac{dx}{dy} = g'(y) = F(x), \\ D^{(2)}(x)(y) &= \frac{d^2x}{dy^2} = -\frac{f''(x)}{[f'(x)]^2} \frac{dx}{dy} = F'(x)F(x), \\ D^{(3)}(x)(y) &= \frac{d^3x}{dy^3} = F''(x)[F(x)]^2 + [F'(x)]^2F(x). \end{aligned}$$

Finalmente obtenemos

$$\mathcal{L}(x(y); h_1) = \frac{h_1^3}{2}[(F'(x))^2]F(x) + O(h_1^4), \quad (6.40)$$

con lo que el método tiene orden de precisión dos, que es una unidad inferior al de Euler que corresponde al orden de convergencia 3 del método de partida.

En este método sabemos que $h_2 = \frac{h_1^2}{2}F'(x)$ y conocemos $F(x)$ del problema de valor inicial. El interés de este método es básicamente teórico, ya que para aplicarlo necesitamos conocer la función f cuando el problema de valor inicial sólo nos proporciona $\frac{1}{f'(x)}$.

A continuación cambiamos el enfoque del problema, construyendo una fórmula de cuadratura en torno al método de resolución de ecuaciones no lineales. Consideramos de nuevo el método de Traub con su fórmula iterativa

$$\begin{aligned} x_k^* &= x_k - \frac{f(x_k)}{f'(x_k)}, \\ x_{k+1} &= x_k^* - \frac{f(x_k^*)}{f'(x_k)}. \end{aligned}$$

Reescribimos el primer paso del método como sigue:

$$(x_k^* - x_k)f'(x_k) + f(x_k) = 0.$$

Asumiendo que $\xi \approx x_k^*$

$$f(\xi) - f(x_k) = (\xi - x_k)f'(x_k),$$

de donde deducimos

$$\int_{x_k}^{\xi} f'(t)dt \approx (\xi - x_k)f'(x_k).$$

Aplicamos esta fórmula de cuadratura en (6.3), obtenemos el primer paso del método que nos permitirá aproximar la solución del PVI:

$$y_k^* = y_k + h_1 f(x_k). \quad (6.41)$$

donde denotamos por $h_1 = \xi - x_k$, que dependerá del iterado x_k en que estamos. Continuando con el siguiente paso, de la misma manera obtenemos:

$$(x_{k+1} - x_k^*)f'(x_k) + f(x_k^*) = 0.$$

Dado que $\xi \approx x_{k+1}$,

$$f(\xi) - f(x_k^*) = (\xi - x_k^*)f'(x_k),$$

por lo que

$$\int_{x_k^*}^{\xi} f'(t)dt \approx (\xi - x_k^*)f'(x_k).$$

Aplicando de nuevo en (6.3) esta fórmula de cuadratura, obtenemos el segundo paso del método para PVI

$$y_{k+1} = y_k^* + h_2 f(x_k) = y_k + (h_1 + h_2)f(x_k). \quad (6.42)$$

donde denotamos con $h_2 = \xi - x_k^*$, que dependerá también del iterado en el que estamos y de h_1 . Para analizar el orden de precisión del método planteamos el operador asociado como:

$$\begin{aligned} \mathcal{M}(y(x), h_1) &= y(x + h_1) - [y(x) + h_1 f(x, y) + h_2 f(x_k, y_k)] \\ &= y(x) + h_1 f(x, y) + \frac{h_1^2}{2} f'(x_k, y_k) + \frac{h_1^3}{6} f''(x_k, y_k) \\ &\quad - y(x) - h_1 f(x, y) - h_2 f(x, y) + O(h_1^3). \end{aligned} \quad (6.43)$$

donde notamos que si tomamos $h_2 = \frac{h_1^2 f'(x_k + h_1, y_k + h_1 f(x_k, y_k))}{2 f(x, y)}$ el orden de precisión del método será dos. Finalmente el método tendrá la fórmula iterativa:

$$y_{k+1} = y_k + h_1 f(x_k, y_k) + \frac{h_1^2}{2} f'(x_k + h_1, y_k + h_1 f(x_k, y_k)).$$

6.2.2. Método de derivada congelada

Consideramos otro método iterativo de resolución de ecuaciones no lineales: el método de la derivada congelada, con orden de convergencia cuatro. Este esquema tampoco proviene de ninguna fórmula de cuadratura; su expresión iterativa es:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\ z_k &= y_k - \frac{f(y_k)}{f'(x_k)}, \\ x_{k+1} &= z_k - \frac{f(z_k)}{f'(x_k)} \\ &= x_k - \frac{f(x_k) + f(y_k) + f(z_k)}{f'(x_k)}. \end{aligned}$$

Siguiendo la notación introducida en (6.32) a (6.34), transferimos este método al PVI equivalente, tal y como hemos visto con el método de Traub, obteniendo que la solución en el siguiente instante se estimaría por

$$x_1 = z_0 + h_3 F(x_0) = x_0 + (h_1 + h_2 + h_3)F(x_0),$$

donde $F(x) = \frac{1}{f'(x)}$. Los $y_0^* = x_0 + h_1 F(x_0)$, $z_0 = y_0^* + h_2 F(x_0)$, $h_1 = -f(x_0)$, $h_2 = -f(y_0)$ y $h_3 = -f(z_0)$ son variables (dependen del paso).

Definimos el operador asociado al método como:

$$\mathcal{L}(x(y); h_1) = x(y + h_1) - x(y) - HD(x)(y) \quad (6.44)$$

donde $H = h_1 + h_2 + h_3$, $h_1 = -f(x)$, $h_2 = -f(y)$ y $h_3 = -f(z)$. Desarrollamos $x(y + h_1)$ en torno a y :

$$x(y + h_1) = x(y) + h_1 D(x)(y) + \frac{h_1^2}{2} D^{(2)}(x)(y) + \frac{h_1^3}{6} D^{(3)}(x)(y) + \frac{h_1^4}{24} D^{(4)}(x)(y) + O(h_1^5) \quad (6.45)$$

y lo sustituimos en (6.44)

$$\mathcal{L}(x(y); h_1) = -(h_2 + h_3)D(x)(y) + \frac{h_1^2}{2} D^{(2)}(x)(y) + \frac{h_1^3}{6} D^{(3)}(x)(y) + \frac{h_1^4}{24} D^{(4)}(x)(y) + O(h_1^5). \quad (6.46)$$

Obtenemos el desarrollo de Taylor de $f(y)$, en el que sustituyendo $y - x = h_1 D(x)(y)$:

$$\begin{aligned} f(y) &= f(x) + f'(x)(y - x) + \frac{1}{2} f^{(2)}(x)(y - x)^2 + \frac{1}{6} f^{(3)}(x)(y - x)^3 + \frac{1}{24} f^{(4)}(x)(y - x)^4 + O(h_1^5) \\ &= f(x) + f'(x)h_1 D(x)(y) + \frac{1}{2} f^{(2)}(x)(h_1 D(x)(y))^2 + \frac{1}{6} f^{(3)}(x)(h_1 D(x)(y))^3 \\ &\quad + \frac{1}{24} f^{(4)}(x)(h_1 D(x)(y))^4 + O(h_1^5) \\ &= \frac{h_1^2}{2} f^{(2)}(x)(D(x)(y))^2 + \frac{h_1^3}{6} f^{(3)}(x)(D(x)(y))^3 + \frac{h_1^4}{24} f^{(4)}(x)(D(x)(y))^4 + O(h_1^5). \end{aligned}$$

Del mismo modo que en la sección anterior, derivando obtenemos

$$\begin{aligned} f'(x) &= \frac{1}{F(x)} = \frac{1}{D(x)(y)}, \\ f^{(2)}(x) &= -\frac{F'(x)}{[D(x)(y)]^2}, \\ f^{(3)}(x) &= -\frac{F^{(2)}(x)}{(D(x)(y))^2} + 2\frac{F'(x)}{(D(x)(y))^3}, \\ f^{(4)}(x) &= -\frac{F^{(3)}(x)}{(D(x)(y))^2} + 6\frac{F^{(2)}(x)F'(x)}{(D(x)(y))^3} - 6\frac{(F'(x))^3}{(D(x)(y))^4}. \end{aligned}$$

y

$$\begin{aligned} D(x)(y) &= F(x), \\ D^{(2)}(x)(y) &= F'(x)F(x), \\ D^{(3)}(x)(y) &= F''(x)[F(x)]^2 + [F'(x)]^2 F(x), \\ D^{(4)}(x)(y) &= F'''(x)[F(x)]^3 + 4F''(x)F'(x)[F(x)]^2 + (F'(x))^3 F(x). \end{aligned}$$

Entonces,

$$\begin{aligned} h_2 &= -\frac{h_1^2}{2} f^{(2)}(x)(D(x)(y))^2 - \frac{h_1^3}{6} f^{(3)}(x)(D(x)(y))^3 - \frac{h_1^4}{24} f^{(4)}(x)(D(x)(y))^4 + O(h_1^5) \\ &= \frac{h_1^2}{2} F'(x) + \frac{h_1^3}{6} \left(F^{(2)}(x)D(x)(y) - 2(F'(x))^2 \right) \\ &\quad + \frac{h_1^4}{24} \left(F^{(3)}(x)(D(x)(y))^2 - 6F^{(2)}(x)F'(x)(D(x)(y)) + 6(F'(x))^3 \right) + O(h_1^5). \end{aligned} \quad (6.47)$$

Ahora debemos obtener el desarrollo de Taylor de $f(z)$, tomando $z - x = (h_1 + h_2)D(x)(y)$

$$\begin{aligned}
 f(z) &= f(x) + f'(x)(z - x) + \frac{1}{2}f^{(2)}(x)(z - x)^2 + \frac{1}{6}f^{(3)}(x)(z - x)^3 + \frac{1}{24}f^{(4)}(x)(z - x)^4 + O(h_1^5) \\
 &= \frac{h_1^2}{2}F'(x) + \frac{h_1^3}{6}\left(F^{(2)}(x)D(x)(y) - 2(F'(x))^2\right) \\
 &+ \frac{h_1^4}{24}\left(F^{(3)}(x)(D(x)(y))^2 - 6F^{(2)}(x)F'(x)(D(x)(y)) + 6(F'(x))^3\right) \\
 &- \frac{h_1^2}{2}F'(x) - \frac{h_1^3}{2}(F'(x))^2 - \frac{h_1^4}{24}\left(4F^{(2)}(x)F'(x)D(x)(y) - 5(F'(x))^3\right) \\
 &- \frac{h_1^3}{6}\left(F^{(2)}(x)D(x)(y) - 2(F'(x))^2\right) - \frac{h_1^4}{4}\left(F^{(2)}(x)F'(x)D(x)(y) - 2(F'(x))^3\right) \\
 &- \frac{h_1^4}{24}\left(F^{(3)}(x)(D(x)(y))^2 + 6F^{(2)}(x)F'(x)D(x)(y) - 6(F'(x))^3\right) + O(h_1^5) \\
 &= -\frac{h_1^3}{2}(F'(x))^2 - \frac{h_1^4}{24}\left(10F^{(2)}(x)F'(x)D(x)(y) - (F'(x))^3\right) + O(h_1^5).
 \end{aligned}$$

Entonces, h_3 queda:

$$h_3 = \frac{h_1^3}{2}(F'(x))^2 + \frac{h_1^4}{24}\left(10F^{(2)}(x)F'(x)D(x)(y) - (F'(x))^3\right) + O[h_1^5]. \quad (6.48)$$

Ahora, busquemos $h_2 + h_3$:

$$\begin{aligned}
 h_2 + h_3 &= \frac{h_1^2}{2}F'(x) + \frac{h_1^3}{6}\left(F^{(2)}(x)D(x)(y) - 2(F'(x))^2\right) \\
 &+ \frac{h_1^4}{24}\left(F^{(3)}(x)(D(x)(y))^2 - 6F^{(2)}(x)F'(x)(D(x)(y)) + 6(F'(x))^3\right) \\
 &+ \frac{h_1^3}{2}(F'(x))^2 + \frac{h_1^4}{24}\left(10F^{(2)}(x)F'(x)D(x)(y) - (F'(x))^3\right) + O(h_1^5) \\
 &= \frac{h_1^2}{2}F'(x) + \frac{h_1^3}{6}\left(F^{(2)}(x)D(x)(y) + (F'(x))^2\right) \\
 &+ \frac{h_1^4}{24}\left(F^{(3)}(x)(D(x)(y))^2 + 4F^{(2)}F'(x)(D(x)(y)) + 5(F'(x))^3\right) + O(h_1^5).
 \end{aligned} \quad (6.49)$$

Sustituimos (6.49) en (6.46)

$$\mathcal{L}((x(y); h_1) = \frac{h_1^4}{6}(F'(x))^3F(x) + O(h_1^5). \quad (6.50)$$

Deducimos de nuevo este mismo método trasladándolo de manera "clásica" a problema de valor inicial. Tomamos de nuevo el método de Traub con su fórmula iterativa

$$\begin{aligned}
 x_k^* &= x_k - \frac{f(x_k)}{f'(x_k)}, \\
 x_k^{**} &= x_k^* - \frac{f(x_k^*)}{f'(x_k^*)}, \\
 x_{k+1} &= x_k^{**} - \frac{f(x_k^{**})}{f'(x_k^{**})}.
 \end{aligned}$$

Siguiendo sus pasos hacemos la transformación:

$$\begin{aligned}
 (x_k^* - x_k)f'(x_k) + f(x_k) &= 0 \\
 f(\xi) - f(x_k) &= (\xi - x_k)f'(x_k) \\
 \int_{x_k}^{\xi} f'(t)dt &\approx (\xi - x_k)f'(x_k)
 \end{aligned}$$

de donde dándole la forma de problema de valor inicial, obtenemos para el primer paso

$$y_k^* = y_k + h_1 f'(x_k). \quad (6.51)$$

donde denotamos con $h_1 = \xi - x_k$ y que dependerá del paso en que estamos. Continuando con el siguiente paso de la misma manera obtenemos

$$\begin{aligned} (x_k^{**} - x_k^*)f'(x_k) + f(x_k^*) &= 0, \\ f(\xi) - f(x_k^*) &= (\xi - x_k^*)f'(x_k), \\ \int_{x_k^*}^{\xi} f'(t)dt &\approx (\xi - x_k^*)f'(x_k), \end{aligned}$$

de donde resulta el segundo paso

$$y_k^{**} = y_k^* + h_2 f'(x_k) = y_k + (h_1 + h_2)f'(x_k), \quad (6.52)$$

donde denotamos con $h_2 = \xi - x_k^*$ y que dependerá también del paso en que estamos. En el siguiente paso obtenemos

$$\begin{aligned} (x_{k+1} - x_k^{**})f'(x_k) + f(x_k^{**}) &= 0, \\ f(\xi) - f(x_k^{**}) &= (\xi - x_k^{**})f'(x_k), \\ \int_{x_k^{**}}^{\xi} f'(t)dt &\approx (\xi - x_k^{**})f'(x_k), \end{aligned}$$

de donde finalmente tenemos la expresión para el problema de valor inicial

$$y_{k+1} = y_k^{**} + h_3 f'(x_k) = y_k + (h_1 + h_2 + h_3)f'(x_k), \quad (6.53)$$

donde denotamos con $h_3 = \xi - x_k^{**}$ y que dependerá también del paso en que estamos.

Analizamos el orden de precisión del método. Para tales fines planteamos el operador asociado al método como:

$$\begin{aligned} \mathcal{M}(y(x), h_1) &= y(x + h_1) - [y(x) + h_1 f(x, y) + h_2 f(x, y) + h_3 f(x, y)] \\ &= y(x) + h_1 f(x, y) + \frac{h_1^2}{2} f'(x_k, y_k) + \frac{h_1^3}{6} f''(x_k, y_k) + \frac{h_1^4}{24} f^{(3)}(x_k, y_k) \\ &\quad - y(x) - h_1 f(x, y) - h_2 f(x, y) - h_3 f(x, y). \end{aligned}$$

Comparando estos resultados con los obtenidos en el caso del método de Euler-Traub vemos que si tomamos h_2 de la misma forma, es decir,

$$h_2 = \frac{h_1^2}{2} \frac{f' \left(x_k + \frac{h_1}{2}, y_k + \frac{h_1}{2} f(x_k, y_k) \right)}{f(x, y)}$$

obtenemos un método de orden de precisión dos. Para tener orden de precisión tres debemos tomar $h_3 = \frac{h_1^3}{6} \frac{f''(x_k, y_k)}{f(x, y)}$. Para no calcular la f'' la aproximamos según la relación:

$$f''(x_k, y_k) \approx \frac{f'(x_k + h_1, y_k + h_1 f(x_k, y_k)) - f'(x_k, y_k)}{f(x_k + h_1, y_k + h_1 f(x_k, y_k)) - f(x_k, y_k)}.$$

Finalmente la fórmula del método queda:

$$y_{k+1} = y_k + h_1 f(x_k, y_k) + \frac{h_1^2}{2} f' \left(x_k + \frac{h_1}{2}, y_k + \frac{h_1}{2} f(x_k, y_k) \right) + \frac{h_1^3}{6} f''(x_k + h_1, y_k + h_1 f(x_k, y_k)).$$

LÍNEAS FUTURAS Y CONCLUSIONES

La aportación principal de esta memoria es la generalización de la cuadratura de Gauss para el diseño de métodos iterativos tanto para la resolución de ecuaciones y sistemas de ecuaciones no lineales como para la resolución de PVI. Utilizando la pseudocomposición podemos predeterminar el orden de convergencia de un método iterativo para ecuaciones o sistemas de ecuaciones no lineales partiendo de un método iterativo como predictor con orden de convergencia conocido p , aplicando la cuadratura de Gauss como corrector y obtener un método iterativo con orden de convergencia $\min\{p+q, 3q\}$, donde q es el orden del penúltimo paso del predictor. Queda pendiente por determinar si es posible generalizar la pseudocomposición para cualquier corrector que no provenga de la cuadratura de Gauss, y el orden de convergencia del método resultante.

Es conocido que, para aproximar las soluciones de determinadas ecuaciones, es interesante aplicar procesos iterativos que tengan alto orden de convergencia. En primer lugar, siguiendo las ideas desarrolladas en los Capítulos 3 y 4 queda abierta la línea de investigación enfocada al diseño de métodos de órdenes más altos óptimos. La mayor parte de los métodos desarrollados en estos dos capítulos, sean óptimos o no, no pueden ser utilizados para sistemas de ecuaciones no lineales.

Por otra parte, si la derivada de la función $f(x)$ se anula en algún iterado, o está proxima a cero $f'(x_k) \approx 0$, la mayoría de los métodos iterativos tienen problemas de estabilidad, o incluso de convergencia. Para abordar este problema, Wu et al. [84]-[87] han diseñado distintas variantes del método de Newton. Una de estas variantes tiene la forma

$$x_{k+1} = x_k - \frac{f(x_k)}{v_k f(x_k) + f'(x_k)}, \quad k = 0, 1, \dots \quad (7.1)$$

Si escogemos $v_k = 0$ el método se reduce a método clásico de Newton. Si $v_k \neq 0$, el denominador de la fórmula (7.1) nunca será cero aunque $f'(x_k) = 0$. Además, Kou et al. en [58] extiende este método al contexto de sistemas no lineales, permitiendo que la matriz Jacobiana sea singular en algunos puntos. En ambos casos los métodos obtenidos tienen orden de convergencia dos. Otra técnica utilizada para resolver este tipo de problemas son los métodos libres de derivada que presentan variantes del método de Steffensen que tiene orden de convergencia dos y dos evaluaciones de la función. Siguiendo estas ideas, como otra vía de investigación se pretende diseñar métodos óptimos para la resolución de ecuaciones y sistemas de ecuaciones con las particularidades mencionadas. En particular, utilizando la composición de métodos y aproximando las derivadas que aparezcan por diferentes técnicas, pensamos que se podrían conseguir métodos óptimos libres de derivadas del orden que se desee.

En esta memoria, todas las demostraciones de la convergencia de los métodos se han hecho desde un punto de vista local. Pensamos que un estudio de la convergencia semilocal de algunos de estos métodos nos permitiría comprender mejor el comportamiento de los mismos, en términos de cotas del error y radios de convergencia.

Motivados por los resultados numéricos obtenidos en el Capítulo 5, nos planteamos el diseño de métodos iterativos para la resolución de sistemas de ecuaciones, más eficientes (menor cantidad de Jacobianas y menos sistemas lineales por resolver). Queda también por estudiar el efecto del número de sumas y restas sobre el tiempo de ejecución de los algoritmos y su influencia sobre el índice de eficiencia computacional.

En el Capítulo 6 se han desarrollado métodos para la resolución de problemas de valor inicial basados en métodos iterativos de resolución de ecuaciones no lineales que no provienen de cuadraturas. Utilizando esta

experiencia pretendemos continuar este camino hacia la mejora de los métodos obtenidos y el diseño de otros. La mejora podría obtenerse de la eliminación de la derivada de la función $f(x, y(x))$ en la determinación de los pasos $h_i, i = 1, 2, \dots$

ANEXO 1: M-file con las ecuaciones de las pruebas numéricas

```
function [fx,dfx]=ecuaciones(x)
global numero
switch numero
    case 1
        fx=sin(x)-x^2+1;
        dfx=cos(x)-2*x;
        %fx=subs(fx);
        %dfx=subs(dfx);
    case 2
        fx=x^2-exp(x)-3*x+2;
        dfx=2*x-exp(x)-3;
    case 3
        fx=cos(x)-x;
        dfx=-sin(x)-1;
    case 4
        fx=(x-1)^3-1;
        dfx=3*(x-1)^2;
    case 5
        fx=x^3-10;
        dfx=3*x^2;
    case 6
        fx=cos(x)-x*exp(x)+x^2;
        dfx=-sin(x)-exp(x)-x*exp(x)+2*x;
    case 7
        fx=exp(x)-1.5-atan(x);
        dfx=exp(x)-1/(1+x^2);
    case 8
        fx=x^3+4*x^2-10;
        dfx=3*x^2+8*x;
    case 9
        fx=8*x-cos(x)-2*x^2;
        dfx=8+sin(x)-4*x;
    case 10
        fx=atan(x);
        dfx=1/(1+x^2);
    case 11
        fx=exp(x)-4*x^2;
        dfx=exp(x)-8*x;
    case 12
```

```
    fx=(sin(x)-x/2)^2;
    dfx=2*(sin(x)-x/2)*(cos(x)-1/2);
case 13
    fx=x*exp(x^2)-sin(x^2)+3*cos(x)+5;
    dfx=exp(x^2)+2*x^2*exp(x^2)-2*x*cos(x^2)-3*sin(x);
case 14
    fx=(x^2+2*x+5)^(1/2)-2*sin(x)-x^2+3;
    dfx=(2*x+2)/(2*(x^2+2*x+5)^(1/2))-2*cos(x)-2*x;
case 15
    fx=x^4+sin(vpa(pi)/(x^2))-5;
    dfx=4*x^3-((2*vpa(pi))/(x^3))*cos(vpa(pi)/(x^2));
case 16
    fx=10*x*exp(-x^2)-1;
    dfx=10*exp(-x^2)-20*x^2*exp(-x^2);
case 17
    fx=exp(-x)+cos(x);
    dfx=-exp(-x)-sin(x);
case 18
    fx=sqrt(x^4+8)*sin(vpa(pi)/(x^2+2))+x^3/(x^4+1)+(-sqrt(6)+8/17);
    dfx=2*x^3/sqrt(x^4+8)*sin(vpa(pi)/(x^2+2))
        -2*vpa(pi)*x*sqrt(x^4+8)*cos(vpa(pi)/(x^2+2))/((x^2+2)^2)+3*x^2/(x^4+1)
        -4*x^6/((x^4+1)^2);
end
```

ANEXO 2: M-file de los métodos desarrollados con cuadratura de Gauss para ecuaciones no lineales

1. Familia NeG

a). NeG-LD1

```
function [x0,iter,p,incr,incr2] = NeGLD1(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;

[fx,dfx] = feval(f,x0);
while incr2>tol && incr>tol && iter<maxiter
    x = x0 - fx/dfx;
    y = (x0 + x)/2;
    [~,dfy] = feval(f,y);
    x = x0 - fx/dfy;
    incr = abs(double(x - x0));
    Inc = [Inc incr];
    x0 = x;
    [fx, dfx] = feval(f,x0);
    incr2 = abs(double(fx));
    iter = iter + 1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));
```

b). NeG-LD2

```
function [x0,iter,p,incr,incr2] = NeGLD2(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;

q=sqrt(sym(3));
a=vpa((3+q)/6);
b=vpa((3-q)/6);

[fx,dfx] = feval(f,x0);

while incr2>tol && incr>tol && iter<maxiter
    x = x0 - fx/dfx;
    y = a*x0 + b*x;
    z = b*x0 + a*x;
    [~,dfy] = feval(f,y);
    [~,dfz] = feval(f,z);
    delta=2*fx/(dfy+dfz);
    xn = x0 - delta;
    incr = abs(double(delta));
    Inc = [Inc incr];
    [fx, dfx] = feval(f,xn);
    x0 = xn;
    incr2 = abs(double(fx));
    iter = iter + 1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));
```

c). NeG-LT2

```

function [x0,iter,p,incr,incr2] = NeGLT2(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;

[fx,dfx] = feval(f,x0);

while incr2>tol && incr>tol && iter<maxiter
    y = x0 - fx/dfx;
    [~, dfy] = feval(f,y);
    x = x0 - 2*fx/(dfy+dfx);
    incr = abs(double(x - x0));
    Inc = [Inc incr];
    [fx, dfx] = feval(f,x);
    x0 = x;
    incr2 = abs(double(fx));
    iter = iter + 1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

d). NeG-LR2

```

function [x0,iter,p,incr,incr2] = NeGR2(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;

[fx,dfx] = feval(f,x0);

while incr2>tol && incr>tol && iter<maxiter
    x = x0 - fx/dfx; size(x0),size(d)
    y = (x0 + 2*x)/3;
    [~, dfy] = feval(f,y);
    xn = x0 -4*fx/(3*dfy+dfx);
    incr = abs(double(x - x0));
    Inc = [Inc incr];
    [fx, dfx] = feval(f,xn);
    x0 = xn;
    incr2 = abs(double(fx));
    iter = iter + 1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

2. Família TrG

a). TrG-LD1

```

function [x0,iter,p,incr,incr2] = TrGLD1(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;

[fx,dfx] = feval(f,x0);
while incr2>tol && incr>tol && iter<maxiter

```

```

    y = x0 - fx/dfx;
    fy = feval(f,y);
    z = x0 - (fx + fy)/dfx;
    u = (y + z)/2;
    [~,dfu] = feval(f,u);
    xn = y - fy/dfu;
    incr = abs(double(xn - x0));
    Inc = [Inc incr];
    x0 = xn;
    [fx, dfx] = feval(f,x0);
    incr2 = abs(double(fx));
    iter = iter + 1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

b). TrG-LD2

```

function [x0,iter,p,incr,incr2] = TrGLD2(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;

q=sqrt(sym(3));
a=vpa((3+q)/6);
b=vpa((3-q)/6);

[fx,dfx] = feval(f,x0);
while incr2>tol && incr>tol && iter<maxiter
    y = x0 - fx/dfx;
    fy = feval(f,y);
    z = x0 - (fx + fy)/dfx;
    u1 = a*y + b*z;
    u2 = b*y + a*z;
    [~,dfu1] = feval(f,u1);
    [~,dfu2] = feval(f,u2);
    delta = 2*fy/(dfu1+dfu2);
    xn = y - delta;
    incr = abs(double(xn - x0));
    Inc = [Inc incr];
    x0 = xn;
    [fx, dfx] = feval(f,xn);
    incr2 = abs(double(fx));
    iter = iter + 1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

a). TrG-LT2

```

function [x0,iter,p,incr,incr2] = TrGLT2(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;

[fx,dfx] = feval(f,x0);

while incr2>tol && incr>tol && iter<maxiter

```

```

y = x0 - fx/dfx;
[fy,dfy] = feval(f,y);
z = x0 - (fx + fy)/dfx;
[~,dfz] = feval(f,z);
delta = 2*fy/(dfz+dfy);
xn = y - delta;
incr = abs(double(xn - x0));
Inc = [Inc incr];
x0 = xn;
[fx, dfx] = feval(f,xn);
incr2 = abs(double(fx));
iter = iter + 1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

d). TrG-LR2

```

function [x0,iter,p,incr,incr2] = TrGR2(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;

[fx,dfx] = feval(f,x0);

while incr2>tol && incr>tol && iter<maxiter
y = x0 - fx/dfx;
[fy,dfy] = feval(f,y);
z = x0 - (fx + fy)/dfx;
u = (y + 2*z)/3;
[~,dfu] = feval(f,u);
delta = 4*fy/(3*dfu+dfy);
xn = y - delta;
incr = abs(double(xn - x0));
Inc = [Inc incr];
x0 = xn;
[fx, dfx] = feval(f,xn);
incr2 = abs(double(fx));
iter = iter + 1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

3. Familia OsG

a). OsG-LD1

```

function [x0,iter,p,incr,incr2] = OsGLD1(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;

[fx,dfx] = feval(f,x0);
while incr2>tol && incr>tol && iter<maxiter
a = fx/dfx;
y = x0 - a;
fy = feval(f,y);
b = fy/(fx-2*fy);

```

```

    c = b*a;
    z = y - c;
    u = (y + z)/2;
    [~,dfu] = feval(f,u);
    x = y - fy/dfu;
    incr = abs(double(x - x0));
    Inc = [Inc incr];
    x0 = x;
    [fx, dfx] = feval(f,x0);
    incr2 = abs(double(fx));
    iter = iter + 1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

b). OsG-LD2

```

function [x0,iter,p,incr,incr2] = OsGLD2(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;

q=sqrt(sym(3));
a=vpa((3+q)/6);
b=vpa((3-q)/6);

[fx,dfx] = feval(f,x0);
while incr2>tol && incr>tol && iter<maxiter
    m = fx/dfx;
    y = x0 - m;
    fy = feval(f,y);
    n = fy/(fx-2*fy);
    c = n*m;
    z = y - c;
    u1 = a*y + b*z;
    u2 = b*y + a*z;
    [~,dfu1] = feval(f,u1);
    [~,dfu2] = feval(f,u2);
    delta = 2*fy/(dfu1+dfu2);
    xn = y - delta;
    incr = abs(double(xn - x0));
    Inc = [Inc incr];
    x0 = xn;
    [fx, dfx] = feval(f,xn);
    incr2 = abs(double(fx));
    iter = iter + 1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2))

```

c). OsG-LT2

```

function [x0,iter,p,incr,incr2] = OsGLT2(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;

[fx,dfx] = feval(f,x0);

```

```

while incr2>tol && incr>tol && iter<maxiter
    a = fx/dfx;
    y = x0 - a;
    [fy,dfy] = feval(f,y);
    b = fy/(fx-2*fy);
    c = b*a;
    z = y - c;
    [~,dfz] = feval(f,z);
    delta = 2*fy/(dfz+dfy);
    xn = y - delta;
    incr = abs(double(xn - x0));
    Inc = [Inc incr];
    x0 = xn;
    [fx, dfx] = feval(f,xn);
    incr2 = abs(double(fx));
    iter = iter + 1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

d). OsG-LR2

```

function [x0,iter,p,incr,incr2] = OsGR2(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;

[fx,dfx] = feval(f,x0);

while incr2>tol && incr>tol && iter<maxiter
    a = fx/dfx;
    y = x0 - a;
    [fy,dfy] = feval(f,y);
    b = fy/(fx-2*fy);
    c = b*a;
    z = y - c;
    u = (y + 2*z)/3;
    [~,dfu] = feval(f,u);
    delta = 4*fy/(3*dfu+dfy);
    xn = y - delta;
    incr = abs(double(xn - x0));
    Inc = [Inc incr];
    x0 = xn;
    [fx, dfx] = feval(f,xn);
    incr2 = abs(double(fx));
    iter = iter + 1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

4. Familia KuG

a). KuG-LD1

```

function [x0,iter,p,incr,incr2] = KuGLD1(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];

```

```

iter = 0;

[fx,dfx] = feval(f,x0);
while incr2>tol && incr>tol && iter<maxiter
    a = fx/dfx;
    y = x0 - a;
    fy = feval(f,y);
    b = (fx + 2*fy)/fx;
    c = fy/dfx;
    d = b*c;
    z = y - d;
    u = (y + z)/2;
    [~,dfu] = feval(f,u);
    x = y - fy/dfu;
    incr = abs(double(x - x0));
    Inc = [Inc incr];
    x0 = x;
    [fx, dfx] = feval(f,x0);
    incr2 = abs(double(fx));
    iter = iter + 1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

b). KuG-LD2

```

function [x0,iter,p,incr,incr2] = KuGLD2(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;

q=sqrt(sym(3));
a=vpa((3+q)/6);
b=vpa((3-q)/6);

[fx,dfx] = feval(f,x0);
while incr2>tol && incr>tol && iter<maxiter
    m = fx/dfx;
    y = x0 - m;
    fy = feval(f,y);
    n = (fx + 2*fy)/fx;
    c = fy/dfx;
    d = n*c;
    z = y - d;
    u1 = a*y + b*z;
    u2 = b*y + a*z;
    [~,dfu1] = feval(f,u1);
    [~,dfu2] = feval(f,u2);
    delta = 2*fy/(dfu1+dfu2);
    xn = y - delta;
    incr = abs(double(xn - x0));
    Inc = [Inc incr];
    x0 = xn;
    [fx, dfx] = feval(f,xn);
    incr2 = abs(double(fx));
    iter = iter + 1;
end

```

```
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));
```

c). KuG-LT2

```
function [x0,iter,p,incr,incr2] = KuGLT2(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;

[fx,dfx] = feval(f,x0);

while incr2>tol && incr>tol && iter<maxiter
    a = fx/dfx;
    y = x0 - a;
    [fy,dfy] = feval(f,y);
    b = (fx + 2*fy)/fx;
    c = fy/dfx;
    d = b*c;
    z = y - d;
    [~,dfz] = feval(f,z);
    delta = 2*fy/(dfz+dfy);
    xn = y - delta;
    incr = abs(double(xn - x0));
    Inc = [Inc incr];
    x0 = xn;
    [fx, dfx] = feval(f,xn);
    incr2 = abs(double(fx));
    iter = iter + 1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));
```

d). KuG-LR2

```
function [x0,iter,p,incr,incr2] = KuGR2(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;

[fx,dfx] = feval(f,x0);

while incr2>tol && incr>tol && iter<maxiter
    a = fx/dfx;
    y = x0 - a;
    [fy,dfy] = feval(f,y);
    b = (fx + 2*fy)/fx;
    c = fy/dfx;
    d = b*c;
    z = y - d;
    u = (y + 2*z)/3;
    [~,dfu] = feval(f,u);
    delta = 4*fy/(3*dfu+dfy);
    xn = y - delta;
    incr = abs(double(xn - x0));
    Inc = [Inc incr];
    x0 = xn;
    [fx, dfx] = feval(f,xn);
```

```

    incr2 = abs(double(fx));
    iter = iter + 1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

a). DCoG-LD1

```

function [x0,iter,p,incr,incr2] = DCoGLD1(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;

[fx,dfx] = feval(f,x0);
while incr2>tol && incr>tol && iter<maxiter
    a = fx/dfx;
    y = x0 - a;
    fy = feval(f,y);
    b = (fx + fy)/dfx;
    z = x0 - b;
    fz = feval(f,z);
    u = x0 - (fx + fy + fz)/dfx;
    v = (u + z)/2;
    [~,dfv] = feval(f,v);
    x = z - fz/dfv;
    incr = abs(double(x - x0));
    Inc = [Inc incr];
    x0 = x;
    [fx, dfx] = feval(f,x0);
    incr2 = abs(double(fx));
    iter = iter + 1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

b). DCoG-LD2

```

function [x0,iter,p,incr,incr2] = DCoGLD2(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;

q=sqrt(sym(3));
a=vpa((3+q)/6);
b=vpa((3-q)/6);

[fx,dfx] = feval(f,x0);
while incr2>tol && incr>tol && iter<maxiter
    m = fx/dfx;
    y = x0 - m;
    fy = feval(f,y);
    n = (fx + fy)/dfx;
    z = x0 - n;
    fz = feval(f,z);
    u = x0 - (fx + fy + fz)/dfx;
    u1 = a*z + b*u;
    u2 = b*z + a*u;
    [~,dfu1] = feval(f,u1);

```

```

    [~,dfu2] = feval(f,u2);
    delta = 2*fz/(dfu1+dfu2);
    xn = z - delta;
    incr = abs(double(xn - x0));
    Inc = [Inc incr];
    x0 = xn;
    [fx, dfx] = feval(f,xn);
    incr2 = abs(double(fx));
    iter = iter + 1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

c). DCoG-LT2

```

function [x0,iter,p,incr,incr2] = DCoGLT2(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;

[fx,dfx] = feval(f,x0);

while incr2>tol && incr>tol && iter<maxiter
    a = fx/dfx;
    y = x0 - a;
    fy = feval(f,y);
    b = (fx + fy)/dfx;
    z = x0 - b;
    fz = feval(f,z);
    u = x0 - (fx + fy + fz)/dfx;
    [~,dfz] = feval(f,z);
    [~,dfu] = feval(f,u);
    delta = 2*fz/(dfz+dfu);
    xn = z - delta;
    incr = abs(double(xn - x0));
    Inc = [Inc incr];
    x0 = xn;
    [fx, dfx] = feval(f,xn);
    incr2 = abs(double(fx));
    iter = iter + 1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

d). DCoG-LR2

```

function [x0,iter,p,incr,incr2] = DCoGR2(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;

[fx,dfx] = feval(f,x0);

while incr2>tol && incr>tol && iter<maxiter
    a = fx/dfx;
    y = x0 - a;
    fy = feval(f,y);
    b = (fx + fy)/dfx;

```

```
z = x0 - b;
fz = feval(f,z);
u = x0 - (fx + fy + fz)/dfx;
v = (z + 2*u)/3;
[~,dfz] = feval(f,z);
[~,dfv] = feval(f,v);
delta = 4*fz/(3*dfv+dfz);
xn = z - delta;
incr = abs(double(xn - x0));
Inc = [Inc incr];
x0 = xn;
[fx, dfx] = feval(f,xn);
incr2 = abs(double(fx));
iter = iter + 1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));
```

ANEXO 3: M-file de los métodos desarrollados para ecuaciones no lineales

1. Método M7

```
function [x0,iter,p,incr,incr2] = M7(x0,f,maxiter,tol)
iter = 0; incr = tol+1; incr2=incr;
Inc=[];

[fx,dfx] = feval(f,x0);
while incr2>tol && incr>tol && iter < maxiter
    delta = fx/dfx;
    iter = iter+1;
    y = x0 - delta;
    [fy] = feval(f,y);
    H1 = fy/(fx - 2*fy);
    z = x0 - (1 + H1)*delta;
    [fz] = feval(f,z);
    H2 = fz/(fy - fz);
    H3 = fz/dfx;
    H = (1 + H1 + H2/2);
    xn = z - H^2*H3;
    incr = abs(double(xn - x0));
    Inc = [Inc incr];

    [fx,dfx] = feval(f,xn);
    incr2 = abs(double(fx));
    x0 = xn;
end

p = log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));
```

2. Método M8A

```
function [x0,iter,p,incr,incr2] = M8A(x0,f,maxiter,tol)
iter = 0; incr = tol+1; incr2=incr;
Inc=[];

[fx,dfx] = feval(f,x0);
while incr2>tol && incr>tol && iter < maxiter
    delta = - fx/dfx ;
    iter = iter+1;
    y = x0 + 2*delta/3;
    z = y - delta/6;
    [~,dfy] = feval(f,y);
    H1 = fx/(dfx - 3*dfy);
    u = z +H1;
    [fu] = feval(f,u);
    H2 = fu/(dfx - 3*dfy);
    v = u + 2*H2;
    [fv] = feval(f,v);
    H3 =(1/2)*(5*dfx - 3*dfy)/dfx;
    xn = v - H3*fv/dfx;
    incr = abs(double(xn - x0));
    Inc = [Inc incr];

    [fx,dfx] = feval(f,xn);
```

```

    incr2 = abs(double(fx));
    x0 = xn;
end

p = log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

3. Método M8B

```

function [x0,iter,p,incr,incr2] = M8B(x0,f,maxiter,tol)
iter = 0; incr = tol+1; incr2=incr;
Inc=[];

[fx,dfx] = feval(f,x0);
while incr2>tol && incr>tol && iter < maxiter
    a = fx/dfx;
    iter = iter+1;
    y = x0 - a;
    [fy] = feval(f,y);
    H1 = fy/(fx - 2*fy);
    z = x0 - (1 + H1)*fx/dfx;
    [fz] = feval(f,z);
    H2 = fz/(fy - 2*fz);
    H3 = 1+3*fz/fx;
    H = (1 + H1 +H2/2)^2;
    xn = z - H*H3*fz/dfx;
    incr = abs(double(xn - x0));
    Inc = [Inc incr];

    [fx,dfx] = feval(f,xn);
    incr2 = abs(double(fx));
    x0 = xn;
end

p = log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

4. Método M8C

```

function [x0,iter,p,incr,incr2] = M8C(x0,f,maxiter,tol)
iter = 0; incr = tol+1; incr2=incr;
Inc=[];

[fx,dfx] = feval(f,x0);
while incr2>tol && incr>tol && iter < maxiter
    delta = fx/dfx;
    iter = iter+1;
    y = x0 - delta;
    [fy] = feval(f,y);
    H1 = fy/(fx - 2*fy);
    z = x0 - (1 + H1)*delta;
    [fz] = feval(f,z);
    H2 = fz/(fy - fz);
    H3 = fz/dfx;
    H = (1 + H1 + H2/2);
    u = z - H^2*H3;
    H4 = fy/fx;
    xn = u - 3*H2*H4*H3;

```

```

    incr = abs(double(xn - x0));
    Inc = [Inc incr];

    [fx,dfx] = feval(f,xn);
    incr2 = abs(double(fx));
    x0 = xn;
end

p = log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

5. Método M8D

```

function [x0,iter,p,incr,incr2] = M8D(x0,f,maxiter,tol)
iter = 0; incr = tol+1; incr2=incr;
Inc=[];

[fx,dfx] = feval(f,x0);
while incr2>tol && incr>tol && iter < maxiter
    delta = fx/dfx;
    iter = iter+1;
    y = x0 - delta;
    [fy] = feval(f,y);
    H1 = fy/(fx - 2*fy);
    z = x0 - (1 + H1)*delta;
    [fz] = feval(f,z);
    H2 = fz/(fy - fz);
    H3 = fz/dfx;
    H = (1 + H1 + H2/2);
    u = z - H^2*H3;
    H4 = (u - z)/(y - x0);
    xn = u - 3*H4*H3;
    incr = abs(double(xn - x0));
    Inc = [Inc incr];

    [fx,dfx] = feval(f,xn);
    incr2 = abs(double(fx));
    x0 = xn;
end

p = log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

ANEXO 4: M-file de los métodos utilizados para comparación para ecuaciones no lineales

1. Método de Newton (NC)

```
function [x0,iter,p,incr,incr2]=newton(x0,f,maxiter,tol)
incr=tol+1;
incr2=incr;
Inc=[];
iter=0;

[fx,dfx]=feval(f,x0);
while incr2>tol && incr>tol && iter<maxiter
    x=x0-fx/dfx;
    incr=abs(double(x-x0));
    Inc=[Inc incr];
    x0=x;
    [fx,dfx]=feval(f,x0);
    incr2=abs(double(fx));
    iter=iter+1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));
```

2. Método de Traub (TR)

```
function [x0,iter,p,incr,incr2]=traub(x0,f,maxiter,tol)
incr=tol+1;
incr2=incr;
Inc=[];
iter=0;

[fx,dfx]=feval(f,x0);
while incr2>tol && incr>tol && iter<maxiter
    y=x0-fx/dfx;
    fy=feval(f,y);
    x=x0-(fx+fy)/dfx;
    incr=abs(double(x-x0));
    Inc=[Inc incr];
    x0=x;
    [fx,dfx]=feval(f,x);
    incr2=abs(double(fx));
    iter=iter+1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));
```

3. Método de Ostrowski (OS)

```
function [x0,iter,p,incr,incr2] = ostrowski(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;

[fx,dfx] = feval(f,x0);

while incr2>tol && incr>tol && iter<maxiter
    a = fx/dfx;
```

```

    y = x0 - a;
    fy = feval(f,y);
    b = fy/(fx - 2*fy);
    c = b*a;
    xn = y - c;
    incr = abs(double(xn - x0));
    Inc = [Inc incr];
    x0 = xn;
    [fx, dfx] = feval(f,xn);
    incr2 = abs(double(fx));
    iter = iter + 1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

4. Método de Jarratt

```

function [x0,iter,p,incr,incr2] = jarratt(x0,f,maxiter,tol)
incr=tol+1;
incr2=incr;
Inc=[];
iter=0;

[fx,dfx]=feval(f,x0);
while incr2>tol && incr>tol && iter<maxiter
    m=fx/dfx;
    y=x0-2*m/3;
    [~,dfy] = feval(f,y);
    a1=3*dfy+dfx;
    a2=3*dfy-dfx;
    a=a1/a2;
    x=x0-a*m/2;
    incr=abs(double(x-x0));
    Inc=[Inc incr];
    x0=x;
    [fx,dfx]=feval(f,x0);
    incr2=abs(double(fx));
    iter=iter+1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));
incr=abs(double(x-x0));

```

5. Método DC

```

function [x0,iter,p,incr,incr2] = DC(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;

[fx,dfx] = feval(f,x0);
while incr2>tol && incr>tol && iter<maxiter
    a = fx/dfx;
    y = x0 - a;
    fy = feval(f,y);
    b = (fx + fy)/dfx;
    z = x0 - b;

```

```

    fz = feval(f,z);
    x = x0 - (fx + fy + fz)/dfx;
    incr = abs(double(x - x0));
    Inc = [Inc incr];
    x0 = x;
    [fx, dfx] = feval(f,x0);
    incr2 = abs(double(fx));
    iter = iter + 1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

6. Método R1

```

function [x0,iter,p, incr,incr2] = R1(x0,f, maxiter,tol)
iter = 0; incr = tol+1; incr2=incr;
Inc=[];

[fx,dfx] = feval(f,x0);
while incr2>tol && incr>tol && iter < maxiter
    delta = - fx/dfx ;
    iter = iter+1;
    y = x0 + 2*delta/3;
    [~,dfy] = feval(f,y);
    H1=(3*dfy+dfx)/(-dfx+3*dfy);
    z=x0+H1*delta/2;

    [fz] = feval(f,z);
    H3 = fz/dfx;

    xn = z - H3;
    incr = abs(double(xn - z));
    Inc = [Inc incr];

    [fx,dfx] = feval(f,xn);
    incr2 = abs(double(fx));
    x0=xn;
end

p = log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

7. Método RR1

```

function [x0,iter,p,incr,incr2] = RR1(x0,f,maxiter,tol)
iter = 0; incr = tol+1; incr2=incr;
Inc=[];

[fx,dfx] = feval(f,x0);
while incr2>tol && incr>tol && iter < maxiter
    delta = - fx/dfx ;
    iter = iter+1;
    y = x0 + 2*delta/3;
    [~,dfy] = feval(f,y);
    H1=(3*dfy+dfx)/(-dfx+3*dfy);
    z=x0 + H1*delta/2;

    [fz] = feval(f,z);

```

```

H3 = 2*fz/(3*dfy-dfx);

xn = z - H3;
incr = abs(double(xn - z));
Inc = [Inc incr];

[fx,dfx] = feval(f,xn);
incr2 = abs(double(fx));
x0=xn;
end

p = log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

8. Método LW8

```

function [x0,iter,p,incr,incr2] = LW8(x0,f,maxiter,tol)
iter = 0; incr = tol+1; incr2=incr;
Inc=[];

[fx,dfx] = feval(f,x0);
while incr2>tol && incr>tol && iter < maxiter
    delta = - fx/dfx ;
    iter = iter+1;
    y = x0 + delta;
    [fy] = feval(f,y);
    H1=fy/(fx-2*fy);
    z = x0 +(1+H1)*delta;

    [fz] = feval(f,z);
    H3=fz/(fy-3*fz);

    xn = z-((1+H1)^2+(1+4*H1)*H3)*fz/dfx;
    incr = abs(double(xn - z));
    Inc = [Inc incr];

    [fx,dfx] = feval(f,xn);
    incr2 = abs(double(fx));
    x0 = xn;
end

p = log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

9. Método BRW8

```

function [x0,iter,p,incr,incr2] = BRW8(x0,f,maxiter,tol)
iter = 0; incr = tol+1; incr2=incr;
Inc=[];

[fx,dfx] = feval(f,x0);

a=1;
while incr>tol && iter < maxiter %&& incr2>tol
    delta = - fx/dfx ;
    iter = iter+1;
    y = x0 + delta;
    [fy] = feval(f,y);

```

```
delta2 = - fy/dfx ;
C1=(2*fx-fy)/(2*fx-5*fy);
z=y+C1*delta2;
[fz] = feval(f,z);
C2=(fx+(2+a)*fz)/(fx+a*fz);
d1=(fz-fy)/(z-y);
d2=(fz-fx)/(z-x0);
d3=(d2-dfx)/(z-x0);
C3=fz/(d1+d3*(z-y));
xn=z-C2*C3;
incr = abs(double(xn-x0));
Inc = [Inc incr];

[fx,dfx] = feval(f,xn);
%incr2 = abs(double(fx));
x0=xn;
end

p = log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));
```

ANEXO 5: M-file con las sistemas de ecuaciones no lineales utilizadas

```

function [F,dF] = misistemas(x)

global N
switch N

case 1  %problema (a)

    F(1) = exp(x(1))*exp(x(2)) + (x(1))*cos(x(2));
    F(2) = x(1) + x(2) - 1;

    dF(1,1)=exp(x(1))*exp(x(2)) + cos(x(2));
    dF(1,2)=exp(x(1))*exp(x(2)) - (x(1))*sin(x(2));
    dF(2,1)=1;
    dF(2,2)=1;

    F = [F(1) F(2)];
    dF = [dF(1,1) dF(1,2); dF(2,1) dF(2,2)];

case 2 %problema (b)

    F(4) = x(2)*x(3)+x(4)*(x(2)+x(3));
    F(2) = x(1)*x(3)+x(4)*(x(1)+x(3));
    F(3) = x(1)*x(2)+x(4)*(x(1)+x(2));
    F(1) = x(1)*x(2)+x(1)*x(3)+x(2)*x(3)-1;

    dF(1,1) = x(2)+x(3);
    dF(1,2) = x(1)+x(3);
    dF(1,3) = x(1)+x(2);
    dF(1,4) = 0;
    dF(2,1) = x(3)+x(4);
    dF(2,2) = 0;
    dF(2,3) = x(1)+x(4);
    dF(2,4) = x(1)+x(3);
    dF(3,1) = x(2)+x(4);
    dF(3,2) = x(1)+x(4);
    dF(3,3) = 0;
    dF(3,4) = x(1)+x(2);
    dF(4,1) = 0;
    dF(4,2) = x(3)+x(4);
    dF(4,3) = x(2)+x(4);
    dF(4,4) = x(2)+x(3);

    F = [F(1) F(2) F(3) F(4)];
    dF = [dF(1,1) dF(1,2) dF(1,3) dF(1,4); dF(2,1) dF(2,2) dF(2,3) dF(2,4);
    dF(3,1) dF(3,2) dF(3,3) dF(3,4);
    dF(4,1) dF(4,2) dF(4,3) dF(4,4)];

case 3 %problema (c)

    F(1) = x(1)^2 + x(2)^2 - 1;

```

```

F(2) = x(1)^2 - x(2)^2 - 1/2;

dF(1,1) = 2*x(1);
dF(1,2) = 2*x(2);
dF(2,1) = 2*x(1);
dF(2,2) = -2*x(2);

F = [F(1) F(2)];
dF = [dF(1,1) dF(1,2); dF(2,1) dF(2,2)];

```

case 4 %problema (d)

```

n=length(x);
for i=1:n-1
    y(i)= x(i)*x(i+1)-1;
end
y(n)=x(n)*x(1)-1;
%size(x)
a=x(2:n);
a=[a; x(1)]; a=a'; %size(a),
b=x(1:n-1); %size(b)
dy=diag(a)+diag(b,1);
dy(n,1)=x(n);
F=y;
dF=dy;

```

case 5 %problema (e)

```

F(1) = x(1)^2 - x(1) - x(2)^2 - 1;
F(2) = -sin(x(1)) + x(2);

dF(1,1) = 2*x(1) - 1;
dF(1,2) = -2*x(2);
dF(2,1) = -cos(x(1));
dF(2,2) = 1;

F = [F(1) F(2)];
dF = [dF(1,1) dF(1,2); dF(2,1) dF(2,2)];

```

case 6 %problema (f)

```

F(1) = x(1)^2 + x(2)^2 - 4;
F(2) = exp(x(1)) + x(2) - 1;

dF(1,1) = 2*x(1);
dF(1,2) = 2*x(2);
dF(2,1) = exp(x(1));
dF(2,2) = 1;

F = [F(1) F(2)];
dF = [dF(1,1) dF(1,2); dF(2,1) dF(2,2)];

```

case 7 %problema (g)

$$F(1) = x(1)^2 + x(2)^2 + x(3)^2 - 9;$$

$$F(2) = x(1)*x(2)*x(3) - 1;$$

$$F(3) = x(1) + x(2) - x(3)^2;$$

$$dF(1,1) = 2*x(1);$$

$$dF(1,2) = 2*x(2);$$

$$dF(1,3) = 2*x(3);$$

$$dF(2,1) = x(2)*x(3);$$

$$dF(2,2) = x(1)*x(3);$$

$$dF(2,3) = x(1)*x(2);$$

$$dF(3,1) = 1;$$

$$dF(3,2) = 1;$$

$$dF(3,3) = -2*x(3);$$

$$F = [F(1) F(2) F(3)];$$

$$dF = [dF(1,1) dF(1,2) dF(1,3); dF(2,1) dF(2,2) dF(2,3); dF(3,1) dF(3,2) dF(3,3)];$$

case 8 %problema (h)

$$F(1) = \sin(x(1)) + x(2)^2 + \log_{10}(x(3)) - 7;$$

$$F(2) = 3*x(1) + 2^x(2) - x(3)^{-3};$$

$$F(3) = x(1) + x(2) + x(3) - 5;$$

$$dF(1,1) = \cos(x(1));$$

$$dF(1,2) = 2*x(2);$$

$$dF(1,3) = 1/(x(3)*\log(10));$$

$$dF(2,1) = 3;$$

$$dF(2,2) = 2^x(2)*\log(2);$$

$$dF(2,3) = 3/x(3)^4;$$

$$dF(3,1) = 1;$$

$$dF(3,2) = 1;$$

$$dF(3,3) = 1;$$

$$F = [F(1) F(2) F(3)];$$

$$dF = [dF(1,1) dF(1,2) dF(1,3); dF(2,1) dF(2,2) dF(2,3); dF(3,1) dF(3,2) dF(3,3)];$$

case 9 %problema (i)

$$F(1) = (x(1)-1)^6 - x(2);$$

$$F(2) = x(2) - 1;$$

$$dF(1,1) = 6*(x(1)-1)^5;$$

$$dF(1,2) = -1;$$

$$dF(2,1) = 0;$$

$$dF(2,2) = 1;$$

$$F = [F(1) F(2)];$$

$$dF = [dF(1,1) dF(1,2); dF(2,1) dF(2,2)];$$

case 10 %problema (j)

```

F(1) = log(x(1)^2) - 2*log(cos(x(2)));
F(2) = x(1)*tan(x(1)/sqrt(2) + x(2)) - sqrt(2);

dF(1,1) = 2/x(1);
dF(1,2) = 2*tan(x(2));
dF(2,1) = tan(x(2)+(2^(1/2)*x(1))/2)+(2^(1/2)*x(1)*(tan(x(2)+(2^(1/2)*x(1))/2)^2+1))/2;
dF(2,2) = x(1)*(tan(x(2) + (2^(1/2)*x(1))/2)^2 + 1);

F = [F(1) F(2)];
dF = [dF(1,1) dF(1,2); dF(2,1) dF(2,2)];

```

case 11 %problema (k)

```

F(1) = x(1) + exp(x(2)) - cos(x(2));
F(2) = 3*x(1) - x(2) - sin(x(2));

dF(1,1) = 1;
dF(1,2) = exp(x(2)) + sin(x(2));
dF(2,1) = 3;
dF(2,2) = -1 - cos(x(2));

F = [F(1) F(2)];
dF = [dF(1,1) dF(1,2); dF(2,1) dF(2,2)];

```

case 12 %problema (l)

```

F(1) = cos(x(2)) - sin(x(1));
F(2) = x(3)^x(1) - 1/x(2);
F(3) = exp(x(1)) - x(3)^2;

dF(1,1) = -cos(x(1));
dF(1,2) = -sin(x(2));
dF(1,3) = 0;
dF(2,1) = x(3)^x(1)*log(x(3));
dF(2,2) = 1/x(2)^2;
dF(2,3) = x(1)*x(3)^(x(1) - 1);
dF(3,1) = exp(x(1));
dF(3,2) = 0;
dF(3,3) = -2*x(3);

F = [F(1) F(2) F(3)];
dF = [dF(1,1) dF(1,2) dF(1,3); dF(2,1) dF(2,2) dF(2,3); dF(3,1) dF(3,2) dF(3,3)];

```

case 14 %problema (n)

```

F(1) = 4*x(1) - x(2) + x(3) - x(1)*x(4);
F(2) = -x(1) + 3*x(2) - 2*x(3) - x(2)*x(4);
F(3) = x(1) - 2*x(2) + 3*x(3) - x(3)*x(4);
F(4) = x(1)^2 + x(2)^2 + x(3)^2 + x(4)^2 - 1;

dF(1,1) = 4 - x(4);
dF(1,2) = -1;
dF(1,3) = 1;

```

```
dF(1,4) = -x(1);  
dF(2,1) = -1;  
dF(2,2) = 3 - x(4);  
dF(2,3) = -2;  
dF(2,4) = -x(2);  
dF(3,1) = 1;  
dF(3,2) = -2;  
dF(3,3) = 3 - x(4);  
dF(3,4) = -x(3);  
dF(4,1) = 2*x(1);  
dF(4,2) = 2*x(2);  
dF(4,3) = 2*x(3);  
dF(4,4) = 2*x(4);
```

```
F = [F(1) F(2) F(4) F(3)];
```

```
dF = [dF(1,1) dF(1,2) dF(1,3) dF(1,4); dF(2,1) dF(2,2) dF(2,3) dF(2,4); dF(3,1)  
dF(3,2) dF(3,3) dF(3,4); dF(4,1) dF(4,2) dF(4,3) dF(4,4)];
```

```
end
```

ANEXO 6: Métodos de las familias con cuadratura de Gauss para sistema de ecuaciones no lineales

1. Familia NeG-S

a). NeG-LD1S

```
function [x0,iter,p,incr,incr2] = NeGLD1S(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;
x=x0(:);
x0=x0(:);

[fx,dfx] = feval(f,x0);
while (incr+incr2)>tol && iter<maxiter && incr>tol
    d = dfx\fx'; size(x0),size(d)
    y = x0 - d;
    z = (x0 + y)/2;
    [~,dfz] = feval(f,z);
    dd=dfz\fx';
    xn = x0 - dd;
    incr = norm(double(x0-xn));
    Inc = [Inc incr];
    x = [x0, xn];
    x0 = xn;
    [fx, dfx] = feval(f,x0);
    incr2 = norm(double(fx));
    iter = iter + 1;
    Inc(iter)=incr;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));
```

b). NeG-LD2S

```
function [x0,iter,p,incr,incr2] = NeGLD2S(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;
x=x0(:);
x0=x0(:);

q=sqrt(sym(3));
a=vpa((3+q)/6);
b=vpa((3-q)/6);

[fx,dfx] = feval(f,x0);

while (incr+incr2)>tol && iter<maxiter && incr>tol
    d = dfx\fx'; size(x0),size(d)
    y = x0 - d;
    u = a*x0 + b*y;
    v = b*x0 + a*y;
    [~,dfu] = feval(f,u);
    [~,dfv] = feval(f,v);
    delta=2*((dfu+dfv)\fx');
```

```

    xn = x0 - delta;
    incr = norm(double(delta));
    Inc = [Inc incr];
    x = [x, xn];
    x0 = xn;
    [fx, dfx] = feval(f,xn);
    incr2 = norm(double(fx));
    iter = iter + 1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

c). NeG-LT2S

```

function [x0,iter,p,incr,incr2] = NeGLT2S(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;
x0=x0(:);

[fx,dfx] = feval(f,x0);

while (incr+incr2)>tol && iter<maxiter && incr>tol
    d = dfx\fx'; size(x0),size(d)
    y = x0 - d;
    [~, dfy] = feval(f,y);
    x = x0 - 2*((dfy+dfx)\fx');
    incr = norm(double(x - x0));
    Inc = [Inc incr];
    [fx, dfx] = feval(f,x);
    x0 = x;
    incr2 = norm(double(fx));
    iter = iter + 1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

d). NeG-R2S

```

function [x0, iter,p,incr,incr2] = NeGR2S(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;
x=x0(:);
x0=x0(:);

[fx,dfx] = feval(f,x0);
while (incr+incr2)>tol && iter<maxiter && incr>tol
    d = dfx\fx'; size(x0),size(d)
    y = x0 - d;
    z = (x0 + 2*y)/3;
    [~,dfz] = feval(f,z);
    dd = (3*dfz+dfx)\fx'; size(dd)
    xn = x0 -4*dd;
    incr = norm(double(x0 - xn));
    Inc = [Inc incr];
    [fx, dfx] = feval(f,xn);
    x0 = xn;

```

```

    incr2 = norm(double(fx));
    iter = iter + 1;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

2. Familia TrG-S

a). TrG-LD1S

```

function [x0,iter,p,incr,incr2] = TrGLD1S(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;
x=x0(:);
x0=x0(:);

[fx,dfx] = feval(f,x0);
while (incr+incr2)>tol && iter<maxiter && incr>tol
    d = dfx\fx'; size(x0),size(d)
    y = x0 - d;
    fy = feval(f,y);
    dd = dfx\ (fx+fy)';
    z = x0 - dd;
    u = (y + z)/2;
    [~,dfu] = feval(f,u);
    ddd = dfu\fy';
    xn = y - ddd;
    incr = norm(double(x0-xn));
    Inc = [Inc incr];
    x = [x, xn];
    x0 = xn;
    [fx, dfx] = feval(f,x0);
    incr2 = norm(double(fx));
    iter = iter + 1;
    Inc(iter)=incr;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

b). TrG-LD2S

```

function [x0,iter,p,incr,incr2] = TrGLD2S(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;
x=x0(:);
x0=x0(:);

q=sqrt(sym(3));
a=vpa((3+q)/6);
b=vpa((3-q)/6);

[fx,dfx] = feval(f,x0);
while (incr+incr2)>tol && iter<maxiter && incr>tol
    d = dfx\fx'; size(x0),size(d)
    y = x0 - d;
    fy = feval(f,y);

```

```

    dd = dfx\'(fx+fy)';
    z = x0 - dd;
    u = a*y + b*z;
    v = b*y + a*z;
    [~,dfu] = feval(f,u);
    [~,dfv] = feval(f,v);
    ddd = 2*((dfu+dfv)\fy');
    xn = y - ddd;
    incr = norm(double(x0-xn));
    Inc = [Inc incr];
    x = [x, xn];
    x0 = xn;
    [fx, dfx] = feval(f,x0);
    incr2 = norm(double(fx));
    iter = iter + 1;
    Inc(iter)=incr;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

c). TrG-LT2S

```

function [x0,iter,p,incr,incr2] = TrGLT2S(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;
x=x0(:);
x0=x0(:);

[fx,dfx] = feval(f,x0);
while (incr+incr2)>tol && iter<maxiter && incr>tol
    d = dfx\'fx'; size(x0),size(d)
    y = x0 - d;
    [fy, dfy] = feval(f,y);
    dd = dfx\'(fx+fy)';
    z = x0 - dd;
    [~, dfz] = feval(f,z);
    ddd = (dfy + dfz)\fy';
    xn = y - 2*ddd;
    incr = norm(double(x0-xn));
    Inc = [Inc incr];
    x = [x, xn];
    x0 = xn;
    [fx, dfx] = feval(f,x0);
    incr2 = norm(double(fx));
    iter = iter + 1;
    Inc(iter)=incr;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

d). TrG-R2S

```

function [x0,iter,p,incr,incr2] = TrGR2S(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;
x=x0(:);

```

```
x0=x0(:);

[fx,dfx] = feval(f,x0);
while (incr+incr2)>tol && iter<maxiter && incr>tol
    d = dfx\fx'; size(x0),size(d)
    y = x0 - d;
    [fy, dfy] = feval(f,y);
    dd = dfx\'(fx+fy)';
    z = x0 - dd;
    u = (2*z + y)/3;
    [~, dfu] = feval(f,u);
    ddd = (dfy + 3*dfu)\fy';
    xn = y - 4*ddd;
    incr = norm(double(x0-xn));
    Inc = [Inc incr];
    x = [x, xn];
    x0 = xn;
    [fx, dfx] = feval(f,x0);
    incr2 = norm(double(fx));
    iter = iter + 1;
    Inc(iter)=incr;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));
```

ANEXO 8: M-file con métodos de sistema de ecuaciones no lineales

```

function [x0,iter,p,incr,incr2] = M5S(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;
x=x0(:);
x0=x0(:);

[fx,dfx] = feval(f,x0);
while (incr+incr2)>tol && iter<maxiter && incr>tol
    d = dfx\fx'; size(x0),size(d)
    y = x0 - d;
    [fy, dfy] = feval(f,y);
    a = (dfx - 5*dfy)\(3*dfx + dfy);
    b = dfx\fy';
    dd = a*b;
    xn = y + dd;
    incr = norm(double(x0-xn));
    Inc = [Inc incr];
    x = [x, xn];
    x0 = xn;
    [fx, dfx] = feval(f,x0);
    incr2 = norm(double(fx));
    iter = iter + 1;
    Inc(iter)=incr;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

function [x0,iter,p,incr,incr2] = M8S(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;
x=x0(:);
x0=x0(:);

[fx,dfx] = feval(f,x0);
while (incr+incr2)>tol && iter<maxiter && incr>tol
    d = dfx\fx'; size(x0),size(d)
    y = x0 - 2*d/3;
    z = y + d/6;
    [~, dfy] = feval(f,y);
    m = dfx-3*dfy;
    dd = m\fx';
    u = z + dd;
    fu = feval(f,u);
    n = fx + 2*fu;
    ddd = m\n';
    v = z + ddd;
    fv = feval(f,v);
    a = dfx\((5*dfx - 3*dfy));
    b = dfx\fv';
    xn = v - (a*b)/2;
    incr = norm(double(x0-xn));

```

```

    Inc = [Inc incr];
    x = [x, xn];
    x0 = xn;
    [fx, dfx] = feval(f,x0);
    incr2 = norm(double(fx));
    iter = iter + 1;
    Inc(iter)=incr;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

function [x0,iter,p,incr,incr2] = M10S(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;
x=x0(:);
x0=x0(:);

[fx,dfx] = feval(f,x0);
while (incr+incr2)>tol && iter<maxiter && incr>tol
    d = dfx\fx'; size(x0),size(d)
    y = x0 - d/2;
    [fy, dfy] = feval(f,y);
    dd = (dfx-2*dfy)\(3*fx-4*fy)';
    z = x0 + dd;
    fz = feval(f,z);
    ddd = (dfx-2*dfy)\fz';
    u = z + ddd;
    v = (u + z)/2;
    [~, dfv] = feval(f,v);
    dddd = dfv\fz';
    xn = z - dddd;
    incr = norm(double(x0-xn));
    Inc = [Inc incr];
    %x = [x, xn];
    x0 = xn;
    [fx, dfx] = feval(f,x0);
    incr2 = norm(double(fx));
    iter = iter + 1;
    Inc(iter)=incr;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

function [x0,iter,p,incr,incr2] = M14S(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;
x=x0(:);
x0=x0(:);

[fx,dfx] = feval(f,x0);
while (incr+incr2)>tol && iter<maxiter && incr>tol
    d = dfx\fx'; size(x0),size(d)
    y = x0 - 2*d/3;
    z = y + d/6;

```

```
[~, dfy] = feval(f,y);
m = dfx-3*dfy;
dd = m\fx';
u = z + dd;
fu = feval(f,u);
n = fx + 2*fu;
ddd = m\n';
v = z + ddd;
fv = feval(f,v);
a = dfx\((5*dfx - 3*dfy));
b = dfx\fv';
w = v - (a*b)/2;
q = (w + v)/2;
[~, dfq] = feval(f,q);
p = dfq\fv';
xn = v - p;
incr = norm(double(x0-xn));
Inc = [Inc incr];
x = [x, xn];
x0 = xn;
[fx, dfx] = feval(f,x0);
incr2 = norm(double(fx));
iter = iter + 1;
Inc(iter)=incr;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));
```

ANEXO 9: M-file con métodos utilizados para comparación en sistemas no lineales

```

function [x0,iter,p,incr,incr2] = NewtonS(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;
x=x0(:);
x0=x0(:);

[fx,dfx] = feval(f,x0);
while (incr+incr2)>tol && iter<maxiter && incr>tol
    d = dfx\fx'; size(x0),size(d)
    xn = x0 - d;
    incr = norm(double(x0-xn));
    Inc = [Inc incr];
    x = [x0, xn];
    x0 = xn;
    [fx, dfx] = feval(f,x0);
    incr2 = norm(double(fx));
    iter = iter + 1;
    Inc(iter)=incr;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

function [x0,iter,p,incr,incr2] = TraubS(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;
x=x0(:);
x0=x0(:);

[fx,dfx] = feval(f,x0);
while (incr+incr2)>tol && iter<maxiter && incr>tol
    d = dfx\fx'; size(x0),size(d)
    y = x0 - d;
    fy = feval(f,y);
    dd = dfx\((fx+fy)');
    xn = x0 - dd;
    incr = norm(double(xn - x0));
    Inc = [Inc incr];
    x = [x, xn];
    x0 = xn;
    [fx, dfx] = feval(f,x0);
    incr2 = norm(double(fx));
    iter = iter + 1;
    Inc(iter)=incr;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

function [x0, iter, p,incr, incr2] = JarrattS(x0, f, maxiter, tol)
iter = 0;
incr = tol+1;
incr2=incr;
x=x0(:);

```

```

x0=x0(:);
Inc=[];

[fx,dfx] = feval(f,x0);
while (incr+incr2)>tol && iter<maxiter && incr>tol
    d = dfx\fx'; size(x0),size(d)
    y = x0 - (2/3)*d;
    [~,dfy] = feval(f,y);
    dd = ((3*dfy - dfx)\(3*dfy + dfx))*d;
    xn = x0 - (1/2)*dd;
    incr = norm(double(xn - x0));
    Inc = [Inc incr];
    x = [x, xn];
    x0 = xn;
    [fx,dfx] = feval(f,x0);
    incr2 = norm(double(fx));
    iter = iter+1;
    Inc(iter) = incr;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

```

function [x0,iter,p,incr,incr2] = DCongS(x0,f,maxiter,tol)
incr = tol + 1;
incr2 = incr;
Inc = [];
iter = 0;
x=x0(:);
x0=x0(:);

```

```

[fx,dfx] = feval(f,x0);
while (incr+incr2)>tol && iter<maxiter && incr>tol
    d = dfx\fx'; size(x0),size(d)
    y = x0 - d;
    fy = feval(f,y);
    dd = dfx\((fx+fy)');
    z = x0 - dd;
    fz = feval(f,z);
    a = fx+fy+fz;
    ddd = dfx\a';
    u = x0 - ddd;
    v = (u + z)/2;
    [~,dfv] = feval(f,v);
    dddd = dfv\fz';
    xn = z - dddd;
    incr = norm(double(x0 - xn));
    Inc = [Inc incr];
    x = [x, xn];
    x0 = xn;
    [fx, dfx] = feval(f,x0);
    incr2 = norm(double(fx));
    iter = iter + 1;
    Inc(iter)=incr;
end
p=log(Inc(3:end)./Inc(2:end-1))./log(Inc(2:end-1)./Inc(1:end-2));

```

ANEXO 10: M-file de cálculo y gráficas de los Índices

```

function [IECS,IES]=IndicesS

n = (1:1:10);
n = (10:10:110);
n = (100:50:550);

% INDICE DE EFICIENCIA COMPUTACIONAL

iec_NS = 2.^(1./((1/3)*n.^3+2*n.^2+(2/3)*n));
iec_TS = 3.^(1./((1/3)*n.^3+3*n.^2+(5/3)*n));
iec_JS = 4.^(1./((2/3)*n.^3+5*n.^2+(1/3)*n));
iec_DCongS = 5.^(1./((1/3)*n.^3+4*n.^2+(8/3)*n));
iec_aliciaS = 5.^(1./((1/3)*n.^3 + 4*n.^2 + (7/3)*n));
iec_NeGLD1S = 3.^(1./((2/3)*n.^3+4*n.^2+(1/3)*n));
iec_NeGLD2S = 3.^(1./((2/3)*n.^3+5*n.^2+(1/3)*n));
iec_TrGLD1S = 5.^(1./((2/3)*n.^3+5*n.^2+(4/3)*n));
iec_TrGLD2S = 5.^(1./((2/3)*n.^3+6*n.^2+(4/3)*n));
iec_M40A = 4.^(1./((2/3)*n.^3 + 4*n.^2 + (1/3)*n));
iec_M5S = 5.^(1./((2/3)*n.^3 + 7*n.^2 + (4/3)*n));
iec_M60A = 6.^(1./((2/3)*n.^3 + 5*n.^2 + (4/3)*n));
iec_M8 = 8.^(1./((2/3)*n.^3 + 8*n.^2 + (7/3)*n));
iec_M140A = 14.^(1./((n.^3 + 9*n.^2 + 2*n)));
iec_M4D = 4.^(1./((2/3)*n.^3 + 4*n.^2 + (5/3)*n));
iec_M6D = 6.^(1./((2/3)*n.^3 + 5*n.^2 + (7/3)*n));
iec_M10 = 10.^(1./((n.^3 + 7*n.^2 + 2*n)));
iec_M14D = 14.^(1./((2/3)*n.^3 + 8*n.^2 + (7/3)*n));
iec_DCGS = 7.^(1./((2/3)*n.^3+6*n.^2+(7/3)*n));

% INDICE DE EFICIENCIA

ie_NS = 2.^(1./((n.^2 + n)));
ie_TS = 3.^(1./((n.^2 + 2*n)));
ie_JS = 4.^(1./((2*n.^2 + n)));
ie_DCongS = 5.^(1./((n.^2 + 3*n)));
ie_aliciaS = 5.^(1./((n.^2 + 3*n)));
ie_NeGLD1S = 3.^(1./((2*n.^2 + n)));
ie_NeGLD2S = 3.^(1./((3*n.^2 + n)));
ie_TrGLD1S = 5.^(1./((2*n.^2 + 2*n)));
ie_TrGLD2S = 5.^(1./((3*n.^2 + 2*n)));
ie_M5S = 5.^(1./((2*n.^2 + 2*n)));
ie_M40A = 4.^(1./((2*n.^2 + n)));
ie_M60A = 6.^(1./((2*n.^2 + 2*n)));
ie_M8 = 8.^(1./((2*n.^2 + 3*n)));
ie_M140A = 14.^(1./((3*n.^2 + 3*n)));
ie_M4D = 4.^(1./((2*n.^2 + 2*n)));b
ie_M6D = 6.^(1./((2*n.^2 + 3*n)));
ie_M10 = 10.^(1./((3*n.^2 + 3*n)));
ie_M14D = 14.^(1./((3*n.^2 + 3*n)));
ie_DCGS = 7.^(1./((2*n.^2 + 3*n)));

IECS=[iec_NS; iec_TS; iec_JS; iec_DCongS; iec_M140A; iec_M14D; iec_DCGS]; IECS = IECS';
IES=[ie_NS; ie_TS; ie_JS; ie_DCongS; ie_M140A; ie_M14D; ie_DCGS]; IES = IES';

```

```
subplot(2,1,1), plot(n',IES),title('Indice de eficiencia'),  
legend('I_{NC}','I_{TR}','I_{JT}','I_{DCongS}','I_{M140A}','I_{M14D}','I_{DCGS}'),  
xlabel('Tamano del sistema, n')  
ylabel('Indice de eficiencia, I')
```

```
subplot(2,1,2), plot(n',IECS),title('Indice de eficiencia computacional'),  
legend('IC_{NC}','IC_{TR}','IC_{JT}','IC_{DCongS}','IC_{M140A}','IC_{M14D}','IC_{DCGS}'),  
xlabel('Tamano del sistema, n')  
ylabel('Indice de eficiencia computacional, IC')
```

ANEXO 11 Gráficas de las funciones de las pruebas numéricas de 1 a 18

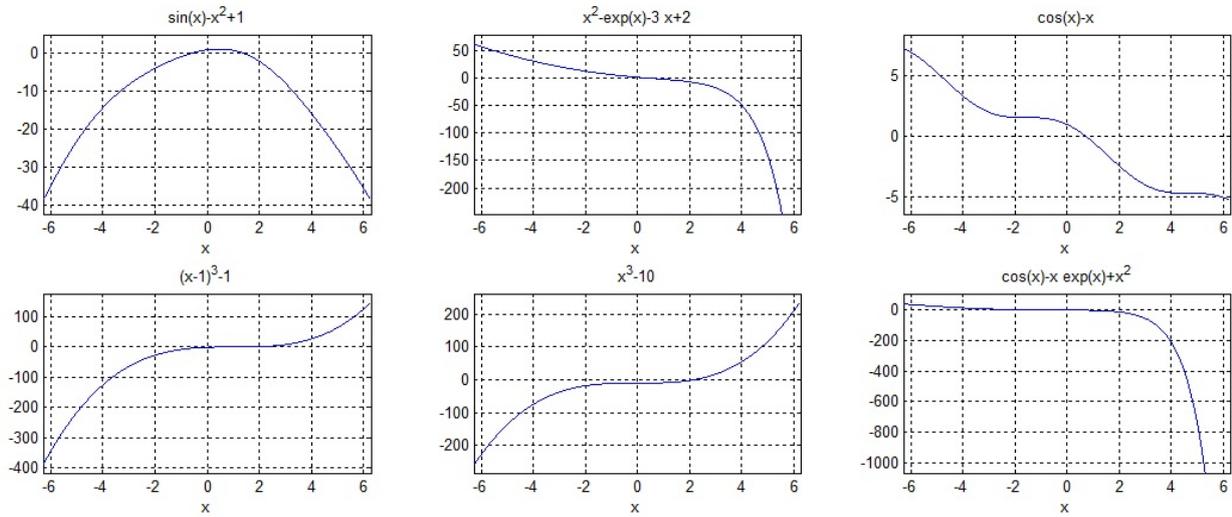


Figura 8.1: Gráficas de las funciones de las pruebas numéricas de 1 a 6

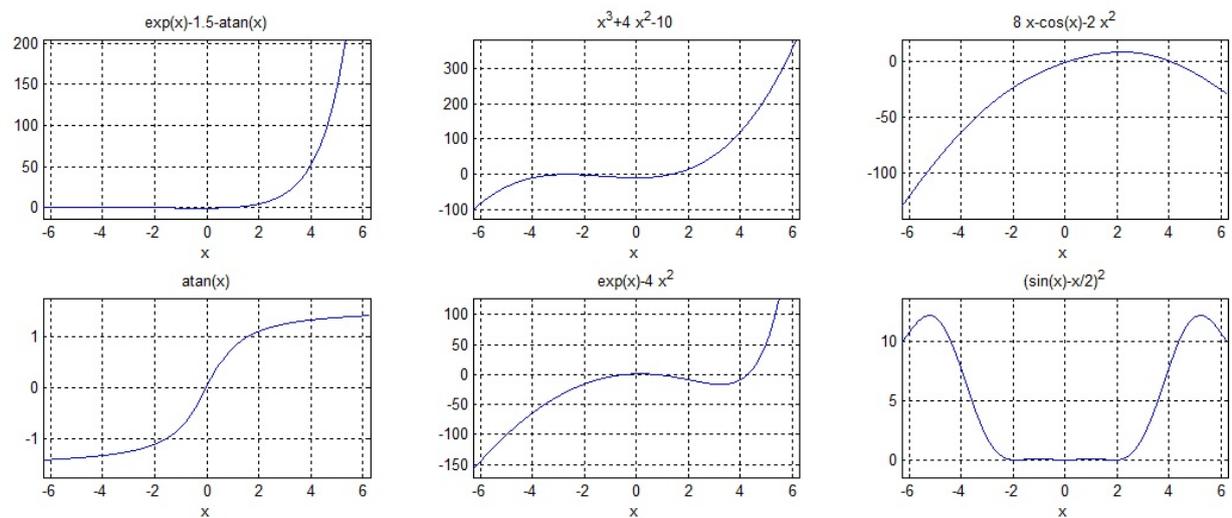


Figura 8.2: Gráficas de las funciones de las pruebas numéricas de 6 a 12

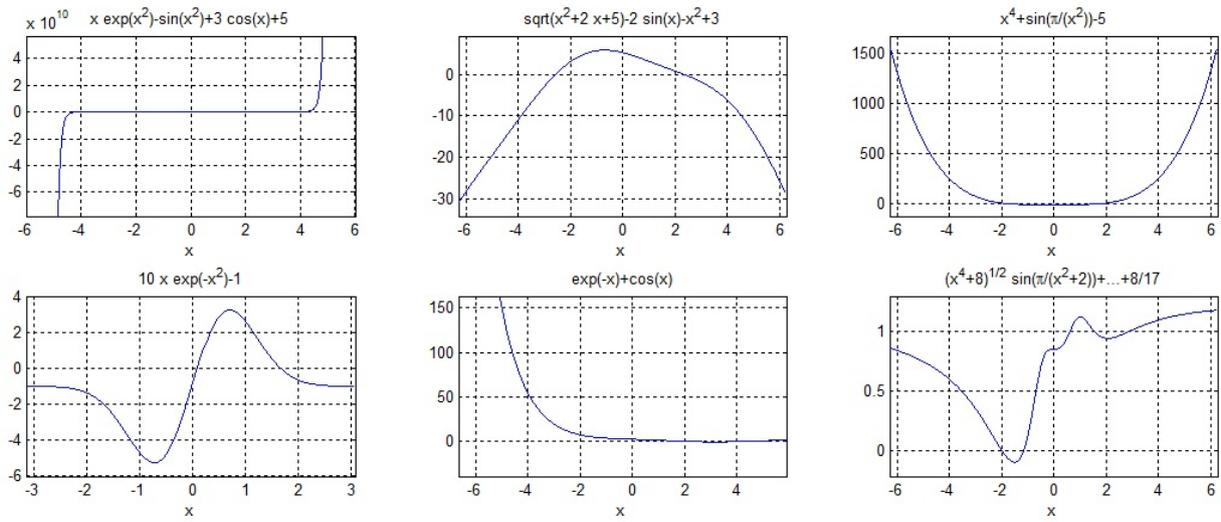


Figura 8.3: Gráficas de las funciones de las pruebas numéricas de 13 a 18

Bibliografía

- [1] Abbasbandy S., *Improving Newton-Raphson method for nonlinear equations by modified Adomian decomposition method*, Applied Mathematics and Computation, 145 (2003) 887-893.
- [2] Adomian G., *Solving Frontier Problem of Physics: The Decomposition Method*, Kluwer Academic Publishers, Dordrecht, 1994.
- [3] Alarcón V, Amat S., Busquier S., López D.J., *A Steffensen's type method in Banach spaces with applications on boundary-value problems*, Journal of Computational and Applied Mathematics, 216 (2008) 243-250.
- [4] Argyros I.K., *Concerning the "terra incognita" between convergence regions of two Newton methods*, Nonlinear Analysis, 62 (2005) 179-184.
- [5] Arroyo V., Cordero A., Torregrosa J.R., Vassileva M.P., *Artificial Satellites Preliminary Orbit Determination by modified high-order Gauss methods*, International Journal of Computer Mathematics, doi: 10.1080/00207160.2011.560266.
- [6] Babajee D.K.R., Dauhoo M.Z., Darvishi M.T., Karami A., Barati A., *Analysis of two Chebyshev-like third order methods free from second derivatives for solving systems of nonlinear equations*, Journal of Computational and Applied Mathematics, 233 (2010) 2001-2012.
- [7] Bi W., Ren H., Wu Q., *Third-step iterative methods with eighth-order convergence for solving nonlinear equations*, Journal of Computational and Applied Mathematics, 255 (2009) 105-112
- [8] Burden R., Faires J.D., *Numerical Analysis*, seventh ed., PWS Publishing Company, Boston, 2001.
- [9] Butcher J.C., *Initial Value Problems: Numerical Methods and Mathematics*, Computers Mathematics and Applications, 28 (1994) 1-16.
- [10] Chun C., *Construction of Newton-like iterative methods for solving nonlinear equations*, Numerical Mathematics, 104, (2006) 297-315.
- [11] Chun C., *A family of composite fourth-order iterative methods for solving nonlinear equations*, Applied Mathematics and Computation, 187 (2007) 951-956.
- [12] Chun C., *Some improvements of Jarratt's method with sixth-order convergence*, Applied Mathematics and Computation 190, (2007) 1432-1437.
- [13] Chun C., Yoon Mee Ham, *Some sixth-order variants of Ostrowski root-finding methods*, Applied Mathematics and Computation 193, (2007) 389-394.
- [14] Coddington E.A., *An introduction to ordinary differential equations*, Prentice-Hall Mathematics Series, 1961.
- [15] Collette J.P., *Historia de las matemáticas*, Siglo XXI, 1998.
- [16] Cordero A., Torregrosa J.R., *Variants of Newton's Method for functions of several variables*, Applied Mathematics and Computation, 183 (2006) 199-208.

- [17] Cordero A., Torregrosa J.R., *Variants of Newton's Method using fifth-order quadrature formulas*, Applied Mathematics and Computation, 190 (2007) 686-698.
- [18] Cordero A., Torregrosa J.R., *On interpolation variants of Newton's method for functions of several variables*, Journal of Computational and Applied Mathematics, 234 (2010) 34-43.
- [19] Cordero A., Martínez E., Torregrosa J.R., *Iterative methods of order four and five for systems of nonlinear equations*, Journal of Computational and Applied Mathematics, 231 (2009) 541-551.
- [20] Cordero A., Torregrosa J.R., *On interpolation variants of Newton's method for functions of several variables*, Journal of Computational and Applied Mathematics, 234 (2010) 34-35.
- [21] Cordero A., Hueso J.L., Martínez E., Torregrosa J.R., *A modified Newton-Jarratt's composition*, Numerical Algorithm, 55 (2010) 87-99.
- [22] Cordero A., Hueso J.L., Martínez E., Torregrosa J.R., *Efficient high-order methods based on golden-ratio for nonlinear systems*, Applied Mathematics and Computation, 217 (2011) 4548-4556.
- [23] Cordero A., Torregrosa J.R., Vassileva M.P., *Tree-step iterative method with a optimal eighth order of convergence*, Journal of Computational and Applied Mathematics, 235 (2011) 3189-3194.
- [24] Cordero A., Torregrosa J.R., Vassileva M.P., *A family of modified Ostrowski's method with a optimal eighth order of convergence*, Applied Mathematics Letters, doi: 10.1016/j.aml.2011.06.002.
- [25] Danby J.M.A., *Fundamentals of Celestial Mechanics*, The MacMillan Company, 1962.
- [26] Dahlquist G., *Convergence and stability in the numerical integration of ordinary differential equations*, Math. Scand. 4 (1956) 33-53.
- [27] Darvishi M.T., Barati A., *A fourth-order method from quadrature formulae to solve systems of nonlinear equations*, Applied Mathematics and Computation, 188 (2007) 257-261.
- [28] Darvishi M.T., Barati A., *A third-order Newton-type method to solve systems of nonlinear equations*, Applied Mathematics and Computation, 187 (2007) 630-635.
- [29] Darvishi M.T., Barati A., *Super cubic iterative methods to solve systems of nonlinear equations*, Applied Mathematics and Computation, 188 (2007) 1678-1685.
- [30] Ding H., Zhang Y., Wang S., Yang X., *A note on some quadrature based three-step iterative methods for non-linear equations*, Applied Mathematics and Computation, 215 (2009) 53-57.
- [31] Escobal P.R., *Methods of Orbit Determination*, Robert E. Krieger Publishing Company, 1975.
- [32] Ezquerro J.A., Gutiérrez, J.M., Hernández, M.A., Salanova, M.A., *Chebyshev-like methods and quadratic equations*, Revue d'Analyse Numérique et de Théorie de l'Approximation, 28 (2000) 23-35.
- [33] Ezquerro J.A., Hernández M.A., *On Halley-type iterations with free second derivative*, Journal of Computational and Applied Mathematics, 170 (2004) 455-459
- [34] Ezquerro J.A., Hernández M.A., *An optimization of Chebyshev's method*, Journal of Complexity, 25 (2009) 343-361.
- [35] Ezquerro J.A., Hernández M.A., Romero N., *Newton-type methods of high order and domains of semilocal and global convergence*, Applied Mathematics and Computation 214 (2009) 142-154.
- [36] Fowler D., Robson E. *Square Root Approximations in Old Babylonian Mathematics: YBC 7289 in Context*, Historia Mathematica, 25 (1998) 366-378.
- [37] Frontini M., Sormani E., *Some variants of Newton's method with third order convergence and multiple roots*, Journal of Computational and Applied Mathematics, 156 (2003) 345-354.
- [38] Frontini M., Sormani E., *Third-order methods from quadrature formulae for solving systems of nonlinear equations*, Applied Mathematics and Computation, 149 (2004) 771-782.

- [39] Gerlach J., *Accelerated convergence in Newton's method*, SIAM Review, 36(2), 272-276 (1994)
- [40] Grau-Sánchez M., Peris J.M., Gutiérrez J.M., *Accelerated iterative methods for finding solutions of a system of nonlinear equations*, Applied Mathematics and Computation, 190 (2007) 1815-1823.
- [41] Grau-Sánchez M., Noguera M., Díaz-Barrero J.L., *Adams-like techniques for zero-finder methods*, Applied Mathematics and Computation, 211 (2009) 130-136.
- [42] Grau-Sánchez M., Gutiérrez J.M., *Zero-finder methods derived from Obreshkov's techniques*, Applied Mathematics and Computation, 215 (2009) 2992-3001.
- [43] Gutierrez J.M., Hernandez M.A., *A family of Chebyshev-Halley type methods in Banach spaces*, Bull. Aust. Math. Soc., 55 (1997) 113-130.
- [44] Hasanov V.I., Ivanov I.G., Nedzhibov G., *A new modification of Newton method*, Applied Mathematical Engineering, 27 (2002) 278-286.
- [45] Homeier H.H.H., *On Newton-tipe methods with cubic convergence*, Journal of Computational and Applied Mathematics, 176 (2005), 425-432.
- [46] Henrici H., *Discrete variable methods in ordinary differential equations*, John Wiley and Sons, New York, (1962).
- [47] Henrici H., *Error propagation for difference methods*, John Wiley and Sons, New York, (1963).
- [48] Henrici P., *Elements of Numerical Analysis*, John Wiley and Sons, New York, (1964).
- [49] Hueso J.L., Martínez E., Torregrosa J.R., *Third order iterative methods free from second derivative for nonlinear systems*, Applied Mathematics and Computation, 215 (2009) 58-65.
- [50] Hueso J.L., Martínez E., Torregrosa J.R., *Third and fourth order iterative methods free from second derivative for nonlinear systems*, Applied Mathematics and Computation, 211 (2009) 190-197.
- [51] Jarratt P., *Some fourth order multipoint methods for solving equations*, Mathematics and Computation, 20 (1966) 434-437.
- [52] Jiang D., Han D., *Some one-parameter families of third-order methods for solving nonlinear equations*, Applied Mathematics and Computation, 195 (2008) 392-396.
- [53] Kantorovic L.V., *Functional Analysis and Applied Mathematics*. Uspehi Mat. Nauk, 3 (1948), 89-185 (Russian). Tr. by C. D. Benster, Natl. Bur. Std. Rept. No. 1509, Washington, 1952.
- [54] Kanwar V., Tomar S.K., *Modified families of Newton, Halley and Chebyshev methods*, Applied Mathematics and Computation, 192 (2007) 20-26.
- [55] King R., *A family of fourth order methods for nonlinear equations*, SIAM J. Numer. Anal., 10 (1973) 876-879.
- [56] Kolmogorov A.N., Fomin S.V., *Introductory Real Analysis*, Dover Publications, Inc., New York, 1970.
- [57] Kou J., Li Y., Wang X., *A modification of Newton method with third-order convergence*, Applied Mathematics and Computation, 181 (2006) 1106-1111.
- [58] Kou J., Li Y., Wang X., *Efficient continuation Newton-like method for solving system of nonlinear equations*, Applied Mathematics and Computation, 174 (2006) 846-853.
- [59] Kou J., *On Chebyshev-Halley methods with sixth-order convergence for solving non-linear equations*, Applied Mathematics and Computation, 190 (2007) 126-131.
- [60] Kou J., Li Y., Wang X., *A composite fourth-order iterative method for solving non-linear equations*, Applied Mathematics and Computation, 184 (2007) 471-475.
- [61] Kou J., Li Y., *An improvement of Jarratt method*, Applied Mathematics and Computation, 189 (2007) 1816-1821.

- [62] Kou J., Wang X., *Sixth-order variants of Chebyshev-Halley methods for solving non-linear equations*, Applied Mathematics and Computation, 190 (2007) 1839-1843.
- [63] Kou J., Wang X., *Some improvements of Ostrowski's method*, Applied Mathematics Letters, 23 (2010) 92-96.
- [64] Kung H.T., Traub J.F., *Optimal order of one-point and multipoint iteration*, Journal ACM, 21 (1974) 643-651.
- [65] Liu L., Wang X., *Eighth-order methods with high efficiency index for solving nonlinear equations*, Applied Mathematics and Computation, 215 (2010) 3449-3454.
- [66] Martínez E., Cordero A., Hueso J., Torregrosa J.R., *Una nueva familia de métodos iterativos para ecuaciones lineales*, XXI Congreso de Ecuaciones Diferenciales y Aplicaciones, XI Congreso de Matemática Aplicada Ciudad Real, 21-25 septiembre 2009 (pp. 1-8).
- [67] Neta B., *A sixth order family of methods for nonlinear equations*, International Journal of Computational Mathematics, 7 (1979) 157-161.
- [68] Nedzhibov G.H., *A family of multi-point iterative methods for solving systems of nonlinear equations*, Journal of Computational and Applied Mathematics, 222 (2008), 244-250.
- [69] Noor M.A., Waseem M. *Some iterative methods free from second derivatives for nonlinear equations*, Computers and Mathematics with Applications, 57 (2009) 101-106.
- [70] Ortega J.M., *Numerical Analysis. A Second Course*, SIAM, 1990.
- [71] Ortega J.M., Rheinboldt W.C., *Iterative Solutions of Nonlinear Equations in Several Variables*, Academic Press, Inc., 1970.
- [72] Ostrowski A.M., *Solutions of equations and systems of equations*, Academic Press, New York-London, 1966.
- [73] Özban A.Y., *Some new variants of Newton's method*, Applied Mathematics Letters, 17 (2004) 677-682.
- [74] Potra F.A., Pták V., *Nondiscrete induction and iterative processes*, Research Notes in Mathematics, 103 (1984) Pitman, Boston.
- [75] Romero N.A., Ezquerro J.A., Hernandez M.A., *Aproximación de soluciones de algunas ecuaciones integrales de Hammerstein mediante métodos iterativos tipo Newton*, XXI Congreso de ecuaciones diferenciales y aplicaciones, Universidad de Castilla-La Mancha, (2009)
- [76] Rheinboldt, W.C., *Methods for solving systems of nonlinear equations*, SIAM, Philadelphia, PA, USA, 1974.
- [77] Sharma J.R., Guha R.K., *A family of modified Ostrowski methods with accelerated sixth order convergence*, Applied Mathematics and Computation, 190 (2007) 111-115.
- [78] Sharma J.R., Sharma R., *A new family of modified Ostrowski's methods with accelerated eighth order convergence*, Numerical Algorithms, 54 (2010) 445-458.
- [79] Sevilla M.J., *Mecánica Celeste Clásica*, Instituto de Astronomía y Geodesia. Facultad de Ciencias Matemáticas. Universidad Complutense de Madrid, 1989.
- [80] Tierney J.A., *Differential equations*, Allyn and Bacon, Boston, MA, USA, 1985.
- [81] Traub J.F., *Iterative methods for the solution of equations*, Chelsea Publishing Company, New York, 1982.
- [82] Weerakoon S., Fernando T.G.I., *A variant of Newton's method with accelerated third-order convergence*, Applied Mathematics Letters, 13(8) (2000) 87-93.
- [83] Wang X., Liu L., *Modified Ostrowski's method with eighth-order convergence and high efficiency index*, Applied Mathematics Letters, 2010.

-
- [84] Wu X., *A new continuation Newton-like method and its deformation*, Applied Mathematics and Computation, 112 (2000) 75-78.
- [85] Wu X., Wu H., *On class of quadratic convergence iteration formulae without derivatives*, Applied Mathematics and Computation, 107 (2000) 77-80.
- [86] Wu X., Fu D., *New high order iteration methods without employing derivatives for solving nonlinear equations*, Computers and Mathematics with Applications, 41 (2001) 489-495.
- [87] Wu X., *Note on the improvement of Newton's method for system of nonlinear equation*, Applied Mathematics and Computation, 189 (2007) 1476-1479.