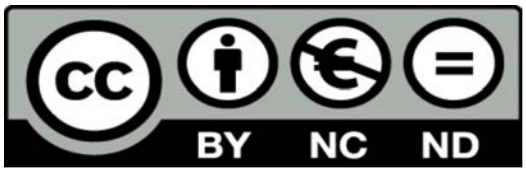
UNIVERSIDAD DE LA RIOJA

TESIS DOCTORAL

Título
Estudio de métodos tipo secante: convergencia, estabilidad y accesibilidad
Autor/es
Alejandro Moysi Amieva
Director/es
Angel Alberto Magreñán Ruiz
Facultad
Facultad de Ciencia y Tecnología
Titulación
Departamento
Matemáticas y Computación
Curso Académico
2024-2025



publicaciones.unirioja.es
E-mail: publicaciones@unirioja.es

TESIS DOCTORAL
2025

Programa de Doctorado en Matemáticas y Computación

ESTUDIO DE MÉTODOS TIPO SECANTE: CONVERGENCIA, ESTABILIDAD Y ACCESIBILIDAD.

Alejandro Moysi Amieva

Director: Ángel Alberto Magreñán Ruiz

Prólogo

Esta tesis doctoral se ha realizado siguiendo la normativa de la Universidad de La Rioja dentro del «Programa de Doctorado en Matemáticas y Computación» (Plan 782D, según el R.D. 99/2011), y esta memoria está estructura en apartados tal como se detalla en la citada normativa.

Los capítulos presentados abordan diversos aspectos teóricos y aplicados de los métodos iterativos tipo secante, que desempeñan un papel crucial en la resolución de ecuaciones no lineales en distintos campos de las matemáticas y la computación científica. Estos métodos, al no requerir la evaluación explícita de derivadas, son particularmente útiles en contextos donde el cálculo de derivadas es costoso o incluso impracticable. A lo largo de estos capítulos, se analiza la evolución y optimización de estos métodos iterativos, desde sus formulaciones clásicas hasta versiones mejoradas con mayor eficiencia y accesibilidad.

El segundo capítulo, "Sobre el conjunto de aproximaciones iniciales para el método de la secante", se centra en la importancia de la selección de valores iniciales en la aplicabilidad del método de la secante. Se parte del hecho de que, aunque este método ofrece una convergencia superlineal y evita la necesidad de calcular derivadas, su desempeño puede ser sensible a la elección de los puntos iniciales. Para abordar este desafío, se introduce una familia uniparamétrica de métodos iterativos que interpolan entre el método de la secante y el método de Newton, permitiendo mejorar la velocidad de convergencia sin comprometer la eficiencia computacional. Además, se propone una estrategia basada en la descomposición del operador no lineal en dos componentes: una diferenciable y otra continua pero no diferenciable. A través de esta descomposición, se consigue una mejor accesibilidad del método, ampliando la región de convergencia de las iteraciones. El capítulo también incluye un análisis teórico de la convergencia local, definiendo condiciones que garantizan la existencia de una bola de convergencia y permitiendo estimar el radio de esta región en función de las propiedades del operador.

El tercer capítulo, "Una mejora significativa de una familia de métodos tipo secante", amplía el estudio previo al introducir una familia biparamétrica de métodos iterativos. Partiendo de la familia uniparamétrica analizada en el primer capítulo, se desarrolla una versión que alcanza convergencia cuadrática, combinando la flexibilidad del método de la secante con la robustez del método de Newton. En particular, se demuestra que el uso de diferencias divididas simétricas permite mejorar la aproximación de la derivada del operador involucrado, lo que se traduce en un incremento en la velocidad de convergencia sin elevar significativamente el costo computacional. Se realiza un análisis numérico detallado de la eficiencia compu-

tacional, estableciendo comparaciones con métodos previos y mostrando que la nueva familia de métodos iterativos ofrece ventajas sustanciales en términos de accesibilidad y estabilidad. Además, se examina el comportamiento dinámico de estos métodos mediante el estudio de cuencas de atracción en el plano complejo, proporcionando una representación visual de las regiones de convergencia y permitiendo evaluar experimentalmente la accesibilidad de los métodos iterativos propuestos.

El cuarto capítulo, "Un procedimiento para obtener convergencia cuadrática a partir del método de la secante", se enfoca en el desarrollo de una familia de métodos iterativos con memoria, diseñados para mantener la eficiencia computacional del método de la secante pero alcanzar un segundo orden de convergencia. Se explora en profundidad la relación entre la elección de parámetros en la iteración y la velocidad de convergencia, analizando cómo la utilización de diferencias divididas mejoradas permite recuperar la convergencia cuadrática característica del método de Newton. En este contexto, se realiza una comparación detallada entre los métodos iterativos desarrollados y el método de la secante clásico, concluyendo que la familia propuesta no solo mejora la velocidad de convergencia, sino que también ofrece una mejor accesibilidad a las soluciones, ampliando la región de aproximaciones iniciales que garantizan la convergencia. Además del análisis teórico, el capítulo incluye experimentos numéricos en ecuaciones no lineales e integrales de tipo Hammerstein, lo que permite validar en la práctica las ventajas del enfoque propuesto.

En conjunto, estos capítulos ofrecen un estudio exhaustivo sobre la evolución de los métodos tipo secante, desde su formulación original hasta propuestas avanzadas que combinan flexibilidad, eficiencia y robustez. Se abordan tanto aspectos teóricos, como el análisis de convergencia y accesibilidad, como aplicaciones prácticas, incluyendo su implementación en problemas concretos. Además, se proporciona un marco metodológico para evaluar la eficacia de estos métodos, considerando métricas como la eficiencia computacional y la accesibilidad dinámica. A lo largo del texto, se demuestra que la optimización de estos métodos no solo es relevante desde un punto de vista teórico, sino que tiene un impacto significativo en la resolución efectiva de problemas matemáticos y computacionales de gran interés

Agradecimientos

Realizar esta tesis ha sido un desafío que ha requerido un gran esfuerzo para compaginar la vida laboral, académica y familiar. Ha supuesto un verdadero reto, lleno de momentos de aprendizaje, dedicación y sacrificio. Sin embargo, llegar a su culminación no habría sido posible sin el apoyo y la confianza de mi director de tesis, Ángel Alberto Magreñán Ruiz, cuya orientación y paciencia han sido fundamentales en cada etapa del proceso. Su guía me ha permitido avanzar con seguridad y determinación, incluso en los momentos más difíciles.

También quiero expresar mi más profundo agradecimiento a Miguel Ángel Hernández Verón y José Antonio Ezquerro Fernández, con quienes he tenido la oportunidad de trabajar de manera cercana a lo largo de esta etapa. Gracias a su conocimiento, apoyo y colaboración, no solo he podido completar esta tesis, sino también desarrollar publicaciones que han enriquecido mi formación investigadora, lo que ha sido clave para alcanzar este logro.

Por último, pero no menos importante, quiero dedicar unas palabras a mi familia, especialmente a mi esposa y a mi hija. Han sido un pilar inquebrantable en este camino, brindándome su amor, paciencia y comprensión en cada momento. Su apoyo incondicional me ha dado la fuerza necesaria para seguir adelante, y esta meta alcanzada es también, en gran parte, gracias a ellas. A todos, mi más sincero agradecimiento

Logroño, a 31 de marzo de 2025

A mi mujer y mi hija

Índice general

Índice de figuras

Índice de tablas

Resumen

El presente trabajo de investigación se enmarca dentro del estudio y desarrollo de métodos numéricos iterativos para la resolución de ecuaciones no lineales. En particular, se centra en el análisis y optimización de los métodos tipo secante, con el propósito de mejorar su eficiencia y estabilidad en comparación con las técnicas tradicionales. La motivación de este estudio radica en la importancia que tiene la resolución de ecuaciones no lineales en múltiples campos científicos y de la ingeniería, donde frecuentemente no es posible obtener soluciones exactas y es necesario recurrir a algoritmos iterativos.

En este contexto, la tesis desarrolla una nueva familia de métodos iterativos basada en la reformulación del método de la secante mediante la introducción de parámetros de control y diferencias divididas simétricas. Se demuestra que estas modificaciones permiten mejorar la velocidad de convergencia sin aumentar el costo computacional. A través de un análisis teórico riguroso, se estudia la convergencia local de los métodos propuestos, estableciendo condiciones bajo las cuales el esquema iterativo alcanza un orden de convergencia cuadrático. Además, se explora el comportamiento dinámico de estos métodos mediante herramientas de análisis complejas, identificando la existencia de cuencas de atracción y zonas de inestabilidad.

El desarrollo de estos métodos ha requerido la implementación de diversas simulaciones numéricas en Python, utilizando bibliotecas especializadas para la resolución de ecuaciones no lineales. A través de estas simulaciones, se ha comparado el rendimiento de los métodos propuestos frente a los esquemas clásicos, verificando que las nuevas formulaciones presentan una mayor robustez y accesibilidad. Se han aplicado estos algoritmos en la resolución de ecuaciones integrales y problemas en los que la función objetivo no es diferenciable, evidenciando su utilidad en contextos más amplios.

Desde una perspectiva teórica, el trabajo también aborda la relación entre los métodos tipo secante y los esquemas iterativos basados en operadores en espacios de Banach. Se estudia cómo la descomposición del operador en términos de una parte diferenciable y otra continua no diferenciable permite mejorar la aplicabilidad de estos algoritmos en problemas donde los métodos clásicos presentan limitaciones. Este enfoque ha permitido ampliar la aplicabilidad de los métodos tipo secante a ecuaciones más generales, manteniendo una estructura computacional sencilla y eficiente.

A lo largo del desarrollo de la tesis, se ha prestado especial atención al análisis gráfico del comportamiento de los métodos iterativos, explorando sus propiedades

dinámicas en el plano complejo. Se han utilizado representaciones visuales para comprender la distribución de las cuencas de atracción de las raíces y los efectos de los parámetros introducidos en la estabilidad de las iteraciones. Estas representaciones han sido clave para caracterizar el impacto de las mejoras propuestas y han permitido establecer una comparación visual con otros métodos iterativos.

Los resultados obtenidos evidencian que la optimización de los métodos tipo secante mediante ajustes paramétricos y técnicas de diferencias divididas permite obtener esquemas más eficientes, con mayores regiones de convergencia y mejor comportamiento dinámico. La implementación de estos métodos en entornos computacionales confirma su utilidad práctica y su potencial para ser utilizados en problemas matemáticos y de ingeniería en los que la resolución de ecuaciones no lineales es un desafío recurrente.

Capítulo 1

Introducción

Algunos de los problemas más estudiados en Matemáticas, Química, Física o Ingeniería pueden expresarse como ecuaciones no lineales [18, 137]. Ejemplos comunes incluyen ecuaciones integrales no lineales en contextos como la teoría de la elasticidad, electrostática, teoría del potencial, problemas de transferencia de calor radiativo y sistemas de ecuaciones no lineales derivados de la discretización de problemas. La principal dificultad a la hora de encontrar las soluciones de este tipo de ecuaciones es que en muchas ocasiones no se pueden encontrar de forma directa y por ello se debe recurrir a métodos iterativos.

En este capítulo se va a presentar la gran mayoría de las definiciones, resultados o aplicaciones, que se van a usar en los capítulos posteriores. Para ello, se lleva a cabo un análisis detallado de diversos métodos iterativos empleados en la resolución de ecuaciones. Además, se profundiza en los principios teóricos que sustentan las técnicas que se van a usar, permitiendo una comprensión más amplia de su aplicabilidad y de los contextos en los que resultan más eficaces.

La estructura del capítulo es la siguiente:

- En la Sección 1.1, se muestran los conceptos básicos de los métodos que utilizan derivadas.
- En la Sección 1.2, se presenta los conceptos necesarios sobre los métodos libres de derivadas, ya que la tesis se centra en este tipo de algoritmos.
- En la Sección 1.3, se presentan los preliminares de la dinámica compleja, donde diferentes conceptos se ilustrarán con ejemplos.
- En la Sección 1.4, se presentan los conceptos previos necesarios de la dinámica real a partir de los presentados en la sección anterior.
- En la Sección 1.5, se presentan los conceptos relacionados con la convergencia en espacios de Banach.
- En la Sección 1.6, se justifica la elección del lenguaje Python, para depués en la Sección 1.7 presentar los códigos usados en este capítulo introductorio.
- Por último, en la Sección 1.8, se presentan la organización de los capítulos de esta tesis.

1.1. Métodos iterativos con derivadas

En esta sección se introducirán algunos de los métodos que requieren de la utilización de derivadas y que han sido objeto de numerosos estudios en los últimos años [37, 51, 66, 67, 69, 74, 94, 95, 96, 104, 118, 164, 169, 170], así como las referencias que aparecen en dichos estudios.

1.1.1. El método de Newton

El método de Newton, también conocido como método de Newton-Raphson, es un proceso iterativo utilizado para aproximar soluciones x^* de ecuaciones no lineales de la forma

$$f(x) = 0. \tag{1.1.1}$$

A lo largo de este capítulo se considerará la función

$$f : [a, b] \longrightarrow \mathbb{R},$$

aunque cabe señalar que la metodología puede ser extendida al caso de funciones con variable compleja.

El método parte de una estimación inicial

$$x_0$$

cercana a la raíz

$$x^*$$

y a partir de ella define una sucesión que tiene como objetivo acercarse progresivamente a dicha solución buscada.

La sucesión generada por el método de Newton-Raphson viene dada por la siguiente fórmula iterativa

$$\begin{cases} \text{Dado } x_0 \in [a, b], \\ \\ x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}, \qquad n \geq 0, \end{cases} \tag{1.1.2}$$

donde n representa el número de iteración. Esto es que la primera iteración se denominará como x_1, la segunda x_2 y así sucesivamente.

Cada paso del método busca mejorar la aproximación, siempre que la derivada $f'(x_n)$ no sea cero, lo que garantiza la validez de la fórmula. Para lograr la convergencia de las iteraciones hacia la solución buscada se deben imponer una serie de condiciones que se presentarán durante el capítulo.

A continuación, se explorará el desarrollo histórico de este procedimiento y algunas de las técnicas alternativas que han llevado a su formulación definitiva, proporcionando un contexto más amplio de su evolución.

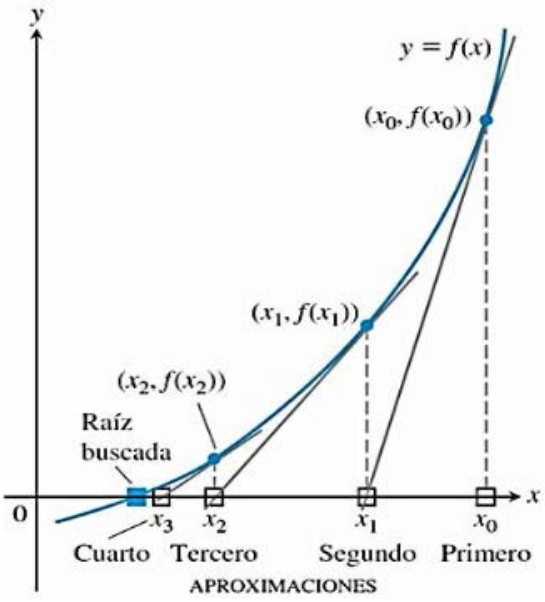


Figura 1.1: Descripción gráfica del método de Newton-Raphson

A continuación, se presentan los principales hitos históricos que condujeron a la construcción del método de Newton en la forma en que lo conocemos hoy en día:

- François **Viète** (1540-1603), considerado uno de los fundadores del álgebra moderna, fue un precursor clave en la resolución de ecuaciones. En el año 1600, logró obtener de manera general las raíces positivas de polinomios de grados 2 a 6, sentando las bases para desarrollos posteriores en métodos de solución de ecuaciones.

- Isaac **Newton** (1643-1727) basó su trabajo en el de Viète para desarrollar un método dirigido a resolver ecuaciones no lineales. En 1669, mejoró su enfoque al introducir una técnica para linealizar polinomios. Por ejemplo, para la ecuación $f(x) = x^3 + 2x - 5 = 0$, observó que la parte entera de la raíz sería 2, eligiendo así $x_0 = 2$ como primera aproximación. Luego, evaluó la función en $x = 2 + p$, obteniendo:

$$p^3 + 6p^2 + 10p - 1 = 0 \longleftrightarrow 10p - 1 \approx 0 \longleftrightarrow p \approx \frac{1}{10},$$

  lo que llevó a la siguiente aproximación $x_1 = 2 + \frac{1}{10} = 2.1$. Repitió el proceso evaluando en $x = 2.1 + q$, resultando en:

$$q^3 + \frac{63}{10}q^2 + \frac{61}{1000} = 0 \longleftrightarrow \frac{1123}{100}q + \frac{61}{1000} \approx 0 \longleftrightarrow q \approx -0.0054,$$

  y estableció $x_2 = 2.1 - 0.0054 = 2.0946$. A pesar de no usar explícitamente derivadas, puede observarse que

$$p = x_1 - x_0 = -\frac{f(x_0)}{f'(x_0)},$$

$$q = x_2 - x_1 = -\frac{f(x_1)}{f'(x_1)}.$$

- Joseph **Raphson** (1648-1715) simplificó el procedimiento en 1690 e introdujo la noción de iteración. Para ecuaciones de la forma $f(a) = a^3 - ba - c = 0$, propuso que si g es una aproximación inicial, una mejora viene dada por $g+x$, con
$$x = \frac{c + bg - g^3}{3g^2 - b},$$
equivalente a $g + x = g - \frac{f(g)}{f'(g)}$.

- Thomas **Simpson** (1710-1761) amplió el método al incorporar el cálculo diferencial y resolvió por primera vez una ecuación no polinómica:
$$\sqrt{1-x} + \sqrt{1-x^2} + \sqrt{1-3x^3} - 2 = 0.$$
Además, extendió su aplicación a sistemas de dos ecuaciones con dos incógnitas:
$$\begin{cases} y + \sqrt{y^2 - x^2} = 0 \\ x + \sqrt{yy + x} - 12 = 0. \end{cases}$$

Con las bases del método de Newton establecidas, diversos investigadores contribuyeron a profundizar en sus características y aplicaciones:

- Joseph-Louis **Lagrange** (1736-1813) en 1808 señaló la necesidad de una buena estimación inicial para la eficacia del método, y destacó problemas al tratar raíces múltiples o cercanas.

- Jean Baptiste Joseph **Fourier** (1768-1830) modernizó la notación del método en 1818 y analizó, por primera vez, la velocidad de convergencia.

- Augustin-Louis **Cauchy** (1789-1857) en 1829 proporcionó condiciones para las derivadas f' y f'' que aseguran la convergencia del método.

- Arthur **Cayley** (1821-1895) en 1879 introdujo el concepto de cuenca de atracción en el plano complejo, estudiando regiones de convergencia para el polinomio $p(z) = (z - a)(z - b)$, con $a \neq b$ y $a, b \in \bar{\mathbb{C}}$.

- Henry Burchard **Fine** (1858-1928) y Albert Arnold **Bennett** (1888-1971) retomaron el estudio de sistemas de ecuaciones no lineales, publicando en 1916 trabajos relevantes que no se habían abordado desde Simpson.

- Leonid Vitálievich **Kantoróvich** (1912-1986) propuso una extensión del método de Newton para resolver ecuaciones funcionales en espacios de Banach, abriendo nuevas posibilidades en análisis funcional.

La expresión que define el método de Newton puede derivarse utilizando diversos enfoques. A continuación, se presenta la deducción del algoritmo del método de Newton partiendo del desarrollo de Taylor, lo cual proporciona una base matemática sólida para entender su funcionamiento.

Sea $x_0 \in [a, b]$ una estimación inicial de la solución de la ecuación (1.1.1), a la cual nos referiremos como x^*. El objetivo es encontrar un término corrector F tal que $x_0 + h$ sea la raíz buscada, es decir, se cumpla

$$f(x_0 + h) = f(x^*) = 0.$$

Si desarrollamos $f(x_0 + h)$ en serie de Taylor alrededor de x_0, se obtiene la siguiente expresión:

$$0 = f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{h^2}{2!} f''(x_0) + \cdots . \tag{1.1.3}$$

La solución exacta de esta ecuación proporcionaría el término corrector F para el cual $x_0 + h = x^*$. Sin embargo, para simplificar el cálculo, se puede considerar una versión linealizada de la ecuación 1.1.3, que consiste en tomar únicamente los términos lineales:

$$f(x_0) + hf'(x_0) = 0.$$

De esta ecuación linealizada, se obtiene de manera sencilla una aproximación para el corrector:

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

Aunque este valor de F no representa necesariamente la solución exacta de la ecuación original 1.1.3, en muchas situaciones puede servir como una buena aproximación, siempre que se cumplan ciertas condiciones de suavidad y proximidad de x_0 a la raíz x^*. Sumando x_0 a ambos lados de la ecuación, se obtiene una nueva aproximación de la raíz:

$$x_0 + h = x_0 - \frac{f(x_0)}{f'(x_0)},$$

lo cual ofrece una mejora respecto a la aproximación inicial x_0. Si denominamos $x_1 = x_0 + h$, podemos reescribir la expresión como

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

Repitiendo este proceso iterativamente, se obtiene la fórmula general del método de Newton (1.1.2), la cual permite refinar sucesivamente la estimación de x^* mediante iteraciones adicionales.

Cabe destacar que existen otras formas de deducir la expresión del método de Newton. Por ejemplo, se puede obtener a partir de su construcción geométrica, conocida como el método de la tangente, que se basa en aproximar la función con una recta tangente en cada paso. Otra opción es derivar el método utilizando la técnica de iteración del punto fijo, lo cual también ofrece una interpretación interesante desde el punto de vista de análisis numérico.
A continuación, se presenta una implementación en Python del método para la ecuación $x^2 - x - 2$:

$$x_{n+1} = x_n - \frac{x_n^2 - x_n - 2}{2x_n - 1}.$$

```
# -*- coding: utf-8 -*-
    #  f, funcion a evaluar
    #  dfdx, derivada respecto a x de la funcion
    #  x, valor inicial, semilla
    #  eps, error admisible
def newton(f, dfdx, x, eps):
    i = 0
    while abs(f(x)) > eps and i < 100:
        try:
            x = x - float(f(x)) / dfdx(x)
            i += 1
            #Se muestra evolucion iteraciones intermedias
            print('Iteracion: %i, Valor x: %.16f' % (i, x))
        except ZeroDivisionError:
            print('Error, derivada cero para x= ', x)
            break

    # Si no encuentra solucion
    if abs(f(x)) > eps:
        i = -1
    return x, i

# Funcion y su derivada, definida inline
f = lambda x: x**2-x-2
dfdx = lambda x: 2*x-1


# Valores iniciales y error
x0 = -4
eps=1E-15

# Ejecutar el metodo de Newton
solucion, itera = newton(f, dfdx, x0, eps)

# Imprimir resultados
if itera > 0:  # Solucion encontrada
    print('N  iteraciones: %i' % itera)
    print('Solucion: %.16f' % solucion)
else:
    print('Solucion no encontrada.')
```

Las raices de este polinomio son bien conocidas, $x = -1$ y $x = 2$. Se muestran los resultados obtenidos para varios puntos iniciales:

- Valor inicial, $x_0 = -4$, tolerancia= 10^{-15}.
  - Número de iteraciones hasta converger= 6.
  - Raíz a la que converge $x = -1$.

Figura 1.2: Iteraciones del método de Newton aplicado al polinomio $x^2 - x - 2$ tomando como valor inicial $x_0 = -4$.

- Valor inicial, $x_0 = -0.5$, tolerancia= 10^{-15}.
  - Número de iteraciones hasta converger= 5.
  - Raíz a la que converge $x = -1$.

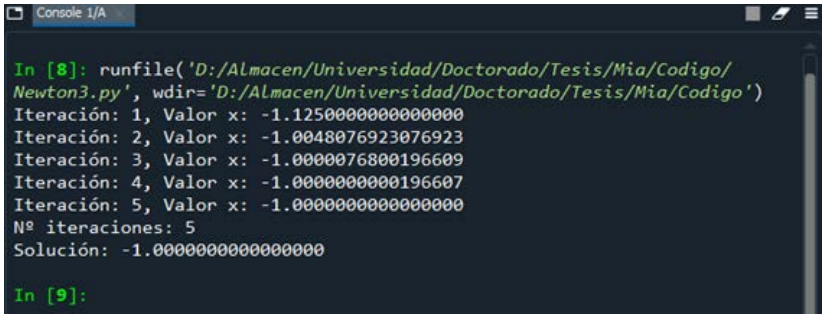


Figura 1.3: Iteraciones del método de Newton aplicado al polinomio $x^2 - x - 2$ tomando como valor inicial $x_0 = -0.5$.

- Valor inicial, $x_0 = 0$, tolerancia= 10^{-15}.
  - Número de iteraciones hasta converger= 6.
  - Raíz a la que converge $x = -1$.

Figura 1.4: Iteraciones del método de Newton aplicado al polinomio $x^2 - x - 2$ tomando como valor inicial $x_0 = 0$.

- Valor inicial, $x_0 = 1$, tolerancia= 10^{-15}.
  - Número de iteraciones hasta converger= 6.
  - Raíz a la que converge $x = 2$.

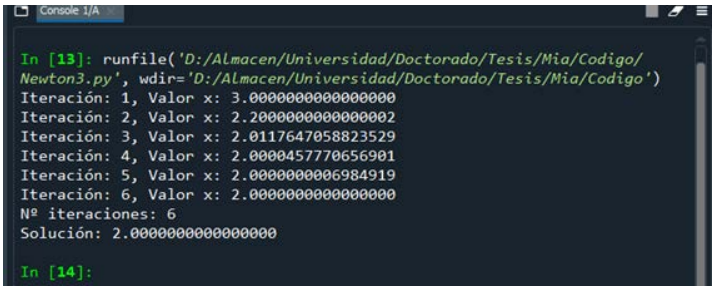


Figura 1.5: Iteraciones del método de Newton aplicado al polinomio $x^2 - x - 2$ tomando como valor inicial $x_0 = 1$.

- Valor inicial, $x_0 = 5$, tolerancia= 10^{-15}.
  - Número de iteraciones hasta converger= 6.
  - Raíz a la que converge $x = 2$.

Figura 1.6: Iteraciones del método de Newton aplicado al polinomio $x^2 - x - 2$ tomando como valor inicial $x_0 = 5$.

1.1.2. El método de Chebyshev

Otro método iterativo para aproximar la solución de ecuaciones no lineales (1.1.1) es el método de Chebyshev, que se caracteriza por su convergencia cúbica. El algoritmo correspondiente se define como

$$\begin{cases} \text{Dado } x_0 \in [a,b], \\ x_{n+1} = x_n - \left(1 + \dfrac{1}{2} L_f(x_n)\right) \dfrac{f(x_n)}{f'(x_n)}, \quad n \geq 0, \end{cases} \tag{1.1.4}$$

donde $L_f(x)$ está dado por

$$L_f(x) = \frac{f(x) f''(x)}{f'(x)^2}.$$

La cantidad $L_f(x)$ se conoce como el grado de convexidad logarítmico de la función f [88]. Este término adicional corrige la aproximación considerando la curvatura local de la función, lo que permite alcanzar una velocidad de convergencia cúbica, es decir, el error se reduce de forma proporcional al cubo del error en cada iteración, siempre que la función sea suficientemente regular y la aproximación inicial sea adecuada.

El algoritmo del método de Chebyshev fue propuesto inicialmente por el francés Irénée-Jules Bienaymé (1796-1878), pero fue el matemático ruso Pafnuty Chebyshev (1821-1894) quien lo formalizó y demostró en 1867, motivo por el cual lleva su nombre. Chebyshev es ampliamente reconocido por sus contribuciones a diversas ramas de las matemáticas, incluyendo la teoría de la probabilidad, el análisis numérico y la teoría de aproximación de funciones. Sus trabajos en polinomios ortogonales, que también llevan su nombre, han sido fundamentales para el desarrollo de métodos numéricos en la resolución de ecuaciones no lineales y problemas de optimización.

De manera similar al método de Newton, existen varios enfoques para deducir el algoritmo del método de Chebyshev. Algunos de estos procedimientos incluyen la interpolación cuadrática inversa, el método de las parábolas tangentes, la interpretación geométrica, la interpolación exponencial o los métodos desarrollados por Obreshkov. A continuación, se detalla el procedimiento basado en la interpolación cuadrática inversa. Para una explicación más completa sobre las otras técnicas, se puede consultar [47].

Dada la función $y = f(x)$, sea $x = \phi(y)$ la función inversa de f. Si aproximamos $\phi(y)$ en un punto y_0, entonces tenemos:

$$\phi(y) \approx \phi(y_0) + \phi'(y_0)(y - y_0) + \frac{1}{2}\phi''(y_0)(y - y_0)^2.$$

Buscar una raíz $x^* \in [a,b]$ de la ecuación $f(x) = 0$ equivale a encontrar la imagen del cero a través de la función ϕ, es decir, $\phi(0)$. Si x_0 es un valor cercano a $x^* \in [a,b]$ y $f(x_0) = y_0$, entonces se tiene:

$$\phi(0) \approx \phi(y_0) - \phi'(y_0) y_0 + \frac{1}{2}\phi''(y_0) y_0^2.$$

A partir de esta aproximación, se puede deducir una nueva estimación de x^*, denotada por x_1:

$$x_1 = x_0 - \phi'(y_0) f(x_0) + \frac{1}{2}\phi''(y_0) f(x_0)^2. \tag{1.1.5}$$

Para proceder, calculamos $\phi'(y)$ y $\phi''(y)$ y los sustituimos en la ecuación anterior. Sabiendo que $x = \phi(y)$, $y' = f'(x)$ y $y'' = f''(x)$, y aplicando el teorema de la función inversa, obtenemos:

$$\phi'(y) = \frac{dx}{dy} = \frac{1}{\frac{dy}{dx}} = \frac{1}{y'} = \frac{1}{f'(x)},$$

y

$$\phi''(y) = \frac{d\phi'(y)}{dy} = \frac{d\left(\frac{1}{f'(x)}\right)}{dx} \cdot \frac{dx}{dy} = -\frac{y''}{(y')^3}.$$

Sustituyendo $\phi'(y)$ y $\phi''(y)$ en (1.1.5), se obtiene:

$$x_1 = x_0 - \left(1 + \frac{1}{2}L_f(x_0)\right)\frac{f(x_0)}{f'(x_0)}.$$

Repitiendo el procedimiento para obtener x_1, se obtiene la siguiente aproximación x_2. Por inducción, podemos generalizar:

$$\begin{cases} \text{Dado } x_0 \in [a,b], \\ x_{n+1} = x_n - \left(1 + \frac{1}{2}L_f(x_n)\right)\frac{f(x_n)}{f'(x_n)}, \quad n \geq 0, \end{cases} \tag{1.1.6}$$

que corresponde al algoritmo del método de Chebyshev descrito en (1.1.4).

El método de Chebyshev es especialmente útil cuando se busca una convergencia más rápida que la del método de Newton, ya que alcanza la convergencia cúbica bajo condiciones adecuadas. Sin embargo, es más complejo en su implementación debido a la necesidad de calcular la segunda derivada de la función. En comparación con el método de Halley, ambos comparten la misma tasa de convergencia, pero difieren en la forma de la corrección aplicada.

El método se usa comúnmente en problemas donde se necesita alta precisión con un número limitado de iteraciones, siendo una herramienta útil en física, ingeniería y análisis numérico.

A continuación, se presenta una implementación en Python del método para la ecuación $x^2 - x - 2$:

$$x_{n+1} = x_n - \left(1 + \frac{1}{2} \cdot \frac{(x_n^2 - x_n - 2) \cdot 2}{(2x_n - 1)^2}\right)\frac{x_n^2 - x_n - 2}{2x_n - 1}.$$

```
# -*- coding: utf-8 -*-
#  f: funcion a evaluar
#  dfdx: derivada respecto a x de la funcion
#  d2fdx2: segunda derivada de la funcion
#  x: valor inicial, semilla
#  eps: error admisible
def chebyshev(f, dfdx, d2fdx2, x, eps):
    # Metodo de Chebyshev
    i = 0
    while abs(f(x)) > eps and i < 100:
        try:
            # Metodo de Chebyshev
            fx = f(x)
            fpx = dfdx(x)
            fppx = d2fdx2(x)
            x = x - (fx / fpx) * (1 + 0.5 * (fx * fppx / (fpx**2)))
            i += 1
            #Se muestra evolucion iteraciones intermedias
            print('Iteracion: %i, Valor x: %.16f' % (i, x))
        except ZeroDivisionError:
            print('Error, derivada cero para x= ', x)
            break

    # Si no encuentra solucion
    if abs(f(x)) > eps:
        i = -1
    return x, i

# Funcion, su primera y segunda derivada definidas inline
f = lambda x: x**2 - x - 2
dfdx = lambda x: 2*x - 1
d2fdx2 = lambda x: 2

# Valores iniciales y error
x0 = -4
eps = 1E-15

# Ejecutar el metodo de Chebyshev
solucion, itera = chebyshev(f, dfdx, d2fdx2, x0, eps)

# Imprimir resultados
if itera > 0:  # Solucion encontrada
    print('N  iteraciones: %i' % itera)
    print('Solucion: %.16f' % solucion)
else:
    print('Solucion no encontrada.')
```

Las raices de este polinomio son bien conocidas, $x = -1$ y $x = 2$. Se muestran los resultados obtenidos para varios puntos iniciales:

- Valor inicial, $x_0 = -4$, tolerancia= 10^{-15}.
  - Número de iteraciones hasta converger= 4.
  - Raíz a la que converge $x = -1$.

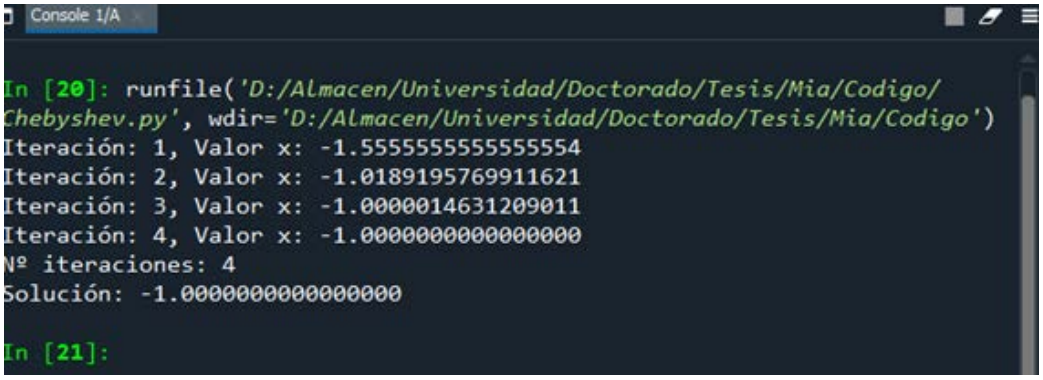


Figura 1.7: Iteraciones del método de Chebyshev aplicado al polinomio $x^2 - x - 2$ tomando como valor inicial $x_0 = -4$.

- Valor inicial, $x_0 = -0.5$, tolerancia= 10^{-15}.
  - Número de iteraciones hasta converger= 4.
  - Raíz a la que converge $x = -1$.

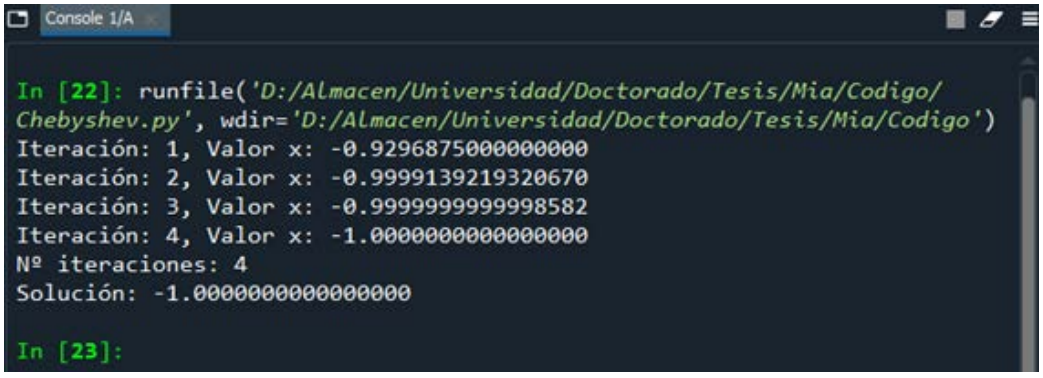


Figura 1.8: Iteraciones del método de Chebyshev aplicado al polinomio $x^2 - x - 2$ tomando como valor inicial $x_0 = -0.5$.

- Valor inicial, $x_0 = 0$, tolerancia= 10^{-15}.
  - Número de iteraciones hasta converger= 1.
  - Raíz a la que converge $x = 2$.

Figura 1.9: Iteraciones del método de Chebyshev aplicado al polinomio $x^2 - x - 2$ tomando como valor inicial $x_0 = 0$.

- Valor inicial, $x_0 = 1$, tolerancia= 10^{-15}.
  - Número de iteraciones hasta converger= 1.
  - Raíz a la que converge $x = 2$.

Figura 1.10: Iteraciones del método de Chebyshev aplicado al polinomio $x^2 - x - 2$ tomando como valor inicial $x_0 = 1$.

- Valor inicial, $x_0 = 5$, tolerancia= 10^{-15}.
  - Número de iteraciones hasta converger= 4.
  - Raíz a la que converge $x = 2$.

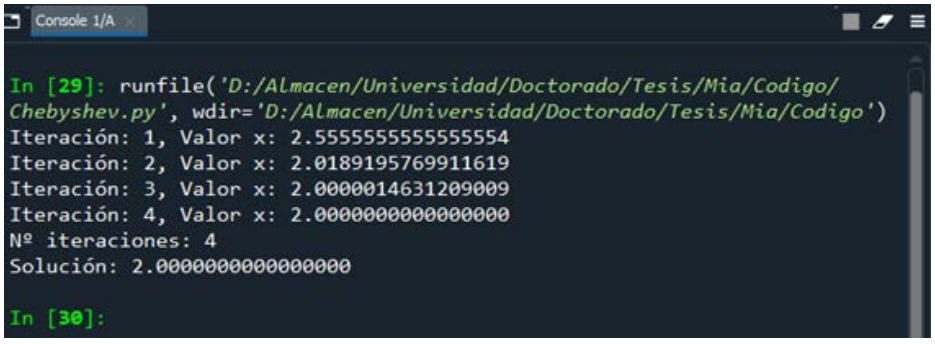


Figura 1.11: Iteraciones del método de Chebyshev aplicado al polinomio $x^2 - x - 2$ tomando como valor inicial $x_0 = 5$.

1.1.3. El método de Halley

El método de Halley es un procedimiento iterativo de convergencia cúbica utilizado para resolver ecuaciones no lineales de la forma (1.1.1). El algoritmo del método se define como:

$$\begin{cases} \text{Dado } x_0 \in [a,b], \\ x_{n+1} = x_n - \dfrac{2}{2 - L_f(x_n)} \dfrac{f(x_n)}{f'(x_n)}, \qquad n \geq 0, \end{cases} \tag{1.1.7}$$

donde

$$L_f(x) = \frac{f(x)f''(x)}{f'(x)^2}.$$

El término $L_f(x)$ es conocido como el grado de convexidad logarítmico de la función f, y permite ajustar la iteración para obtener una mejor aproximación de la raíz. El método de Halley alcanza una velocidad de convergencia cúbica, lo que significa que el error en la aproximación disminuye proporcionalmente al cubo del error en cada iteración, siempre que las condiciones iniciales sean adecuadas y la función sea lo suficientemente regular.

El método de Halley lleva el nombre del astrónomo y matemático inglés Edmond Halley (1656-1742), famoso por predecir la órbita del cometa que también lleva su nombre. Halley propuso este método en 1694, extendiendo el trabajo de Newton sobre la resolución de ecuaciones no lineales. Aunque Halley es más conocido por su trabajo en astronomía, también realizó importantes contribuciones a las matemáticas, incluyendo el desarrollo de técnicas numéricas avanzadas para su época.

La deducción del método se basa en mejorar la corrección proporcionada por el método de Newton utilizando información adicional de la segunda derivada de la función. Esta mejora en la precisión permite alcanzar la convergencia cúbica, superando la convergencia cuadrática del método de Newton.

Al igual que en los métodos anteriores, existen diversos enfoques para deducir el algoritmo del método de Halley. A continuación, se presenta su deducción utilizando el desarrollo de Taylor.

Consideremos el polinomio de Taylor de segundo grado centrado en x_n:

$$f(x) = f(x_n) + f'(x_n)(x - x_n) + \frac{f''(x_n)}{2!}(x - x_n)^2,$$

donde x_n es una aproximación de la raíz de $f(x) = 0$. El objetivo es determinar el punto x_{n+1} donde la gráfica de $f(x)$ corta el eje X. Para ello, se resuelve la ecuación

$$0 = f(x_n) + f'(x_n)(x_{n+1} - x_n) + \frac{f''(x_n)}{2!}(x_{n+1} - x_n)^2.$$

Factorizando $(x_{n+1} - x_n)$, se obtiene

$$0 = f(x_n) + (x_{n+1} - x_n)\left(f'(x_n) + \frac{1}{2}f''(x_n)(x_{n+1} - x_n)\right),$$

lo que, al simplificar, lleva a

$$x_{n+1} - x_n = \frac{f(x_n)}{f'(x_n) + \frac{1}{2} f''(x_n)(x_{n+1} - x_n)}.$$

Aproximando la diferencia $x_{n+1} - x_n$ por la corrección de Newton $\frac{-f(x_n)}{f'(x_n)}$, se obtiene:

$$x_{n+1} = x_n - \frac{2f(x_n) f'(x_n)}{2f'(x_n)^2 - f(x_n) f''(x_n)} = x_n - \frac{2}{2 - L_f(x_n)} \frac{f(x_n)}{f'(x_n)},$$

lo cual corresponde a la función de iteración del método de Halley descrita en (1.1.7).

Además del enfoque basado en el desarrollo de Taylor, existen otras formas de deducir el método de Halley. Entre ellas se incluyen:

- **Construcción geométrica**: Utilizando la geometría de la función y las tangentes, se puede llegar a una fórmula equivalente.
- **Método de las hipérbolas tangentes**: Basado en la construcción de curvas hiperbólicas que aproximan la función.
- **Interpolación exponencial y racional**: Técnicas que emplean funciones exponenciales o fracciones racionales para obtener aproximaciones mejoradas.
- **Fracciones continuas**: Utilizando desarrollos en fracciones continuas para refinar la aproximación.
- **Aceleración del método de Newton**: Consiste en modificar la corrección de Newton para alcanzar la convergencia cúbica.
- **Iteraciones de Laguerre**: Métodos derivados que aprovechan la estructura del polinomio de Laguerre para iterar hacia la solución.

El método de Halley se destaca por su convergencia cúbica, lo que lo hace más eficiente que el método de Newton cuando se busca alta precisión en pocas iteraciones. A diferencia del método de Chebyshev, que también alcanza convergencia cúbica, el método de Halley aplica una corrección diferente, lo que puede hacerlo más adecuado en situaciones donde la función presenta curvaturas abruptas. Sin embargo, su implementación es más compleja, ya que requiere el cálculo de la segunda derivada de la función.

Este método se utiliza en diversas áreas como la física, la ingeniería y las ciencias computacionales, donde se necesitan soluciones precisas y eficientes para ecuaciones no lineales.

A continuación, se presenta una implementación en Python del método para la ecuación $x^2 - x - 2$:

$$x_{n+1} = x_n - \frac{2}{2 - \frac{(x_n^2 - x_n - 2) \cdot 2}{(2x_n - 1)^2}} \cdot \frac{x_n^2 - x_n - 2}{2x_n - 1}.$$

```
# -*- coding: utf-8 -*-
#  f: funcion a evaluar
#  dfdx: derivada respecto a x de la funcion
#  d2fdx2: segunda derivada de la funcion
#  x: valor inicial, semilla
#  eps: error admisible
def halley(f, dfdx, d2fdx2, x, eps):
    # Metodo de Halley
    i = 0
    while abs(f(x)) > eps and i < 100:
        try:
            # Calcular L_f(x)
            fx = f(x)
            fpx = dfdx(x)
            fppx = d2fdx2(x)
            Lf = (fx * fppx) / (fpx**2)

            # Metodo de Halley
            x = x - (2 / (2 - Lf)) * (fx / fpx)
            i += 1
            #Se muestra evolucion iteraciones intermedias
            print('Iteraci n: %i, Valor x: %.16f' % (i, x))
        except ZeroDivisionError:
            print('Error, divisi n por cero para x= ', x)
            break

    # Si no encuentra solucion
    if abs(f(x)) > eps:
        i = -1
    return x, i

# Funcion, su primera y segunda derivada definidas inline
f = lambda x: x**2 - x - 2
dfdx = lambda x: 2*x - 1
d2fdx2 = lambda x: 2

# Valores iniciales y error
x0 = 5
eps = 1E-15

# Ejecutar el metodo de Halley
solucion, itera = halley(f, dfdx, d2fdx2, x0, eps)

# Imprimir resultados
if itera > 0:  # Solucion encontrada
    print('N  iteraciones: %i' % itera)
    print('Solucion: %.16f' % solucion)
else:
    print('Solucion no encontrada.')
```

Las raices de este polinomio son bien conocidas, $x = -1$ y $x = 2$. Se muestran los resultados obtenidos para varios puntos iniciales:

- Valor inicial, $x_0 = -4$, tolerancia= 10^{-15}.
  - Número de iteraciones hasta converger= 4.
  - Raíz a la que converge $x = -1$.

Figura 1.12: Iteraciones del método de Halley aplicado al polinomio $x^2 - x - 2$ tomando como valor inicial $x_0 = -4$.

- Valor inicial, $x_0 = -0.5$, tolerancia= 10^{-15}.
  - Número de iteraciones hasta converger= 3.
  - Raíz a la que converge $x = -1$.

Figura 1.13: Iteraciones del método de Halley aplicado al polinomio $x^2 - x - 2$ tomando como valor inicial $x_0 = -0.5$.

- Valor inicial, $x_0 = 0$, tolerancia= 10^{-15}.
  - Número de iteraciones hasta converger= 4.
  - Raíz a la que converge $x = -1$.

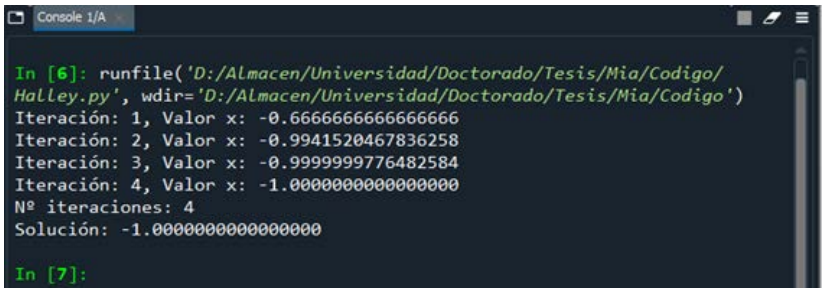


Figura 1.14: Iteraciones del método de Halley aplicado al polinomio $x^2 - x - 2$ tomando como valor inicial $x_0 = 0$.

- Valor inicial, $x_0 = 1$, tolerancia= 10^{-15}.
  - Número de iteraciones hasta converger= 4.
  - Raíz a la que converge $x = 2$.

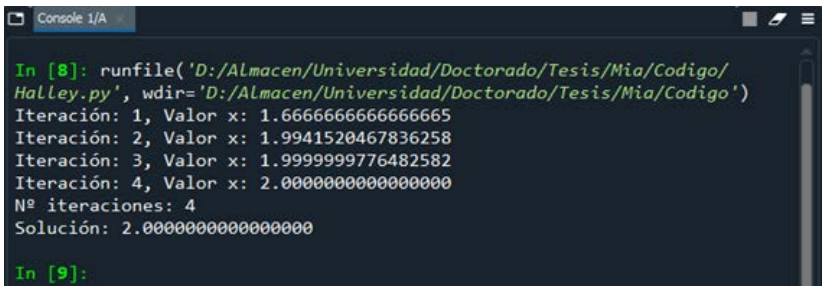


Figura 1.15: Iteraciones del método de Halley aplicado al polinomio $x^2 - x - 2$ tomando como valor inicial $x_0 = 1$.

- Valor inicial, $x_0 = 5$, tolerancia= 10^{-15}.
  - Número de iteraciones hasta converger= 4.
  - Raíz a la que converge $x = 2$.

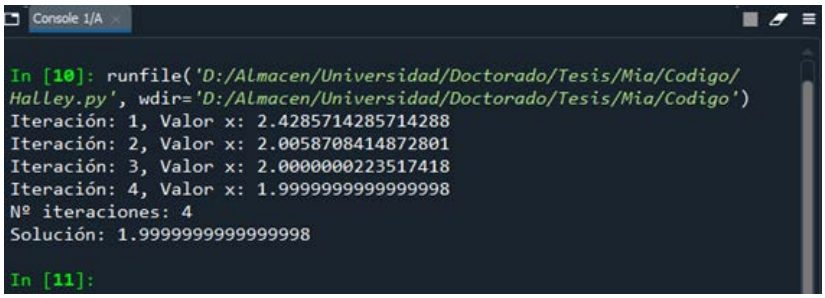


Figura 1.16: Iteraciones del método de Halley aplicado al polinomio $x^2 - x - 2$ tomando como valor inicial $x_0 = 5$.

1.1.4. El método de Super Halley

El método de Super Halley es una extensión del método de Halley que también es un procedimiento iterativo de convergencia cúbica, diseñado para resolver ecuaciones no lineales de la forma

$$f(x) = 0. \tag{1.1.8}$$

Este método mejora el ajuste de la iteración al incorporar un término adicional relacionado con la convexidad de la función, logrando mayor estabilidad en ciertos escenarios. La fórmula iterativa del método de Super Halley se define como:

$$\begin{cases} \text{Dado } x_0 \in [a,b], \\ x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)} \left(1 - \dfrac{L_f(x_n)}{2} \dfrac{f(x_n)}{f'(x_n)}\right), \qquad n \geq 0, \end{cases} \tag{1.1.9}$$

donde

$$L_f(x) = \frac{f(x)f''(x)}{f'(x)^2}.$$

El término $L_f(x)$, conocido como el grado de convexidad logarítmico, también juega un papel central en este método al ajustar la iteración para mejorar la aproximación a la raíz. Como en el método de Halley, la convergencia del método de Super Halley es cúbica, siempre que las condiciones iniciales y la regularidad de la función sean adecuadas.

El método de Super Halley fue propuesto como una alternativa más robusta al método de Halley, en contextos donde el comportamiento de la función puede comprometer la estabilidad numérica del método original. Al introducir un término de corrección adicional, Super Halley busca mitigar los efectos de valores extremos de la convexidad logarítmica que podrían ralentizar la convergencia o producir oscilaciones.

Deducción del método

La deducción del método se basa en el desarrollo de Taylor de segundo grado, al igual que en el método de Halley, pero incluye un término adicional en la corrección de Newton para capturar mejor el comportamiento local de la función. A continuación, se describe brevemente este razonamiento.

Consideremos el polinomio de Taylor de segundo grado centrado en x_n:

$$f(x) = f(x_n) + f'(x_n)(x - x_n) + \frac{f''(x_n)}{2!}(x - x_n)^2,$$

y una corrección iterativa ajustada que mejora la convergencia cúbica. Para determinar x_{n+1}, se modifica la ecuación iterativa de Halley al introducir un factor de corrección adicional:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \left(1 - \frac{L_f(x_n)}{2} \frac{f(x_n)}{f'(x_n)}\right).$$

Esta fórmula permite mejorar la convergencia en casos donde el método de Halley puede ser inestable debido a altos valores de $L_f(x)$ o variaciones abruptas en la función.

Diferencias con el método de Halley

El método de Super Halley comparte muchas características con el método de Halley, como la convergencia cúbica y la dependencia de la segunda derivada. Sin embargo, la principal diferencia radica en la corrección adicional basada en $L_f(x_n)$, que hace al método más adecuado en problemas donde la función tiene cambios significativos en su curvatura. Esto lo convierte en una herramienta valiosa en aplicaciones como:

- **Optimización no lineal**: Donde la forma de la función objetivo puede ser compleja y presentar múltiples curvaturas.
- **Resolución de ecuaciones no lineales en sistemas dinámicos**: Donde la estabilidad es crítica.
- **Modelado en física y ciencias computacionales**: Especialmente en modelos con términos de alta no linealidad.

Consideraciones prácticas

Aunque el método de Super Halley es más robusto que el de Halley, su implementación puede ser ligeramente más costosa en términos computacionales debido al término adicional en la iteración. Requiere evaluar no solo las derivadas primera y segunda de la función, sino también realizar operaciones adicionales para calcular la corrección ajustada. Sin embargo, en muchos casos, este costo adicional se compensa con una mayor estabilidad y eficiencia global.

Como el método de Halley y otros métodos cúbicos, el éxito del método de Super Halley depende de contar con buenas condiciones iniciales (x_0), una función bien definida y derivadas continuas en el intervalo de interés. Es menos propenso a fallar por puntos de inflexión cercanos a la raíz, pero sigue siendo sensible a singularidades en $f'(x)$.

Este método representa un refinamiento adicional dentro de la familia de métodos iterativos basados en Newton, ofreciendo una poderosa herramienta para problemas que requieren alta precisión y estabilidad numérica.

A continuación, se presenta una implementación en Python del método para la ecuación $x^2 - x - 2$:

$$x_{n+1} = x_n - \frac{x_n^2 - x_n - 2}{2x_n - 1}\left(1 - \frac{\frac{(x_n^2 - x_n - 2)\cdot 2}{(2x_n-1)^2}}{2} \cdot \frac{x_n^2 - x_n - 2}{2x_n - 1}\right).$$

```
# -*- coding: utf-8 -*-
#  f: funcion a evaluar
#  dfdx: derivada respecto a x de la funcion
#  d2fdx2: segunda derivada de la funcion
#  x: valor inicial, semilla
#  eps: error admisible
def superHalley(f, dfdx, d2fdx2, x, eps):
    # Metodo de Super Halley
    i = 0
    while abs(f(x)) > eps and i < 100:
        try:
            # Calcular L_f(x)
            fx = f(x)
            fpx = dfdx(x)
            fppx = d2fdx2(x)
            Lf = (fx * fppx) / (fpx**2)

            # M todo de Super Halley
            x = x - (fx / fpx) * (1 - (Lf / 2) * (fx / fpx))
            i += 1
            #Se muestra evolucion iteraciones intermedias
            print('Iteraci n: %i, Valor x: %.16f' % (i, x))
        except ZeroDivisionError:
            print('Error, division por cero para x= ', x)
            break

    # Si no encuentra solucion
    if abs(f(x)) > eps:
        i = -1
    return x, i

# Funcion, su primera y segunda derivada definidas inline
f = lambda x: x**2 - x - 2
dfdx = lambda x: 2*x - 1
d2fdx2 = lambda x: 2

# Valores iniciales y error
x0 = -4
eps = 1E-15

# Ejecutar el metodo de Super Halley
solucion, itera = superHalley(f, dfdx, d2fdx2, x0, eps)

# Imprimir resultados
if itera > 0:  # Solucion encontrada
    print('N  iteraciones: %i' % itera)
    print('Solucion: %.16f' % solucion)
else:
    print('Solucion no encontrada.')
```

Las raices de este polinomio son bien conocidas, $x = -1$ y $x = 2$. Se muestran los resultados obtenidos para varios puntos iniciales:

- Valor inicial, $x_0 = -4$, tolerancia= 10^{-15}.
  - Número de iteraciones hasta converger= 5.
  - Raíz a la que converge $x = -1$.

Figura 1.17: Iteraciones del método de Super Halley aplicado al polinomio x^2-x-2 tomando como valor inicial $x_0 = -4$.

- Valor inicial, $x_0 = -0.5$, tolerancia= 10^{-15}.
  - Número de iteraciones hasta converger= 5.
  - Raíz a la que converge $x = -1$.

Figura 1.18: Iteraciones del método de Super Halley aplicado al polinomio x^2-x-2 tomando como valor inicial $x_0 = -0.5$.

- Valor inicial, $x_0 = 0$, tolerancia= 10^{-15}.
  - Número de iteraciones hasta converger= 6.
  - Raíz a la que converge $x = 2$.

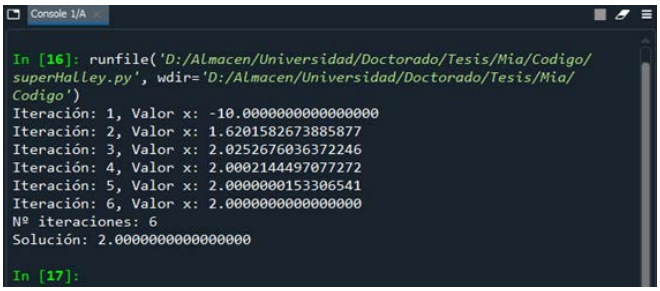


Figura 1.19: Iteraciones del método de Super Halley aplicado al polinomio x^2-x-2 tomando como valor inicial $x_0 = 0$.

- Valor inicial, $x_0 = 1$, tolerancia$= 10^{-15}$.
  - Número de iteraciones hasta converger$= 6$.
  - Raíz a la que converge $x = -1$.

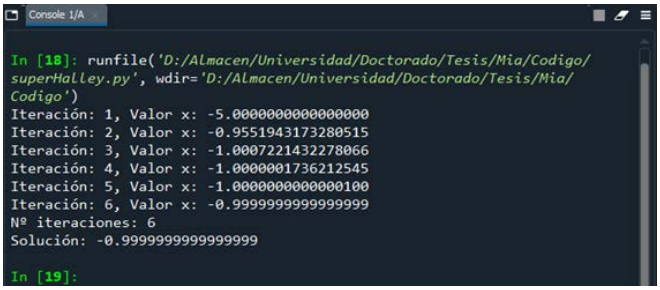


Figura 1.20: Iteraciones del método de Super Halley aplicado al polinomio x^2-x-2 tomando como valor inicial $x_0 = 1$.

- Valor inicial, $x_0 = 5$, tolerancia$= 10^{-15}$.
  - Número de iteraciones hasta converger$= 4$.
  - Raíz a la que converge $x = 2$.

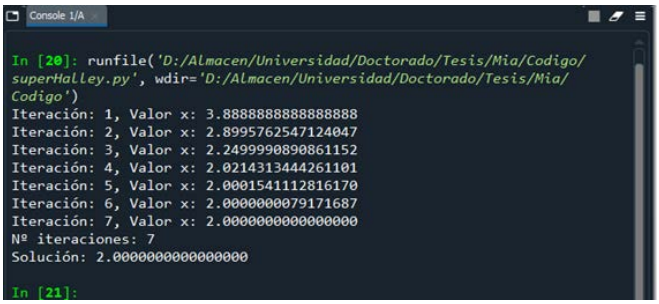


Figura 1.21: Iteraciones del método de Super Halley aplicado al polinomio x^2-x-2 tomando como valor inicial $x_0 = 5$.

1.2. Métodos libres de derivadas

Todos los métodos de la sección anterior tienen un mismo inconveniente que es la evaluación de derivadas. Esto limita el número de problemas a los que se pueden aplicar y que se puedan resolver ya que si el operador involucrado en el problema tiene una derivada que es muy costosa de calcular o incluso no es derivable no se van a poder aplicar. Para paliar estos problemas de aplicabilidad existen los métodos libres de derivadas que utilizan aproximaciones a las derivadas. Entre los métodos libres de derivadas se encuentran los que se describen a continuación.

1.2.1. El método la Secante

El método de la secante es un procedimiento iterativo utilizado para aproximar soluciones x^* de ecuaciones no lineales de la forma (1.1.1) y que ha sido estudiado por numeroso investigadores [84, 144, 148, 158]. A diferencia del método de Newton-Raphson, el método de la secante no requiere la evaluación de la derivada de la función en cada iteración. En su lugar, se emplea una aproximación basada en dos puntos cercanos a la raíz. Este enfoque puede ser ventajoso en situaciones donde el cálculo de la derivada es complicado o la función no es diferenciable en algunos puntos.

La sucesión generada por el método de la secante está dada por

$$\begin{cases} \text{Dado } x_0,\, x_1 \in [a,b], \\ x_{n+1} = x_n - \dfrac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}, \qquad n \geq 1, \end{cases} \tag{1.2.1}$$

donde n representa el número de iteración. Este método busca mejorar la aproximación de la raíz utilizando los dos últimos valores de la sucesión para construir una línea secante que aproxima la función.

El método de la secante tiene sus raíces en técnicas de aproximación que se remontan a la antigüedad. Los matemáticos babilonios y griegos ya utilizaban procedimientos similares para encontrar raíces cuadradas y resolver ecuaciones cúbicas. En el siglo XVII, el método fue redescubierto y formalizado en un contexto más moderno de análisis numérico.

En 1690, Joseph Raphson introdujo un método de iteración que fue una simplificación del método de Newton, lo que llevó a desarrollos paralelos del método de la secante. Aunque Raphson no utilizó el enfoque de la secante explícitamente, su trabajo influyó en métodos posteriores que evitaban el cálculo de derivadas.

El método de la secante puede derivarse a partir de una aproximación lineal de la función $f(x)$. Supongamos que se tienen dos puntos iniciales, x_0 y x_1, donde $f(x_0)$ y $f(x_1)$ son los valores de la función en esos puntos. La línea secante que pasa por estos dos puntos se puede expresar como

$$\frac{f(x_1) - f(x_0)}{x_1 - x_0} \approx f'(x),$$

aproximando la derivada de $f(x)$. La idea principal es utilizar esta pendiente aproximada para iterar hacia la raíz. La ecuación de la recta que pasa por los puntos

$(x_0, f(x_0))$ y $(x_1, f(x_1))$ es

$$f(x_1) \approx f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0).$$

Para hallar la intersección de esta línea con el eje X, se resuelve

$$0 = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x_1 - x_0),$$

lo que lleva a

$$x_2 = x_1 - \frac{f(x_1)(x_1 - x_0)}{f(x_1) - f(x_0)}.$$

Repitiendo el procedimiento para obtener $x_2, x_3, \ldots$, se obtiene la expresión general del método de la secante, tal como se muestra en (1.2.1).

Comparación con otros métodos

El método de la secante presenta una velocidad de convergencia de orden ≈ 1.618 (la razón áurea), lo que lo hace menos eficiente que el método de Newton-Raphson, cuya convergencia es cuadrática. Sin embargo, al no requerir el cálculo de la derivada en cada iteración, puede ser más conveniente en ciertos problemas prácticos.

Este método también se puede interpretar como un caso particular del método de interpolación lineal, y se utiliza frecuentemente como una alternativa a los métodos que requieren derivadas. Además, puede combinarse con otras técnicas, como el método de bisección, para mejorar la robustez del proceso iterativo.

Al igual que con otros métodos de aproximación de raíces, es importante contar con buenas estimaciones iniciales para garantizar la convergencia y evitar situaciones donde la secante se vuelva casi horizontal, lo que podría conducir a iteraciones divergentes.

A continuación, se presenta una implementación en Python del método para la ecuación $x^2 - x - 2$:

$$x_{n+1} = x_n - \frac{(x_n^2 - x_n - 2)(x_n - x_{n-1})}{(x_n^2 - x_n - 2) - (x_{n-1}^2 - x_{n-1} - 2)}.$$

```
# -*- coding: utf-8 -*-
    #  f, funcion a evaluar
    #  x0, x1 valores iniciales, semilla
    #  eps, error admisible

def secante(f, x0, x1, eps):
    # Metodo de la secante
    fx0 = f(x0)  # Evaluacion inicial
    fx1 = f(x1)
    iter = 0  # Contador de iteraciones
    while abs(fx1) > eps and iter < 100:
        try:
            denominador = float(fx1 - fx0) / (x1 - x0)
            x = x1 - float(fx1) / denominador
        except ZeroDivisionError:
            print('Division por cero en x= ', x)
            break  # Salir del bucle en caso de error
        x0, x1 = x1, x  # Actualizacion de valores
        fx0, fx1 = f(x0), f(x1)
        iter += 1
        #Se muestra evolucion iteraciones intermedias
        print('Iteraci n: %i, Valor x: %.16f' % (iter, x1))

    # Si no encuentra solucion
    if abs(fx1) > eps:
        iter = -1
    return x, iter

# Funcion, definida inline
f = lambda x: x**2-x-2

# Valores iniciales
x0 = -4
x1 = -0.5
eps=1E-15

# Llamada al metodo de la secante
solucion, itera = secante(f, x0, x1, eps)

# Resultados
if itera > 0:
    print('Numero de iteraciones: %d,' % itera)
    print('Una solucion es: %.16f' % solucion)
else:
    print('Solucion no encontrada')
```

Las raices de este polinomio son bien conocidas, $x = -1$ y $x = 2$. Se muestran los resultados obtenidos para varios puntos iniciales, en este caso se necesita partir de 2 puntos iniciales, se utilizará por tanto la combinación de los empleados en los ejemplos anteriores:

- Valores iniciales, $x_0 = -4$ y $x_1 = -0.5$, tolerancia= 10^{-15}
  - Número de iteraciones hasta converger= 7
  - Raíz a la que converge $x = -1$.

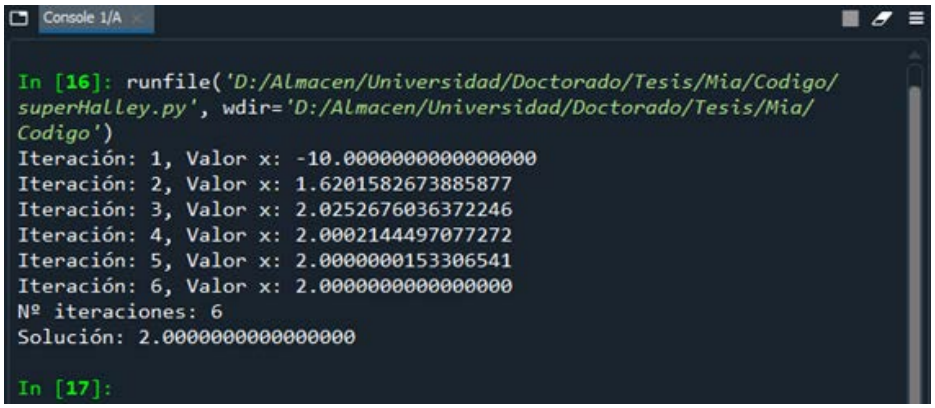


Figura 1.22: Iteraciones del método de la Secante aplicado al polinomio $x^2 - x - 2$ tomando como valores iniciales $x_0 = -4$ y $x_1 = -0.5$.

- Valores iniciales, $x_0 = -4$ y $x_1 = 0$, tolerancia= 10^{-15}
  - Número de iteraciones hasta converger= 8
  - Raíz a la que converge $x = -1$.

Figura 1.23: Iteraciones del método de la Secante aplicado al polinomio $x^2 - x - 2$ tomando como valores iniciales $x_0 = -4$ y $x_1 = 0$.

- Valores iniciales, $x_0 = -4$ y $x_1 = 1$, tolerancia= 10^{-15}
  - Número de iteraciones hasta converger= 11.
  - Raíz a la que converge $x = 2$.

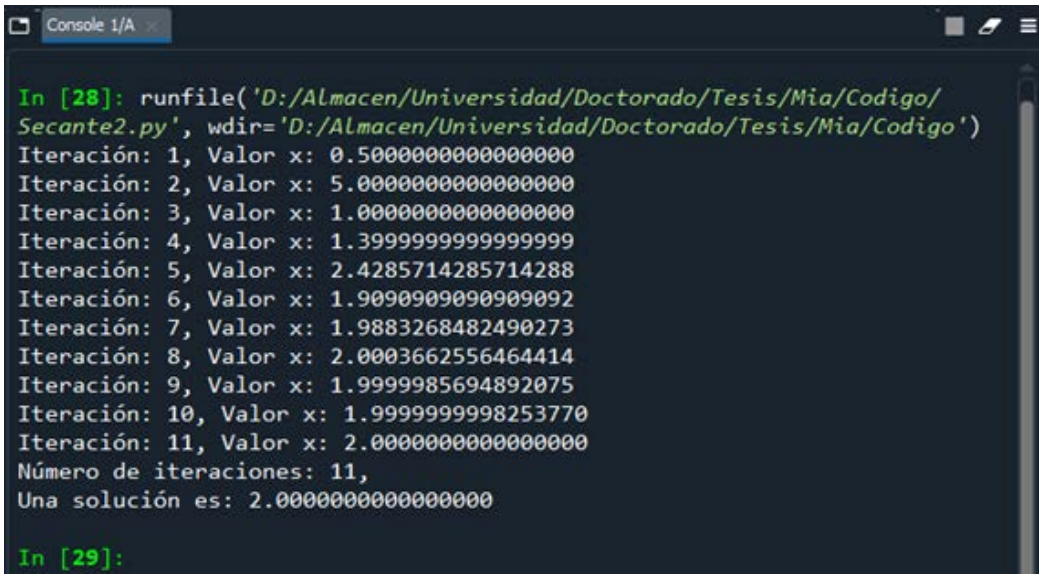


Figura 1.24: Iteraciones del método de la Secante aplicado al polinomio $x^2 - x - 2$ tomando como valores iniciales $x_0 = -4$ y $x_1 = 1$.

- Valores iniciales, $x_0 = -0.5$ y $x_1 = 0$, tolerancia= 10^{-15}
  - Número de iteraciones hasta converger= 8.
  - Raíz a la que converge $x = -1$.

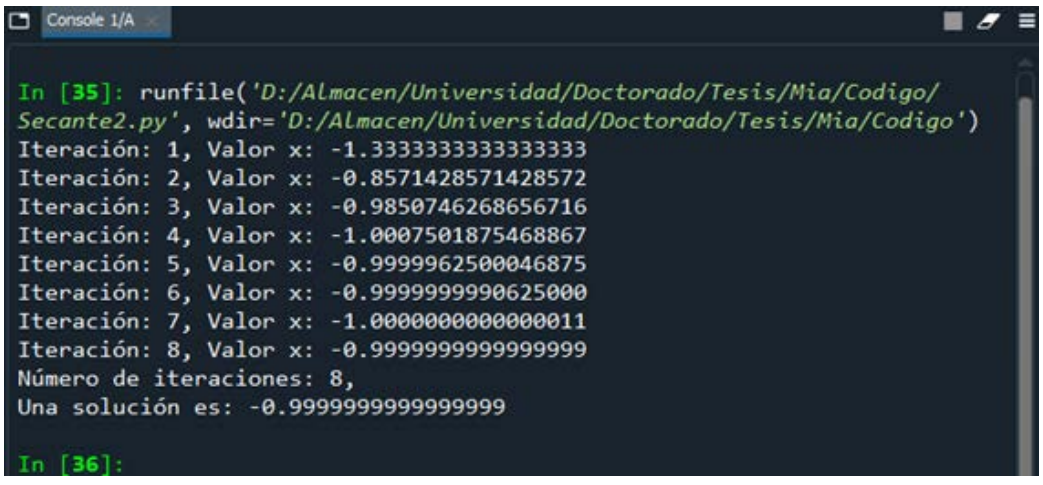


Figura 1.25: Iteraciones del método de la Secante aplicado al polinomio $x^2 - x - 2$ tomando como valores iniciales $x_0 = -0.5$ y $x_1 = 0$.

- Valores iniciales, $x_0 = -0.5$ y $x_1 = 1$, tolerancia= 10^{-15}
  - Número de iteraciones hasta converger= 11.
  - Raíz a la que converge $x = -1$.

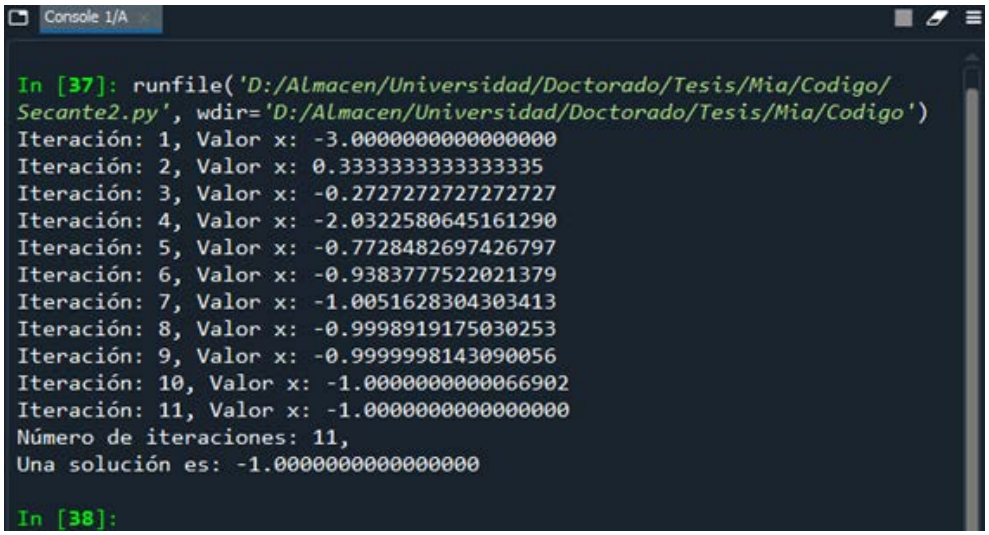


Figura 1.26: Iteraciones del método de la Secante aplicado al polinomio $x^2 - x - 2$ tomando como valores iniciales $x_0 = -0.5$ y $x_1 = 1$.

- Valores iniciales, $x_0 = -0.5$ y $x_1 = 5$, tolerancia= 10^{-15}
  - Número de iteraciones hasta converger= 15.
  - Raíz a la que converge $x = -1$.

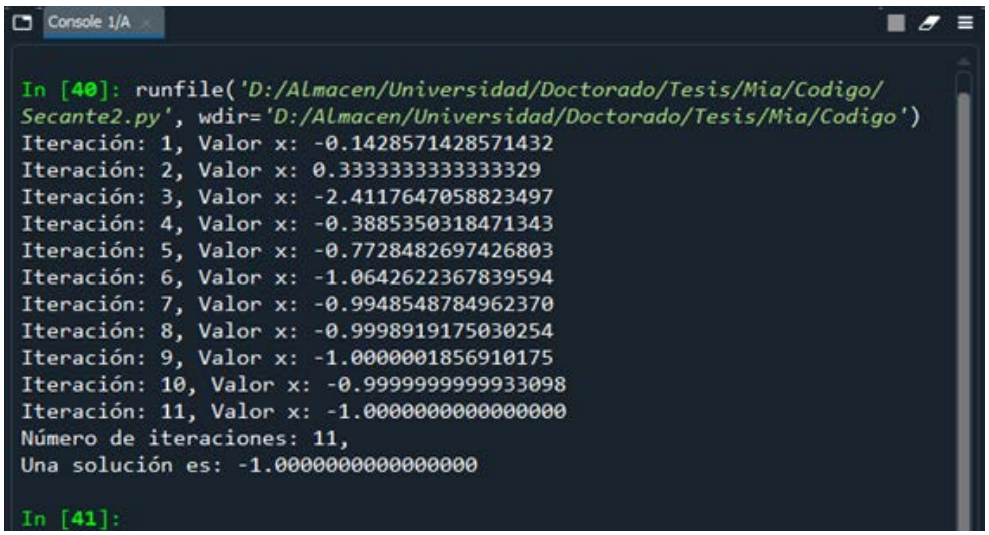


Figura 1.27: Iteraciones del método de la Secante aplicado al polinomio $x^2 - x - 2$ tomando como valores iniciales $x_0 = -0.5$ y $x_1 = 5$.

- Valores iniciales, $x_0 = 0$ y $x_1 = 5$, tolerancia= 10^{-15}
  - Número de iteraciones hasta converger= 11.
  - Raíz a la que converge $x = 2$.

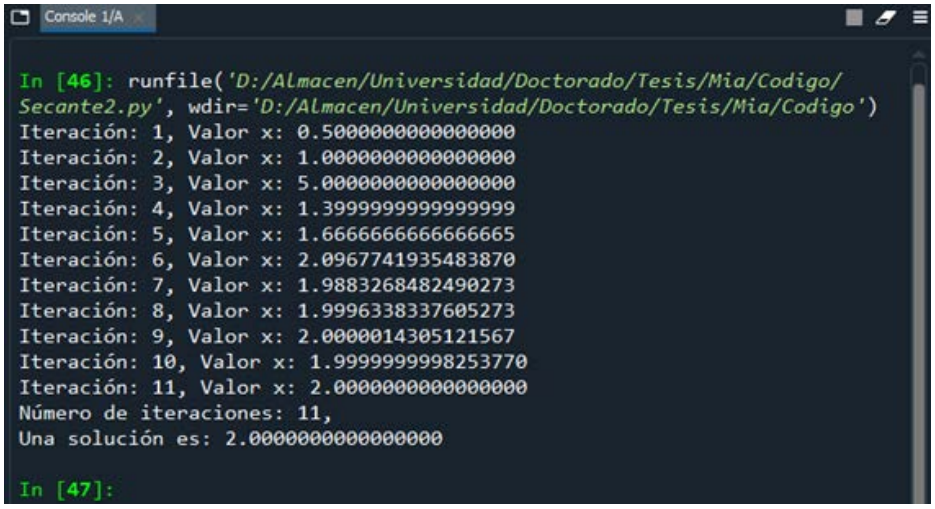


Figura 1.28: Iteraciones del método de la Secante aplicado al polinomio $x^2 - x - 2$ tomando como valores iniciales $x_0 = 0$ y $x_1 = 5$.

Valores iniciales, $x_0 = 1$ y $x_1 = 5$, tolerancia= 10^{-15}, Número de iteraciones hasta converger= 8 , raíz a la que converge $x = 2$.

- Valores iniciales, $x_0 = 1$ y $x_1 = 5$, tolerancia= 10^{-15}
  - Número de iteraciones hasta converger= 8.
  - Raíz a la que converge $x = 2$.

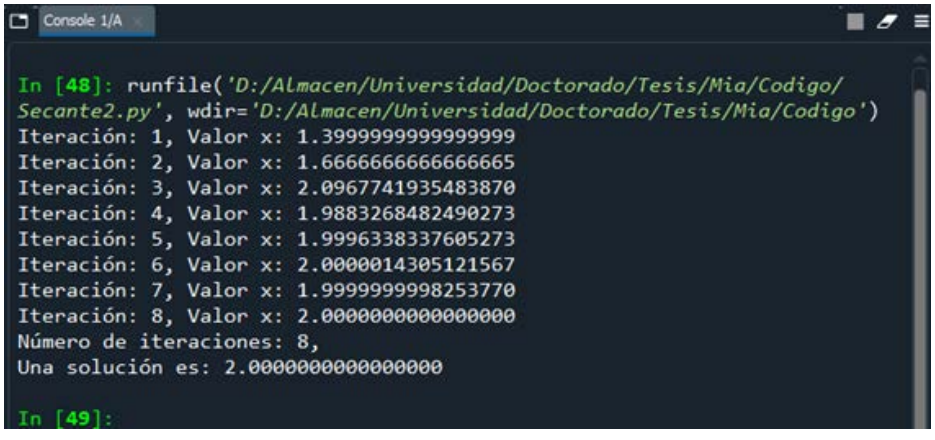


Figura 1.29: Iteraciones del método de la Secante aplicado al polinomio $x^2 - x - 2$ tomando como valores iniciales $x_0 = 1$ y $x_1 = 5$.

1.2.2. El método de Steffensen

El método de Steffensen es un procedimiento iterativo utilizado para aproximar soluciones x^* de ecuaciones no lineales de la forma (1.1.1). A diferencia del método de Newton-Raphson, el método de Steffensen no requiere el cálculo explícito de la derivada de la función. En su lugar, utiliza una técnica de aceleración de convergencia mediante la transformación de Aitken. Esta transformación permite que el método alcance convergencia cuadrática a partir de una función iterativa $g(x)$, donde $g(x) \approx x^*$, evitando así el uso de derivadas. Este enfoque es útil cuando la evaluación de derivadas es compleja o la función no es diferenciable en todos los puntos.

La sucesión generada por el método de Steffensen se define como:

$$\begin{cases} \text{Dado } x_0 \in [a,b], \\ x_{n+1} = x_n - \dfrac{[g(x_n) - x_n]^2}{g(g(x_n)) - 2g(x_n) + x_n}, \qquad n \geq 0, \end{cases} \tag{1.2.2}$$

donde n representa el número de iteración y $g(x) = x + f(x)$ en la mayoría de los casos prácticos. Este método mejora la aproximación de la raíz utilizando dos evaluaciones de la función g en cada iteración para obtener un paso que converge cuadráticamente hacia la solución.

El método de Steffensen tiene sus raíces en el trabajo de Alexander Aitken, quien desarrolló la idea de la transformación de Aitken en la década de 1920. Esta transformación permite reducir la cantidad de iteraciones necesarias para la convergencia en métodos iterativos, optimizando el proceso y minimizando el número de evaluaciones de funciones. El método de Steffensen es, de hecho, una aplicación directa de esta transformación en el contexto de aproximación de raíces.

Derivación del Método

El método de Steffensen se basa en una técnica de aceleración de la convergencia. Partiendo de un punto inicial x_0, el método construye una sucesión utilizando la función auxiliar $g(x)$ que define la transformación iterativa. Supongamos que $g(x)$ es tal que $g(x) \approx x^*$, entonces se define el siguiente paso iterativo:

$$x_{n+1} = x_n - \frac{[g(x_n) - x_n]^2}{g(g(x_n)) - 2g(x_n) + x_n}.$$

Este paso permite alcanzar convergencia cuadrática al calcular la corrección x_{n+1} sin necesidad de derivadas, aprovechando únicamente las evaluaciones de $g(x)$.

La transformación de Aitken proporciona la expresión x_{n+1} optimizando la sucesión de $g(x)$, de manera que:

$$x_{n+1} = x_n - \frac{[g(x_n) - x_n]^2}{g(g(x_n)) - 2g(x_n) + x_n},$$

reduciendo significativamente la cantidad de iteraciones necesarias para alcanzar la precisión deseada.

Comparación con otros métodos

El método de Steffensen tiene una velocidad de convergencia cuadrática, al igual que el método de Newton-Raphson, lo cual lo convierte en una alternativa atractiva. Sin embargo, mientras que el método de Newton-Raphson requiere el cálculo de la derivada de $f(x)$, el método de Steffensen se limita a evaluar la función iterativa $g(x)$, resultando en un método más adecuado para funciones complicadas o no diferenciables en ciertos puntos.

En comparación con el método de la secante, el método de Steffensen requiere una evaluación adicional de $g(x)$, pero ofrece una convergencia más rápida. Además, se puede considerar como una versión acelerada del método de punto fijo, y su uso es especialmente recomendable en problemas donde la velocidad de convergencia es crucial y el cálculo de derivadas es complejo o indeseable.

Al igual que otros métodos iterativos, es importante contar con una buena estimación inicial para asegurar la convergencia y evitar puntos donde el denominador en (1.2.2) se aproxime a cero, lo cual podría ocasionar resultados indeterminados o divergentes.

A continuación, se presenta una implementación en Python del método para la ecuación $x^2 - x - 2$:

$$x_{n+1} = x_n - \frac{\left[(x_n^2 - 1) - x_n\right]^2}{[(x_n^2 - 1)^2 - 1] - 2(x_n^2 - 1) + x_n}.$$

```
# -*- coding: utf-8 -*-
# Metodo de Steffensen
# f: funcion no lineal
# x0: valor inicial (semilla)
# eps: error tolerable

def steffensen(f, x0, eps):
    # Definimos g(x) = x + f(x)
    g = lambda x: x + f(x)
    iter = 0  # Contador de iteraciones

    while abs(f(x0)) > eps and iter < 100:
        try:
            gx = g(x0)  # g(x_n)
            ggx = g(gx)  # g(g(x_n))
            denominador = ggx - 2 * gx + x0
            # F rmula de Steffensen
            x = x0 - ((gx - x0) ** 2) / denominador
        except ZeroDivisionError:
            print('Divisi n por cero en x= ', x)
            break  # Salir del bucle en caso de error

        x0 = x  # Actualizacion
        iter += 1
        #Se muestra evolucion iteraciones intermedias
        print('Iteracion: %i, Valor x: %.16f' % (iter, x0))
    # Si no encuentra solucion dentro del error tolerable
    if abs(f(x0)) > eps:
        iter = -1
    return x0, iter


# Funcion, definida inline
f = lambda x: x**2 - x - 2

# Valores iniciales
x0 = -4 # Semilla inicial
eps = 1E-15  # Tolerancia al error

# Llamada al metodo
solucion, itera = steffensen(f, x0, eps)

# Resultados
if itera > 0:
    print(f'Numero de iteraciones: {itera}')
    print(f'Una solucion es: {solucion:.16f}')
else:
    print('Solucion no encontrada')
```

Las raices de este polinomio son bien conocidas, $x = -1$ y $x = 2$. Se muestran los resultados obtenidos para varios puntos iniciales:

- Valor inicial $x_0 = -4$ y tolerancia= 10^{-15}
  - No converge.

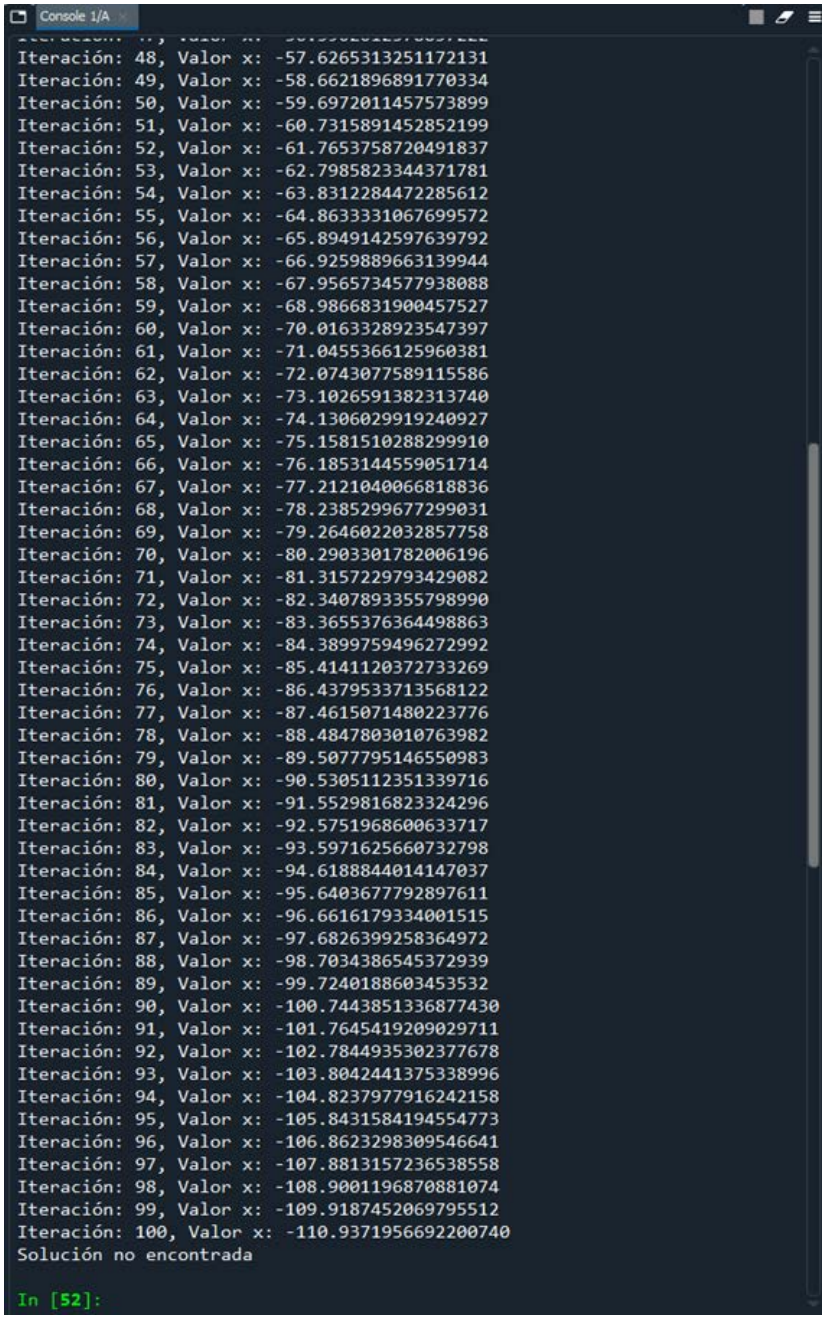


Figura 1.30: Iteraciones del método de Steffensen aplicado al polinomio $x^2 - x - 2$ tomando como valor inicial $x_0 = -4$.

- Valor inicial $x_0 = -0.5$ y tolerancia= 10^{-15}
  - Número de iteraciones hasta converger= 5.
  - Raíz a la que converge $x = -1$.

Figura 1.31: Iteraciones del método de Steffensen aplicado al polinomio $x^2 - x - 2$ tomando como valor inicial $x_0 = -0.5$.

- Valor inicial $x_0 = 0$ y tolerancia= 10^{-15}
  - Número de iteraciones hasta converger= 6.
  - Raíz a la que converge $x = -1$.

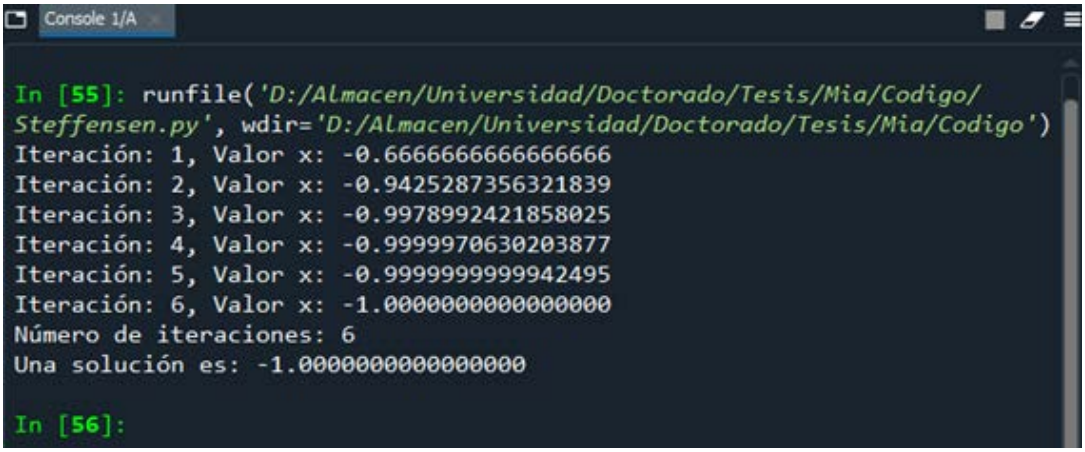


Figura 1.32: Iteraciones del método de Steffensen aplicado al polinomio $x^2 - x - 2$ tomando como valor inicial $x_0 = 0$.

- Valor inicial $x_0 = 1$ y tolerancia= 10^{-15}
  - Número de iteraciones hasta converger= 1.
  - Raíz a la que converge $x = -1$.

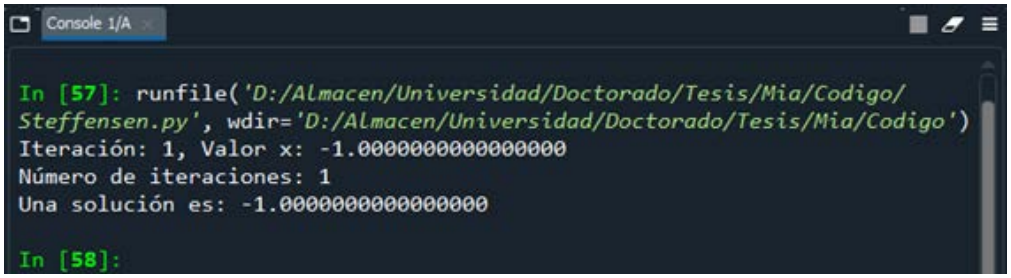


Figura 1.33: Iteraciones del método de Steffensen aplicado al polinomio $x^2 - x - 2$ tomando como valor inicial $x_0 = 1$.

- Valor inicial $x_0 = 5$ y tolerancia= 10^{-15}
  - Número de iteraciones hasta converger= 11.
  - Raíz a la que converge $x = 2$.

Figura 1.34: Iteraciones del método de Steffensen aplicado al polinomio $x^2 - x - 2$ tomando como valor inicial $x_0 = 2$.

1.2.3. Métodos tipo secante

En el ámbito de los métodos numéricos, se ha desarrollado una intensa labor de investigación orientada a mejorar las herramientas disponibles para la resolución de ecuaciones no lineales, con el objetivo de aumentar tanto la estabilidad como la velocidad de convergencia de los algoritmos. Esta línea de trabajo ha dado lugar a numerosos avances en los métodos clásicos, como los basados en Newton, la secante o Steffensen, a través de la incorporación de nuevas estrategias que optimizan su rendimiento y permiten abordar problemas más complejos de forma eficiente.

Uno de los enfoques más destacados en esta área es la búsqueda de métodos que no solo mantengan un alto orden de convergencia, sino que también sean más flexibles en su implementación, reduciendo el número de evaluaciones funcionales o evitando el uso de derivadas, lo cual es especialmente útil en situaciones donde estas últimas son difíciles de calcular. Estas mejoras no solo incrementan la rapidez del proceso iterativo, sino que también contribuyen a minimizar los errores numéricos asociados y a ampliar el rango de aplicabilidad de los métodos.

En este contexto, se presenta el siguiente método uniparamétrico basado en la secante, como una propuesta que combina simplicidad con eficiencia. Este método será descrito con mayor detalle en un capítulo posterior de la tesis, destacándose por sus propiedades de convergencia y su capacidad para ser adaptado a diferentes problemas mediante el ajuste de un parámetro que actúa como elemento de control del comportamiento del algoritmo.

El método iterativo uniparamétrico se define de la siguiente manera:

$$\begin{cases} w_{-1}, w_0 \text{ dados en } \Omega, \quad \lambda \in [0,1], \\ u_n = \lambda w_n + (1-\lambda) w_{n-1}, \quad n \geq 0, \\ w_{n+1} = w_n - [u_n, 2w_n - u_n; F]^{-1} F(w_n), \end{cases}$$

donde:

- λ es un parámetro que controla el balance entre las iteraciones actuales y anteriores.
- $[u, v; F]$ representa una diferencia dividida de primer orden, definida como:

$$[u, v; F](u - v) = F(u) - F(v).$$

Características y Propiedades

- **Familia de Métodos:**
  - Para $\lambda = 0$, el método se reduce al método de Kurchatov.
  - Para $\lambda = 1$ y F diferenciable, el método se convierte en el método de Newton.
- **Convergencia Cuadrática:** La utilización de diferencias divididas simétricas, como $[u_n, 2w_n - u_n; F]$, mejora la aproximación de la derivada, logrando una convergencia de al menos segundo orden para $\lambda \in [0, 1]$.

- **Eficiencia y Flexibilidad:**
  - El costo computacional es similar al de los métodos de la secante y Newton.
  - La inclusión del parámetro λ permite ajustar el método según las características del problema, acercándose al método de Newton a medida que λ se aproxima a 1.
- **Velocidad de Convergencia:** Al usar diferencias divididas simétricas, se mejora la velocidad de convergencia respecto a métodos tradicionales de tipo secante.

Este método es una alternativa poderosa y flexible para resolver ecuaciones no lineales, combinando la estabilidad de los métodos basados en diferencias divididas con la rapidez de convergencia propia de esquemas cuadráticos.

A continuación, se presenta una implementación en Python del método para la ecuación $x^2 - x - 2$:

$$u_n = \lambda x_n + (1 - \lambda)x_{n-1},$$

$$[u_n, 2x_n - u_n; F] = \frac{(2x_n - u_n)^2 - (2x_n - u_n) - 2 - (u_n^2 - u_n - 2)}{(2x_n - u_n) - u_n},$$

$$x_{n+1} = x_n - \frac{x_n^2 - x_n - 2}{[u_n, 2x_n - u_n; F]}.$$

```
# -*- coding: utf-8 -*-
# Metodo uniparametrico basado en la secante
# F: funcion no lineal
# x0, x1: valores iniciales
# eps: error tolerable
# L: parametro del m todo

def tipo_secante_unipar(f, x0, x1, L, eps):
    iter = 0  # Contador de iteraciones

    while abs(f(x1)) > eps and iter < 100:
        try:
            # Calculo de U(x1, x0), vaiable intermedia
            U = L * x1 + (1 - L) * x0

            # Calculo de DF(x1, U), Diferencia dividida
            DF = (f(2 * x1 - U) - f(U)) / (2 * x1 - 2 * U)

            # Calculo de M(X, Y)
            x = x1 - f(x1) / DF
        except ZeroDivisionError:
            print('Division por cero en x= ', x)
            break  # Salir del bucle en caso de error
        x0, x1 = x1, x  # Actualizar valores
        iter += 1
        #Se muestra evolucion iteraciones intermedias
        print('Iteraci n: %i, Valor x: %.16f' % (iter, x1))
    # Si no encuentra solucion dentro del error tolerable
    if abs(f(x1)) > eps:
        iter = -1
    return x1, iter

# Funcion, definida inline
f = lambda x: x**2-x-2

# Valores iniciales y parametros
x0 = -4  # Primer valor inicial
x1 = -0.5  # Segundo valor inicial
L = 0.6  # Parametro del m todo
eps = 1E-15  # Tolerancia al error

# Llamada al metodo uniparametrico
solucion, itera = tipo_secante_unipar(f, x0, x1, L, eps)

# Resultados
if itera > 0:
    print('Numero de iteraciones: %d,' % itera)
    print('Una solucion es: %.16f' % solucion)
else:
    print('Solucion no encontrada')
```

Las raices de este polinomio son bien conocidas, $x = -1$ y $x = 2$. Se muestran los resultados obtenidos para varios puntos iniciales, en este caso se necesita partir de 2 puntos iniciales, se utilizará por tanto la combinación de los empleados en los ejemplos anteriores:

- Valores iniciales, $x_0 = -4$ y $x_1 = -0.5$, tolerancia= 10^{-15}
  - Número de iteraciones hasta converger= 5
  - Raíz a la que converge $x = -1$.

Figura 1.35: Iteraciones de la familia uniparamétrica de tipo Secante aplicada al polinomio $x^2 - x - 2$ tomando como valores iniciales $x_0 = -4$ y $x_1 = -0.5$. Parámetro $\lambda = 0.6$.

- Valores iniciales, $x_0 = -4$ y $x_1 = 0$, tolerancia= 10^{-15}
  - Número de iteraciones hasta converger= 6
  - Raíz a la que converge $x = -1$.

Figura 1.36: Iteraciones de la familia uniparamétrica de tipo Secante aplicada al polinomio $x^2 - x - 2$ tomando como valores iniciales $x_0 = -4$ y $x_1 = 0$. Parámetro $\lambda = 0.6$.

- Valores iniciales, $x_0 = -4$ y $x_1 = 1$, tolerancia= 10^{-15}
  - Número de iteraciones hasta converger= 6.
  - Raíz a la que converge $x = 2$.

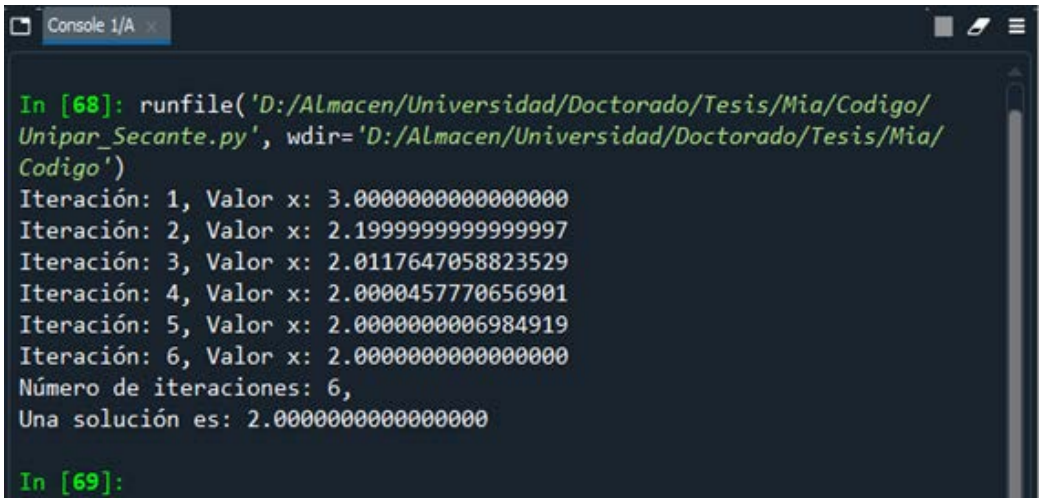


Figura 1.37: Iteraciones de la familia uniparamétrica de tipo Secante aplicada al polinomio $x^2 - x - 2$ tomando como valores iniciales $x_0 = -4$ y $x_1 = 1$. Parámetro $\lambda = 0.6$.

- Valores iniciales, $x_0 = -4$ y $x_1 = 5$, tolerancia= 10^{-15}
  - Número de iteraciones hasta converger= 6.
  - Raíz a la que converge $x = 2$.

Figura 1.38: Iteraciones de la familia uniparamétrica de tipo Secante aplicada al polinomio $x^2 - x - 2$ tomando como valores iniciales $x_0 = -4$ y $x_1 = 5$. Parámetro $\lambda = 0.6$.

- Valores iniciales, $x_0 = -0.5$ y $x_1 = 0$, tolerancia= 10^{-15}
  - Número de iteraciones hasta converger= 6.
  - Raíz a la que converge $x = -1$.

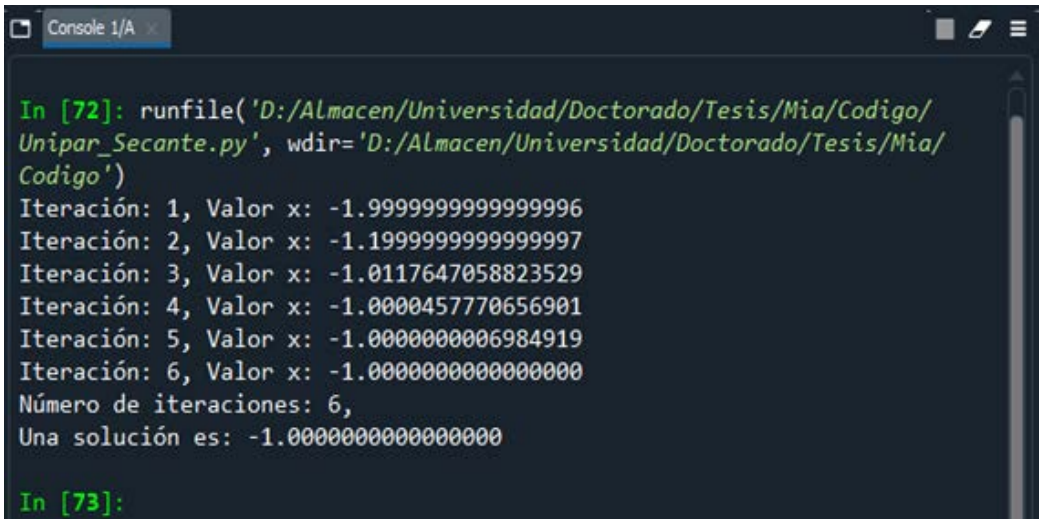


Figura 1.39: Iteraciones de la familia uniparamétrica de tipo Secante aplicada al polinomio $x^2 - x - 2$ tomando como valores iniciales $x_0 = -0.5$ y $x_1 = 0$. Parámetro $\lambda = 0.6$.

- Valores iniciales, $x_0 = -0.5$ y $x_1 = 1$, tolerancia= 10^{-15}
  - Número de iteraciones hasta converger= 6.
  - Raíz a la que converge $x = -1$.

- Valores iniciales, $x_0 = -0.5$ y $x_1 = 5$, tolerancia= 10^{-15}
  - Número de iteraciones hasta converger= 6.
  - Raíz a la que converge $x = 2$.

- Valores iniciales, $x_0 = 0$ y $x_1 = 1$, tolerancia= 10^{-15}
  - Número de iteraciones hasta converger= 6.
  - Raíz a la que converge $x = 2$.

- Valores iniciales, $x_0 = 0$ y $x_1 = 5$, tolerancia= 10^{-15}
  - Número de iteraciones hasta converger= 6.
  - Raíz a la que converge $x = 2$.

- Valores iniciales, $x_0 = 1$ y $x_1 = 5$, tolerancia= 10^{-15}
  - Número de iteraciones hasta converger= 6.
  - Raíz a la que converge $x = 2$.

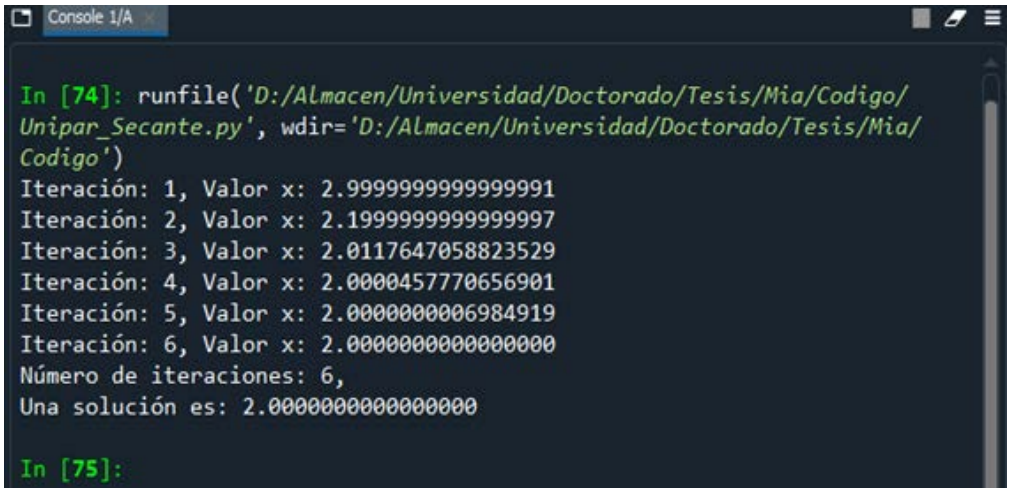


Figura 1.40: Iteraciones de la familia uniparamétrica de tipo Secante aplicada al polinomio $x^2 - x - 2$ tomando como valores iniciales $x_0 = -0.5$ y $x_1 = 1$. Parámetro $\lambda = 0.6$.

Figura 1.41: Iteraciones de la familia uniparamétrica de tipo Secante aplicada al polinomio $x^2 - x - 2$ tomando como valores iniciales $x_0 = -0.5$ y $x_1 = 5$. Parámetro $\lambda = 0.6$.

1.3. Nociones básicas de dinámica compleja

En esta sección introductoria se presentan algunas nociones y resultados fundamentales acerca de las funciones de variable compleja, con especial atención en el caso de las aplicaciones racionales. Sea $R : \bar{\mathbb{C}} \to \bar{\mathbb{C}}$ una función racional definida sobre la esfera de Riemann, donde $\bar{\mathbb{C}} = \mathbb{C} \cup \infty$. Dicha función puede expresarse en la forma $R(z) = \frac{P(z)}{Q(z)}$, donde $P(z)$ y $Q(z)$ son polinomios complejos sin factores en común.

En el contexto del espacio compacto $\bar{\mathbb{C}}$, todo polinomio de grado n tiene exactamente n raíces, considerando sus respectivas multiplicidades. Esta característica

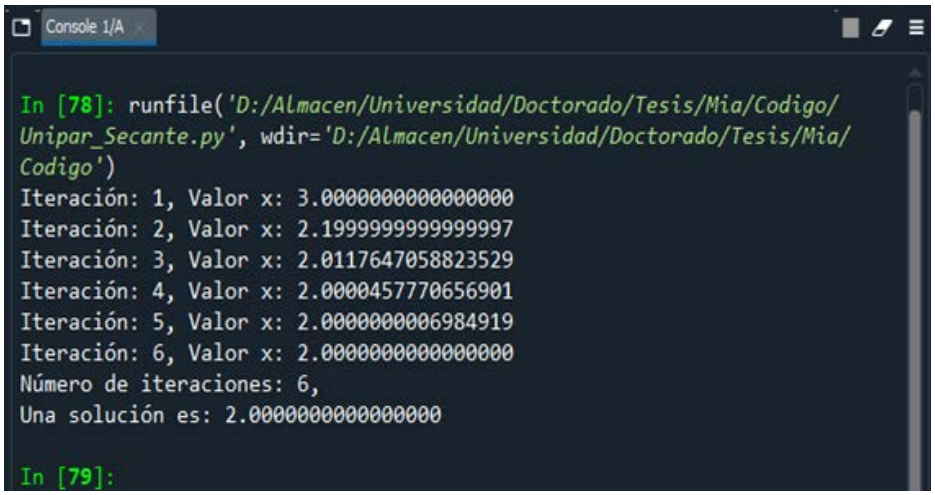


Figura 1.42: Iteraciones de la familia uniparamétrica de tipo Secante aplicada al polinomio $x^2 - x - 2$ tomando como valores iniciales $x_0 = 0$ y $x_1 = 1$. Parámetro $\lambda = 0.6$.

Figura 1.43: Iteraciones de la familia uniparamétrica de tipo Secante aplicada al polinomio $x^2 - x - 2$ tomando como valores iniciales $x_0 = 0$ y $x_1 = 5$. Parámetro $\lambda = 0.6$.

permite clasificar los polinomios en función del número de raíces en el plano complejo. Dado que el objetivo principal de esta tesis es el estudio de métodos tipo secante aplicados a la solución de ecuaciones polinómicas, se pondrá un énfasis especial en el análisis de los polinomios. Para una exploración más profunda de estas propiedades y otros aspectos relacionados con la dinámica en el plano complejo, se recomienda consultar la bibliografía pertinente, como [2, 3, 23, 25, 27, 28, 29, 33, 43, 35, 37, 38, 41, 46, 48, 64, 70, 78, 79, 81, 90, 92, 99, 104, 108, 112, 120, 121, 127, 128, 129, 130, 141, 146, 149, 151, 152, 154, 156, 157, 163, 165, 166, 168], entre otras fuentes.

Figura 1.44: Iteraciones de la familia uniparamétrica de tipo Secante aplicada al polinomio $x^2 - x - 2$ tomando como valores iniciales $x_0 = 1$ y $x_1 = 5$. Parámetro $\lambda = 0.6$.

1.3.1. Puntos fijos, puntos periódicos y puntos críticos

El grado de la función racional $R(z)$ se define como el mayor de los grados de los polinomios $P(z)$ y $Q(z)$. A partir de este punto, se asumirá que todas las aplicaciones racionales consideradas en este análisis, salvo que se especifique lo contrario, tendrán un grado igual o superior a 2.

En el contexto del estudio de la dinámica de funciones racionales, uno de los problemas principales es comprender el comportamiento de las sucesiones $\{z_n\}$ que se generan de manera recursiva a partir de un valor inicial dado, $z_0 \in \bar{\mathbb{C}}$. En este caso, cada término de la sucesión se define como $z_k = R^k(z_0)$, donde $R^k(z)$ representa la composición iterada de la función $R(z)$ consigo misma k veces, es decir, $R^k(z) = \overbrace{R \circ \cdots \circ R}^{k}$. El análisis se centra en determinar si esta sucesión converge, lo cual implica que existe un límite al que los términos de la sucesión tienden. Si tal límite existe, se deduce que corresponde a un punto fijo de la función $R(z)$, dado que el valor alcanzado por la sucesión permanece inalterado bajo la aplicación de R.

Definición 1.1. *Decimos que un punto $z \in \bar{\mathbb{C}}$, es un punto fijo de $R(z)$ si verifica $R(z) = z$.*

Los puntos fijos pueden clasificarse en diferentes categorías de acuerdo con su multiplicador, el cual se define de la siguiente manera.

Definición 1.2. *Si z es un punto fijo de $R(z)$, entonces llamaremos multiplicador o autovalor de z al valor $\mu = R'(z)$.*

Los puntos fijos de una función racional representan casos particulares dentro de los puntos periódicos, ya que corresponden específicamente a aquellos con periodo 1. La definición general de los puntos periódicos es la siguiente.

Definición 1.3. *Decimos que un punto $z_0 \in \bar{\mathbb{C}}$, es un punto periódico de periodo p, si $R^p(z_0) = z_0$ y $R^n(z_0) \neq z_0$ para todo $n < p$.*

La órbita de un punto periódico z con periodo n está formada por los n puntos distintos dados por

$$\mathcal{O}(z) = \{z, R(z), R^2(z), \ldots, R^{n-1}(z)\}.$$

Definición 1.4. *La órbita de un punto periódico de periodo n recibe el nombre de n-ciclo.*

Otro tipo relevante de puntos son aquellos que, aunque no son periódicos, alguna de sus iteraciones sí lo es; estos se denominan puntos eventualmente periódicos.

Definición 1.5. *Un punto z decimos que es eventualmente periódico si cumple que $R^k(z) = R^{k+p}(z)$ para algunos enteros positivos p y k.*

A los puntos periódicos se les asocia un multiplicador o valor propio, el cual se define de la siguiente forma:

Definición 1.6. *Si z es un punto periódico de periodo p, entonces llamaremos multiplicador del p-ciclo al valor de la derivada $\mu = (R^p)'(z)$.*

Tanto los puntos fijos como los puntos periódicos pueden clasificarse según su carácter, el cual está determinado por el valor propio asociado.

Definición 1.7. *Sea z_0 un punto fijo de $R(z)$ con multiplicador μ, entonces diremos que z_0 es:*

(i) *Superatractor, si $\mu = 0$.*

(ii) *Atractor, si $|\mu| < 1$.*

(iii) *Indiferente, si $|\mu| = 1$.*

(iv) *Repulsor, si $|\mu| > 1$.*

Los puntos periódicos de periodo p de una función racional $R(z)$ se clasifican, de manera análoga a los puntos fijos, en las siguientes categorías:

(i) Superatractores, si $\mu = 0$.

(ii) Atractores, si $|\mu| < 1$.

(iii) Indiferentes, si $|\mu| = 1$.

(iv) Repulsores, si $|\mu| > 1$.

Es importante destacar el papel fundamental del autovalor μ asociado a un punto periódico de periodo p (o, de manera equivalente, al p-ciclo). La clasificación de los puntos en función del valor de su multiplicador proporciona indicaciones sobre el comportamiento en un entorno cercano a dichos puntos. Así, en el caso

de un punto fijo atractor, cualquier punto en su vecindad tenderá a acercarse a él con cada iteración, mientras que en un punto fijo repulsor, los puntos en su entorno se alejarán progresivamente. El análisis de los puntos fijos indiferentes resulta considerablemente más complejo, ya que presenta diversas posibilidades para el comportamiento local y se sitúa en la frontera entre los atractores y los repulsores. Para demostrar esta afirmación, consideremos que z_0 es un punto fijo atractor de $R(z)$ (la demostración es análoga para puntos periódicos de cualquier periodo p), lo que implica que $|R'(z_0)| < 1$. Entonces, para un punto z lo suficientemente próximo a z_0, existirá un valor $\beta < 1$ que cumplirá la siguiente condición:

$$\frac{|R(z) - R(z_0)|}{|z - z_0|} < \beta < 1,$$

Dado que z_0 es un punto fijo, se cumple que $R(z_0) = z_0$. Además, la desigualdad mencionada anteriormente se puede expresar de la siguiente manera:

$$|R(z) - z_0| < \beta |z - z_0|,$$

Por lo tanto, $R(z)$ se encuentra más cerca de z_0 que el punto z. Si se iteran k veces, se obtiene que

$$|R(z) - z_0| < \beta^k |z - z_0|,$$

y, como $\beta < 1$, tenemos que

$$\lim_{n \to \infty} R^n(z) = z_0.$$

A partir de lo anterior, se puede concluir que cualquier punto z perteneciente a un entorno del punto fijo z_0 tenderá hacia z_0 bajo la iteración de la función $R(z)$. Este resultado indica que, en tales condiciones, el punto fijo actúa como un atractor local en el espacio complejo.

Por otro lado, si z_0 es un punto fijo de tipo repulsor (de manera análoga, esto también se aplica a puntos periódicos de cualquier periodo p) de la función $R(z)$, lo que ocurre cuando se cumple la condición $|R'(z_0)| > 1$, el análisis sigue un procedimiento similar, pero con una implicación diferente. En este caso, la desigualdad obtenida es

$$\frac{|R(z) - R(z_0)|}{|z - z_0|} > 1,$$

A partir de esta expresión, sin necesidad de realizar más manipulaciones algebraicas, se deduce directamente que $|R(z) - z_0| > |z - z_0|$, lo que implica que la iteración de $R(z)$ provoca que los puntos cercanos a z_0 se alejen progresivamente de él. Esto confirma que, en este caso, z_0 no actúa como un atractor, sino que los puntos en su entorno inmediato son expulsados en cada iteración.

Por último, al enfocarse en los puntos indiferentes, es decir, aquellos z_0 que satisfacen $|R'(z_0)| = 1$, el comportamiento de las órbitas de los puntos en el entorno de estos puntos fijos indiferentes es considerablemente más complejo que en los casos anteriores, ya que representan la frontera entre atractores y repulsores. El

multiplicador asociado a los puntos fijos indiferentes, como se ha mencionado, tiene la forma $|\mu| = 1 = e^{i\theta}$, donde $\theta \in [0, 2\pi)$.

El comportamiento de estos puntos dependerá de si θ es un número racional, es decir, $\theta = \frac{p}{q}$ con p y q coprimos, o si es un número irracional. Para un estudio más detallado de este tipo de puntos, se puede consultar [30], [42] y [150], entre otros. Los puntos fijos indiferentes se clasifican en los siguientes tipos:

- Parabólicos, si existe algún $q \in \mathbb{Z}$ tal que $\mu^q = 1$.
- Neutrales o indiferentes, cuando $|\mu| = 1$ y $\mu^q \neq 1$ para todos los $q \in \mathbb{Z}$. En este caso, se presentan dos posibilidades:
  - Si el punto es linealizable, se denomina punto de Siegel.
  - Si no es linealizable, se clasifica como punto de Cremer.

El análisis del carácter del ∞ como punto fijo requiere un estudio particular.

Definición 1.8. *El punto ∞ es un punto fijo de $R(z)$ si y sólo si $z = 0$ es un punto fijo de la función*

$$F : z \to \frac{1}{R(\frac{1}{z})}.$$

Si ∞ es un punto fijo de $R(z)$, entonces su multiplicador asociado es $\mu = F'(0)$. En particular tenemos que ∞ es un superatractor si y sólo si $F'(0) = 0$.

Además de los diversos tipos de puntos fijos existentes, el concepto de cuenca de atracción también es de gran relevancia.

Definición 1.9. *Llamaremos cuenca de atracción de un punto fijo ω de $R(z)$ al conjunto de puntos del plano cuyas iteraciones por $R(z)$ convergen hacia el punto fijo, es decir,*

$$\{z \in \bar{\mathbb{C}} | \lim_{n\to\infty} R^n(z) = \omega\}.$$

La Figura 1.45 muestra las cuencas de atracción correspondientes a las raíces del polinomio $p(z) = z^3 - 1$ al aplicar el método de Newton, es decir,

$$R(z) = z - \frac{z^3 - 1}{3z^2} = \frac{2z^3 + 1}{3z^2}.$$

Dado que las cuencas de atracción pueden poseer un número infinito de componentes, se introduce el siguiente concepto.

Definición 1.10. *La cuenca de atracción inmediata de un punto fijo ω de $R(z)$ es la componente conexa que contiene el punto fijo.*

Asimismo, en el análisis de la dinámica de funciones racionales, los puntos críticos desempeñan un papel fundamental, ya que, en cierta medida, pueden caracterizar el comportamiento de las diferentes órbitas.

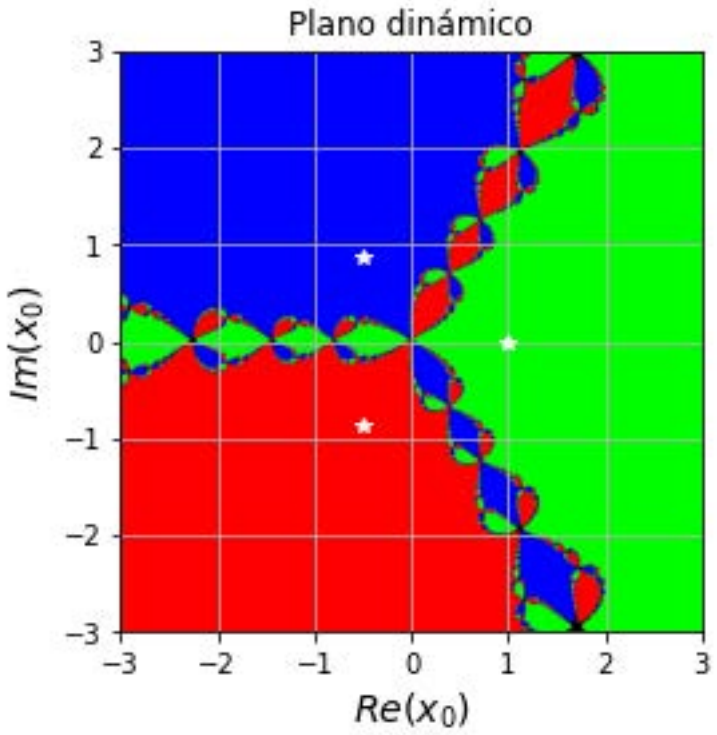


Figura 1.45: Cuencas de atracción asociadas a las raíces del polinomio $p(z) = z^3 - 1$ al aplicarle el método de Newton. En verde aparece la cuenca de $z = 1$, en azul la cuenca de $z = e^{2\pi i/3}$ y en rojo la cuenca de $z = e^{4\pi i/3}$. La imagen ha sido generada con Python.

Definición 1.11. *Sea $R(z)$ una función racional de grado d. Un punto $w \in \bar{\mathbb{C}}$, para el cual la cardinalidad de $R^{-1}(w)$ es menor que d, es llamado un valor crítico de $R(z)$. Un punto $z \in R^{-1}(w)$ que es una raíz de $R(z) - w$, con multiplicidad mayor que 1, es llamado un punto crítico de $R(z)$. En particular, z es un punto crítico de una función holomorfa $p(z)$ si $p'(z) = 0$.*

Como ejemplo, consideremos la función $R(z) = \frac{z}{(z-1)^2}$, la cual posee un punto crítico en $z = -1$, dado que su derivada se anula en ese valor:

$$R'(z) = \frac{z+1}{(1-z)^3} = 0.$$

El valor crítico asociado a $z = -1$ es $-\frac{1}{4}$. Además, al considerar $\omega = \infty$, se obtiene que

$$R^{-1}(\omega) = 1,$$

lo que implica que $\omega = \infty$ es otro valor crítico, esta vez vinculado al punto crítico $z = 1$.

Los puntos críticos desempeñan un papel clave en el estudio de la dinámica de una función $p(z)$, ya que en estos puntos la función $p(z)$ deja de comportarse como un homeomorfismo local. En otras palabras, si p_1 es un punto crítico de $p(z)$,

entonces no es posible definir ninguna rama de la función inversa de $p(z)$ en un entorno del valor crítico $p(p_1)$.

Estos dos conceptos se confunden con frecuencia, por lo que es fundamental distinguir claramente entre punto crítico y valor crítico. Por otro lado, la siguiente nota establece una relación entre los puntos críticos y los ciclos atractores.

Nota 1.1. *Si el máximo de los grados del denominador y el numerador de una función racional es d, entonces la función tendrá como mucho* $2d-2$ *puntos críticos y, por lo tanto,* $2d-2$ *ciclos atractores.*

En consecuencia, la presencia de órbitas o ciclos atractores afecta la búsqueda de raíces de $p(z) = 0$ cuando se aplica un método iterativo.

1.3.2. Conjuntos de Julia y Fatou

Esta sección se enfoca en el estudio de las propiedades de los conjuntos de Julia (véase [93]) y de Fatou (véase [65]), los cuales reciben su nombre en honor a sus descubridores, Gaston Julia y Pierre Fatou. Ambos matemáticos mantuvieron una fuerte rivalidad en el marco de *le grand prix des sciences mathématiques* de 1918, un episodio que se encuentra detallado en [64].

Estos conjuntos presentan una relación de complementariedad, ya que la definición de uno es el complemento del otro. De manera intuitiva, el conjunto de Fatou está compuesto por los puntos cuyas órbitas, bajo iteración de una función, exhiben estabilidad, mientras que el conjunto de Julia reúne aquellos puntos cuyas órbitas muestran un comportamiento inestable.

El interés por los conjuntos de Julia y Fatou ha crecido significativamente en la comunidad matemática debido a su relevancia en la dinámica compleja (véanse [64], [116] o [126], entre otros). La diferencia fundamental entre ambos conjuntos está estrechamente vinculada con el concepto de normalidad, el cual se define a continuación.

Definición 1.12. *Dados dos espacios métricos* (X, d) *e* (Y, ρ) *y una familia* τ *de aplicaciones de* X *en* Y*, decimos que* τ *es equicontinua en* x_0 *si para cada* $\varepsilon > 0$ *existe un* $\delta > 0$ *tal que para todo* $x \in X$ *y para todo* $f \in \tau$*, se cumple que si* $d(x_0, x) < \delta$*, entonces* $\rho(f(x_0), f(x)) < \varepsilon$*. Decimos que* τ *es equicontinua en un subespacio* Ω*, si es equicontinua para cada punto* $x \in \Omega \subseteq X$*.*

Definición 1.13. *Suponemos que* $R(z)$ *es una aplicación racional no constante. Se denomina conjunto de Fatou asociado a* $R(z)$*, y se denota por* $\mathcal{F}(R)$*, al mayor subconjunto abierto de* $\bar{\mathbb{C}}$ *en el que la familia*

$$\tau = \{R, R \circ R, \dots, R \circ \cdots \circ R, \dots\}$$

es equicontinua con respecto a una de las métricas en $\bar{\mathbb{C}}$*.*

El complementario del conjunto de Fatou es el conjunto en el que la dinámica es más complicada de entender.

Definición 1.14. *Suponemos que* $R(z)$ *es una aplicación racional no constante. El complementario del conjunto de Fatou, denotado por* $\mathcal{J}(R) = \bar{\mathbb{C}} - \mathcal{F}(R)$*, se denomina conjunto de Julia asociado a* $R(z)$*.*

Las propiedades de los conjuntos de Julia y Fatou han sido estudiadas por numerosos autores en multitud de obras, un ejemplo de ello son el libro de Fagella y Jarque (véase [64]) o el artículo de Peitgen, Saupe y Haeseler (véase [126]). Entre los resultados sobre las propiedades más importantes cabe destacar el siguiente listado.

Sea $R(z)$ una aplicación racional y $\mathcal{J}(R)$ el conjunto de Julia asociado a $R(z)$. Entonces:

(i) $\mathcal{J}(R) \neq \emptyset$ y $\mathcal{J}(R)$ es denso en sí mismo (es perfecto).

(ii) $\mathcal{J}(R) = \mathcal{J}(R^n)$, para todo $n \in \mathbb{N}$.

(iii) $R(\mathcal{J}(R)) = R^{-1}(\mathcal{J}(R)) = \mathcal{J}(R)$, es decir, $\mathcal{J}(R)$ es completamente invariante.

(iv) Si $z_0 \in \mathcal{J}(R)$, entonces la clausura de $\{z \in \bar{\mathbb{C}} | R^n(z) = z_0\} = \mathcal{J}(R)$.

(v) Si γ es un ciclo atractor de $R(z)$, entonces la cuenca de atracción de γ está contenida en $\mathcal{F}(R)$ y, además, su frontera está en $\mathcal{J}(R)$.

(vi) Si $\mathcal{J}(R)$ tiene un interior no vacío, entonces $\mathcal{J}(R) = \bar{\mathbb{C}}$.

(vii) $R(z)$ restringida a su conjunto de Julia es sensible respecto a condiciones iniciales.

(viii) $\mathcal{J}(R)$ es autosimilar.

(ix) Para todo punto $z \in \mathcal{J}(R)$ el conjunto de preimágenes de z es denso en $\mathcal{J}(R)$.

(x) $R(z)$ es topológicamente transitiva en $\mathcal{J}(R)$.

(xi) Si P_R es el conjunto de los puntos periódicos repulsores de $R(z)$, entonces $P_R \subset \mathcal{J}(R)$ y además $\bar{P_R} = \mathcal{J}(R)$.

(xii) Si $R(z)$ es no constante y sea $\tau(z)$ una transformación de Möbius. Definimos una aplicación racional nueva $S(z) = \tau \circ R \circ \tau^{-1}(z)$. Entonces $\mathcal{F}(S) = \tau(\mathcal{F}(R))$ y $\mathcal{J}(S) = \tau(\mathcal{J}(R))$.

Como se mencionó con anterioridad, una cuenca de atracción puede estar compuesta por un número infinito de componentes conexas, y cada una de ellas forma parte del conjunto de Fatou, el cual es un conjunto abierto. Para una clasificación más detallada de las diversas estructuras que pueden aparecer en el conjunto de Fatou, se recomienda consultar las referencias [63], [64], [126] o [153].

Un conjunto X se denomina errante si la intersección $R^n X \cap R^m X$ es vacía para todo $n > m \geq 0$. En el caso de las funciones racionales, la existencia de dominios errantes fue descartada por Sullivan en 1985, como se establece en el siguiente resultado.

Teorema 1.1. *El conjunto de Fatou de una aplicación racional no tiene componentes errantes. Es decir, cada componente del conjunto de Fatou es eventualmente periódica.*

No obstante, I. N. Baker en [21] demostró que los conjuntos de Fatou de ciertas funciones enteras en $\bar{\mathbb{C}}$ pueden presentar dominios errantes.

De manera esquemática, las componentes del conjunto de Fatou se pueden clasificar de la siguiente forma:

1. Cuencas de atracción de puntos fijos atractores. Las órbitas de puntos cercanos a los puntos fijos convergen hacia dichos puntos. Un ejemplo puede verse en la Figura 1.46, en la que se observan las órbitas de los 4 puntos fijos atractores, con colores diferentes.

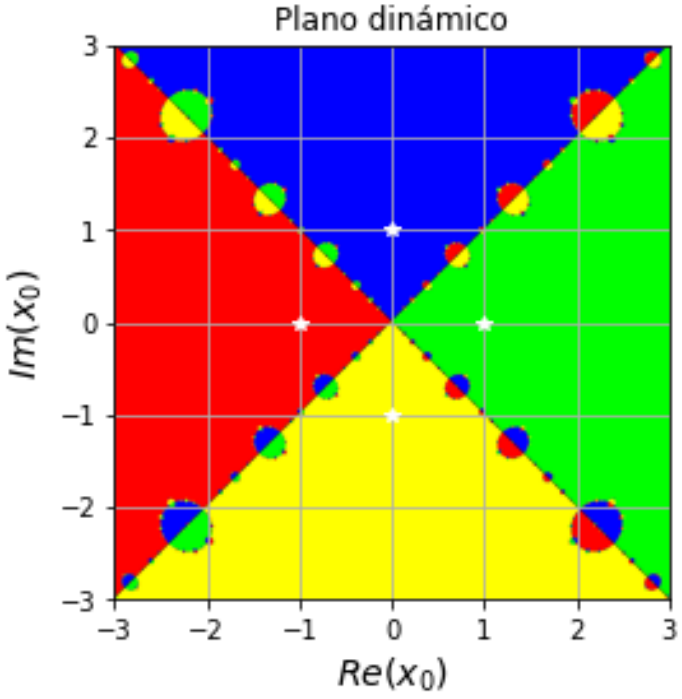


Figura 1.46: Cuencas de atracción asociadas a las raíces del polinomio $p(z) = z^4 - 1$, al aplicarle el conocido método de Halley $\left(H_f(z) = z - \frac{f(z)}{f'(z)} \frac{2}{2 - L_f(z)}\right)$. En verde la cuenca de $z = 1$, en rojo la de $z = -1$, en azul la cuenca de $z = i$ y en amarillo la de $z = -i$.

2. Cuencas de atracción de ciclos periódicos. En este caso, las órbitas convergen hacia una órbita periódica atractora. Por ejemplo, la función $p(z) = z^2 - 1$ presenta una órbita $\{0, -1\}$ de periodo 2 que es atractora.

3. Cuencas de atracción parabólicas. Su comportamiento es muy similar al de las cuencas de atracción de ciclos periódicos, con la diferencia de que los puntos fijos o periódicos se ubican en la frontera de la cuenca, en lugar de estar en su interior. El polinomio $p(z) = z^2 + 0.25$, el cual tiene un punto fijo en $z_0 = 0.5$ con un multiplicador asociado de $\mu = 1$.

Además, aunque en el desarrollo de esta tesis no aparecen, pueden encontrarse las siguientes componentes del conjunto de Fatou:

- Discos de rotación o de Siegel. Un disco de rotación fijo de $P(z)$ se define como un conjunto abierto U, que es conforme equivalente al disco unidad y contiene en su interior un punto fijo z_0. En este caso, la restricción $P|_U$ se encuentra conformemente conjugada a una rotación rígida dada por

  $$R_\theta(z) = e^{2\pi i\theta},$$

  donde $\theta \in \mathbb{R} \setminus \mathbb{Q}$. Esto significa que existe un homeomorfismo conforme $h : U \to D$ tal que

  $$h \circ P(z) = R_\theta \circ h(z).$$

- Anillos de rotación o de Herman. Presentan un comportamiento similar al de los discos de rotación, con la diferencia de que, en lugar de ser conformemente conjugados a un disco, corresponden a una estructura en forma de anillo. A diferencia de los discos de rotación, los anillos no están vinculados a ninguna órbita periódica.

- Dominios parabólicos en el ∞ o de Baker. Un dominio parabólico infinito fijo es un conjunto abierto U en el plano, donde todas las órbitas de los puntos en U tienden al infinito, el cual debe ser una singularidad esencial. Por definición, este tipo de dominios solo puede aparecer en funciones trascendentes.

- Dominios errantes. Un dominio errante es un conjunto abierto U que cumple

  $$F^n(U) \cap F^m(U) = \emptyset, \quad \text{para todos } n, m \in \mathbb{N}.$$

  Esto implica que los dominios errantes no forman parte de componentes que sean periódicas ni eventualmente periódicas dentro del conjunto de Fatou.

Una vez definidos los conjuntos de Julia y Fatou, surge la pregunta de qué tipo de puntos pertenecen a cada uno de ellos.

Nota 1.2. *Los puntos fijos:*

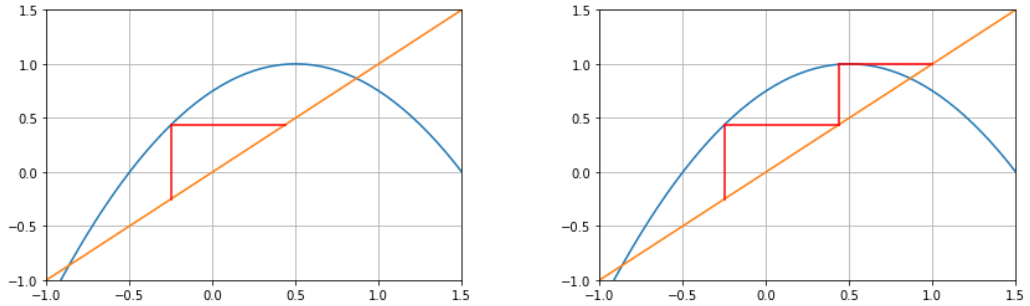
- *Superatractores, atractores y linealizables (Siegel) pertenecen al conjunto de Fatou.*

- *Repulsores, parabólicos y no linealizables (Cremer) pertenecen al conjunto de Julia.*

1.4. Nociones básicas de iteración de funciones reales

Además de los conceptos introducidos en la sección dedicada a las nociones básicas de iteración compleja, en el contexto real surgen otras ideas relevantes que deben considerarse, tales como la presencia de asíntotas, los diagramas de Feigenbaum y los exponentes de Lyapunov, entre otros. Se recomienda consultar la bibliografía pertinente, como [5, 37, 103, 105], entre otras fuentes.

Una de las herramientas fundamentales en este estudio es el análisis gráfico del comportamiento dinámico. Para llevar a cabo este análisis, se requiere un gráfico que represente tanto la función de iteración como la identidad. El procedimiento inicia eligiendo un punto inicial x_0, desde el cual se traza una línea vertical hasta su intersección con la función de iteración. En ese punto de intersección, se dibuja una línea horizontal hasta alcanzar la identidad, tras lo cual se traza nuevamente una línea vertical hasta que vuelva a interceptar la función. La coordenada horizontal de este último punto representa el resultado de la primera iteración del método. Repitiendo este proceso desde la nueva ubicación obtenida, se generan las iteraciones sucesivas. La Figura 1.47 ilustra los dos primeros pasos de este procedimiento, mientras que la Figura 1.48 muestra la evolución completa del proceso con tres y cincuenta iteraciones, respectivamente.



Un paso Dos pasos

Figura 1.47: Aplicación del algoritmo de estudio gráfico de la dinámica con uno y dos pasos.

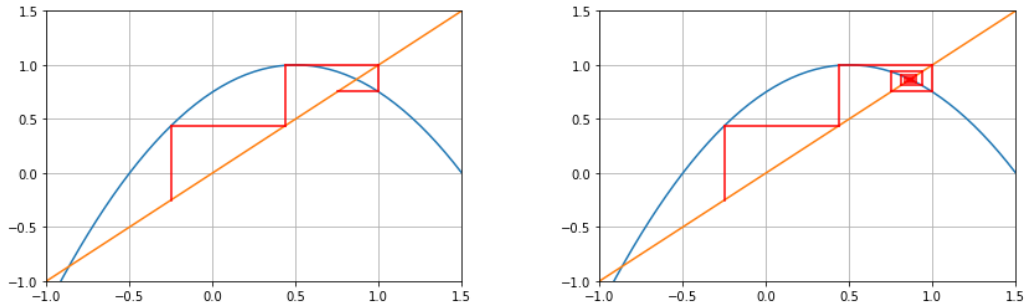
Por otro lado, resulta útil considerar el homeomorfismo $G : \mathbb{R} \to (0, 1)$, el cual permite realizar compactificaciones de métodos en el intervalo $(0, 1)$. Dicho homeomorfismo se define mediante la expresión

$$G(x) = \frac{1}{\pi} \arctan(x) + \frac{1}{2}. \tag{1.4.1}$$

Su función inversa está dada por

$$G^{-1}(x) = \tan\left(\pi x - \frac{\pi}{2}\right). \tag{1.4.2}$$

Sea $\tau : \mathbb{R} \to \mathbb{R}$ un polinomio de grado superior a 1. La compactificación que se introduce en este caso está definida por



Tres pasos Cincuenta pasos

Figura 1.48: Aplicación del algoritmo de estudio gráfico de la dinámica con tres y cincuenta pasos.

$$\widetilde{\tau}(x) = G \circ \tau \circ G^{-1}(x). \tag{1.4.3}$$

Además, utilizando las siguientes notaciones:

$$G(-\infty) = \lim_{x \to -\infty} G(x) = 0,$$

$$G(\infty) = \lim_{x \to \infty} G(x) = 1,$$

y dado que $\tau(x)$ es un polinomio de grado mayor que 1, es posible extender la compactificación a una función bien definida en el intervalo $[0, 1]$. Cabe destacar que esta compactificación introduce dos nuevos puntos fijos en $x = 0$ y $x = 1$. No obstante, dado que estos puntos son repulsivos, su presencia no influye en la dinámica del método.

En lo que respecta a los puntos fijos, existen ciertas diferencias con la dinámica compleja, como se verá en los siguientes teoremas.

Teorema 1.2. *Sea $f : I \to \mathbb{R}$, donde I es un intervalo y $f \in \mathbb{C}^1(I)$. Sea $\xi \in I$ un punto fijo indiferente de $f(x)$, es decir, el módulo de su multiplicador asociado $|f'(\xi)| = 1$, tal que $|f'(x)|$ presenta un máximo local en ξ, entonces ξ es un punto fijo atractor.*

La demostración a este teorema puede verse en [71].

Ejemplo 1.1. *([71]) El 0 es un punto fijo indiferente atractor de la función $f(x) = -x^3 + x$, como puede verse en la Figura 1.49. Notemos que $|f'(x)| = |-3x^2 + 1|$ presenta un máximo local en $x = 0$.*

Teorema 1.3. *Sea $f : I \to \mathbb{R}$, donde I es un intervalo y $f \in \mathbb{C}^1(I)$. Sea $\xi \in I$ un punto fijo indiferente de $f(x)$, es decir, el módulo de su multiplicador asociado $|f'(\xi)| = 1$, tal que $|f'(x)|$ presenta un mínimo local estricto en ξ, entonces ξ es un punto fijo repulsor.*

La demostración a este teorema se puede encontrar también en [71].

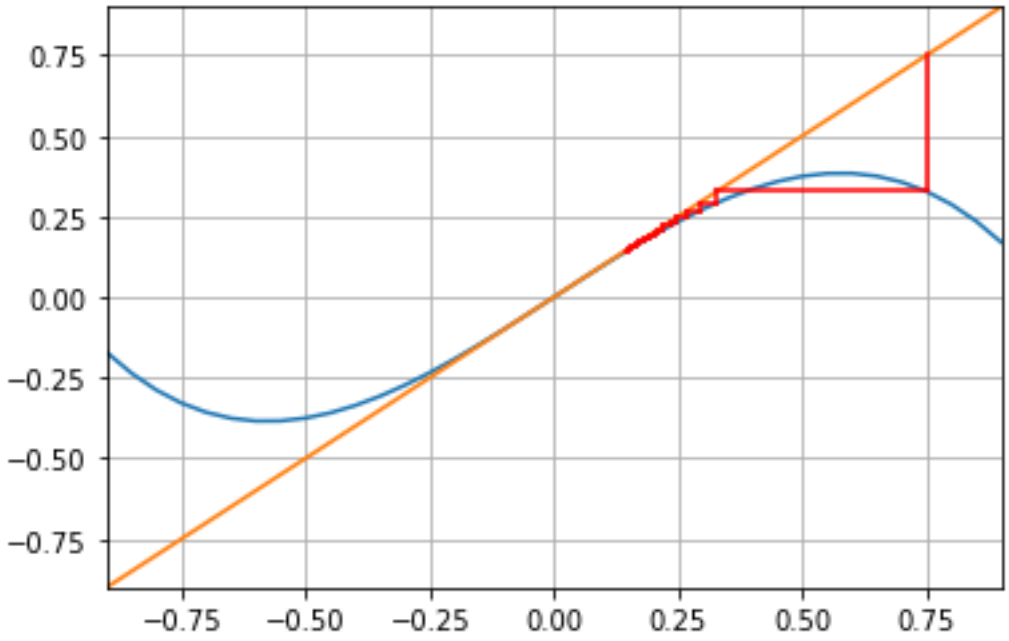


Figura 1.49: Órbitas de $x_0 = 0.75$ bajo la función $f(x) = -x^3 + x$, donde puede verse que el 0 es un punto fijo con multiplicador $\mu = 1$, pero con comportamiento atractor.

Ejemplo 1.2. *([71]) El 0 es un punto fijo indiferente repulsor de la función* $f(x) = x^3 + x$ *como puede verse en la Figura 1.50. En este caso,* $|f'(x)| = 3x^2 + 1$ *tiene un mínimo absoluto en* $x = 0$.

En el estudio de familias uniparamétricas de funciones de iteración, la naturaleza de los puntos periódicos puede modificarse en función del parámetro considerado. En particular, los puntos periódicos indiferentes marcan la transición entre un comportamiento atractor y uno repulsor, o viceversa. Cuando se produce un cambio cualitativo en la dinámica de los puntos periódicos, se dice que ha ocurrido una bifurcación. Entre los tipos de bifurcación más habituales se encuentran los siguientes:

Bifurcación tangente. Se presenta cuando un punto fijo indiferente se descompone en dos puntos fijos, uno de tipo atractor y otro de tipo repulsor, o bien cuando un punto fijo atractor y otro repulsor convergen en un único punto fijo indiferente.

Ejemplo 1.3. *La familia cuadrática* $f_c(x) = x^2 + c$ *presenta una bifurcación tangente en* $c = 0.25$. *Los puntos fijos de* $f_c(x) = x$ *son*

$$x = \frac{1 \pm \sqrt{1 - 4c}}{2}.$$

Por otro lado, como la derivada es $f_c'(x) = 2x$, *se tiene que:*

- *Si* $c < 0.25$, $f_c(x)$ *tiene dos puntos fijos cuyos multiplicadores asociados son* $1 \pm \sqrt{1 - 4c}$ *y, por lo tanto, uno es atractor y el otro es repulsor.*

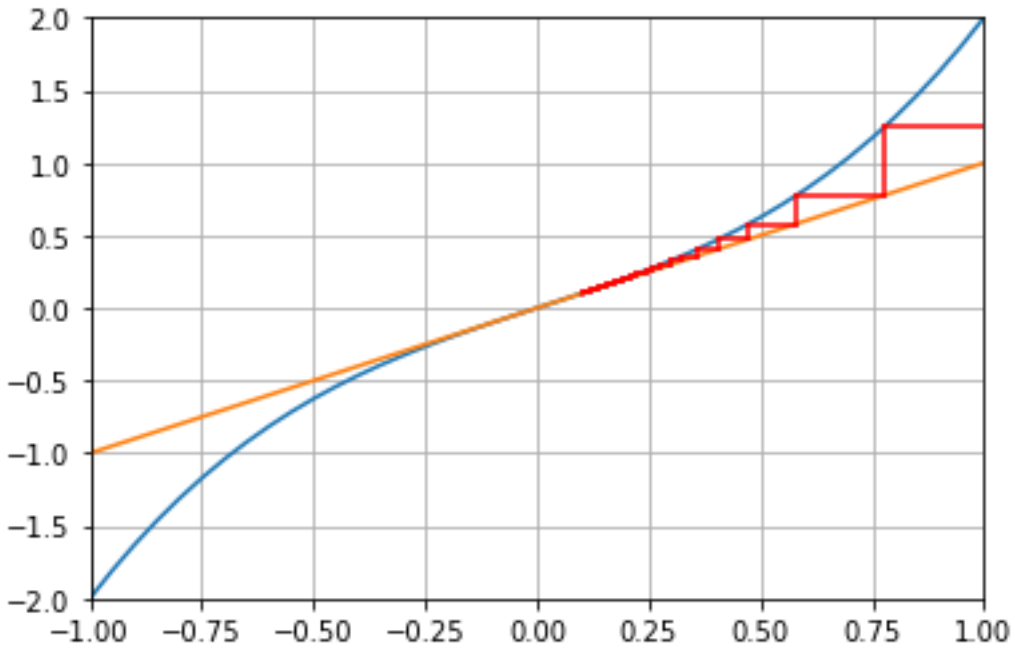


Figura 1.50: Órbitas de $x_0 = 0.1$, bajo la función $f(x) = x^3 + x$, donde puede verse que el 0 es un punto fijo con multiplicador asociado $\mu = 1$, pero con comportamiento repulsor.

- *Si* $c = 0.25$, $f_c(x)$ *tiene un único punto fijo indiferente en* $x = 1$.
- *Si* $c > 0.25$, $f_c(x)$ *no tiene puntos fijos.*

Bifurcación transcrítica. Ocurre cuando dos puntos fijos modifican su comportamiento.

Ejemplo 1.4. *La familia logística* $f_c(x) = cx(1 - x)$, *con* $c > 0$, *presenta una bifurcación transcrítica en* $c = 1$. *Los puntos fijos de* $f_c(x)$, *son* $x_1 = 0$ *y* $x_2 = 1 - \frac{1}{c}$. *Además, como la derivada es* $f_c'(x) = c(1 - 2x)$, *se tiene que:*

- *Si* $c < 1$, x_1 *es atractor y* x_2 *repulsor.*
- *Si* $c = 1$, $x = 0$ *es el único punto fijo indiferente.*
- *Si* $c > 1$, x_1 *es repulsor y* x_2 *atractor.*

Bifurcación horca. Se produce cuando un punto fijo atractor se modifica su comportamiento a repulsor, a la vez que aparecen dos nuevos puntos fijos de caracter atractor.

Ejemplo 1.5. *La familia de funciones* $f_c(x) = c(-x^3 + x)$, *con* $c > 0$, *presenta una bifurcación horca en* $c = 1$. *Los puntos fijos* $f_c(x) = x$, *son* $x_1 = 0$, $x_2 = \sqrt{\frac{c-1}{c}}$ *y* $x_3 = -\sqrt{\frac{c-1}{c}}$. *Ahora bien, como la derivada es* $f_c'(x) = c(-3x^2 + 1)$, *se tiene que:*

- *Si* $c = 1$, $f_c(x)$ *tiene un único punto fijo indiferente en* $x = 0$.
- *Si* $c > 1$, x_1 *es repulsor,* x_2 *atractor y* x_3 *también es atractor.*

Para poder estudiar visualmente el comportamiento y aparición de bifurcaciones, se hará uso del diagrama de Feigenbaum o diagrama de bifurcaciones, que se define en [71] de la siguiente manera:

Definición 1.15. *Un diagrama de bifurcación de un sistema dinámico, es una estratificación de su espacio de parámetros inducida por la equivalencia topológica, junto con los retratos de fase representativos de cada estrato. Las soluciones estables suelen representarse mediante líneas continuas, mientras que las soluciones inestables se representan con líneas punteadas.*

En la Figura 1.51 puede verse un ejemplo de un diagrama de Feigenbaum. Al interpretar el diagrama, no se puede obviar que, ocasionalmente, la escala empleada en el diagrama puede no permitir ver el rango de las bifurcaciones, lo cual no implica que no existan dichas bifurcaciones. Sin embargo, los exponentes de Lyapunov son una herramienta útil que ayuda a superar este problema relacionado con las escalas.

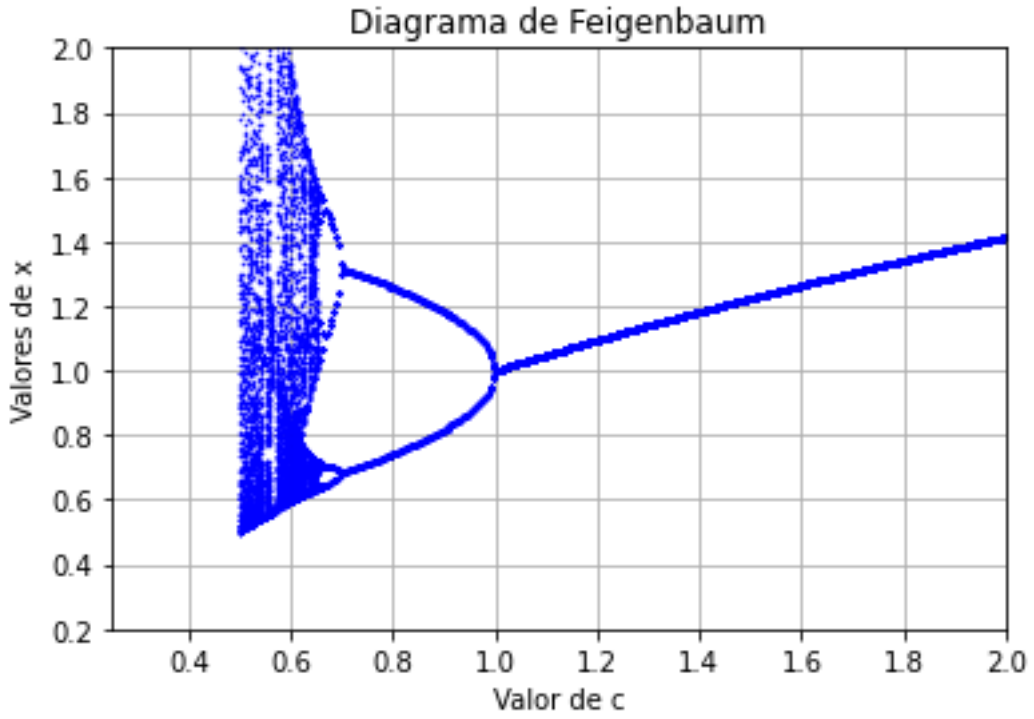


Figura 1.51: Diagrama de Feigenbaum que se obtiene al aplicar el método de dos pasos de Newton a la familia $f(x) = x^3 + cx + 1$, donde se observa que para algunos valores de $c \in (0.94, 1.01)$ se producen bifurcaciones.

Como se mencionó anteriormente, las órbitas próximas a un n-ciclo pueden acercarse o alejarse de este, dependiendo de si el ciclo actúa como un atractor o un repulsor. Además, la rapidez con la que ocurre este acercamiento o alejamiento está

determinada por la derivada de $f^n(x)$ en cualquier punto perteneciente al ciclo. En el caso de un ciclo $\{x_1, x_2, \ldots, x_n\}$, después de n iteraciones a partir de un punto próximo, la distancia entre este punto y el ciclo se habrá multiplicado por

$$|(f^n)'(x_1)| = \cdots = |(f^n)'(x_n)| = |f'(x_1)||f'(x_2)|\cdots|f'(x_n)|$$

Tras cada iteración, la variación media será

$$\sqrt[n]{|f'(x_1)||f'(x_2)|\cdots|f'(x_n)|}.$$

A las órbitas que no presentan ciclos, también se les aplica el razonamiento anterior.

Definición 1.16. *Sea la órbita $\{x_1, x_2, \ldots, x_n\}$, entonces se denomina número de Lyapunov de la órbita al valor*

$$L(x_1) = \lim_{n\to\infty} \sqrt[n]{|f'(x_1)||f'(x_2)|\cdots|f'(x_n)|}.$$

Notemos además, que este valor es el mismo para todos los elementos de la órbita.

El número de Lyapunov mide la tasa de expansión o contracción local asintótica en cada iteración dentro de la proximidad de una órbita.

Nota 1.3. *Si una órbita $\{x_1,\ x_2,\ \ldots,\ x_n\}$ es asintótica a una órbita periódica $\{y_1,\ y_2,\ \ldots,\ y_n\}$, es decir, $\lim_{n\to\infty}|x_n - y_n| = 0$, entonces $L(x_1) = L(y_1)$, siempre que ambos estén definidos.*

Una vez definido el número de Lyapunov, otro concepto relevante es el exponente de Lyapunov, el cual se formaliza de la siguiente manera.

Definición 1.17. *Sea la órbita $\{x_1,\ x_2,\ \ldots,\ x_n\}$, entonces llamaremos exponente de Lyapunov al valor*

$$h(x_1) = \lim_{n\to\infty} \frac{1}{n}\left(\log|f'(x_1)| + \log|f'(x_2)| + \cdots + \log|f'(x_n)|\right) = \log(L(x_1)).$$

Notemos además, que este valor es el mismo para todos los elementos de la órbita.

Cuando el exponente de Lyapunov asociado a x_1 toma un valor negativo, las órbitas de los puntos en su proximidad convergerán hacia la órbita de x_1. Por el contrario, si el exponente es toma un valor positivo, dichas órbitas se alejarán de x_1.

Nota 1.4. *Notar que $h(x)$ no está definida si $f'(x)$ se anula en algún punto de la órbita.*

Una forma de conectar el exponente de Lyapunov con el concepto de caos es la siguiente (véase [26], [71]).

Nota 1.5. *Se tiene que una órbita $\{x_1,\ x_2,\ \ldots,\ x_n\}$ es caótica si*

$$h(x_1) > 0, \text{ o lo que es lo mismo, } L(x_1) > 1.$$

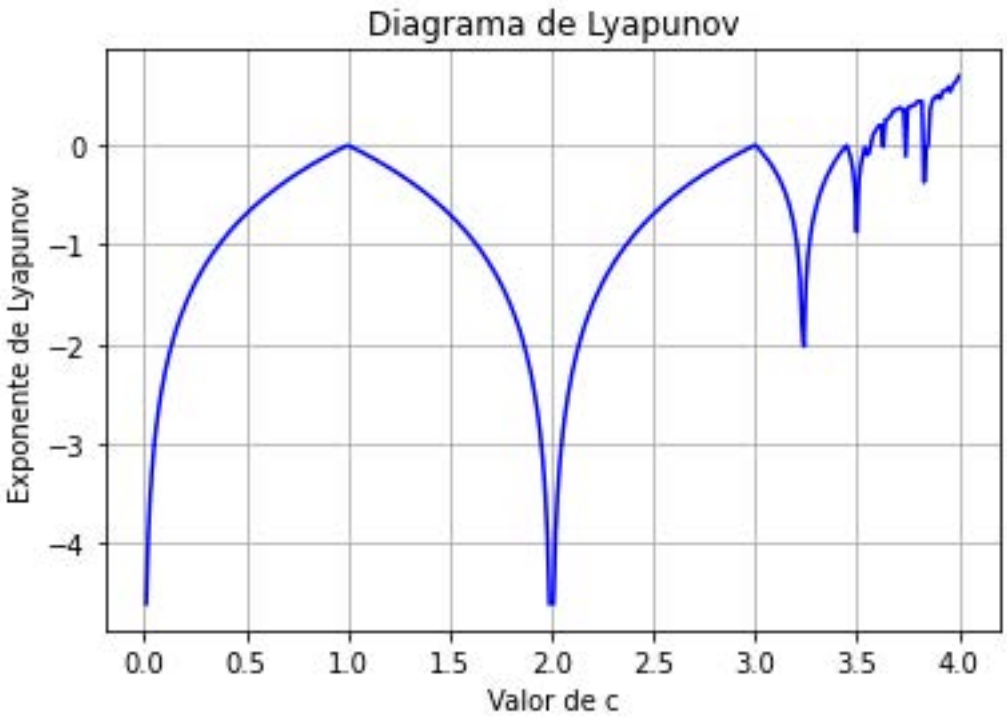


Figura 1.52: Exponentes de Lyapunov asociados al punto $\frac{1}{3}$ en función del parámetro $c \in [0, 4]$ para la función logística.

Ejemplo 1.6. *La familia logística está compuesta por las aplicaciones*

$$f_c : [0, 1] \to [0, 1],$$

definidas por la expresión $f_c(x) = cx(1 - x)$*, donde* $c \in \mathbb{R}$*. En la Figura 1.52 se representan los exponentes de Lyapunov correspondientes a la iteración del punto*

$$x_0 = \frac{1}{3}$$

en función del parámetro c. De manera resumida, se observa lo siguiente:

- *Cuando* $c \in (0, 3]$*, las iteraciones de* x_0 *convergen hacia una de las raíces de la función.*
- *Para* $c \in (3, 1+\sqrt{6}]$*, las iteraciones de* x_0 *terminan convergiendo a un 2-ciclo.*
- *Si* $c \in (1 + \sqrt{6}, c_\infty]$*, donde* $c_\infty \approx 3.5699$*, las iteraciones de* x_0 *conducen a un 4-ciclo.*
- *En el intervalo* $c_\infty < c \leq 4$*, la dinámica se vuelve mayoritariamente caótica.*
- *Cuando* $c \geq 4$*, las iteraciones de* x_0 *divergen.*

Todas las herramientas anteriores, son de gran ayuda, pero tienen sus limitaciones asociadas y, es por eso, que Magreñán en [105] presentó el "Plano de Convergencia", que recoge la misma información que las anteriores, de forma conjunta, y además permite de un vistazo poder observar el comportamiento en cada punto de la recta real.

Esta herramienta, además permite no sólo estudiar el comportamiento real de diferentes métodos iterativos, sino que también da la opción de estudiar métodos multipunto, tomando uno de los ejes del plano como los valores de uno de los puntos y el otro eje como los valores del otro punto. Así, pues el método de la secante puede ser estudiado para cualquier función como puede vers en la Figura 1.53, donde el eje horizontal se corresponde con valores del punto inicial x_0 y los verticales con valores del punto x_{-1}, como se presenta en [105] y se realiza en [31].

Así, cada par de puntos se corresponde con el comportamiento del método para dichos valores iniciales de los puntos iniciales y los colores asociados determinan los comportamientos, que pueden ser convergencia a las raíces, ciclos, divergencia, etc. Por ejemplo, en la Figura puede verse como el verde está asociado a la convergencia a la raíz $x = -1$, el rojo a la raíz $x = 1$ y el azul a la raíz $x = 0$, mientras que el negro en este caso se asocia con divergencia a dichas raíces, pero se podría adaptar de forma que mostrara otros comportamientos pormenorizados. Esta herramienta será de gran ayuda y será utilizada a lo largo de los siguientes capítulos.

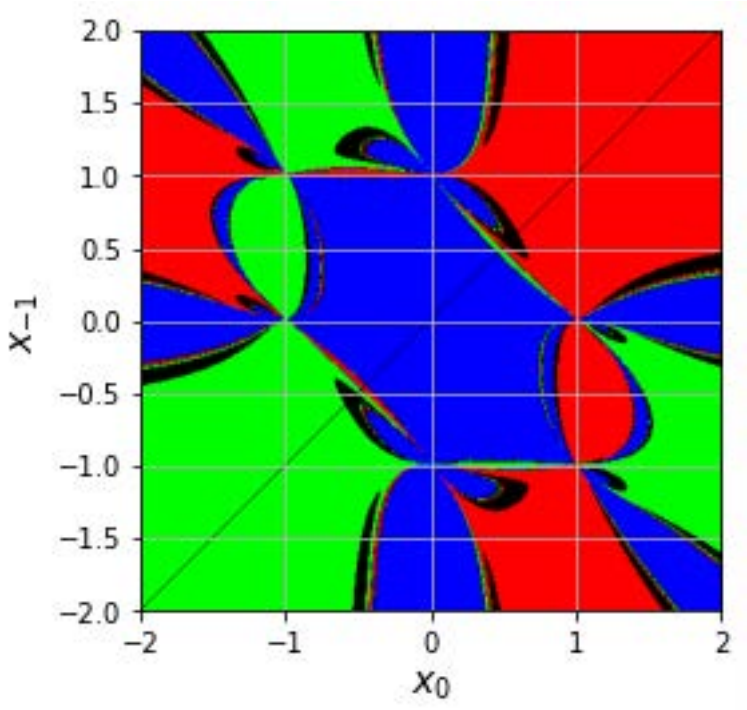


Figura 1.53: Plano de convergencia del método de la secante aplicado al polinomio $p(x) = x^3 + x|x| - 2x$.

1.5. Nociones básicas de operadores definidos en espacios de Banach

Capítulos posteriores están dedicado al estudio de métodos tipo secante aplicados a operadores en espacios de Banach. Por ello, a continuación se presentan algunos conceptos fundamentales que contribuirán a una mejor comprensión de los resultados expuestos en los capítulos posteriores. Para un tratamiento más detallado sobre los espacios de Banach, se puede recurrir a diversas referencias especializadas en la literatura matemática.

Entre las fuentes recomendadas se encuentran las tesis del grupo de Matemática Aplicada de la Universidad de La Rioja: Gutiérrez [76], Ezquerro [49], Rubio [143], Romero [142], de González [72], García-Olivo [68], Diloné [47] y la más reciente de Magreñán [68], que es la que se ha tomado como referencia, junto con los libros de Argyros [10] y [11]. Asimismo, los textos de Kantorovich [97] y Rall [136] ofrecen una presentación de los espacios de Banach desde una perspectiva alineada con la adoptada en este trabajo. Por último, Se recomienda consultar la siguiente bibliografía [12, 13, 15, 14, 16, 17, 20, 24, 44, 51, 52, 55, 56, 73, 80, 82, 84, 89, 101, 109, 110, 111, 122, 123, 124, 125, 131, 133, 134, 159, 158, 160, 161, 162], entre otras fuentes.

Sea $F : X \to Y$ un operador definido entre dos espacios de Banach X e Y. Nos interesa examinar la resolución de la ecuación $F(x) = 0$, bajo ciertas hipótesis exigidas sobre el operador F. A partir de este punto, denotaremos por $\mathcal{L}(X, Y)$ al conjunto de operadores lineales y acotados que actúan entre los espacios X e Y. En el caso particular en que $X = Y$, utilizaremos la notación $\mathcal{L}(X)$.

Sea $L \in \mathcal{L}(X, Y)$, se denotará por $\mathcal{D}(L)$ a su dominio y por $\mathcal{R}(L)$ a su rango. Si existe un operador L^{-1} que actúa sobre $\mathcal{R}(L)$ para llevarlo a $\mathcal{D}(L)$, de forma que

$$L^{-1}Lx = x, \quad \text{para todo } x \in \mathcal{D}(L),$$

$$LL^{-1}y = y, \quad \text{para todo } y \in \mathcal{R}(L),$$

entonces se dice que L es *invertible* y que L^{-1} es el *inverso* de L. Si el operador L^{-1} existe, también es un operador lineal.

Seguidamente, se muestran algunos resultados bien establecidos sobre la inversión de operadores. Es importante destacar que estos resultados se formulan para operadores definidos en un espacio X que actúan sobre sí mismos.

Lema 1.4. (Banach) *Sea $L \in \mathcal{L}(X)$ verificando que*

$$\|L\| \leq k < 1,$$

entonces el operador $I - L$ tiene inverso continuo y, además,

$$\|(I - L)^{-1}\| \leq \frac{1}{1 - k}.$$

Unas ligeras variantes del lema anterior son los siguientes resultados.

Lema 1.5. *Sea $L \in \mathcal{L}(X)$. Existe L^{-1} si y sólo si existe un operador invertible $M \in \mathcal{L}(X)$ tal que*

$$\|I - ML\| < 1.$$

En este caso,

$$L^{-1} = \sum_{n=0}^{\infty} (I - ML)^n M$$

y

$$\|L^{-1}\| \leq \frac{\|M\|}{1 - \|I - ML\|}.$$

Lema 1.6. *Sea $L \in \mathcal{L}(X)$. Existe L^{-1} si y sólo si existe un operador invertible $M \in \mathcal{L}(X)$ tal que*

$$\|M - L\| < \frac{1}{\|M^{-1}\|}.$$

En este caso,

$$L^{-1} = \sum_{n=0}^{\infty} (I - M^{-1}L)^n M^{-1}$$

y

$$\|L^{-1}\| \leq \frac{\|M^{-1}\|}{1 - \|I - M^{-1}L\|} \leq \frac{\|M^{-1}\|}{1 - \|M^{-1}\|\|M - L\|}.$$

A continuación, se introducen algunos conceptos y resultados básicos del cálculo diferencial e integral en espacios de Banach.

Definición 1.18. *Dado $x_0 \in X$, si existe un operador $L \in \mathcal{L}(X, Y)$ de manera que, para todo $x \in X$,*

$$\lim_{h \to 0} \frac{F(x_0 + hx) - F(x_0)}{h} = L(x), \tag{1.5.1}$$

entonces F se dice diferenciable Gâteaux (o diferenciable débilmente) en x_0. En esta situación, el operador lineal L es la derivada de Gâteaux (o la diferencial de Gâteaux) de F en x_0, y se denota $L = dF(x_0)$.

Definición 1.19. *Si el límite de la ecuación* (1.5.1) *es uniforme en el conjunto $\{x \in X;\ \|x\| = 1\}$, entonces F se dice diferenciable Fréchet, o simplemente diferenciable en x_0. En este caso, el operador lineal L se llama derivada de Fréchet (o diferencial de Fréchet) de F en x_0, y se denota $L = F'(x_0)$.*

Equivalentemente, el concepto de diferenciabilidad se puede expresar de la siguiente forma. Si dado $x_0 \in X$, existe un operador lineal y continuo $L = F'(x_0)$ de manera que

$$\lim_{\|v\|\to 0} \frac{\|F(x_0+v) - F(x_0) - Lv\|}{\|v\|} = 0,$$

entonces F es diferenciable Fréchet en x_0.

Obsérvese que si F es un operador diferenciable Fréchet en x_0, entonces también es diferenciable Gâteaux en x_0. Además, $F'(x_0) = dF(x_0)$. Salvo que se indique lo contrario, cuando nos refiramos a un operador diferenciable, lo entenderemos en el sentido de Fréchet.

Es bien conocido que las propiedades de la diferencial en espacios de Banach son análogas a las de la derivada en el caso escalar, salvo el Teorema del Valor Medio. El siguiente resultado generaliza el Teorema del Valor Medio conocido para funciones escalares.

Teorema 1.7. (del Valor Medio) *Sean X e Y dos espacios de Banach. Sea $F : X \to Y$ un operador diferenciable en un conjunto convexo $\Omega \subseteq X$. Entonces, si $x_0, x_1 \in \Omega$, se tiene*

$$\|F(x_1) - F(x_0)\| \leq \sup_{0<\theta<1} \|F'(x_0 + \theta(x_1 - x_0))\| \|x_1 - x_0\|.$$

Corolario 1.8. *Con la notación anterior se tiene que*

$$\begin{aligned} &\|F(x_1) - F(x_0) - F'(x_0)(x_1 - x_0)\| \\ &\leq \sup_{0<\theta<1} \|F'(x_0 + \theta(x_1 - x_0)) - F'(x_0)\| \|x_1 - x_0\|. \end{aligned}$$

A continuación, se discuten algunos aspectos relacionados con el cálculo integral en espacios de Banach.

Definición 1.20. *Sea F definida en un intervalo real $[a, b]$ y con valores en un espacio de Banach Y. Entonces podemos definir su integral como el límite de las siguientes sumas*

$$\sum_{k=0}^{n-1} F(\tau_k)(t_{k+1} - t_k),$$

donde $a = t_0 < t_1 < \cdots < t_n = b$ y $\tau_k \in [t_k, t_{k+1}]$, cuando $\max_k\{t_{k+1} - t_k\} \to 0$. Si el límite dirigido anterior existe, lo llamaremos integral *de F y lo denotaremos*

$$\int_a^b F(t)\, dt.$$

Evidentemente, si la integral existe, es un elemento de Y. Una condición suficiente para que exista dicha integral es que la función F sea continua en el intervalo $[a, b]$.

Las propiedades de esta integral se deducen de las conocidas para la integral de Riemann en el caso real. De entre ellas, destacamos las tres siguientes por su posterior utilidad.

(i) *Si consideramos un operador lineal y acotado $L \in \mathcal{L}(Y, Z)$, entonces*

$$\int_a^b L(F(t))\,dt = L\left(\int_a^b F(t)\,dt\right).$$

(ii) *Si $F(t) = \phi(t)y_0$, donde y_0 es un elemento fijo de Y y $\phi(t)$ es una función real integrable, entonces*

$$\int_a^b F(t)\,dt = y_0 \int_a^b \phi(t)\,dt.$$

(iii) $\left\|\int_a^b F(t)\,dt\right\| \leq \int_a^b \|F(t)\|\,dt.$

Definición 1.21. *Supongamos ahora que T es un operador definido en un segmento $[x_0, x_1] \subseteq X$ y con valores en el espacio $\mathcal{L}(X, Y)$. En este caso, se define su integral*

$$\int_{x_0}^{x_1} T(x)\,dx = \int_0^1 T(x_0 + t(x_1 - x_0))(x_1 - x_0)\,dt.$$

Los siguientes resultados relacionados con el cálculo integral en espacios de Banach pueden encontrarse en los textos de Kantorovich [97] y Rall [136].

Teorema 1.9. *Sea F un operador de X en Y que tiene derivada continua en el segmento $[x_0, x_1] \subseteq X$. Entonces*

$$\int_{x_0}^{x_1} F'(x)\,dx = F(x_1) - F(x_0).$$

Lema 1.10. *Si se verifica*

$$\|F(x_0 + \tau(x_1 - x_0))\| \leq \phi(t_0 + \tau(t_1 - t_0)), \quad \tau \in [0, 1]$$

y

$$\|x_1 - x_0\| \leq t_1 - t_0,$$

entonces

$$\left\|\int_{x_0}^{x_1} F(x)\,dx\right\| \leq \int_{t_0}^{t_1} \phi(t)\,dt.$$

Como variante del lema anterior, tenemos el siguiente resultado.

Corolario 1.11. *Si se cumple la desigualdad*

$$\|F(x)\| \leq \phi(t)$$

para x y t tales que

$$\|x - x_0\| \leq t - t_0,$$

entonces

$$\left\|\int_{x_0}^{x_1} F(x)\,dx\right\| \leq \int_{t_0}^{t_1} \phi(t)\,dt,$$

donde x_1 es un elemento que cumple $\|x_1 - x_0\| \leq t_1 - t_0$.

Si $F : X \to Y$ es un operador dos veces diferenciable en un punto $x_0 \in X$, entonces se tiene que $F''(x_0) \in \mathcal{B}(X^2, Y)$, donde $\mathcal{B}(X^2, Y)$ representa el conjunto de operadores bilineales acotados de $X^2 = X \times X$ en Y, con la norma definida como

$$\|B\| = \sup_{\|x_1\|,\|x_2\|\leq 1} \|B(x_1, x_2)\|, \quad B \in \mathcal{B}(X^2, Y).$$

Se sabe (véase [34]) que entre los espacios $\mathcal{B}(X^2, Y)$ y $\mathcal{L}(X, \mathcal{L}(X, Y))$ existe una isometría. De ahora en adelante, teniendo en cuenta esta isometría, se identificarán los elementos de ambos espacios. También se sabe, según [136], que $F''(x_0)$ es un operador bilineal simétrico, es decir,

$$F''(x_0)x_1x_2 = F''(x_0)x_2x_1 \quad \text{para todo } x_1, x_2 \in X.$$

Para un operador bilineal de $\mathbb{R}^2 \times \mathbb{R}^2$ en $\mathbb{R}^2$, se utiliza la notación basada en una matriz de bloques, según lo presentado en [10] y [11].

$$B = \left(\begin{array}{cc} b_1^{11} & b_1^{12} \\ b_1^{21} & b_1^{22} \\ \hline b_2^{11} & b_2^{12} \\ b_2^{21} & b_2^{22} \end{array}\right). \tag{1.5.2}$$

Entonces, si $x = (x_1, x_2)$ y $y = (y_1, y_2)$, se cumple que

$$\begin{aligned} B(x, y) &= \begin{pmatrix} x_1, x_2 \end{pmatrix} \left(\begin{array}{cc} b_1^{11} & b_1^{12} \\ b_1^{21} & b_1^{22} \\ \hline b_2^{11} & b_2^{12} \\ b_2^{21} & b_2^{22} \end{array}\right) \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \\ &= \begin{pmatrix} b_1^{11}x_1 + b_1^{21}x_2 & b_1^{12}x_1 + b_1^{22}x_2 \\ b_2^{11}x_1 + b_2^{21}x_2 & b_2^{12}x_1 + b_2^{22}x_2 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \\ &= \begin{pmatrix} b_1^{11}x_1y_1 + b_1^{21}x_2y_1 + b_1^{12}x_1y_2 + b_1^{22}x_2y_2 \\ b_2^{11}x_1y_1 + b_2^{21}x_2y_1 + b_2^{12}x_1y_2 + b_2^{22}x_2y_2 \end{pmatrix}. \end{aligned}$$

En $\mathbb{R}^2$ consideramos la norma del máximo

$$\|(x_1, x_2)\| = \text{máx}\{|x_1|, |x_2|\},$$

y en $\mathcal{L}(\mathbb{R}^2, \mathbb{R}^2)$ la norma dada por

$$\|L\| = \text{máx}\{|a_{11}| + |a_{12}|, |a_{21}| + |a_{22}|\},$$

con

$$L = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}.$$

De este modo, se puede comprobar directamente [135] que la norma del operador bilineal B, definido por la ecuación (1..52), viene dada por

$$\|B\| = \sup_{\|x\|=1} \max_i \sum_{j=1}^{2} \left| \sum_{k=1}^{2} b_i^{jk} x_k \right|.$$

Además, se obtiene la siguiente estimación para la norma anterior

$$\|B\| \leq \max \{|b_1^{11}| + |b_1^{12}| + |b_1^{21}| + |b_1^{22}|, |b_2^{11}| + |b_2^{12}| + |b_2^{21}| + |b_2^{22}|\}.$$

La composición de una aplicación bilineal B con una aplicación lineal L da lugar a una nueva aplicación bilineal, la cual se denota como LB. De acuerdo con la notación previamente establecida, la matriz que representa a LB está dada por

$$LB = \left(\begin{array}{cc} a_{11}b_1^{11} + a_{12}b_2^{11} & a_{11}b_1^{12} + a_{12}b_2^{12} \\ a_{11}b_1^{21} + a_{12}b_2^{21} & a_{11}b_1^{22} + a_{12}b_2^{22} \\ \hline a_{21}b_1^{11} + a_{22}b_2^{11} & a_{21}b_1^{12} + a_{22}b_2^{12} \\ a_{21}b_1^{21} + a_{22}b_2^{21} & a_{21}b_1^{22} + a_{22}b_2^{22} \end{array} \right).$$

Si F es un operador que mapea $\mathbb{R}^2$ en $\mathbb{R}^2$ y asocia a cada par $(x, y) \in \mathbb{R}^2$ un nuevo par $(f(x, y), g(x, y)) \in \mathbb{R}^2$, entonces su segunda derivada en un punto (x_0, y_0) es un operador que actúa de $\mathbb{R}^2 \times \mathbb{R}^2$ en $\mathbb{R}^2$. Este operador puede representarse mediante una matriz de la forma (1.2):

$$\left(\begin{array}{cc} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \\ \hline g_{xx} & g_{xy} \\ g_{yx} & g_{yy} \end{array} \right),$$

donde las derivadas parciales indicadas se encuentran evaluadas en el punto (x_0, y_0).

Se finaliza esta sección en la que se dan una introducción a los espacios de Banach con la aplicación del Teorema de Taylor.

Teorema 1.12. (Taylor) *Supongamos que F es un operador n-veces diferenciable en la bola $B(x_0, r)$, $r > 0$, y que $F^{(n)}$ es integrable en el segmento $[x_0, x_1]$ con $x_1 \in B(x_0, r)$. Entonces*

$$F(x_1) = F(x_0) + \sum_{k=1}^{n-1} \frac{1}{k!} F^{(k)}(x_0)(x_1 - x_0)^k + R_n(x_0, x_1)$$

donde

$$\begin{aligned} R_n(x_0, x_1) &= \frac{1}{(n-1)!} \int_{x_0}^{x_1} F^{(n)}(x)(x_1 - x)^{n-1}\, dx \\ &= \frac{1}{(n-1)!} \int_0^1 F^{(n)}(x_0 + t(x_1 - x_0))(x_1 - x_0)^n (1-t)^{n-1}\, dt. \end{aligned}$$

1.6. Elección del Entorno de Desarrollo: Python como Herramienta para Métodos Numéricos

En el contexto de la presente investigación, dedicada al estudio y desarrollo de métodos numéricos para la resolución de ecuaciones no lineales, se ha optado por la utilización del lenguaje de programación Python como entorno principal de desarrollo. Este capítulo expone los motivos de dicha elección, así como las ventajas específicas que Python aporta a este tipo de investigaciones frente a otros entornos de cálculo comúnmente utilizados en trabajos similares, como MATLAB o Mathematica.

1.6.1. Ventajas de Python frente a Otros Entornos de Cálculo Científico

Python es un lenguaje de programación de código abierto, ampliamente reconocido en la comunidad científica y con una creciente base de usuarios y desarrolladores. Este enfoque de software libre ofrece importantes ventajas en términos de coste y accesibilidad, ya que permite a los investigadores y a la comunidad científica utilizar y compartir herramientas sin necesidad de licencias costosas. Además, Python es un lenguaje multiplataforma, compatible con sistemas operativos como Windows, macOS y Linux, lo cual facilita la portabilidad de los proyectos y permite a los usuarios replicar los experimentos y realizar análisis sin restricciones en el entorno operativo.

Otra ventaja significativa de Python es su capacidad de integración con otras herramientas y lenguajes de programación, gracias a sus numerosas bibliotecas y APIs. Esto permite, entre otras cosas, la incorporación de scripts de Python en aplicaciones como Excel, que recientemente ha integrado soporte para el lenguaje de forma nativa. Esta incorporación abre nuevas posibilidades, como la ejecución de cálculos complejos en la nube mediante servicios como Azure, lo que expande el ámbito de aplicación de Python y aumenta la flexibilidad y escalabilidad del procesamiento de datos y análisis numéricos.

1.6.2. Herramientas y Librerías Especializadas para Cálculo Numérico en Python

Python dispone de una amplia gama de bibliotecas especializadas para el cálculo numérico y la ciencia de datos, tales como `NumPy`, `SciPy` y `SymPy`. Estas bibliotecas proporcionan una infraestructura sólida para la implementación de métodos numéricos, permitiendo la creación de algoritmos eficaces para la resolución de ecuaciones no lineales. `NumPy`, por ejemplo, facilita el manejo de arrays multidimensionales y la realización de operaciones algebraicas vectorizadas, características esenciales para mejorar el rendimiento computacional en cálculos iterativos.

Por otro lado, `SciPy` ofrece módulos específicos para la resolución de ecuaciones no lineales, incluyendo algoritmos como el método de Newton, el método de Halley, la secante y otros métodos libres de derivadas. `SymPy`, una biblioteca de álgebra simbólica, permite realizar diferenciación y manipulación simbólica, lo cual resulta

útil en la implementación de ciertos métodos que requieren el cálculo de derivadas o la simplificación de expresiones algebraicas. Estas herramientas permiten la construcción de soluciones robustas y precisas, además de ofrecer flexibilidad en la implementación de métodos numéricos específicos adaptados a las necesidades de la investigación.

1.6.3. Implementación de Métodos Numéricos en Python

En este trabajo, se han desarrollado e implementado varios métodos numéricos para la resolución de ecuaciones no lineales, tales como el método de Newton, el método de Halley, y el método de la secante, entre otros. Estos métodos se han programado utilizando las bibliotecas mencionadas anteriormente, aprovechando sus características de optimización de cálculos y manipulación algebraica. Python permite, además, una implementación clara y legible de estos algoritmos, lo cual facilita la validación del código y la verificación de resultados. La elección de Python para este propósito se basa en su flexibilidad y en su adaptabilidad a diferentes tipos de métodos numéricos, lo cual permite realizar comparaciones y ajustes de manera ágil.

Los métodos implementados han sido diseñados no solo para resolver ecuaciones no lineales en diferentes contextos y con diferentes condiciones iniciales, sino también para evaluar la sensibilidad de estos métodos a través de diagramas de órbitas, diagramas de Lyapunov y planos dinámicos. Python ofrece herramientas adicionales para la visualización de estos resultados mediante bibliotecas como `Matplotlib` y `Seaborn`, que permiten una representación gráfica detallada y personalizable de los comportamientos de convergencia y de estabilidad de los métodos empleados.

1.6.4. Python como Herramienta Escalable para Investigación Científica

La capacidad de Python para integrarse con plataformas de computación en la nube, tales como Microsoft Azure, y su reciente inclusión como lenguaje nativo en Microsoft Excel, ofrece nuevas posibilidades en términos de escalabilidad y accesibilidad de la computación científica. Estas características permiten que el código desarrollado pueda ejecutarse en infraestructuras de alto rendimiento y procesar grandes volúmenes de datos, lo cual es particularmente útil en experimentos numéricos de alta complejidad.

Python se posiciona así como una herramienta versátil que, al ser abierta y extensible, se adapta a las necesidades cambiantes de los proyectos de investigación. La posibilidad de acceder a los recursos en la nube y de automatizar procesos en entornos como Excel amplía las aplicaciones de los métodos numéricos aquí desarrollados, permitiendo su uso no solo en un contexto académico, sino también en el ámbito industrial y empresarial.

1.6.5. Conclusión

La elección de Python como entorno de desarrollo para esta investigación ofrece múltiples ventajas que justifican su utilización frente a otras opciones tradicionales como MATLAB o Mathematica. La combinación de su carácter abierto, su vasta comunidad de usuarios y desarrolladores, y sus potentes bibliotecas científicas, hace de Python una herramienta ideal para el desarrollo de métodos numéricos complejos y su aplicación en la resolución de ecuaciones no lineales. A ello se suma su reciente integración en entornos de computación en la nube y su compatibilidad multiplataforma, lo cual refuerza su utilidad y capacidad de expansión para la investigación científica actual y futura.

1.7. Código en Python

En este capítulo se recoge el código desarrollado en Python que constituye una de las principales herramientas computacionales utilizadas en esta investigación. A lo largo de su implementación, se han considerado cuidadosamente aspectos clave de la estructura y eficiencia del código, así como la elección de parámetros y configuraciones específicos, de manera que estos permitan una ejecución precisa y reproducible de los experimentos numéricos planteados. Este código no solo facilita la obtención y visualización de los resultados, sino que también permite replicar y validar el análisis teórico expuesto en capítulos anteriores. Asimismo, se incluyen comentarios y explicaciones detalladas sobre cada fragmento de código para asegurar que cualquier lector o investigador interesado pueda comprender el flujo de trabajo y realizar adaptaciones o extensiones en futuras investigaciones, brindando así una base sólida para exploraciones adicionales en el área estudiada.

1.7.1. Plano dinámico método de Newton

```
# Plano Din mico  m todo de Newton z^3-1
import numpy as np
import matplotlib.pyplot as plt

F = lambda Z: (Z**3 - 1)
DF = lambda Z: 3 * Z**2
M = lambda Z, L: Z - L * F(Z) / DF(Z)

# L, par metro de amortiguamiento
# reZ, parte real de los puntos del plano din mico
# imZ, parte imaginaria de los puntos del plano din mico
# maxiter, n mero m ximo de iteraciones
# Z, valor final de la  rbita  de cada punto del plano complejo
# I, imagen del plano din mico
# dZ, control de la tolerancia
# tol, valor de tolerancia establecido
# h, tama o de paso del mallado

h = 0.01
tol = 1E-4
maxiter = 20
L = 1
reZ = np.arange(-3, 3 + h, h)
imZ = np.arange(-3, 3 + h, h)
RZ, IZ = np.meshgrid(reZ, imZ)
Z = RZ + 1j * IZ
itera = 1

# R G B para los colores del plano din mico
# I, para combinar R G B
R = np.zeros([len(reZ), len(imZ)])
G = np.zeros([len(reZ), len(imZ)])
B = np.zeros([len(reZ), len(imZ)])
I = np.zeros([len(reZ), len(imZ), 3])
IT = maxiter * np.ones([len(reZ), len(imZ)])

# Puntos fijos
ZF1 = -1/2 + (3)**0.5 / 2 * 1j
ZF2 = 1
ZF3 = -1/2 - (3)**0.5 / 2 * 1j

while itera <= maxiter:
    Z0 = Z
    Z = M(Z0, L)
    itera = itera + 1
    for fil in range(len(reZ)):
        for col in range(len(imZ)):
            Zfc = Z[fil, col]
            if np.abs(Zfc - ZF1) < tol:
                R[fil, col] = 1
            if np.abs(Zfc - ZF2) < tol:
                G[fil, col] = 1
            if np.abs(Zfc - ZF3) < tol:
                B[fil, col] = 1
I[..., 0] = R
I[..., 1] = G
```

```
I[..., 2] = B

# Se muestra el plano din mico
plt.figure()
plt.title('Plano din mico')
plt.plot(np.real(ZF1), np.imag(ZF1), 'w*')
plt.plot(np.real(ZF2), np.imag(ZF2), 'w*')
plt.plot(np.real(ZF3), np.imag(ZF3), 'w*')
plt.xlabel("$Re(x_{0})$", fontsize=14)
plt.ylabel("$Im(x_{0})$", fontsize=14)
plt.grid()
plt.imshow(I, extent=[reZ[0], reZ[-1], imZ[0], imZ[-1]])
```

1.7.2. Plano dinámico método de Halley

```
#  Plano Din mico  m todo de Halley z^4-1
import numpy as np
import matplotlib.pyplot as plt

F= lambda Z: (Z**4-1)
DF= lambda Z: 4*Z**3
DF2= lambda Z: 12*Z**2
LF= lambda Z:  F(Z)*DF2(Z)/DF(Z)**2
M= lambda Z, L: Z-L*(2/(2-LF(Z)))*F(Z)/DF(Z)

#  L, par metro de amortiguamiento
#  reZ, parte real de los puntos del plano din mico
#  imZ, parte imaginaria de los puntos del plano din mico
#  maxiter, n mero  m ximo de iteraciones
#  Z, valor final de la  rbita  de cada punto del plano complejo
#  I, imagen del plano din mico
#  dZ, control de la tolerancia
#  tol, valor de tolerancia establecido
#  h, tama o de paso del mallado

h=0.01
tol=1E-4
maxiter=20
L=1
reZ=np.arange(-3,3+h,h)
imZ=np.arange(-3,3+h,h)
RZ, IZ=np.meshgrid(reZ,imZ)
Z=RZ+1j*IZ
itera=1
#  R G B para los colores del plano din mico
#  I, para combinar R G B
R=np.zeros([len(reZ),len(imZ)])
G=np.zeros([len(reZ),len(imZ)])
B=np.zeros([len(reZ),len(imZ)])
I=np.zeros([len(reZ),len(imZ),3])
IT=maxiter*np.ones([len(reZ),len(imZ)])
#  Puntos fijos
ZF1=-1
ZF2=1
ZF3=-1j
ZF4=1j

while itera<=maxiter:
    Z0=Z
    Z=M(Z0,L)
    itera=itera+1
    for fil in range(len(reZ)):
        for col in range(len(imZ)):
            Zfc=Z[fil,col]
            if np.abs(Zfc-ZF1)<tol:
                R[fil,col]=1
            if np.abs(Zfc-ZF2)<tol:
                G[fil,col]=1
            if np.abs(Zfc-ZF3)<tol:
                B[fil,col]=1
            if np.abs(Zfc-ZF4)<tol:
```

```
                R[fil,col]=1
                G[fil,col]=1
I[...,0]=R
I[...,1]=G
I[...,2]=B

# Se muestra el plano din mico
plt.figure()
plt.title('Plano din mico')
plt.plot(np.real(ZF1),np.imag(ZF1),'w*')
plt.plot(np.real(ZF2),np.imag(ZF2),'w*')
plt.plot(np.real(ZF3),np.imag(ZF3),'w*')
plt.plot(np.real(ZF4),np.imag(ZF4),'w*')
plt.xlabel("$Re(x_{0})$", fontsize=14)
plt.ylabel("$Im(x_{0})$", fontsize=14)
plt.grid()
plt.imshow(I, extent=[reZ[0],reZ[-1],imZ[0],imZ[-1]])
```

1.7.3. Diagrama de órbita

```
#  Importaci n de paquetes
import numpy as np
import matplotlib.pyplot as plt

#  Funci n que hace las iteraciones para dibujar el diagrama
#  Llamable desde un script exterior
def orbita(f,x,x0,n):
    f1= lambda x: x  #  Funci n f(x)=x para dibujar en el diagrama
    y=f(x)
    y1=f1(x)
    plt.plot(x,y,x,y1)
    plt.xlim(x[0],x[-1])
    plt.ylim(x[0],x[-1])
    plt.grid()
    X=x0  #  Se asigna el valor de la semilla
    for i in range(n):
        Y=f(X)  # Valor de la funci n correspondiente a la semilla
        plt.plot((X, X), (X, Y),'r')  #  Recta vertical desde y=x
            hasta funci n
        plt.plot((X, Y), (Y, Y),'r')  #  Recta horizontal desde
            funci n hasta y=0
        X=Y  #  Actualizaci n de valor X para continuar bucle for
    plt.show()

#  Funci n sobre la que calcular el diagrama
f= lambda x: -(x-0.5)**2+1#  Funci n a evaluar
n= 50  #  N mero de iteraciones
a=-1 # Inicio dominio para graficar
b= 1.5  # Fin domino para graficar
h= 0.05  # Tama o de paso
x=np.arange(a,b+h,h)
x0=-0.25 #  Semilla
orbita(f,x,x0,n)
```

1.7.4. Diagrama de Feigenbaum aplicado a la función Logística

```
import numpy as np
import matplotlib.pyplot as plt

maxiter = 500  # N mero m ximo de iteraciones
F = lambda c, X: c * X * (1 - X)  # Familia log stica
h=0.005 #Tama o de paso
c = np.arange(2.4, 4+h, h)  # Par metro de la familia log stica
plt.axis([c[0], c[-1], 0, 1]) #Ejes del gr fico
x0 = 0.1 #Semilla
X0 = np.full(len(c), x0)  # Semilla

for it in range(maxiter):
    X = F(c, X0)
    if it > maxiter/2: #Se grafica a partir de la mitad de las
        iteraciones
        plt.plot(c, X, "b.", markersize=1)

    X0 = X

plt.xlabel('Valor de c')
plt.ylabel('Valores de x')
plt.title('Diagrama de Feigenbaum')
plt.grid(True)
plt.show()
```

1.7.5. Diagrama de Feigenbaum aplicado al método de Newton

```
import numpy as np
import matplotlib.pyplot as plt

maxiter = 500  # N mero m ximo de iteraciones
F = lambda c, X: X**2-c # Familia
DF= lambda c, X: 2*X-1
M= lambda c, X: X-F(c,X)/DF(c,X)

h=0.005 #Tama o de paso
c = np.arange(0.25, 2+h, h)  # Par metro de la familia log stica
plt.axis([c[0], c[-1], 0.2, 2]) #Ejes del gr fico
x0 = 1 #Semilla
X0 = np.full(len(c), x0)  # Semilla

for it in range(maxiter):
    X = M(c, X0)
    if it > maxiter/2: #Se grafica a partir de la mitad de las
        iteraciones
         plt.plot(c, X, "b.", markersize=1)

    X0 = X

plt.xlabel('Valor de c')
plt.ylabel('Valores de x')
plt.title('Diagrama de Feigenbaum')
plt.grid(True)
plt.show()
plt.show()
```

1.7.6. Diagrama de Lyapunov

```
import numpy as np
import matplotlib.pyplot as plt

# f funcion que define el sistema din mico
F = lambda x, c: c * x * (1 - x)

# Par metros
h=0.01 #Tama o de paso
c_rango = np.arange(0, 4+h, h)  # Rango de c
x0 = 1/3  # Semilla
maxiter = 1000  # N de iteraciones
iter_transi = 100  # N mero de iteraciones de transici n

# Array para valores de Lyapunov
lyapunov = np.zeros(len(c_rango))

# Calcula el diagrama de Lyapunov
for i, c in enumerate(c_rango):
    x = x0
    sum_lyapunov = 0
    for j in range(maxiter + iter_transi):
        x = F(x, c)
        if j >= iter_transi: #Se consideran las que iteran por encima
             de iter_tranis
            sum_lyapunov += np.log(np.abs(c - 2 * c * x)) # log valor
                absoluto de la derivada de f(x)
    lyapunov[i] = sum_lyapunov/maxiter

# Grafica el diagrama de Lyapunov
plt.plot(c_rango, lyapunov,"b")
plt.xlabel('Valor de c')
plt.ylabel('Exponente de Lyapunov')
plt.title('Diagrama de Lyapunov')
plt.grid()
plt.show()
```

1.8. Organización de la tesis

Los capítulos presentados a continuación abordan diversos aspectos teóricos y aplicados de los métodos iterativos tipo secante, que desempeñan un papel crucial en la resolución de ecuaciones no lineales en distintos campos de las matemáticas y la computación científica. Estos métodos, al no requerir la evaluación explícita de derivadas, son particularmente útiles en contextos donde el cálculo de derivadas es costoso o incluso impracticable. A lo largo de estos capítulos, se analiza la evolución y optimización de estos métodos iterativos, desde sus formulaciones clásicas hasta versiones mejoradas con mayor eficiencia y accesibilidad.

El Capítulo 2 "Aproximaciones iniciales para una familia de métodos tipo secante", se centra en la importancia de la selección de valores iniciales en la aplicabilidad del método de la secante. Se parte del hecho de que, aunque este método ofrece una convergencia superlineal y evita la necesidad de calcular derivadas, su desempeño puede ser sensible a la elección de los puntos iniciales. Para abordar este desafío, se introduce una familia uniparamétrica de métodos iterativos que interpolan entre el método de la secante y el método de Newton, permitiendo mejorar la velocidad de convergencia sin comprometer la eficiencia computacional. Además, se propone una estrategia basada en la descomposición del operador no lineal en dos componentes: una diferenciable y otra continua pero no diferenciable. A través de esta descomposición, se consigue una mejor accesibilidad del método, ampliando la región de convergencia de las iteraciones. El capítulo también incluye un análisis teórico de la convergencia local, definiendo condiciones que garantizan la existencia de una bola de convergencia y permitiendo estimar el radio de esta región en función de las propiedades del operador. Los resultados de este capítulo han sido publicados en [117].

El Capítulo 3 "Una mejora significativa de una familia de métodos tipo secante", amplía el estudio previo al introducir una familia biparamétrica de métodos iterativos. Partiendo de la familia uniparamétrica analizada en el primer capítulo, se desarrolla una versión que alcanza convergencia cuadrática, combinando la flexibilidad del método de la secante con la robustez del método de Newton. En particular, se demuestra que el uso de diferencias divididas simétricas permite mejorar la aproximación de la derivada del operador involucrado, lo que se traduce en un incremento en la velocidad de convergencia sin elevar significativamente el costo computacional. Se realiza un análisis numérico detallado de la eficiencia computacional, estableciendo comparaciones con métodos previos y mostrando que la nueva familia de métodos iterativos ofrece ventajas sustanciales en términos de accesibilidad y estabilidad. Además, se examina el comportamiento dinámico de estos métodos mediante el estudio de cuencas de atracción en el plano complejo, proporcionando una representación visual de las regiones de convergencia y permitiendo evaluar experimentalmente la accesibilidad de los métodos iterativos propuestos. Los resultados de este capítulo han sido publicados en [59].

El Capítulo 4 "Un procedimiento para obtener convergencia cuadrática a partir del método de la secante", se enfoca en el desarrollo de una familia de métodos iterativos con memoria, diseñados para mantener la eficiencia computacional del método de la secante pero alcanzar un segundo orden de convergencia. Se explo-

ra en profundidad la relación entre la elección de parámetros en la iteración y la velocidad de convergencia, analizando cómo la utilización de diferencias divididas mejoradas permite recuperar la convergencia cuadrática característica del método de Newton. En este contexto, se realiza una comparación detallada entre los métodos iterativos desarrollados y el método de la secante clásico, concluyendo que la familia propuesta no solo mejora la velocidad de convergencia, sino que también ofrece una mejor accesibilidad a las soluciones, ampliando la región de aproximaciones iniciales que garantizan la convergencia. Además del análisis teórico, el capítulo incluye experimentos numéricos en ecuaciones no lineales e integrales de tipo Hammerstein, lo que permite validar en la práctica las ventajas del enfoque propuesto. Los resultados de este capítulo han sido publicados en [60].

En conjunto, estos capítulos ofrecen un estudio exhaustivo sobre la evolución de los métodos tipo secante, desde su formulación original hasta propuestas avanzadas que combinan flexibilidad, eficiencia y robustez. Se abordan tanto aspectos teóricos, como el análisis de convergencia y accesibilidad, como aplicaciones prácticas, incluyendo su implementación en problemas concretos. Además, se proporciona un marco metodológico para evaluar la eficacia de estos métodos, considerando métricas como la eficiencia computacional y la accesibilidad dinámica. A lo largo de la presente tesis, se demuestra que la optimización de estos métodos no solo es relevante desde un punto de vista teórico, sino que tiene un impacto significativo en la resolución efectiva de problemas matemáticos y computacionales de gran interés

Capítulo 2

Aproximaciones iniciales para una familia de métodos tipo secante

Los resultados de este capítulo han sido publicados en [117]:

- "On the set of initial guesses for the secant method".
- DOI: `https://doi.org/10.1002/mma.9052`.
- Revista "Mathematical Methods in the Applied Science".

2.1. Introducción

Como ya se ha comentado previamente, muchos problemas en ciencias pueden expresarse como ecuaciones no lineales [18, 103, 137]. Estos problemas se formulan frecuentemente como una ecuación del estilo

$$F(x) = 0,$$

donde F es un operador al que hay que dotar de una estructura general. Para ello, se considera el operador

$$F : \Omega \subseteq X \to Y,$$

donde X e Y son espacios de Banach, y Ω es un subconjunto abierto, no vacío y convexo de X. Dado que las raíces de $F(x) = 0$ no suelen obtenerse de manera analítica, se emplean métodos iterativos para resolver el problema.

Si el operador F es diferenciable, uno de los métodos más utilizados es el método de Newton [57], conocido por su eficiencia computacional, que viene definido como:

$$\begin{cases} x_0 \text{ dado en } \Omega, \\ x_{n+1} = x_n - [F'(x_n)]^{-1} F(x_n), \quad n \geq 0, \end{cases} \tag{2.1.1}$$

y tiene convergencia cuadrática.

Sin embargo, la aplicabilidad del método de Newton depende de la existencia de la derivada de F y del coste asociado al cómputo de dicha derivada y de la inversa de dicha derivada. Así en los casos en que dicha derivada es muy costosa o incluso en los casos donde F no es diferenciable, se deben buscar alternativas, y la más intuitiva e inmediata es el uso del método de la secante [6] que viene definido por:

$$\begin{cases} x_{-1}, x_0 \text{ dados en } \Omega, \\ x_{n+1} = x_n - [x_{n-1}, x_n; F]^{-1} F(x_n), \quad n \geq 0, \end{cases} \tag{2.1.2}$$

En este método, teniendo encuenta propiedades del operador F, el uso de diferencias divididas de primer orden juegan un papel crucial, ya que su uso permite la aproximación de la derivada.

Definición 2.1. *Se denomina diferencia dividida de primer orden de F en los puntos distintos x e y al operador lineal y acotado de X a Y, que denotamos $[x, y; F]$, que satisface la condición*

$$[x, y; F](x - y) = F(x) - F(y).$$

Para una discusión más detallada sobre la existencia de diferencias divididas en espacios lineales, puede consultarse [22] y [75].

Por otro lado, el método de la secante tiene convergencia superlineal, con un orden de convergencia R de al menos

$$\frac{1}{2}(1 + \sqrt{5}).$$

Debido a la pérdida de velocidad de convergencia que puede presentarse al utilizar el método de la secante en comparación con el método de Newton, en [85] se introduce una familia uniparamétrica de métodos iterativos tipo secante:

$$\begin{cases} x_{-1}, x_0 \text{ dados en } \Omega, \\ y_n = \theta x_n + (1 - \theta) x_{n-1}, \quad \theta \in [0, 1), \\ x_{n+1} = x_n - [y_n, x_n; F]^{-1} F(x_n), \quad n \geq 0. \end{cases} \tag{2.1.3}$$

Esta familia se puede interpretar como una interpolación entre el método de la secante ($\theta = 0$) y el método de Newton ($\theta = 1$) para operadores diferenciables. Se debe notar que si F es diferenciable, entonces

$$[x, x; F] = F'(x).$$

Como se muestra en [87], el orden de convergencia R de (2.1.3) es, para todo θ, al menos igual al del método de la secante. Además, en la práctica, cuanto más cercanos estén x_n e y_n, mayor es la velocidad de convergencia. De hecho, la velocidad de convergencia de (2.1.3) incrementa conforme θ se aproxima a 1, acercándose a la velocidad del método de Newton.

Otro desafío al aplicar el método de la secante es su limitada accesibilidad, es decir, el pequeño conjunto de aproximaciones iniciales que garantizan su convergencia. Para mejorar esta accesibilidad, se utiliza una técnica de descomposición del operador F, expresando F como:

$$F(x) = G(x) + H(x),$$

donde G y F son operadores no lineales, con G diferenciable y F continuo pero no diferenciable. Para aproximar una solución de $F(x) = 0$, se propone la siguiente familia de métodos iterativos tipo Newton-secante:

$$\begin{cases} x_{-1}, x_0 \text{ dados en } \Omega, \\ y_n = \theta x_n + (1-\theta)x_{n-1}, \quad \theta \in [0,1), \\ x_{n+1} = x_n - \left(G'(x_n) + [y_n, x_n; H]\right)^{-1} F(x_n), \quad n \geq 0. \end{cases} \tag{2.1.4}$$

Se debe notar que la iteración (2.1.4) se reduce al método de Newton si F es diferenciable, es decir, si $H(x) = 0$ y a la familia de métodos tipo secante (2.1.3) si no tiene parte diferencial, es decir, si $G(x) = 0$.

Además, cuando $\theta = 1$, tanto (2.1.3) como (2.1.4) se reducen al método de Newton, ya que $y_n = x_n$ y $[x_n, x_n; F] = F'(x_n)$. En este capítulo, se demostrará que la familia (2.1.4) mejora tanto la velocidad de convergencia como la accesibilidad del método de la secante al aplicar la familia (2.1.4).

La estructura del capítulo es la siguiente:

- En la Sección 2.2, se presentará un estudio sobre la convergencia local de la familia de métodos iterativos tipo Newton-secante (2.1.4) y para ello se utilizarán diferentes condiciones y varios lemas técnicos que ayudan a simplificar la demostración del teorema principal. Esta sección termina con un resultado sobre unicidad de la solución.

- En la Sección 2.3, se presentan los casos particulares que tiene la familia de métodos iterativos tipo Newton-secante (2.1.4), como son el método de Newton y los métodos tipo secante, se adaptarán los resultados de convergencia y se aplicarán a diferentes ejemplos a cada uno de dichas casos particulares. Dentro de los ejemplos se usarán ecuaciones integrales no lineales de tipo Hammerstein mixto.

- En la Sección 2.4, se lleva a cabo un estudio dinámico comparativo entre la familia. Para evaluar la accesibilidad de la familia de métodos iterativos, se van a considerar dos enfoques:

  - En el primero de ellos, se va a analizar la convergencia local, lo que proporciona una región de convergencia que permite comparar la accesibilidad del método en función del tamaño de dicha región.
  - En el segundo de los enfoques, se realiza un análisis dinámico, lo cual permite determinar el conjunto de aproximaciones iniciales que garantizan la convergencia del método, es decir, su accesibilidad.

  Además, se usarán dos estrategias diferentes de estudio dinámico, por un lado se fijará uno de los puntos inciales, dejando el otro libre, en el plano complejo y en la segunda se dejarán ambos puntos libres, tomando ambos en la recta real. En ambas estrategias se facilita un código en Python que puede ser replicado.

- Por último, en la Sección 2.5, se presentan las conclusiones que se pueden extraer de este capítulo.

2.2. Convergencia local de la familia de métodos iterativos tipo Newton-secante (2.1.4)

Es fundamental tener presente que la convergencia local de un método iterativo se determina a partir de las condiciones impuestas sobre el operador y la solución de la ecuación a resolver. Este análisis conduce a la obtención de la denominada bola de convergencia para la sucesión generada por el método iterativo. La accesibilidad a la solución se define entonces en función de las aproximaciones iniciales que se encuentran dentro de esta bola de convergencia ([45, 58]).

Para establecer la convergencia local de la iteración (2.1.4), se asumen las siguientes hipótesis:

(CL1) Suponemos que existen dos funciones continuas y no decrecientes,

$$\omega_1 : [0, +\infty) \to \mathbb{R}$$

y

$$h : [0, 1] \to \mathbb{R},$$

tales que

$$\omega_1(tz) \leq h(t)\omega_1(z)$$

para todo $t \in [0, 1]$, $z \in [0, +\infty)$, y se satisface la condición

$$\|G'(x) - G'(y)\| \leq \omega_1(\|x - y\|).$$

(CL2) Para cada par de puntos distintos $p, q \in \Omega$, existe $[p, q; H]$ tal que

$$\|[x, y; H] - [u, v; H]\| \leq \omega_2(\|x - u\|, \|y - v\|)$$

para todos $x, y, u, v \in \Omega$, donde

$$\omega_2 : [0, +\infty) \times [0, +\infty) \to \mathbb{R}$$

es una función continua y no decreciente en ambos argumentos.

(CL3) Dada una solución x^* de la ecuación $F(x) = 0$, se supone que existe $\widetilde{x} \in \Omega$ tal que

$$\|\widetilde{x} - x^*\| = \alpha > 0,$$

y que el operador

$$D^{-1} = \left(G'(x^*) + [x^*, \widetilde{x}; H]\right)^{-1}$$

satisface la condición

$$\|D^{-1}\| \leq \delta.$$

(CL4) Existe $\rho \geq 0$ tal que

$$B(x^*, \rho) \subset \Omega$$

y

$$\ell = \delta\left(\omega_1(\rho) + \omega_2(\rho, \rho + \alpha)\right) < 1.$$

Además de estas condiciones, también usaremos dos lemas que se presentan a continuación. El primero está relacionado con la existencia de los operadores inversos que se encuentran en la formulación.

Lema 2.1. *Bajo las hipótesis (CL1)-(CL2)-(CL3)-(CL4), el operador*

$$\left(G'(x) + [\theta x + (1-\theta)y, x; H]\right)$$

es invertible para cualquier par de puntos distintos $x, y \in B(x^*, \rho)$. Además,

$$\left\| \left(G'(x) + [\theta x + (1-\theta)y, x; H]\right)^{-1} \right\| \leq \frac{\delta}{1-\ell}. \tag{2.2.1}$$

Demostración. En primer lugar, dado que

$$x \in B(x^*, \rho),$$

$$y \in B(x^*, \rho),$$

$$\theta x + (1-\theta) \in B(x^*, \rho),$$

con

$$x \neq \theta x + (1-\theta)y$$

y

$$\theta \in [0, 1),$$

se puede asegurar que el operador definido por

$$[\theta x + (1-\theta)y, x; H]$$

existe.

Después, de la desigualdad

$$\begin{aligned} \left\| I - D^{-1}\left(G'(x) + [\theta x + (1-\theta)y, x; H]\right)\right\| & \\ \leq & \|D^{-1}\| \left(\|G'(x^*) - G'(x)\| + \|[x^*, \widetilde{x}; H]\right) \\ & -\|D^{-1}\| \left([\theta x + (1-\theta)y, x; H]\|\right) \\ \leq & \delta\left(\omega_1(\rho) + \omega_2(\rho, \rho+\alpha)\right) \\ = & \ell \\ < & 1, \end{aligned}$$

y utilizando el lema de Banach para operadores invertibles, se concluye que el operador

$$\left(G'(x) + [\theta x + (1-\theta)y, x; H]\right)^{-1}$$

existe y satisface que

$$\left\| \left(G'(x) + [\theta x + (1-\theta)y, x; H]\right)^{-1} \right\| \leq \frac{\delta}{1-\ell}.$$

□

El segundo lema técnico nos servirá para asegurar que a sucesión $\{x_n\}$ generada por la iteración (2.1.4) está correctamente definida. Para ello, se debe considerar que

$$\theta x_{n-1} + (1-\theta)x_{n-2} \in B(x^*, \rho)$$

siempre que

$$x_{n-1}, x_{n-2} \in B(x^*, \rho)$$

y

$$x_{n-1} \neq x_{n-2},$$

ya que

$$\theta x_{n-1} + (1-\theta)x_{n-2}$$

pertenece al segmento que conecta x_{n-1} y x_{n-2}.

Lema 2.2. *Bajo las hipótesis (CL1)-(CL2)-(CL3)-(CL4), y suponiendo que*

$$x_{n-1}, x_{n-2} \in B(x^*, \rho)$$

con

$$x_{n-1} \neq x_{n-2},$$

la sucesión $\{x_n\}$ generada por (2.1.4) *está correctamente definida y además satisface*

$$\|x_n - x^*\| \leq K\|x_{n-1} - x^*\|,$$

donde

$$K = \frac{\sigma}{1-\ell},$$

$$\sigma = \delta\left(I_h\omega_1(\rho) + \omega_2(\rho, 0)\right)$$

e

$$I_h = \int_0^1 h(t)dt.$$

Demostración. Dado que $x_{n-1} \neq x_{n-2}$ y $\theta \neq 1$, el operador

$$[\theta x_{n-1} + (1-\theta)x_{n-2}, x_{n-1}; H]$$

existe.

Ahora, denotando

$$D_{n-1} = G'(x_{n-1}) + [\theta x_{n-1} + (1-\theta)x_{n-2}, x_{n-1}; H]$$

y aplicando el anterior Lema 2.1, se puede asegurar que D_{n-1} es invertible y que

$$\|D_{n-1}^{-1}\| \leq \frac{\delta}{1-\ell},$$

por lo que x_n está correctamente definido.

De (2.1.4), se deduce que

$$\begin{aligned} x_n - x^* &= x_{n-1} - D_{n-1}^{-1}F(x_{n-1}) - x^* \\ &= D_{n-1}^{-1}\left(G'(x_{n-1}) + [\theta x_{n-1} + (1-\theta)x_{n-2}, x_{n-1}; H]\right)(x_{n-1} - x^*) \\ &\quad -D_{n-1}^{-1}\left(G(x_{n-1}) + H(x_{n-1})\right) \\ &= D_{n-1}^{-1}\left(\int_0^1 \left(G'(x_{n-1} + \tau(x^* - x_{n-1})) - G'(x_{n-1})\right)(x_{n-1} - x^*)d\tau\right) \\ &\quad +D_{n-1}^{-1}\left([\theta x_{n-1} + (1-\theta)x_{n-2}, x_{n-1}; H] - [x^*, x_{n-1}; H]\right)(x_{n-1} - x^*) \end{aligned}$$

y, al tomar normas

$$\begin{aligned} \|x_n - x^*\| &\leq \frac{\delta}{1-\ell}\left(I_h\omega_1(\rho) + \omega_2(\rho, 0)\right)\|x_{n-1} - x^*\| \\ &= \frac{\sigma}{1-\ell}\|x_{n-1} - x^*\| \\ &= K\|x_{n-1} - x^*\|. \end{aligned}$$

Y, por lo tanto, la desmotración está completada. □

Por otro lado, es importante destacar que

$$K < 1$$

siempre que se cumpla

$$\ell + \sigma < 1,$$

es decir:

$$\delta\left((1 + I_h)\omega_1(\rho) + \omega_2(\rho, \alpha + \rho) + \omega_2(\rho, 0)\right) < 1,$$

donde

$$I_h = \int_0^1 h(t)dt.$$

Además, la ecuación

$$\delta\left((1 + I_h)\omega_1(\rho) + \omega_2(\rho, \alpha + \rho) + \omega_2(\rho, 0)\right) - 1 = 0 \tag{2.2.2}$$

posee al menos una raíz real positiva. Denorateremos como r a la menor raíz positiva. Entonces, para cualquier $\rho_\star \in \mathbb{R}_+$ tal que $\rho_\star < r$, se cumple que

$$\ell + \sigma < 1.$$

Teniendo en cuenta todo lo anterior, se puede establecer el siguiente teorema sobre la convergencia local de la familia (2.1.4).

Teorema 2.1. *Bajo las condiciones (CL1)-(CL2)-(CL3)-(CL4), si la ecuación* (2.2.2) *tiene al menos una raíz real positiva y denotando la más pequeña como r, y tomando $\rho_\star \in \mathbb{R}_+$ tal que $\rho_\star < r$ con $B(x^*, \rho_\star) \subset \Omega$, entonces si $x_0 \in B(x^*, \rho_\star)$ y $x_{-1} \in B(x_0, \rho_\star - \eta)$, con $x_{-1} \neq x_0$ y*

$$\eta = \|x_0 - x^*\|,$$

la sucesión generada por (2.1.4) *está bien definida, $x_n \in B(x^*, \rho_\star)$ para todo $n \geq 0$, y converge a la solución x^* de la ecuación $F(x) = 0$.*

Demostración. Dado que $x_{-1} \in B(x_0, \rho_\star - \eta)$, se tiene que $x_{-1} \in B(x^*, \rho_\star)$, puesto que

$$\begin{aligned} \|x_{-1} - x^*\| &\leq \|x_{-1} - x_0\| + \|x_0 - x^*\| \\ &\leq \rho_\star - \eta + \eta \\ &= \rho_\star. \end{aligned}$$

Además,

$$\theta x_0 + (1-\theta)x_{-1} \neq x_0,$$

dado que $\theta \neq 1$. Por lo tanto, del Lema 2.1, se garantiza la existencia del operador

$$D_0^{-1} = \left(G'(x_0) + [\theta x_0 + (1-\theta)x_{-1}, x_0; H]\right)^{-1},$$

con

$$\|D_0^{-1}\| \leq \frac{\delta}{1-\ell}.$$

Por lo tanto, x_1 está correctamente definido y, aplicando el Lema 2.2, se cumple que

$$\begin{aligned} \|x_1 - x^*\| &\leq K\|x_0 - x^*\| \\ &< \|x_0 - x^*\| \\ &< \rho_\star, \end{aligned}$$

lo cual implica que $x_1 \in B(x^*, \rho_\star)$ con $x_0 \neq x_1$, y entonces

$$\theta x_1 + (1-\theta)x_0 \in B(x^*, \rho_\star).$$

Mediante inducción matemática sobre n, es fácil verificar que $x_n \in B(x^*, \rho_\star)$ para todo n. Además, como

$$\theta x_n + (1-\theta)x_{n-1} \in B(x^*, \rho_\star)$$

siempre que $x_n \neq x_{n-1}$, el operador D_{n+1}^{-1} existe y x_{n+1} está correctamente definido. Por lo que del Lema 2.2, se tiene que

$$\|x_{n+1} - x^*\| \leq K\|x_n - x^*\|.$$

Por lo tanto,

$$x_{n+1} \in B(x^*, \rho_\star)$$

para todo $n \geq 0$, y

$$\|x_{n+1} - x^*\| \leq K^{n+1}\|x_0 - x^*\|,$$

lo que demuestra que la sucesión $\{x_n\}$ converge a x^*. □

Nota 2.1. *Una elección sencilla para $\widetilde{x}$ es tomar*

$$\widetilde{x} = x_0,$$

en cuyo caso se tiene que

$$\alpha = \eta.$$

Por otro lado, también se presenta el siguiente resultado relativo a la unicidad de solución.

Teorema 2.2. *Bajo las condiciones (CL1)-(CL2)-(CL3)-(CL4), supóngase que existe $R \geq \rho$ tal que*

$$\delta\left(I_h\omega_1(R) + \omega_2(0, R+\alpha)\right) < 1.$$

Entonces, la solución x^* es única en

$$\overline{B(x^*, R)} \cap \Omega.$$

Demostración. Sea

$$\zeta^* \in \overline{B(x^*, R)} \cap \Omega$$

otra solución de la ecuación $F(x) = 0$, y considérese el operador

$$J = \int_0^1 G'(x^* + \tau(\zeta^* - x^*))d\tau + [x^*, \zeta^*; H].$$

Entonces,

$$\begin{aligned}
\|D^{-1}J - I\| &\leq \|D^{-1}\|\|J - D\| \\
&\leq \delta\left(\int_0^1 \omega_1(\|\tau(\zeta^* - x^*)\|)d\tau + \omega_2(0, R+\alpha)\right) \\
&\leq \delta\left(I_h\omega_1(R) + \omega_2(0, R+\alpha)\right) \\
&< 1,
\end{aligned}$$

se puede garantizar la existencia de J^{-1} mediante el uso del lema de Banach para operadores invertibles. En consecuencia, dado que

$$0 = F(x^*) - F(\zeta^*) = J(x^* - \zeta^*),$$

se concluye que

$$x^* = \zeta^*,$$

lo cual demuestra la unicidad de x^* en

$$\overline{B(x^*, R)} \cap \Omega.$$

□

2.3. Casos particulates y ejemplos

A continuación, se ilustran los resultados teóricos con un ejemplo que aborda una ecuación integral de Hammerstein de segunda especie [132]. Las ecuaciones de Hammerstein se utilizan frecuentemente en la dinámica de fluidos electromagnéticos y se desarrollaron en la década de 1930 como modelos para estudiar problemas de valores en la frontera semilineales, donde el núcleo a menudo corresponde a la función de Green de un operador diferencial. Estas ecuaciones también encuentran aplicaciones en la teoría de la transferencia radiativa, la teoría del transporte de neutrones y la teoría cinética de gases, así como en modelos dinámicos de reactores químicos. A continuación, se muestra cómo se obtienen las bolas de convergencia y cómo se demuestra la unicidad de la solución utilizando los Teoremas 2.1 y 2.2, respectivamente.

Ejemplo 2.1. *Considérese la siguiente ecuación integral no lineal de tipo Hammerstein mixto:*

$$x(s) = f(s) + \int_a^b \mathcal{K}(s,t)\left(\lambda x(t)^2 + \mu|x(t)|\right) dt,\ \ s \in [a,b], \tag{2.3.1}$$

donde

$$\lambda, \mu \in \mathbb{R},$$

$$-\infty < a < b < +\infty,$$

la función $f(s)$ es continua en $[a,b]$ y está dada, el núcleo

$$\mathcal{K}(s,t)$$

es una función conocida en $[a,b] \times [a,b]$, y x es la función a determinar.

Resolver (2.3.1) *equivale a resolver la ecuación $\mathcal{F}(x) = 0$, donde*

$$\mathcal{F} : \mathcal{C}[a,b] \to \mathcal{C}[a,b],$$

viene definido por

$$[\mathcal{F}(x)](s) = f(s) + \int_a^b \mathcal{K}(s,t)\left(\lambda x(t)^2 + \mu|x(t)|\right) dt,\ \ s \in [a,b]. \tag{2.3.2}$$

El núcleo $\mathcal{K}(s,t)$ se toma como la función de Green en $[a,b] \times [a,b]$ y se utiliza una fórmula de cuadratura de Gauss-Legendre con m nodos para discretizar la ecuación (2.3.1)*, transformándola en un problema de dimensión finita:*

$$\int_a^b \chi(t)\, dt = \sum_{i=1}^{m} w_i \chi(t_i),$$

donde los nodos t_i y los pesos w_i se determinan para $i = 1, 2, \ldots, m$. Si se denotan las aproximaciones de $x(t_i)$ y $f(t_i)$ como x_i y f_i respectivamente, para $i = 1, 2, \ldots, m$, entonces la ecuación (2.3.1) *se reduce al siguiente sistema de ecuaciones no lineales:*

$$x_i = f_i + \sum_{j=1}^{m} a_{ij}\left(\lambda x_i^2 + \mu|x_i|\right), \quad j = 1, 2, \ldots, m, \tag{2.3.3}$$

donde

$$a_{ij} = w_j \, \mathcal{K}(t_i, t_j) = \begin{cases} w_j \, \dfrac{(b - t_i)(t_j - a)}{b - a}, & si \quad j \leq i, \\ w_j \, \dfrac{(b - t_j)(t_i - a)}{b - a}, & si \quad j > i. \end{cases}$$

El sistema (2.3.3) *se puede expresar como*

$$F(\mathbf{x}) \equiv \mathbf{x} - \mathbf{f} - A\,\hat{\mathbf{x}} = 0, F : \mathbb{R}^m \to \mathbb{R}^m, \tag{2.3.4}$$

donde

$$\mathbf{x} = (x_1, x_2, \ldots, x_m)^T,$$
$$\mathbf{f} = (f_1, f_2, \ldots, f_m)^T,$$
$$A = (a_{ij})_{i,j=1}^m,$$

y

$$\hat{\mathbf{x}} = \left(\lambda x_1^2 + \mu|x_1|, \lambda x_2^2 + \mu|x_2|, \ldots, \lambda x_m^2 + \mu|x_m|\right)^T.$$

Particularmente, si se toma

$$a = 0,$$
$$b = 1$$

y

$$\lambda = \mu = \frac{1}{2},$$

el sistema (2.3.4) *se reduce a*

$$F(\mathbf{x}) = G(\mathbf{x}) + H(\mathbf{x}), \quad con \quad G(\mathbf{x}) = \mathbf{x} - \mathbf{f} - \frac{1}{2} A\,\dot{\mathbf{x}}, \quad y \quad H(\mathbf{x}) = -\frac{1}{2} A\,\ddot{\mathbf{x}}, \tag{2.3.5}$$

donde

$$\dot{\mathbf{x}} = (x_1^2, x_2^2, \ldots, x_m^2)^T$$

y

$$\ddot{\mathbf{x}} = (|x_1|, |x_2|, \ldots, |x_m|)^T.$$

Por lo tanto, la función F en (2.3.5) *es no lineal y no diferenciable.*

Se utilizan diferencias divididas de primer orden en $\mathbb{R}^m$, que no requieren que F sea diferenciable. Se toma la diferencia dividida de primer orden como

$$[\mathbf{p}, \mathbf{q}; H] = \left([\mathbf{p}, \mathbf{q}; H]_{ij}\right)_{i,j=1}^m,$$

donde

$$[\mathbf{p}, \mathbf{q}; H]_{ij} = \frac{1}{p_j - q_j} \left(H_i(p_1, \ldots, p_{j-1}, p_j, q_{j+1}, \ldots, q_m) - H_i(p_1, \ldots, p_{j-1}, q_j, q_{j+1}, \ldots, q_m)\right),$$

con

$$i, j = 1, 2, \ldots, m,$$
$$\mathbf{p} = (p_1, p_2, \ldots, p_m)^T$$

y

$$\mathbf{q} = (q_1, q_2, \ldots, q_m)^T.$$

Si se establece $\mathbf{f} = 0$ *en* (2.3.5)*, entonces* $\mathbf{x}^* = 0$ *es una solución evidente de* $F(\mathbf{x}) = 0$*, y el operador* F *toma la forma:*

$$F(\mathbf{x}) = \mathbf{x} - \frac{1}{2}A\bar{\mathbf{x}}, \quad \textit{donde} \quad \bar{\mathbf{x}} = \left(x_1^2 + |x_1|, x_2^2 + |x_2|, \ldots, x_m^2 + |x_m|\right)^T.$$

En este caso,

$$G(\mathbf{x}) = \mathbf{x} - \frac{1}{2}A\,\dot{\mathbf{x}},$$

y

$$G'(\mathbf{x}) = I - A\,diag\{x_1, x_2, \ldots, x_m\},$$

donde I *es la matriz identidad, cumpliéndose que:*

$$\|G'(\mathbf{x}) - G'(\mathbf{y})\| \leq \|A\|\|\mathbf{x} - \mathbf{y}\|.$$

Además,

$$H(\mathbf{x}) = -\frac{1}{2}A\,\ddot{\mathbf{x}},$$

y se tiene que

$$[\mathbf{p}, \mathbf{q}; H] = -\frac{1}{2}A\,diag\left\{\frac{|p_1| - |q_1|}{p_1 - q_1}, \frac{|p_2| - |q_2|}{p_2 - q_2}, \ldots, \frac{|p_m| - |q_m|}{p_m - q_m}\right\}.$$

y

$$\|[\mathbf{x}, \mathbf{y}; H] - [\mathbf{u}, \mathbf{v}; H]\| \leq \|A\|.$$

Por lo tanto,

$$\omega_1(z) = \|A\|z,$$

$$I_h = \frac{1}{2}$$

y

$$\omega_2(s, t) = \|A\|.$$

Tomando $m = 8$ *y*

$$\mathbf{x}_0 = \widetilde{\mathbf{x}} = (1, 1, \ldots, 1)^T,$$

según el Teorema 2.1, se tiene que

$$\|A\| = 0.1235\ldots, \alpha = 1, \delta = \|D^{-1}\| = 0.5326\ldots,$$

y la raíz positiva más pequeña de la ecuación (2.2.2) *es*

$$\rho = 8.7968\ldots.$$

Por lo tanto, de acuerdo con el Teorema 2.1, la bola de convergencia es $B(\mathbf{x}^*, \rho)$ *con*

$$\rho < 8.7968\ldots$$

Además, según el Teorema 2.2, la solución es única en

$$\overline{B(\mathbf{x}^*, R)}$$

con $R < 28.3904\ldots$

Finalmente, en el siguiente ejemplo, que también aparece en varios problemas de química, se observa que la iteración (2.1.4) proporciona mejores aproximaciones a la solución en comparación con la familia de métodos tipo secante (2.1.3).

Ejemplo 2.2. *Se considera la ecuación integral* (2.3.1) *con*

$$f(s) = \frac{1}{2},$$
$$a = 0,$$
$$b = 1,$$
$$\lambda = \mu = \frac{3}{4}$$

y el núcleo

$$\mathcal{K}(s,t)$$

como la función de Green en $[0,1]\times[0,1]$. *Siguiendo el mismo proceso de discretización que en el Ejemplo 2.1, se transforma la ecuación integral en el sistema:*

$$F(\mathbf{x}) \equiv \mathbf{x} - \mathbf{u} - \frac{3}{4} A\,\bar{\mathbf{x}} = 0, F : \mathbb{R}^8 \to \mathbb{R}^8, \tag{2.3.6}$$

donde

$$\mathbf{x} = (x_1, x_2, \ldots, x_8)^T,$$
$$\mathbf{u} = \left(\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2}\right)^T,$$
$$A = (a_{ij})_{i,j=1}^{m}$$

y

$$\bar{\mathbf{x}} = \left(x_1^2 + |x_1|, x_2^2 + |x_2|, \ldots, x_8^2 + |x_8|\right)^T.$$

Se utilizan los métodos iterativos dados por (2.1.3) *y* (2.1.4) *con*

$$\theta = \frac{1}{2},$$

comenzando en

$$\mathbf{x}_{-1} = \left(\frac{2}{5}, \frac{2}{5}, \ldots, \frac{2}{5}\right)^T$$

y

$$\mathbf{x}_0 = \left(\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2}\right)^T,$$

para aproximar una solución de (2.3.6). *Comparando los errores* $\|\mathbf{x}_n - \mathbf{x}^*\|$, *mostrados en la Tabla 2.1 con un criterio de parada de* $\|\mathbf{x}_n - \mathbf{x}_{n-1}\| < 10^{-24}$, *se observa que el método* (2.1.4) *proporciona mejores aproximaciones a la solución que el método* (2.1.3). *Los resultados son similares para otros valores de* $\theta \in [0,1)$. *Además, el orden computacional de convergencia ([167]) para la familia de métodos tipo secante* (2.1.3) *se aproxima al orden*

$$\frac{1+\sqrt{5}}{2} = 1.6180\ldots$$

del método de la secante [6], mientras que el de la iteración (2.1.4) *es cercano al orden cuadrático del método de Newton.*

n	$\|\mathbf{x}_n - \mathbf{x}^*\|$	$\|\mathbf{x}_n - \mathbf{x}^*\|$
1	$9.7710\ldots \times 10^{-4}$	$5.8998\ldots \times 10^{-4}$
2	$3.4775\ldots \times 10^{-6}$	$2.8862\ldots \times 10^{-8}$
3	$1.4162\ldots \times 10^{-10}$	$6.8605\ldots \times 10^{-17}$

Tabla 2.1: Errores absolutos obtenidos por los métodos (2.1.3) y (2.1.4) con $\theta = \frac{1}{2}$, respectivamente.

2.3.1. El método de Newton

En el caso de un operador F diferenciable, se tiene que $F(x) = G(x)$ debido a que $H(x) = 0$. En este escenario, la condición (CL2) se vuelve innecesaria y la ecuación (2.2.2) se simplifica a:

$$\delta(1 + I_h)\omega_1(\rho) - 1 = 0. \tag{2.3.7}$$

A partir de esta ecuación, se puede establecer el siguiente resultado de convergencia local para el método de Newton.

Teorema 2.3. *Dado un operador F diferenciable (es decir, $F(x) = G(x)$) y bajo la condición (CL1) y las siguientes hipótesis:*

(N1) *Sea x^* una solución de $F(x) = 0$ tal que exista el operador $[F'(x^*)]^{-1}$, con $\|[F'(x^*)]^{-1}\| \leq \delta$.*

(N2) *La ecuación* (2.3.7) *tiene al menos una raíz real positiva, siendo la más pequeña denotada por r. Además, se considera $\rho_N \in \mathbb{R}_+$ tal que $\rho_N < r$, con $B(x^*, \rho_N) \subset \Omega$.*

Si $x_0 \in B(x^*, \rho_N)$, la sucesión generada por el método de Newton está bien definida, con $x_n \in B(x^*, \rho_N)$ para todo $n \geq 0$, y converge a una solución x^* de la ecuación $F(x) = 0$.

Para ilustrar este teorema, se considera un ejemplo con una ecuación integral no lineal de tipo (2.3.1) donde $\mu = 0$.

Ejemplo 2.3. *Considerando la ecuación* (2.3.1) *con*

$$f(s) = 0,$$

$$a = 0,$$

$$b = 1,$$

$$\lambda = \frac{1}{2},$$

$$\mu = 0$$

y el núcleo

$$\mathcal{K}(s, t)$$

como la función de Green en el intervalo $[0,1]\times[0,1]$*, se procede a discretizarla, obteniendo el sistema*

$$F(\mathbf{x}) \equiv \mathbf{x} - \frac{1}{2}A\bar{\bar{\mathbf{x}}} = 0, \qquad F:\mathbb{R}^8 \to \mathbb{R}^8,$$

donde

$$\mathbf{x} = (x_1, x_2, \ldots, x_8)^T,$$
$$A = (a_{ij})_{i,j=1}^{m},$$

y

$$\bar{\bar{\mathbf{x}}} = (x_1^2, x_2^2, \ldots, x_8^2)^T.$$

Aplicando el Teorema 2.3, se obtiene que

$$\|A\| = 0.1235\ldots, \quad \delta = \|[F'(x^*)]^{-1}\| = 1, \quad I_h = \frac{1}{2},$$

lo que permite satisfacer la ecuación (2.3.7) *si*

$$\rho < 5.3953\ldots.$$

Por tanto, la bola de convergencia es $B(\mathbf{x}^*, \rho_N)$ *con*

$$\rho_N < 5.3953\ldots.$$

Asimismo, según el Teorema 2.2, la solución es única en la bola $\overline{B(\mathbf{x}^*, R)}$ *con*

$$R < 16.1866\ldots.$$

A continuación, se realiza un estudio comparativo de la bola de convergencia para el método de Newton. Para ello, se compara la bola de convergencia obtenida por Dennis y Schnabel en su análisis de la convergencia local del método de Newton en [45]. Bajo el supuesto de que F' es Lipschitz continuo en Ω, se tiene que:

existe $L \geq 0$ tal que $\|F'(x) - F'(y)\| \leq L\|x - y\|$, para todo $x, y \in \Omega$.

Como consecuencia, se obtiene

$$\omega_1(z) = Lz,$$
$$h(t) = t,$$

y

$$I_h = \int_0^1 h(t)dt = \frac{1}{2}.$$

La ecuación (2.3.7) se simplifica a

$$\frac{3}{2}\delta L\rho - 1 = 0,$$

con una única raíz real positiva

$$r = \frac{2}{3L\delta}.$$

Por lo tanto, se puede considerar ρ_N tal que

$$\rho_N < \frac{2}{3L\delta}.$$

En el estudio de Dennis y Schnabel [45], la bola de convergencia $B(x^*, \widetilde{\rho})$ se encuentra dentro de un radio

$$\widetilde{\rho} < \frac{1}{2L\delta}$$

bajo las mismas condiciones, lo que indica que la bola de convergencia obtenida a partir del Teorema 2.3 representa una mejora. Además, Rheinboldt [140] obtiene una bola de convergencia equivalente a la descrita en el Teorema 2.3.

2.3.2. Métodos tipo secante

En el caso de que $G(x) = 0$, se tiene que $F(x) = H(x)$, por lo que la iteración (2.1.4) se reduce a la familia de métodos tipo secante (2.1.3). En este escenario, la condición (CL1) no es necesaria y la ecuación (2.2.2) se simplifica a:

$$\delta(\omega_2(\rho, \rho + \alpha) + \omega_2(\rho, 0)) - 1 = 0. \tag{2.3.8}$$

Bajo estas circunstancias, se puede establecer el siguiente resultado local de convergencia para la familia de métodos tipo secante (2.1.3).

Teorema 2.4. *Bajo las condiciones (CL2), (CL3), y la siguiente hipótesis:*

(S4) *La ecuación* (2.3.8) *tiene al menos una raíz real positiva, siendo la más pequeña denotada por r. Se considera entonces $\rho_S \in \mathbb{R}_+$ tal que $\rho_S < r$, con $B(x^*, \rho_S) \subset \Omega$.*

Si

$$x_0 \in B(x^*, \rho_S)$$

y

$$x_{-1} \in B(x_0, \rho_S - \eta)$$

con $x_0 \neq x_{-1}$, entonces la sucesión generada por la familia de métodos tipo secante (2.1.3) *está bien definida, con*

$$x_n \in B(x^*, \rho_S)$$

para todo $n \geq 0$, y converge a una solución x^* de la ecuación $F(x) = 0$.

Se observa que el enfoque de aplicar directamente la familia de métodos tipo secante (2.1.3) puede ser adecuado cuando $F(x) = H(x)$ y F no es diferenciable. No obstante, como se ilustra en el siguiente ejemplo, la descomposición de $F(x)$ como $F(x) = G(x) + H(x)$ puede mejorar la bola de convergencia en comparación con el caso $F(x) = H(x)$.

Ejemplo 2.4. *Considerando el Ejemplo* (2.1)*, se tiene que:*

$$F(\mathbf{x}) = H(\mathbf{x}) = \mathbf{x} - \mathbf{f} - A\,\hat{\mathbf{x}}.$$

De acuerdo con el Teorema 2.4, se obtiene

$$\omega_2(s,t) = \frac{1}{2}\|A\|(s+t+2),$$

$$\alpha = 1,$$

$$D^{-1} = [x^*, \widetilde{x}; H]^{-1},$$

y

$$\delta = 1.1382\ldots.$$

La ecuación (2.3.8) *se satisface si*

$$\rho < 3.0736\ldots.$$

Por lo tanto, la bola de convergencia es $B(\mathbf{x}^*, \rho_S)$ *con*

$$\rho_S < 3.0736\ldots.$$

Se observa que este radio de la bola de convergencia es menor que el obtenido en el Ejemplo (2.1)*, donde* $F(\mathbf{x})$ *se descompone como*

$$F(\mathbf{x}) = G(\mathbf{x}) + H(\mathbf{x}).$$

Además, según el Teorema 2.2, la solución es única en la bola $\overline{B(\mathbf{x}^*, R)}$ *con*

$$R < 11.2210\ldots,$$

lo cual es menos favorable que en el Ejemplo (2.1)*. Por consiguiente, el enfoque discutido en la Sección 2.2 resulta ser superior al presentado en esta sección.*

Finalmente, se presenta la Tabla 2.2, donde se observa que se obtienen mejores bolas de convergencia $B(\mathbf{x}^*, \rho_S)$ *a medida que el punto auxiliar* $\widetilde{\mathbf{x}}$ *se acerca a* $\mathbf{x}^*$*. Asimismo, se observan mejores bolas de unicidad de solución* $\overline{B(\mathbf{x}^*, R)}$ *a medida que el punto auxiliar* $\widetilde{\mathbf{x}}$ *se aproxima a* $\mathbf{x}^*$*.*

t	ρ_S	R
0.0	4.4070...	13.2210...
0.2	4.4736...	13.4210...
0.4	4.5403...	13.6210...
0.6	4.6070...	13.8210...
0.8	4.6736...	14.0210...
1.0	4.7403...	14.2210...

Tabla 2.2: $B(\mathbf{x}^*, \rho_S)$ y $\overline{B(\mathbf{x}^*, R)}$ cuando $\widetilde{\mathbf{x}} = t\mathbf{x}^* + (1-t)\mathbf{x}_0$, donde $\mathbf{x}^* = \mathbf{0}$.

2.4. Estudio dinámico con diferentes enfoques de la familia (2.1.4)

Primero mostraremos con un ejemplo, donde se analiza el comportamiento dinámico de la familia (2.1.4) según el Teorema 2.1, aplicada a la resolución de una ecuación compleja.

Ejemplo 2.5. *Considérese la función compleja*

$$F(z) = z^3 + z|z| - 2z,$$

la cual no es diferenciable. La función $F(z)$ se puede descomponer como $G(z) = z^3 - 2z$, que es diferenciable, y $H(z) = z|z|$, que no lo es. Es claro que F tiene tres raíces distintas: $z^* = 0$, $z^{**} = -1$ y $z^{***} = 1$.

Tomando el dominio $\Omega = B(z^*, \varepsilon)$, se tiene

$$\omega_1(s) = 6\varepsilon s$$

$$h(t) = t,$$

$$I_h = \frac{1}{2}$$

$$\omega_2(t, s) = 4\varepsilon.$$

Tomando como punto auxiliar

$$\widetilde{z} = \frac{1}{10},$$

entonces

$$\alpha = \frac{1}{10}$$

y

$$\delta = \|D^{-1}\|\frac{10}{19}$$

En este caso, para $\varepsilon = 0.2$ y cualquier $\theta \in [0, 1)$, la raíz positiva más pequeña de la ecuación (2.2.2) es $r = 0.1666\ldots$. Por lo tanto, se toma

$$\rho_\star < 0.1666\ldots.$$

Con $z_0 = 0 \in B(z^*, \rho_\star)$ y $z_{-1} = 0.1 \in B(0, \rho_\star - |z_0|)$, la sucesión $\{z_n\}$ generada por (2.1.4) está bien definida, con $z_n \in B(z^*, \rho_\star) \subseteq \Omega$ para todo $n \geq 0$, y converge hacia la solución $z^* = 0$ de la ecuación $F(z) = 0$.

En la Figura 2.1, se considera

$$z_{-1} = \frac{1}{10}$$

y se deja el otro punto inicial z_0 libre. Se muestra la cuenca de atracción obtenida con una tolerancia de 10^{-20} y un máximo de 6 iteraciones. El color amarillo indica la cuenca de atracción de la raíz $z^* = 0$, mientras que el color negro indica los puntos donde la iteración no converge. Además, en la Figura 2.1, se observa la bola de convergencia (de color rojo), la cual abarca casi todo el dominio Ω, cuyo borde está coloreado de blanco. La bola de convergencia se muestra precisa.

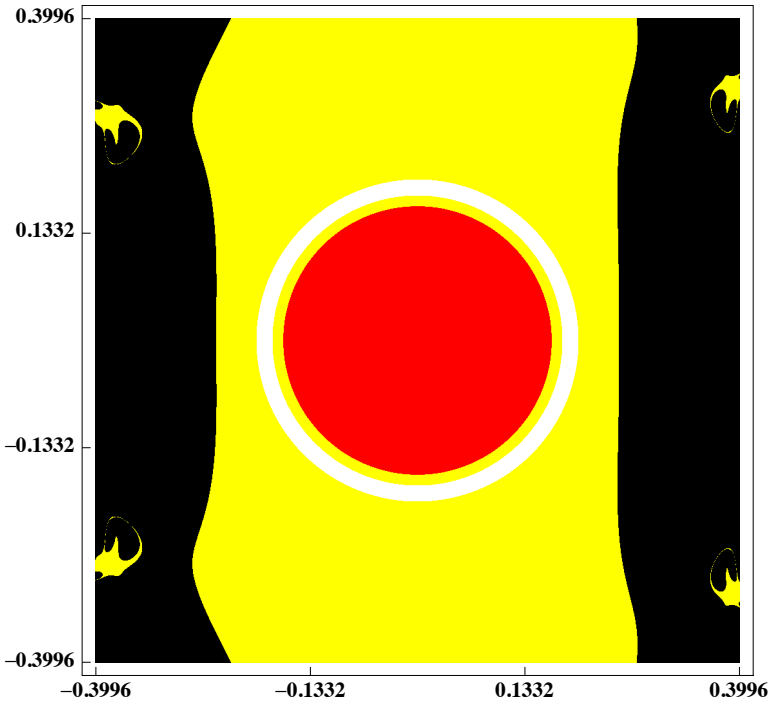


Figura 2.1: La bola de convergencia del método (2.1.4) aplicada a la ecuación $F(z) = z^3 + z|z| - 2z = 0$

Por otro lado, se va a llevar a cabo una comparación experimental de la accesibilidad de las familias de métodos iterativos (2.1.3) y (2.1.4), considerando distintos valores del parámetro θ. Para este análisis, se examina el comportamiento dinámico de los métodos a través del estudio de las cuencas de atracción asociadas a la aplicación de los métodos a una ecuación compleja

$$F(z) = 0,$$

donde $F : \mathbb{C} \to \mathbb{C}$.

El interés en el análisis dinámico ha crecido en los últimos años [1, 39, 77, 102], debido a que ofrece una visión detallada del comportamiento de los métodos iterativos.

En este estudio, se utiliza la función compleja

$$F(z) = z^3 + z|z| - 2z,$$

presentada previamente en el Ejemplo 2.5. Esta función posee tres raíces distintas:

$$z^* = 0,$$

$$z^{**} = -1$$

y

$$z^{***} = 1.$$

A continuación, se muestran los planos dinámicos generados al aproximar estas tres soluciones utilizando los métodos iterativos (2.1.3) y (2.1.4).

La estrategia empleada consiste en asignar un color específico a cada cuenca de atracción correspondiente a una raíz, mientras que los puntos donde la iteración no converge se colorean de negro. Para los gráficos de los planos de convergencia, se asigna el color amarillo a la cuenca de $z^* = 0$, cian a la cuenca de $z^{**} = -1$ y magenta a la cuenca de $z^{***} = 1$. En todos los casos, se ha utilizado una tolerancia de 10^{-6} y un máximo de N iteraciones. Si no se alcanza la tolerancia dentro de las N iteraciones, se concluye que la iteración no converge, coloreándose el punto correspondiente en negro. Por otro lado, si la tolerancia requerida se alcanza en menos de N iteraciones, el punto se colorea según la cuenca de atracción de la raíz correspondiente.

Los gráficos fueron generados utilizando Python, siguiendo un algoritmo adaptado al descrito en [105, 106] y que se aportará en las correspondientes secciones.

Para definir los puntos iniciales (z_{-1}, z_0) dentro del plano de convergencia, se emplearon dos estrategias. La primera mantiene fijo a z_{-1} y deja libre a z_0, mientras que en la segunda, tanto z_{-1} como z_0 son libres.

2.4.1. Primera estrategia: z_{-1} es fijo y z_0 es libre

De acuerdo con estudios previos [40, 43], cuando se utilizan métodos iterativos con memoria, como los métodos (2.1.3) o (2.1.4), es recomendable que los puntos iniciales z_{-1} y z_0 estén lo suficientemente próximos. En este caso, se fija

$$z_{-1} = z_0 - \frac{1}{10},$$

permitiendo que z_0 sea libre. El algoritmo utilizado para el plano de convergencia es el propuesto en [105], donde el eje horizontal representa los valores de la parte real de z_0 y el eje vertical los valores de la parte imaginaria de z_0.

Las Figuras 2.2– 2.7 ilustran el comportamiento dinámico de ambos métodos iterativos para los valores de

$$\theta = 0$$

(método de la secante),

$$\theta = \frac{1}{3}$$

y

$$\theta = \frac{2}{3}.$$

Se observa que el método (2.1.4) presenta un comportamiento dinámico superior al método (2.1.3) cuando $\theta = 0$ y $\theta = \frac{1}{3}$. Además, ambos métodos tienden a tener comportamientos dinámicos similares a medida que θ se aproxima a 1, lo cual es esperable, dado que ambos métodos coinciden con el método de Newton cuando $\theta = 1$. En consecuencia, se concluye que la familia de métodos iterativos (2.1.4) presenta una mejor accesibilidad que la familia de métodos (2.1.3).

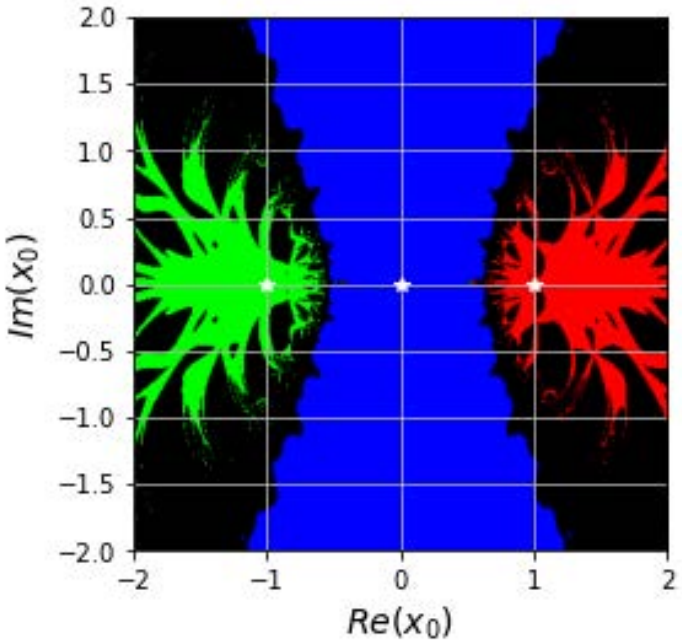


Figura 2.2: Plano dinámico de la familia (2.1.3) aplicada a la ecuación $F(z) = z^3 + z|z| - 2z$ con $N = 10$ y $\theta = 0$.

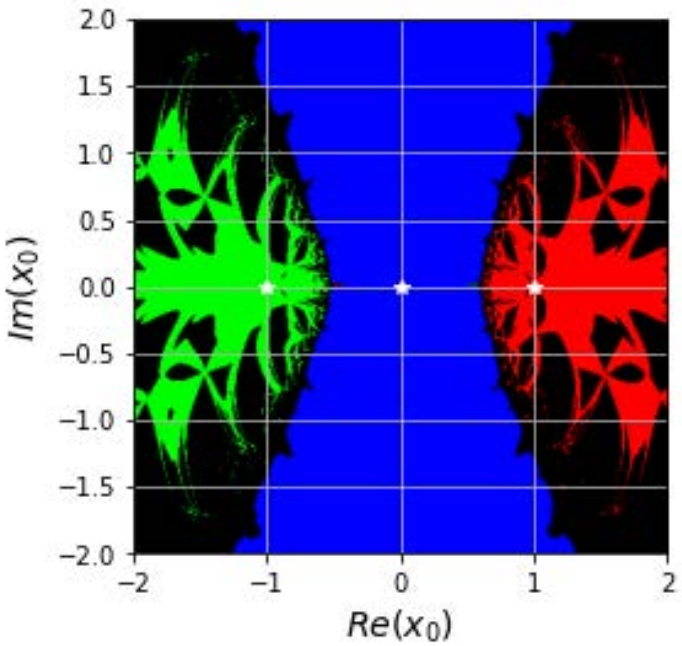


Figura 2.3: Plano dinámico de la familia (2.1.3) aplicada a la ecuación $F(z) = z^3 + z|z| - 2z$ con $N = 10$ y $\theta = 1/3$.

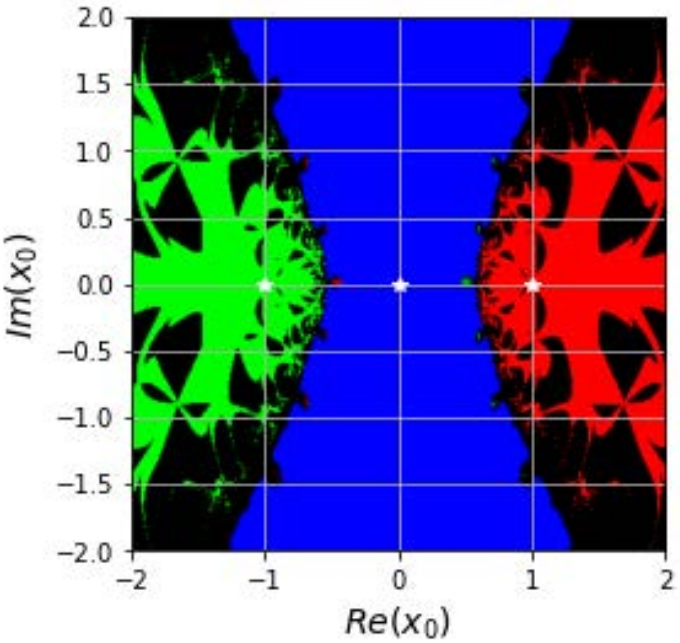


Figura 2.4: Plano dinámido de la familia (2.1.3) aplicada a la ecuación $F(z) = z^3 + z|z| - 2z$ con $N = 10$ y $\theta = 2/3$.

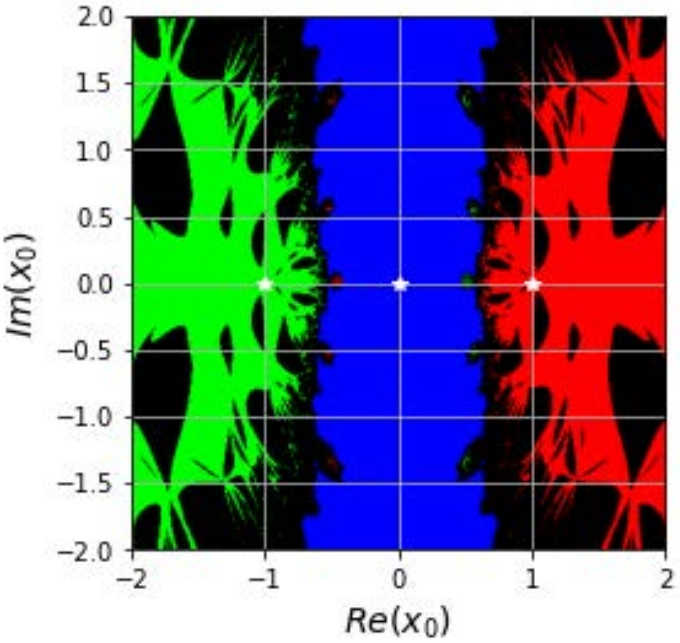


Figura 2.5: Plano dinámico de la familia (2.1.4) aplicada a la ecuación $F(z) = z^3 + z|z| - 2z$ con $N = 10$ y $\theta = 0$.

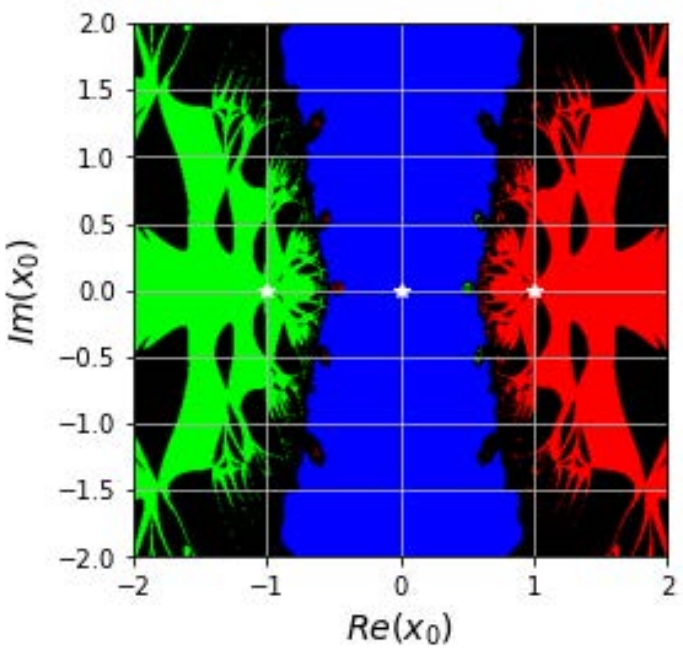


Figura 2.6: Plano dinámico de la familia (2.1.4) aplicada a la ecuación $F(z) = z^3 + z|z| - 2z$ con $N = 10$ y $\theta = 1/3$.

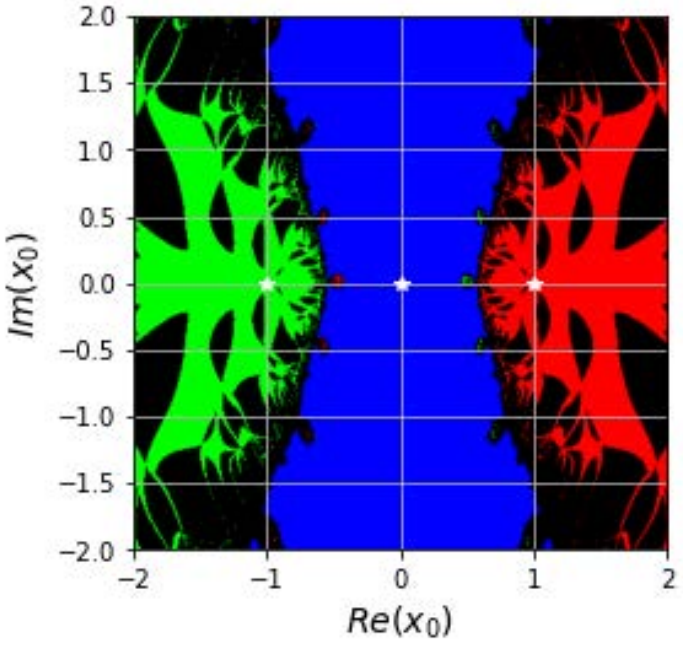


Figura 2.7: Plano dinámido de la familia (2.1.4) aplicada a la ecuación $F(z) = z^3 + z|z| - 2z$ con $N = 10$ y $\theta = 2/3$.

Después de analizar gráficamente la accesibilidad de las familias (2.1.3) y (2.1.4), se estudia su comportamiento de manera numérica. Para ello, se calcula el porcentaje de puntos que convergen a alguna de las raíces después de

$$N = 10$$

iteraciones, utilizando

$$tolerancia = 10^{-6}.$$

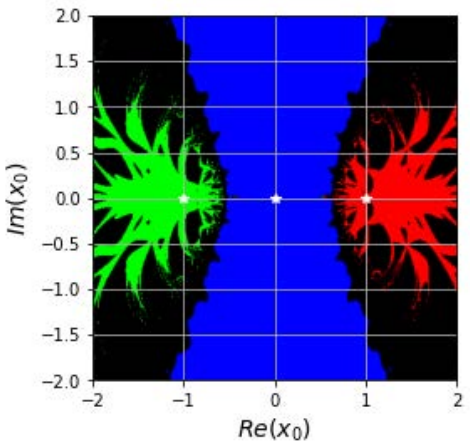
Los resultados se presentan en la Tabla 2.3, donde se observa que la accesibilidad de los métodos de la familia (2.1.4) es superior, excepto en el caso de

$$\theta = \frac{2}{3},$$

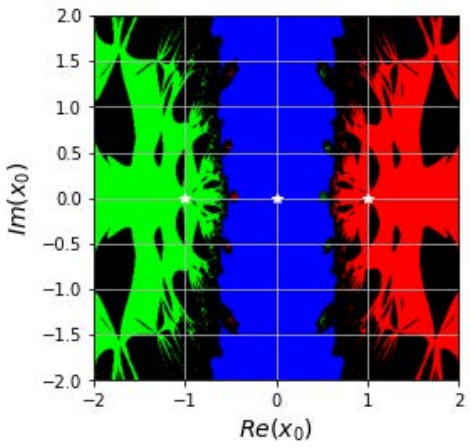
a la de los métodos de la familia (2.1.3). Esta misma información puede ser corroborada en cada caso en las comparativas de las cuencas que se observan en las Figuras 2.8- 2.10.

θ	Familia (2.1.3)	Tiempo en segundos	Método (2.1.4)	Tiempo en segundos
0	57.00 %	1.99140	62.29 %	2.00403
1/3	61.68 %	2.03198	63.82 %	1.96482
2/3	66.87 %	2.05299	63.99 %	2.00774

Tabla 2.3: Porcentaje de puntos de convergencia para $F(z) = z^3 + z|z| - 2z$



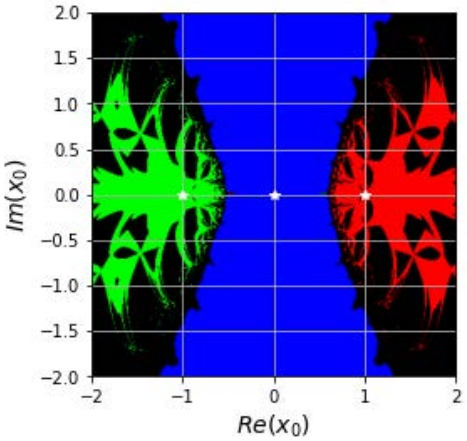
Iteración (2.1.3) con $\theta = 0$ y $N = 10$



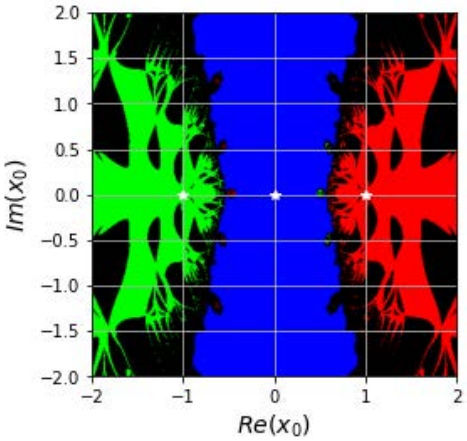
Iteración (2.1.4) con $\theta = 0$ y $N = 10$

Figura 2.8: Comparación de las cuencas de atracción de las tres raíces de $F(z) = z^3 + z|z| - 2z$ cuando se aplica las diferentes familias.

Por último, se adjuntan los códigos en Python para poder dibujar las cuencas de atracción.

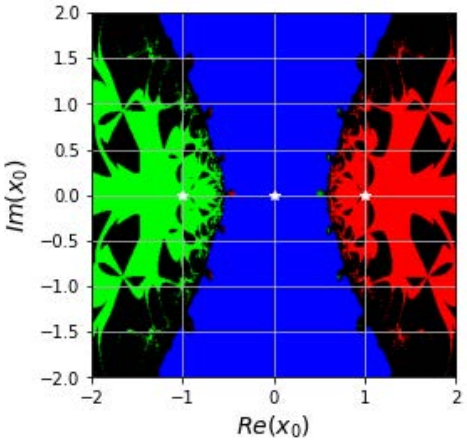


Iteración (2.1.3) con $\theta = 1/3$ y $N = 10$

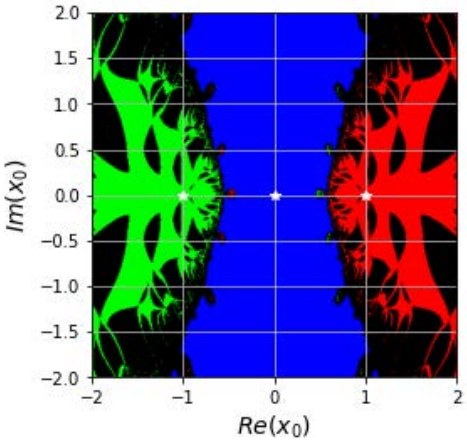


Iteración (2.1.4) con $\theta = 1/3$ y $N = 10$

Figura 2.9: Comparación de las cuencas de atracción de las tres raíces de $F(z) = z^3 + z|z| - 2z$ cuando se aplica las diferentes familias.



Iteración (2.1.3) con $\theta = 2/3$ y $N = 10$



Iteración (2.1.4) con $\theta = 2/3$ y $N = 10$

Figura 2.10: Comparación de las cuencas de atracción de las tres raíces de $F(z) = z^3 + z|z| - 2z$ cuando se aplica las diferentes familias.

Plano dinámico método de la familia(2.1.3) para z_0 complejo

```
# -*- coding: utf-8 -*-
import numpy as np
import matplotlib.pyplot as plt
import time

# Inicio de tiempo de computacion
start_time = time.time()

# Parametros del metodo
L = 2/3

# Funcion

F = lambda X: X**3 + X * np.abs(X) - 2 * X

# Aproximacion de la derivada, diferencia dividida
Y = lambda X, U: L * X + (1 - L) * U
DF = lambda X, U: (F(X) - F(Y(X, U))) / (X - Y(X, U))
m = lambda X, U: X - F(X) / DF(X, U)

# Valores iniciales
h=0.01 #Paso
Rez0 = np.arange(-2, 2+h,h)
Imz0 = np.arange(-2, 2+h, h)
maxiter = 10
tol = 1e-6

puntos = len(Rez0)
ReZ0, ImZ0 = np.meshgrid(Rez0, Imz0)
Z0 = ReZ0 + 1j * ImZ0
Z_1 = Z0 - 0.1
iter = 1
R = np.zeros((puntos, puntos))
G = np.zeros((puntos, puntos))
B = np.zeros((puntos, puntos))
IT = (maxiter + 1) * np.ones_like(Z0)

# Puntos fijos
ZF1 = 1
ZF2 = -1
ZF3 = 0

# Ejecucion del metodo iterativo
while iter <= maxiter:
    Z = m(Z0, Z_1)
    for fil in range(puntos):
        for col in range(puntos):
            Zfc = Z[fil, col]
            if abs(Zfc - ZF1) < tol:
                R[fil, col] = 1
                IT[fil, col] = min(iter, IT[fil, col])
            if abs(Zfc - ZF2) < tol:
                G[fil, col] = 1
                IT[fil, col] = min(iter, IT[fil, col])
            if abs(Zfc - ZF3) < tol:
                B[fil, col] = 1
```

```
                IT[fil, col] = min(iter, IT[fil, col])
    iter += 1
    Z_1 = Z0
    Z0 = Z

# Generacion de la imagen
I = np.zeros((puntos, puntos, 3))
I[:, :, 0] = R
I[:, :, 1] = G
I[:, :, 2] = B

# Representacion de la imagen
plt.imshow(I,origin='lower', extent=[Rez0[0],Rez0[-1],Imz0[0],Imz0
    [-1]])
plt.xlabel("$Re(x_{0})$", fontsize=14)
plt.ylabel("$Im(x_{0})$", fontsize=14)
#plt.title(f"Dynamical Plane, N   Iter={maxiter} lambda={L}")
plt.grid()
#Representacion puntos fijos
plt.plot(np.real(ZF1),np.imag(ZF1),'w*')
plt.plot(np.real(ZF2),np.imag(ZF2),'w*')
plt.plot(np.real(ZF3),np.imag(ZF3),'w*')
plt.show()

# Calculo de estadisticas
print("% de elementos que convergen:")
print(np.count_nonzero(IT <= maxiter) / np.size(IT) * 100)
print("Valor medio de iteraciones para converger:")
print(np.mean(IT[IT <= maxiter]))

# Tiempo de computacion
end_time = time.time()
print("Tiempo de computo:", end_time - start_time, "segundos")
```

Plano dinámico método de la familia(2.1.4) **para** z_0 **complejo**

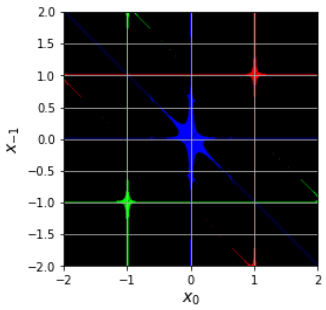
```
# -*- coding: utf-8 -*-
import numpy as np
import matplotlib.pyplot as plt
import time

# Inicio de tiempo de computacion
start_time = time.time()

#Parametro del metodo
L=2/3
#Metodo
#G parte diferenciable e la ecuacion x**3-2x
#H Parte no diferenciable de la ecuacion, X+abs(x)

F = lambda X: X**3 + X * abs(X) - 2 * X
DG = lambda X: 3 * X**2 - 2  # Derivada de G
H = lambda X: X * abs(X)
Y = lambda X, U: L * X + (1 - L) * U  # Aqui supongo que L es una
    constante dada
DH = lambda X, U: (H(Y(X, U)) - H(X)) / (Y(X, U) - X)
M = lambda X, U: X - F(X) / (DG(X) + DH(X, U))


# Valores iniciales
h=0.01 #Paso
Rez0 = np.arange(-2, 2+h,h)
Imz0 = np.arange(-2, 2+h,h)
maxiter = 10
tol = 1e-6

puntos = len(Rez0)
ReZ0, ImZ0 = np.meshgrid(Rez0, Imz0)
Z0 = ReZ0 + 1j * ImZ0
Z_1 = Z0 - 0.1
iter = 1
R = np.zeros((puntos, puntos))
G = np.zeros((puntos, puntos))
B = np.zeros((puntos, puntos))
IT = (maxiter + 1) * np.ones_like(Z0)

# Puntos fijos
ZF1 = 1
ZF2 = -1
ZF3 = 0

# Ejecucion del metodo iterativo
while iter <= maxiter:
    Z = M(Z0, Z_1)
    for fil in range(puntos):
        for col in range(puntos):
            Zfc = Z[fil, col]
            if abs(Zfc - ZF1) < tol:
                R[fil, col] = 1
                IT[fil, col] = min(iter, IT[fil, col])
            if abs(Zfc - ZF2) < tol:
                G[fil, col] = 1
```

```
                IT[fil, col] = min(iter, IT[fil, col])
            if abs(Zfc - ZF3) < tol:
                B[fil, col] = 1
                IT[fil, col] = min(iter, IT[fil, col])
    iter += 1
    Z_1 = Z0
    Z0 = Z

# Generacion de la imagen
I = np.zeros((puntos, puntos, 3))
I[:, :, 0] = R
I[:, :, 1] = G
I[:, :, 2] = B

# Representacion de la imagen
plt.imshow(I,origin='lower', extent=[Rez0[0],Rez0[-1],Imz0[0],Imz0
    [-1]])
plt.xlabel("$Re(x_{0})$", fontsize=14)
plt.ylabel("$Im(x_{0})$", fontsize=14)
#plt.title(f"Dynamical Plane, N   Iter={maxiter}   ={L}")
plt.grid()
#Representacion puntos fijos
plt.plot(np.real(ZF1),np.imag(ZF1),'w*')
plt.plot(np.real(ZF2),np.imag(ZF2),'w*')
plt.plot(np.real(ZF3),np.imag(ZF3),'w*')
plt.show()

# Calculo de estadisticas
print("% de elementos que convergen:")
print(np.count_nonzero(IT <= maxiter) / np.size(IT) * 100)
print("Valor medio de iteraciones para converger:")
print(np.mean(IT[IT <= maxiter]))

# Tiempo de computacion
end_time = time.time()
print("Tiempo de computo:", end_time - start_time, "segundos")
```

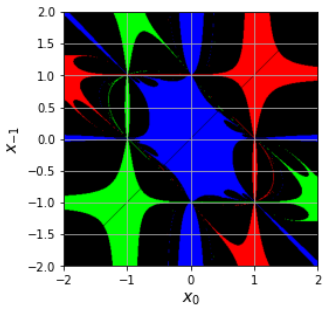
2.4.2. Segunda estrategia: z_{-1} y z_0 son libres

En esta segunda estrategia, se analiza el comportamiento dinámico de las familias (2.1.3) y (2.1.4) cuando tanto z_{-1} como z_0 son libres, siguiendo un enfoque de dinámica real [105]. Dado que las raíces de la ecuación $F(z) = z^3 + z|z| - 2z = 0$ son números reales, se toman z_{-1} y z_0 como números reales. En el plano de convergencia, los valores de z_0 se ubican en el eje horizontal, mientras que los valores de z_{-1} se representan en el eje vertical.

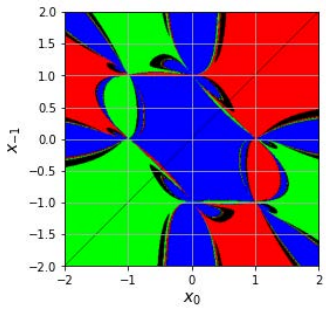
Siguiendo la estrategia mencionada, con una tolerancia de 10^{-6} y un máximo de N iteraciones, se generan los planos de convergencia para ambas familias de métodos iterativos. Las Figuras 2.11–2.16 muestran los comportamientos dinámicos de ambas familias de métodos, (2.1.3) y (2.1.4), para valores de $\theta = 0$, $1/3$, $2/3$ y $N = 10$,20,30. En estas figuras se observa que la familia (2.1.4) presenta un mejor comportamiento dinámico que la familia (2.1.3). A medida que N aumenta y θ se aproxima a 1, los comportamientos dinámicos de ambas familias se vuelven más similares.



Familia (2.1.3) con $\theta = 0$ y $N = 10$

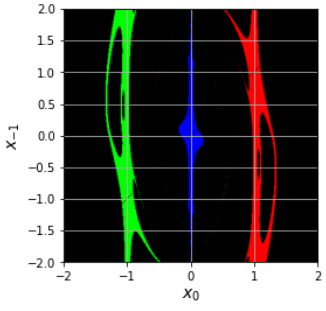


Familia (2.1.3) con $\theta = 0$ y $N = 20$

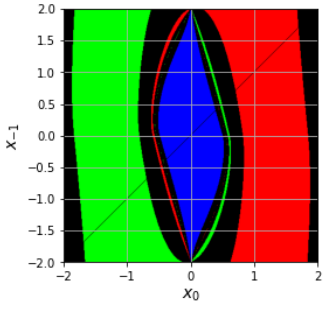


Familia (2.1.3) con $\theta = 0$ y $N = 30$

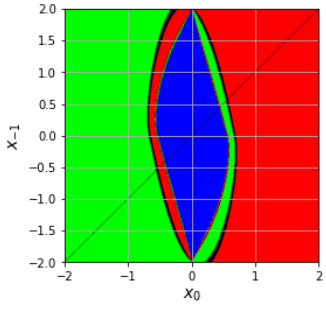
Figura 2.11: Cuencas de atracción de las tres raíces de $F(z) = z^3 + z|z| - 2z$ cuando se aplica la iteración (2.1.3)



Familia (2.1.4) con $\theta = 0$ y $N = 10$

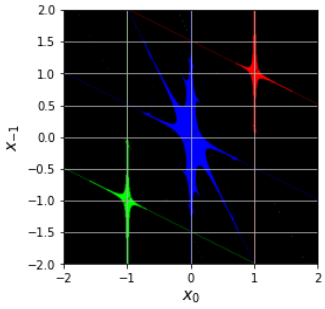


Familia (2.1.4) con $\theta = 0$ y $N = 20$

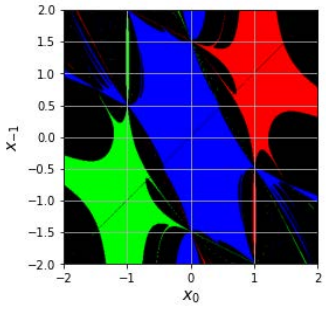


Familia (2.1.4) con $\theta = 0$ y $N = 30$

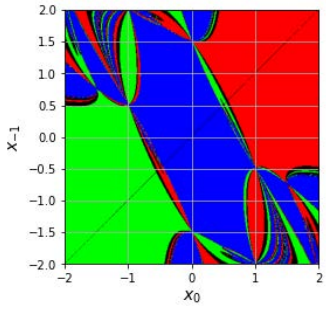
Figura 2.12: Cuencas de atracción de las tres raíces de $F(z) = z^3 + z|z| - 2z$ cuando se aplica la iteración (2.1.4)



Familia (2.1.3) con $\theta = \frac{1}{3}$ y $N = 10$

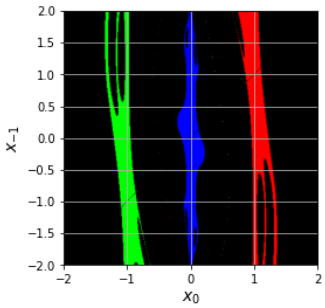


Familia (2.1.3) con $\theta = \frac{1}{3}$ y $N = 20$

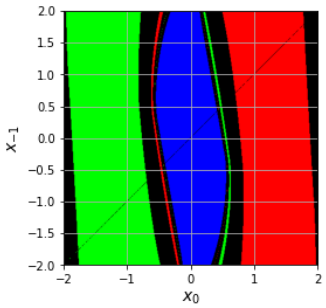


Familia (2.1.3) con $\theta = \frac{1}{3}$ y $N = 30$

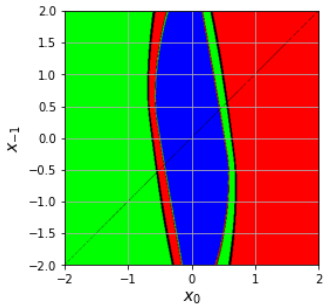
Figura 2.13: Cuencas de atracción de las tres raíces de $F(z) = z^3 + z|z| - 2z$ cuando se aplica la iteración (2.1.3)



Familia (2.1.4) con $\theta = \frac{1}{3}$ y $N = 10$

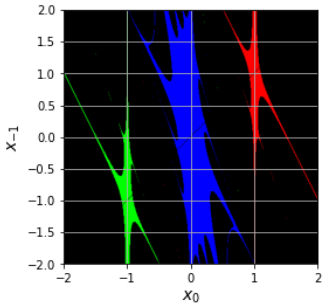


Familia (2.1.4) con $\theta = \frac{1}{3}$ y $N = 20$

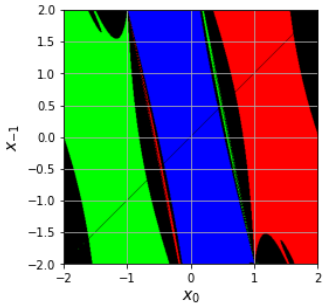


Familia (2.1.4) con $\theta = \frac{1}{3}$ y $N = 30$

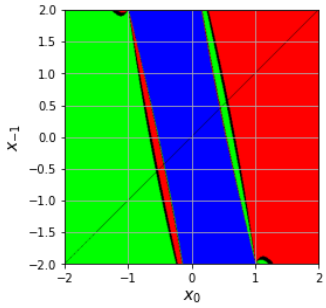
Figura 2.14: Cuencas de atracción de las tres raíces de $F(z) = z^3 + z|z| - 2z$ cuando se aplica la iteración (2.1.4)



Familia eq:sectype con $\theta = \frac{2}{3}$ y $N = 10$

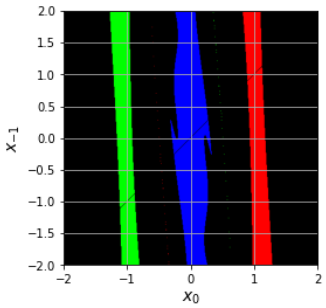


Familia eq:sectype con $\theta = \frac{2}{3}$ y $N = 20$

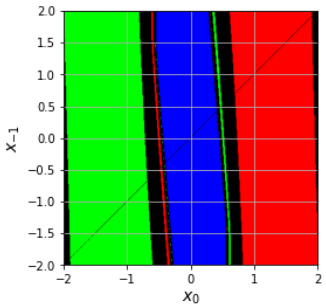


Familia eq:sectype con $\theta = \frac{2}{3}$ y $N = 30$

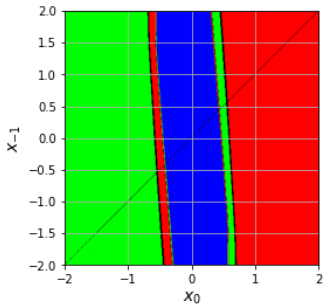
Figura 2.15: Cuencas de atracción de las tres raíces de $F(z) = z^3 + z|z| - 2z$ cuando se aplica la iteración (2.1.3)



Familia (2.1.4) con $\theta = \frac{2}{3}$ y $N = 10$



Familia (2.1.4) con $\theta = \frac{2}{3}$ y $N = 20$



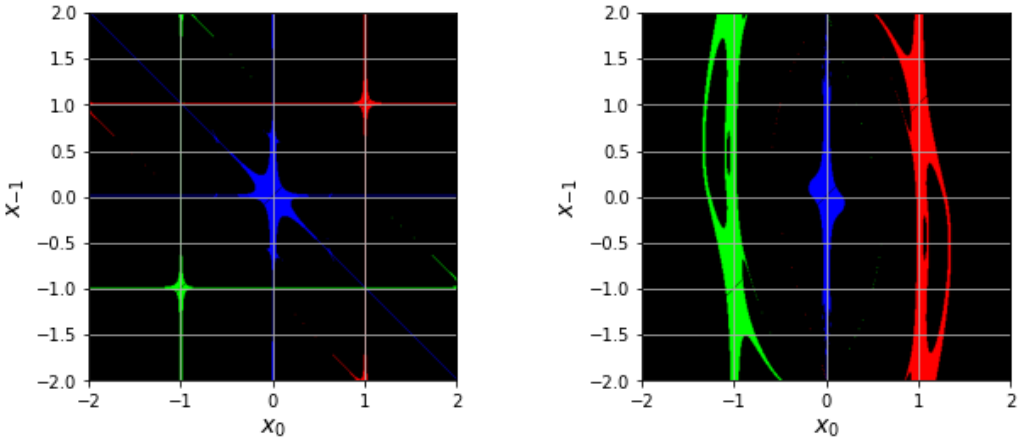
Familia (2.1.4) con $\theta = \frac{2}{3}$ y $N = 30$

Figura 2.16: Cuencas de atracción de las tres raíces de $F(z) = z^3 + z|z| - 2z$ cuando se aplica la iteración (2.1.4)

Tras el análisis gráfico, se examina el comportamiento dinámico real de forma numérica. El porcentaje de puntos que convergen a alguna de las raíces después de N iteraciones, con una tolerancia de 10^{-6}, se presenta en la Tabla 2.4. Los resultados confirman que la accesibilidad de la familia (2.1.4) es superior a la de la familia (2.1.3). Esta comparación también puede verse en las Figuras 2.17-2.25.

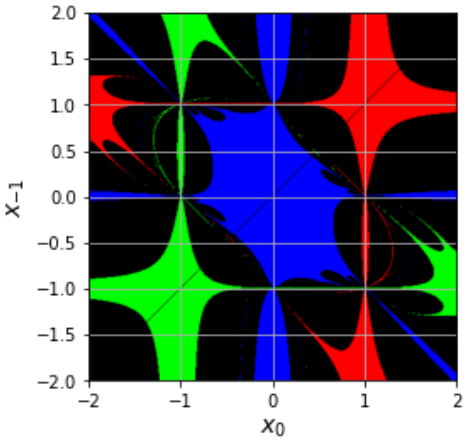
θ	N	Familia (2.1.3)	Familia (2.1.4)
0	10	4.16 %	11.90 %
	20	38.42 %	71.20 %
	30	93.82 %	97.21 %
1/3	10	5.95 %	15.35 %
	20	50.48 %	78.40 %
	30	94.03 %	97.84 %
2/3	10	15.65 %	23.40 %
	20	80.56 %	84.29 %
	30	97.94 %	98.19 %

Tabla 2.4: Porcentaje de puntos de convergencia para $F(z) = z^3 + z|z| - 2z$

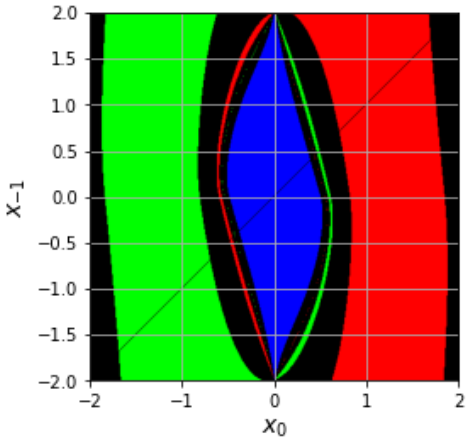


Iteración (2.1.3) con $\theta = 0$ y $N = 10$ Iteración (2.1.4) con $\theta = 0$ y $N = 10$

Figura 2.17: Comparación de las cuencas de atracción de las tres raíces de $F(z) = z^3 + z|z| - 2z$ cuando se aplica las diferentes familias.

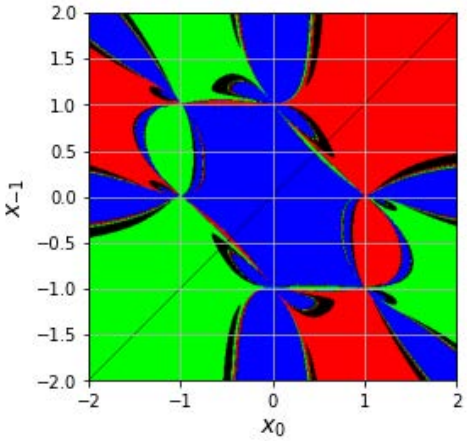


Iteración (2.1.3) con $\theta = 0$ y $N = 20$

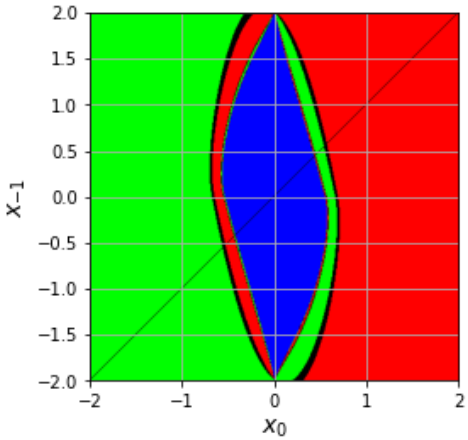


Iteración (2.1.4) con $\theta = 0$ y $N = 20$

Figura 2.18: Comparación de las cuencas de atracción de las tres raíces de $F(z) = z^3 + z|z| - 2z$ cuando se aplica las diferentes familias.

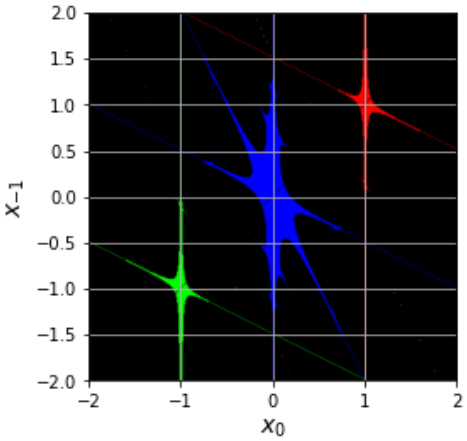


Iteración (2.1.3) con $\theta = 0$ y $N = 30$

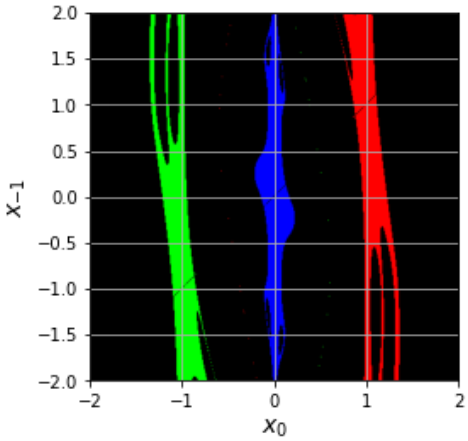


Iteración (2.1.4) con $\theta = 0$ y $N = 30$

Figura 2.19: Comparación de las cuencas de atracción de las tres raíces de $F(z) = z^3 + z|z| - 2z$ cuando se aplica las diferentes familias.

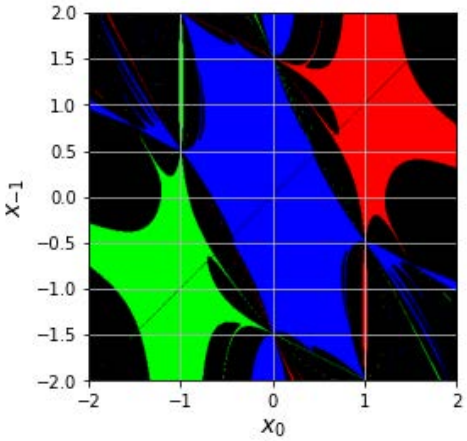


Iteración (2.1.3) con $\theta = 1/3$ y $N = 10$

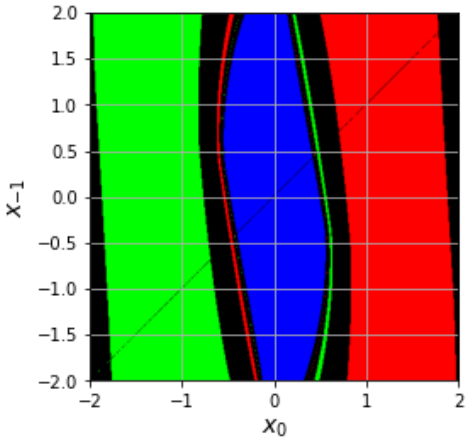


Iteración (2.1.4) con $\theta = 1/3$ y $N = 10$

Figura 2.20: Comparación de las cuencas de atracción de las tres raíces de $F(z) = z^3 + z|z| - 2z$ cuando se aplica las diferentes familias.

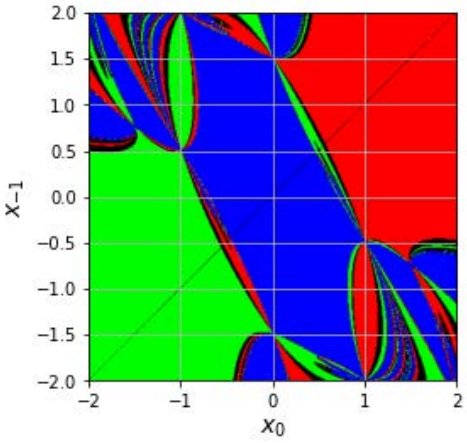


Iteración (2.1.3) con $\theta = 1/3$ y $N = 20$

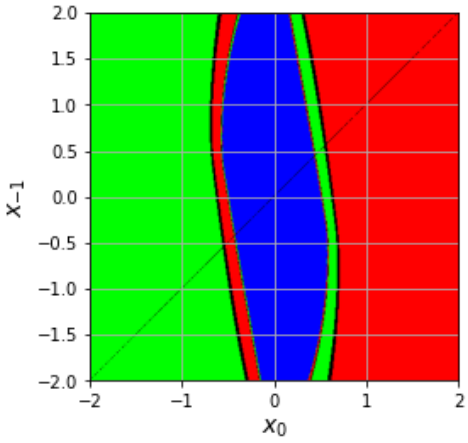


Iteración (2.1.4) con $\theta = 1/3$ y $N = 20$

Figura 2.21: Comparación de las cuencas de atracción de las tres raíces de $F(z) = z^3 + z|z| - 2z$ cuando se aplica las diferentes familias.

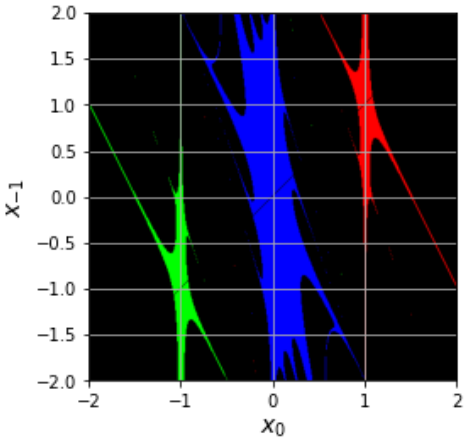


Iteración (2.1.3) con $\theta = 1/3$ y $N = 30$

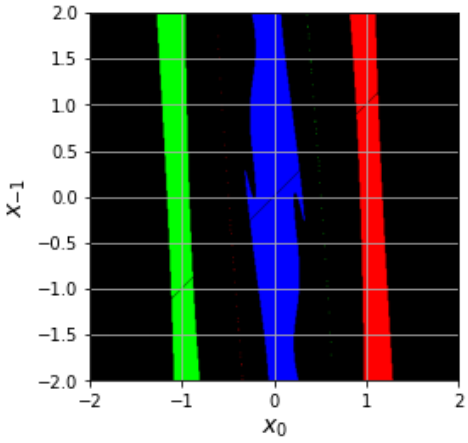


Iteración (2.1.4) con $\theta = 1/3$ y $N = 30$

Figura 2.22: Comparación de las cuencas de atracción de las tres raíces de $F(z) = z^3 + z|z| - 2z$ cuando se aplica las diferentes familias.

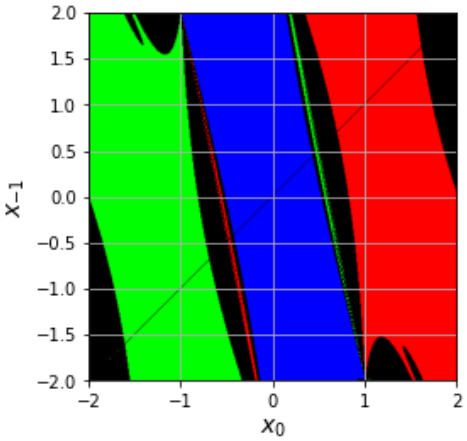


Iteración (2.1.3) con $\theta = 2/3$ y $N = 10$

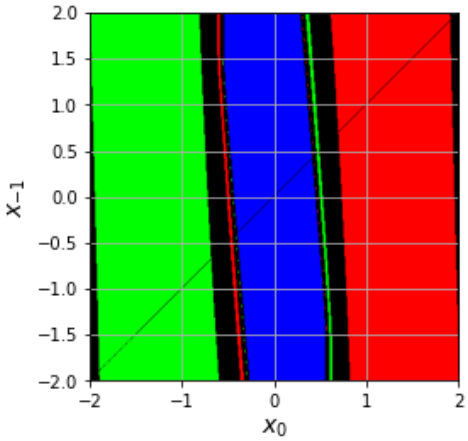


Iteración (2.1.4) con $\theta = 2/3$ y $N = 10$

Figura 2.23: Comparación de las cuencas de atracción de las tres raíces de $F(z) = z^3 + z|z| - 2z$ cuando se aplica las diferentes familias.

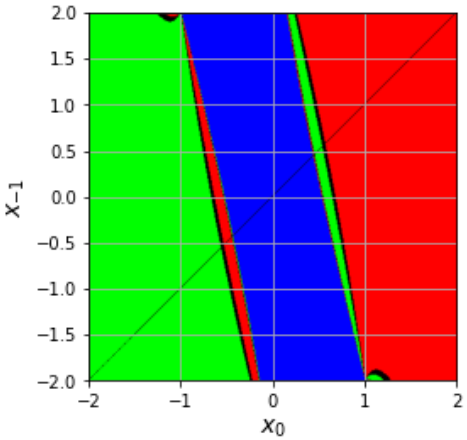


Iteración (2.1.3) con $\theta = 2/3$ y $N = 20$

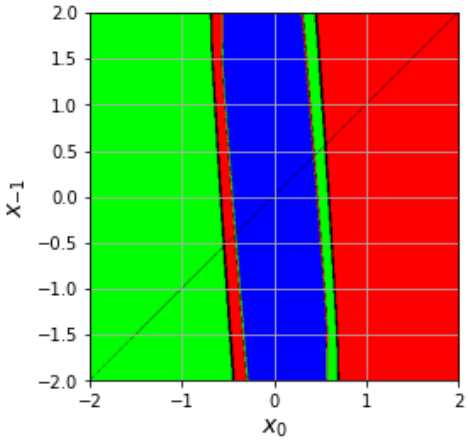


Iteración (2.1.4) con $\theta = 2/3$ y $N = 20$

Figura 2.24: Comparación de las cuencas de atracción de las tres raíces de $F(z) = z^3 + z|z| - 2z$ cuando se aplica las diferentes familias.



Iteración (2.1.3) con $\theta = 2/3$ y $N = 30$



Iteración (2.1.4) con $\theta = 2/3$ y $N = 30$

Figura 2.25: Comparación de las cuencas de atracción de las tres raíces de $F(z) = z^3 + z|z| - 2z$ cuando se aplica las diferentes familias.

Plano dinámico método de la familia(2.1.3) **para** z_0 **y** z_{-1} **reales**

```
# -*- coding: utf-8 -*-

import numpy as np
import matplotlib.pyplot as plt
import time

#Metodo
# Parametros del metodo
L = 2/3

# Funcion

F = lambda X: X**3 + X * np.abs(X) - 2 * X

# Aproximacion de la derivada, diferencia dividida
Y = lambda X, U: L * X + (1 - L) * U
DF = lambda X, U: (F(X) - F(Y(X, U))) / (X - Y(X, U))
M = lambda X, U: X - F(X) / DF(X, U)


tic=time.time() #Inicio tiempo de computacion
h=0.01      #Tama o de paso del mallado
tol=1E-6    #Tolerancia de aproximaci n
maxiter=10 #Numero maximo iteraciones

z0=np.arange(-2,2+h,h)
z_1=np.arange(-2,2+h,h)
Z0, Z_1=np.meshgrid(z0,z_1, indexing='xy')

itera=1 #Inicializacion parametro para iteraciones
#  R G B para los colores del plano dinamico
#  I, para combinar R G B
R=np.zeros([len(z0),len(z_1)])
G=np.zeros([len(z0),len(z_1)])
B=np.zeros([len(z0),len(z_1)])
I=np.zeros([len(z0),len(z_1),3])
IT=(maxiter+1)*np.ones([len(z0),len(z_1)])
#  Puntos fijos
ZF1=1
ZF2=-1
ZF3=0


while itera<=maxiter:
    Z=M(Z0,Z_1)

    for fil in range(len(z0)):
        for col in range(len(z_1)):
            Zfc=Z[fil,col]
            if np.abs(Zfc-ZF1)<tol:
                R[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
            if np.abs(Zfc-ZF2)<tol:
                G[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
```

```
            if np.abs(Zfc-ZF3)<tol:
                B[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
    #linalg.norm
    #Actualizacion de valores
    itera=itera+1
    Z_1=Z0
    Z0=Z

#Generacion de la imagen del plano
I[...,0]=R
I[...,1]=G
I[...,2]=B

# Se muestra el plano dinamico
plt.figure()
#plt.title('Dynamical Plane, Num Iter=%s' %(str(maxiter)))
plt.xlabel("$x_{0}$", fontsize=14)
plt.ylabel("$x_{-1}$", fontsize=14)
plt.grid()
#Se indica origin='lower' para indicar que el origen esta en la
    esquina
#Inferior izquieda
plt.imshow(I,origin='lower', extent=[z0[0],z0[-1],z_1[0],z_1[-1]])

#Numero de iteraciones promedio
print('% de iteraciones que convergen: ', np.size(IT[IT<=maxiter])/np
    .size(IT)*100)
#Valor medio de iteraciones para converger
print('Valor medio de iteraciones para converger: ',np.mean(IT[IT<=
    maxiter]))
#Tiempo de computacion
toc=time.time()
print('Tiempo de computacion:', toc-tic,' segundos')
```

Plano dinámico método de la familia(2.1.4) **para** z_0 **y** z_{-1} **reales**

```
# -*- coding: utf-8 -*-

import numpy as np
import matplotlib.pyplot as plt
import time

#Parametro del metodo
L=2/3
#Metodo
#G parte diferenciable e la ecuacion x**3-2x
#H Parte no diferenciable de la ecuacion, X+abs(x)

F = lambda X: X**3 + X * abs(X) - 2 * X
DG = lambda X: 3 * X**2 - 2  # Derivada de G
H = lambda X: X * abs(X)
Y = lambda X, U: L * X + (1 - L) * U  # Aqui supongo que L es una
    constante dada
DH = lambda X, U: (H(Y(X, U)) - H(X)) / (Y(X, U) - X)
M = lambda X, U: X - F(X) / (DG(X) + DH(X, U))

tic=time.time() #Inicio tiempo de computacion
h=0.01      #Paso del mallado
tol=1E-6    #Tolerancia de aproximacion
maxiter=10 #Numero maximo iteraciones

z0=np.arange(-2,2+h,h)
z_1=np.arange(-2,2+h,h)
Z0, Z_1=np.meshgrid(z0,z_1, indexing='xy')

itera=1 #Inicializacion parametro para iteraciones
#  R G B para los colores del plano dinamico
#  I, para combinar R G B
R=np.zeros([len(z0),len(z_1)])
G=np.zeros([len(z0),len(z_1)])
B=np.zeros([len(z0),len(z_1)])
I=np.zeros([len(z0),len(z_1),3])
IT=(maxiter+1)*np.ones([len(z0),len(z_1)])
#  Puntos fijos
ZF1=1
ZF2=-1
ZF3=0


while itera<=maxiter:
    Z=M(Z0,Z_1)

    for fil in range(len(z0)):
        for col in range(len(z_1)):
            Zfc=Z[fil,col]
            if np.abs(Zfc-ZF1)<tol:
                R[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
            if np.abs(Zfc-ZF2)<tol:
                G[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
```

```
            if np.abs(Zfc-ZF3)<tol:
                B[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
    #linalg.norm
    #Actualizacion de valores
    itera=itera+1
    Z_1=Z0
    Z0=Z

#Generacion de la imagen del plano
I[...,0]=R
I[...,1]=G
I[...,2]=B

# Se muestra el plano dinamico
plt.figure()
#plt.title('Dynamical Plane, Num Iter=%s' %(str(maxiter)))
plt.xlabel("$x_{0}$", fontsize=14)
plt.ylabel("$x_{-1}$", fontsize=14)
plt.grid()
#Se indica origin='lower' para indicar que el origen esta en la
    esquina
#Inferior izquieda
plt.imshow(I,origin='lower', extent=[z0[0],z0[-1],z_1[0],z_1[-1]])

#Numero de iteraciones promedio
print('% de iteraciones que convergen: ', np.size(IT[IT<=maxiter])/np
    .size(IT)*100)
#Valor medio de iteraciones para converger
print('Valor medio de iteraciones para converger: ',np.mean(IT[IT<=
    maxiter]))
#Tiempo de computacion
toc=time.time()
print('Tiempo de computacion:', toc-tic,' segundos')
```

2.5. Conclusiones

En muchas de las ramas de la ciencia, los problemas se centran en encontrar las soluciones de una ecuación no lineal del estilo

$$F(x) = 0.$$

Generalmente, para resolver este tipo de ecuaciones se emplean métodos iterativos. Cuando el operador F es diferenciable, el método de Newton es la alternativa más usada, debido a sus buenas características en cuanto a la velocidad de convergencia, la simplicidad de su uso y la efectividad del mismo. No obstante, si el operador F no es diferenciable o incluso es demasiado costosas de computar, se emplean métodos iterativos que en en lugar de derivadas, emplean diferencias divididas en su algoritmo. Entre estos, el más conocido es la alternativa al método de Newton, el método de la secante. En este capítulo se ha estudiado la familia de métodos iterativos tipo Newton-secante descritos en (2.1.4) y que engloba tanto al método de Newton, como a su alternativa libre de derivadas, el método de la secante.

Dado que la accesibilidad del método de la secante, entendida como el conjunto de aproximaciones iniciales que aseguran la convergencia del método, presenta dificultades en la aplicación, se propone una mejora en este trabajo mediante la técnica de descomposición del operador F. Específicamente, el operador se descompone como

$$F(x) = G(x) + H(x),$$

donde G es diferenciable y F es continuo pero no diferenciable. Teniendo en cuenta esta descomposición, se ha presentado un resultado relativo a la convergencia local para la convergencia de los métodos de dicha familia, exigiendo condiciones considerando que tiene una parte diferenciable y una que no lo es. Además, conjuntamente se obtiene también un resultado relativo a la unicidad de solución de la ecuación, donde se presenta el radio de la bola donde dicha solución es única.

Por otro lado, se presentan ilustran los resultados teóricos con un ejemplo que aborda una ecuación integral de Hammerstein de segunda especie:

$$x(s) = f(s) + \int_a^b \mathcal{K}(s,t)\left(\lambda x(t)^2 + \mu|x(t)|\right) dt,\ s \in [a,b].$$

Para resolver dicha ecuación se considera la forma general de los métodos de la familia así como sus casos particulares, como son el método de la secante y el método de Newton.

Por otro lado, una vez que ya se han probado los resultados teóricos aplicándolos a dicha ecuación, se ha procedido a realizar un estudio de la accesibilidad de los métodos de la familia. Para llevar a cabo este estudio, se analiza desde dos perspectivas diferentes el comportamiento de los mismos cuando se aplican a la ecuación

$$F(z) = z^3 + z|z| - 2z.$$

El primero de los enfoques es fijando uno de los puntos como

$$z_{-1} = z_0 - \frac{1}{10},$$

siendo z_0 un número complejo y tomando diferentes valores del parámetro θ, comparando además el porcentaje de puntos que convergen a las raíces y el tiempo que se requiere para converger. En este caso se ha utilizado una tolerancia de 10^{-6} y un número máximo de iteaciones de $N = 10$.

El segundo de los enfoques es considerando libres, en la recta real, los dos puntos iniciales que necesitan los métodos presentados y se comparan diferentes valores del parámetro θ, en relación al porcentaje de puntos que convergen a las raíces. En este caso se ha utilizado una tolerancia de 10^{-6} y un número máximo de iteaciones de $N = 10$, $N = 20$ y $N = 30$.

Por último, para ambos enfoques se han facilitado códigos para poder replicarlos de forma fácil en Python.

Capítulo 3

Una mejora significativa de una familia de métodos tipo secante

Los resultados de este capítulo han sido publicados en [59]:

- "A significant improvement of a family of secant-type methods".
- DOI: https://doi.org/10.1016/j.cam.2022.115002.
- Revista "Journal of Computational and Applied Mathematics".

3.1. Introducción

Como ya se ha comentado en los capítulos anteriores, el método de Newton, junto a sus variantes, es ampliamente utilizado para resolver ecuaciones con operadores no lineales de la forma $F(x) = 0$. Este tipo de ecuación puede representar una ecuación diferencial, integral o, tras un proceso de discretización, un sistema no lineal. En este capítulo se considera una función no lineal $F : \Omega \subset \mathbb{R}^m \to \mathbb{R}^m$, donde Ω es un dominio convexo abierto en $\mathbb{R}^m$, con

$$F(x) = (F_1(x), F_2(x), \ldots, F_m(x))$$

y

$$F_i : \Omega \subseteq \mathbb{R}^m \to \mathbb{R}$$

para cada $i = 1, 2, \ldots, m$, con

$$x = (x_1, x_2, \ldots, x_m).$$

El método de Newton para aproximar una solución w^* de la ecuación $F(x) = 0$ es el siguiente:

$$\begin{cases} w_0 \text{ dado en } \Omega, \\ w_{n+1} = w_n - [F'(w_n)]^{-1} F(w_n), \quad n \geq 0. \end{cases} \tag{3.1.1}$$

Entre los múltiples beneficios que presenta este método destacan dos en concreto: su convergencia cuadrática y su bajo costo operativo, lo que garantiza una alta eficiencia. En el lado opuesto, la principal limitación es que la derivada $F'(x)$ debe ser evaluada en cada iteración, lo que impide su aplicación en problemas con operadores no diferenciables o donde la derivada es costosa de calcular.

Cuando se dan estas circunstancias, es común aproximar las derivadas mediante diferencias divididas y emplear métodos iterativos basados en dichas aproximaciones. Entre estos métodos, el de la secante, ya presentado anteriormente, se convierte en una modificación directa, ([6, 138]), con la siguiente formulación:

$$\begin{cases} w_0, w_{-1} \text{ dados en } \Omega, \\ w_{n+1} = w_n - [w_{n-1}, w_n; F]^{-1} F(w_n), \quad n \geq 0, \end{cases} \tag{3.1.2}$$

donde $[u, v; F]$ es una diferencia dividida de primer orden en Ω, lo que le permite tener una convergencia superlineal con orden de convergencia R de al menos $\frac{1+\sqrt{5}}{2}$ ([54]).

Como se muestra en [22, 75], el operador $[u, v; F]$, perteneciente al conjunto de funciones lineales acotadas $\mathcal{L}(\mathbb{R}^m, \mathbb{R}^m)$, es una diferencia dividida de primer orden para F en u y v $(u \neq v)$, si se cumple que

$$[u, v; F](u - v) = F(u) - F(v), \quad u, v \in \Omega.$$

En [85], los autores construyeron una familia uniparamétrica de métodos iterativos del tipo secante para resolver $F(x) = 0$:

$$\begin{cases} w_{-1}, w_0 \text{ dados en } \Omega, \quad \lambda \in [0, 1], \\ u_n = \lambda w_n + (1 - \lambda) w_{n-1}, \quad n \geq 0, \\ w_{n+1} = w_n - [u_n, w_n; F]^{-1} F(w_n), \end{cases} \tag{3.1.3}$$

donde el parámetro $\lambda \in [0, 1]$.

Esta familia incluye el método de la secante para $\lambda = 0$ y el método de Newton para $\lambda = 1$ cuando F es diferenciable. Además, el orden de convergencia R de (3.1.3) es, al menos, $\frac{1+\sqrt{5}}{2}$ para $\lambda \in [0, 1)$, de manera similar al método de la secante.

El análisis de los métodos tipo secante de la familia (3.1.3) muestra que, cuando w_n y u_n están más cerca, se incrementa la velocidad de convergencia. De hecho, en [85] se indica que un valor más elevado de λ en (3.1.3) incrementa la velocidad de convergencia, acercándose a la del método de Newton cuando λ está cerca de 1.

La elección de un método iterativo para resolver $F(x) = 0$ generalmente depende de su eficiencia, la cual relaciona la velocidad de convergencia (orden de convergencia R) con su costo computacional. La familia (3.1.3) tiene un coste operacional similar a los métodos de la secante y Newton, pero no mejora la velocidad de convergencia del método de la secante. Por lo tanto, el primer objetivo en este capítulo es modificar la familia (3.1.3) para obtener una convergencia cuadrática, como la del método de Newton.

Es bien conocido que las diferencias divididas simétricas tienden a proporcionar mejores aproximaciones de las derivadas. En lugar de utilizar la diferencia dividida $[u_n, w_n; F]$ para aproximar $F'(w_n)$, se considera

$$[w_n - (w_n - u_n), w_n + (w_n - u_n); F] \approx F'(w_n).$$

Haciendo uso de la dicha diferencia dividida, se construye familia uniparamétrica de métodos iterativos:

$$\begin{cases} w_{-1}, w_0 \text{ dados en } \Omega, \quad \lambda \in [0,1], \\ u_n = \lambda w_n + (1-\lambda) w_{n-1}, \quad n \geq 0, \\ w_{n+1} = w_n - [u_n, 2w_n - u_n; F]^{-1} F(w_n). \end{cases} \tag{3.1.4}$$

El orden de convergencia R de (3.1.4) es al menos cuadrático para $\lambda \in [0,1]$, equivalente al del método de Newton. Además, el método (3.1.4) se reduce al método de Kurchatov ([100, 137, 147]) cuando $\lambda = 0$, y al método de Newton cuando $\lambda = 1$ y F es diferenciable.

Para asegurar un segundo orden de convergencia en la práctica, es necesario contar con una buena aproximación de la derivada primera de F en el método de Newton. En caso contrario, serían necesarias algunas iteraciones adicionales. Básicamente, cuando la norma de $F(x)$ es grande, la aproximación de la diferencia dividida a la derivada primera de F es deficiente.

El segundo objetivo de este capítulo es introducir un parámetro $\texttt{tol} > 0$ que permita ajustar la proximidad de w_n en cada componente de la diferencia dividida y obtener una mejor aproximación de $F'(w_n)$.

De este modo, se tiene:

$$[w_n - \texttt{tol}(w_n - u_n), w_n + \texttt{tol}(w_n - u_n); F] \approx F'(w_n)$$

y se obtiene la siguiente familia biparamétrica de métodos iterativos:

$$\begin{cases} w_{-1}, w_0 \text{ dados en } \Omega, \quad \lambda \in [0,1], \\ u_n = \lambda w_n + (1-\lambda) w_{n-1}, \quad n \geq 0, \\ w_{n+1} = w_n - [w_n - \texttt{tol}(w_n - u_n), w_n + \texttt{tol}(w_n - u_n); F]^{-1} F(w_n). \end{cases} \tag{3.1.5}$$

En este caso, una vez que se fija un $\lambda \in [0,1)$, se puede mejorar el método iterativo ajustando el parámetro $\texttt{tol}$ para obtener una mejor aproximación de $F'(w_n)$. Es importante señalar que (3.1.4) y (3.1.5) son equivalentes si se toma $\texttt{tol} = 1$. Además, para un mismo valor de λ, el método (3.1.5) mejora el método (3.1.4) cuando $\texttt{tol} < 1$, con mejores resultados a medida que $\texttt{tol}$ disminuye.

El principal aporte de este trabajo es la presentación de una familia biparamétrica de métodos iterativos, dada por (3.1.5), que resuelve sistemas no lineales de ecuaciones con mayor eficiencia computacional que la familia (3.1.3).

La estructura del capítulo es la siguiente:

- En la Sección 3.2, se presenta el orden de convergencia del método iterativo (3.1.5).
- En la Sección 3.3, se presenta una comparativa de las eficiencias de las familias (3.1.3), (3.1.4) y (3.1.5), donde puede verse como la familia (3.1.5)es la más eficiente.
- En la Sección 3.4, se estudia la convergencia local de la familia bajo ciertas condiciones sobre la diferencia dividida utilizada en el algoritmo.
- En la Sección 3.5, se proporcionan tres ejemplos que ilustran los objetivos del capítulo.
- En la Sección 3.6, se lleva a cabo un estudio dinámico comparativo entre las familias (3.1.3), (3.1.4) y (3.1.5). Asimismo, se analiza la accesibilidad de la familia desde dos perspectivas. Primero, de forma experimental, mediante un estudio dinámico de los métodos iterativos, y, segundo, a través de las bolas de convergencia obtenidas a partir de un análisis local de la convergencia. Esto permite concluir que el método iterativo (3.1.5) presenta la mejor accesibilidad.
- Por último, en la Sección 3.7, se presentan las conclusiones que se pueden extraer de este capítulo.

3.2. Orden local de convergencia

En esta sección se demuestra que el orden local de convergencia de la familia (3.1.4) es cuadrático. Para ello, se utiliza un resultado conocido ([54]), que establece el siguiente resultado para una familia biparamétrica de métodos iterativos:

$$\begin{cases} w_0, w_{-1} \text{ dados en } \Omega, \quad \gamma, \xi \in \mathbb{R}, \\ u_n = \gamma w_n + (1-\gamma) w_{n-1}, \quad n \geq 0, \\ v_n = \xi w_n + (1-\xi) w_{n-1}, \\ w_{n+1} = w_n - [u_n, v_n; F]^{-1} F(w_n). \end{cases} \tag{3.2.1}$$

Teorema 3.1. *([54], página 204) La familia de iteraciones definida en* (3.2.1) *tiene un orden de convergencia R de al menos 2 si $\gamma + \xi = 2$ y de al menos $\frac{1+\sqrt{5}}{2}$ si $\gamma + \xi \neq 2$. Más concretamente, si $F \in C^4(\Omega)$, $F'(w^*)$ no es singular y*

$$A_k = \frac{1}{k!}[F'(w^*)]^{-1} F^{(k)}(w^*),$$

para $k = 2, 3$, entonces:

$$e_{n+1} = A_2 e_n^2 + (1-\gamma)^2 A_3 e_{n-1}^2 e_n + \psi_1(w^*, e_{n-1}, e_n) \quad si \quad \gamma + \xi = 2, \tag{3.2.2}$$

$$e_{n+1} = (2-\gamma-\xi) A_2 e_{n-1} e_n + (\gamma + \xi - 1) A_2 e_n^2 + \psi_2(w^*, e_{n-1}) \quad si \quad \gamma + \xi \neq 2, \tag{3.2.3}$$

donde

$$e_k = w_k - w^*,$$

$$\|\psi_1(w^*, e_{n-1}, e_n)\| = o(\|e_n\|^2 \, \|e_{n-1}\|)$$

y

$$\|\psi_2(w^*, e_{n-1})\| = o(\|e_{n-1}\|^2).$$

A partir de (3.1.5),

$$w_n - \texttt{tol}(w_n - u_n) = (1 - \texttt{tol}(1-\lambda)) w_n + \texttt{tol}(1-\lambda) w_{n-1},$$

$$w_n + \texttt{tol}(w_n - u_n) = (1 + \texttt{tol}(1-\lambda)) w_n - \texttt{tol}(1-\lambda) w_{n-1}$$

y (3.2.2), se puede deducir que la familia (3.1.5) tiene un orden local de convergencia cuadrático.

3.3. Análisis de la eficiencia de las familias (3.1.3), (3.1.4) y (3.1.5)

Para comparar las eficiencias de las familias (3.1.3), (3.1.4) y (3.1.5), una vez conocidos los órdenes de convergencia R, es necesario conocer el número de operaciones (productos y divisiones) requeridas por los algoritmos de cada familia en cada paso.

Dado que el trabajo práctico se centra en la resolución de sistemas de ecuaciones no lineales, se utiliza la diferencia dividida dada por la matriz

$$[u, v; F] = ([u, v; F]_{ij})_{i,j=1}^{m} \in \mathcal{L}(\mathbb{R}^m, \mathbb{R}^m),$$

donde

$$[u, v; F]_{ij} = \frac{1}{u_j - v_j} \left(F_i(u_1, \ldots, u_{j-1}, u_j, v_{j+1}, \ldots, v_m) - F_i(u_1, \ldots, u_{j-1}, v_j, v_{j+1}, \ldots, v_m)\right),$$

si $u_j \neq v_j$, para $i, j = 1, 2, \ldots, m$ con

$$u = (u_1, u_2, \ldots, u_m)$$

y

$$v = (v_1, v_2, \ldots, v_m)$$

(ver [83]).

Se observa que para calcular w_{n+1} en la familia (3.1.3), es necesario realizar $2m$ operaciones para u_n, $m^2 + 2m$ para $[u_n, w_n; F]$, $(m^3 - m)/3$ para la descomposición LU y m^2 para resolver dos sistemas lineales triangulares. Por lo tanto, el número total de operaciones requeridas es

$$p(m) = \frac{m}{3}\left(m^2 + 6m + 11\right)$$

y la eficiencia computacional, definida como el orden de convergencia del método elevado al inverso del número de operaciones necesarias por iteración (ver [124, 155]), es

$$CE_{(3.1.3)} = \left(\frac{1 + \sqrt{5}}{2}\right)^{\frac{1}{p(m)}}.$$

Se debe tener en cuenta que la familia (3.1.4) requiere las mismas operaciones que la familia (3.1.3) más m operaciones adicionales para calcular $2\,w_n$, por lo que su eficiencia computacional es

$$CE_{(3.1.4)} = 2^{\frac{1}{p(m)+m}}.$$

De manera similar, la familia (3.1.5) requiere las mismas operaciones que la familia (3.1.3) más $2m$ operaciones para calcular $w_n - \texttt{tol}(w_n - u_n)$ y $w_n + \texttt{tol}(w_n - u_n)$, resultando en una eficiencia computacional de

$$CE_{(3.1.5)} = 2^{\frac{1}{p(m)+2m}}.$$

A continuación, se presenta una comparación de la eficiencia para diferentes valores de m en la Figura 3.1, donde se observa que las familias (3.1.4) y (3.1.5) son más eficientes que (3.1.3), y que las familias (3.1.4) y (3.1.5) tienen una eficiencia computacional similar. Para este análisis, se emplea una escala logarítmica.

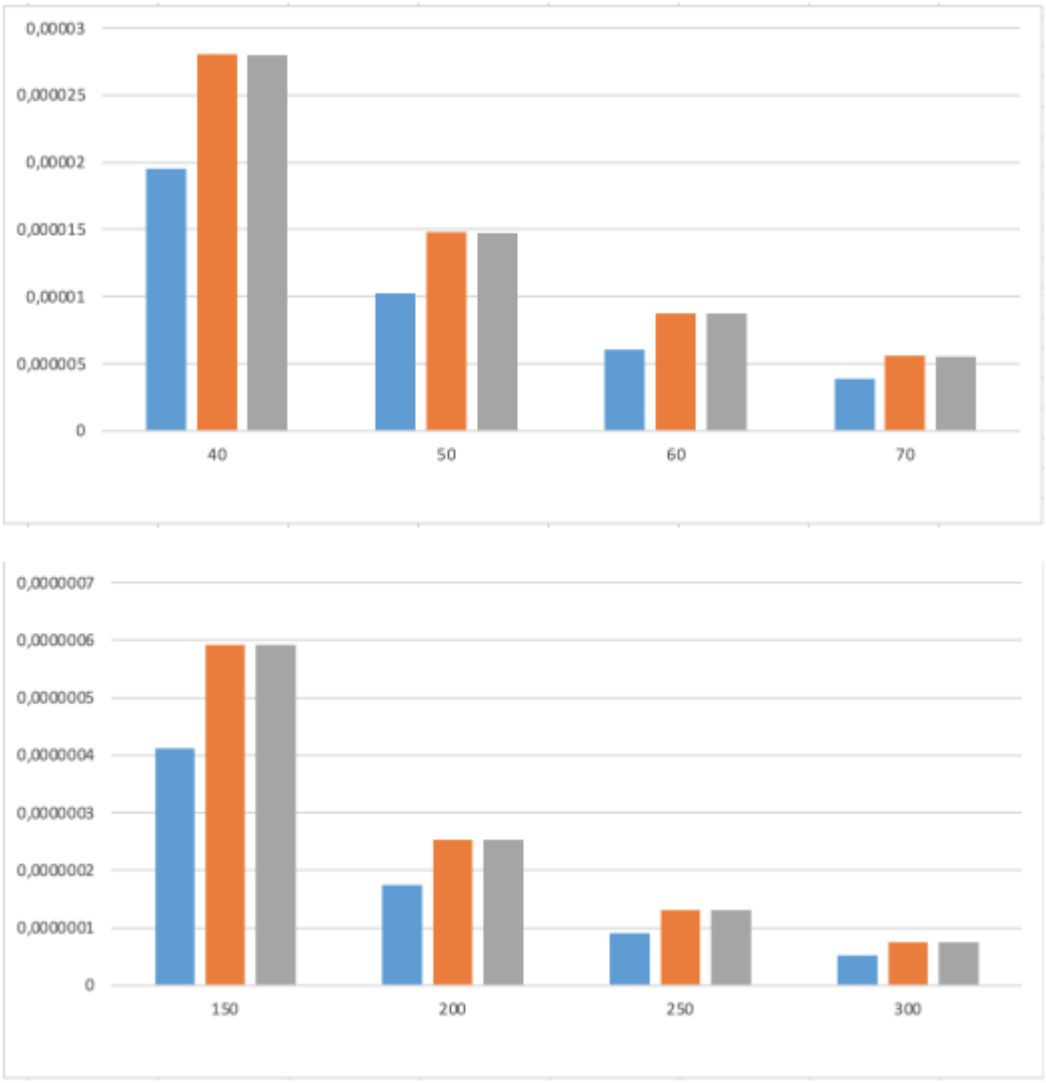


Figura 3.1: En la parte supeior, eficiencia computacional de las familias (3.1.3), (3.1.4) y (3.1.5) para diferentes problemas de tamaño pequeño, $m = 40$, 50, 60 y 70, (eje de abscisas) utilizando una escala logarítmica.En la parte inferior, eficiencia computacional de las familias (3.1.3), (3.1.4) y (3.1.5) para diferentes problemas de tamaño grande, $m = 150$, 200, 250 y 300, (eje de abscisas) utilizando una escala logarítmica.

3.4. Estudio de la convergencia local

En esta sección, nos centramos en el análisis de la convergencia local de la familia (3.1.5). Para el estudio de la convergencia local de métodos iterativos sin derivadas, es común encontrar una pequeña contradicción:

- la existencia del operador $[F'(w^*)]^{-1}$ suele ser requerida en los resultados de convergencia local más conocidos, por lo que se fuerza a que el operador F sea diferenciable en el sentido de Fréchet (ver [19, 36, 91, 98, 139, 148]).

Como consecuencia, la convergencia local de los métodos iterativos para operadores diferenciables en el sentido de Fréchet se analiza considerando dicha contradicción. Sin embargo, en [86], modificando las condiciones en la solución w^* y utilizando un punto auxiliar, se obtiene un resultado de convergencia local para (3.1.3), donde el operador F es no diferenciable y sin requerir la exitencia de la inversa de la derivada, ni la propia derivada en sí. Entonces, para obtener el resultado de convergencia local para (3.1.5), se va a seguir esta idea.

Por otro lado, el estudio local de la convergencia se basa en exigir condiciones en la solución w^*, a partir de ciertas condiciones en el operador F, y proporcionar la llamada bola de convergencia del método iterativo, que muestra la accesibilidad a la solución w^* desde las aproximaciones iniciales que pertenecen a dicha bola.

Así, para obtener resultados de convergencia local para la familia (3.1.5), son necesarias condiciones para la diferencia dividida de primer orden del operador F.

Para hacer el capítulo autocontenido, recordemos que, como aparece en [22], si Ω es un dominio convexo abierto de $\mathbb{R}^m$, existe una diferencia dividida de primer orden de F.

Además, si existe una constante no negativa L tal que

$$\|[x, y; F] - [u, v; F]\| \leq L\ (\|x - u\| + \|y - v\|) \tag{3.4.1}$$

para todo $x, y, u, v \in \Omega$ con $x \neq y$ y $u \neq v$, decimos que F tiene una diferencia dividida de primer orden Lipschitz continua en el dominio Ω.

Se puede generalizar fácilmente este último concepto para la continuidad de tipo Hölder de la siguiente manera: si existe una constante no negativa K tal que se verifica que

$$\|[x, y; F] - [u, v; F]\| \leq K\ (\|x - u\|^p + \|y - v\|^p), \quad p \in [0, 1], \tag{3.4.2}$$

para todo $x, y, u, v \in \Omega$ con $x \neq y$ y $u \neq v$, decimos que F tiene una diferencia dividida de primer orden Hölder continua en el dominio Ω.

Además, si $p = 1$ en (3.4.2), F tiene una diferencia dividida Lipschitz continua en el dominio Ω.

En este capítulo, se van a relajar las condiciones (3.4.1) y (3.4.2) asumiendo que la diferencia dividida $[x, y; F]$ satisface la condición:

$$\|[x, y; F] - [u, v; F]\| \leq \Phi(\|x - u\|, \|y - v\|); \quad x, y, u, v \in \Omega, \tag{3.4.3}$$

donde

$$\Phi : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$$

es una función continua no decreciente en cada uno de sus dos argumentos.

Es fácil ver que la condición (3.4.3) generaliza la condición (3.4.2) considerando

$$\Phi(z_1, z_2) = K(z_1^p + z_2^p).$$

Por otro lado, se sabe que F es un operador diferenciable si $\Phi(0,0) = 0$ ([85]). Pero, si $\Phi(0,0) \neq 0$, el operador F sería no diferenciable. Por lo tanto, bajo la condición (3.4.3), podemos considerar la convergencia local de la familia (3.1.5) en ambas situaciones, cuando F es diferenciable y cuando no lo es.

A partir de ahora, por comodidad, reescribiremos la familia (3.1.5) como:

$$\begin{cases} w_{-1}, w_0 \text{ dados en } \Omega, \quad \lambda \in [0,1], \\ u_n = \lambda w_n + (1-\lambda) w_{n-1}, \quad n \geq 0, \\ a_n = w_n - \texttt{tol}(w_n - u_n), \\ b_n = w_n + \texttt{tol}(w_n - u_n), \\ w_{n+1} = w_n - [a_n, b_n; F]^{-1} F(w_n). \end{cases} \tag{3.4.4}$$

Comenzamos con el caso donde F es un operador no diferenciable, por lo que tomamos $\lambda \in [0,1)$.

Lema 3.1. *Supongamos que existe $w^* \in \Omega$ con*

$$F(w^*) = 0,$$

$$\delta,$$

$$\beta > 0$$

y $\widetilde{w} \in \Omega$, con

$$\|\widetilde{w} - w^*\| = \delta,$$

tal que existe

$$[w^*, \widetilde{w}; F]^{-1} \in \mathcal{L}(\mathbb{R}^m, \mathbb{R}^m)$$

con

$$\|[w^*, \widetilde{w}; F]^{-1}\| \leq \beta.$$

Además, si se cumple la condición (3.4.3) y existe $R > 0$ tal que $B(w^*, R) \subseteq \Omega$ y

$$\beta \Phi_0(R, \delta + R) < 1, \tag{3.4.5}$$

entonces se verifica que:

(I) Si $a, b \in B(w^*, R)$, con $a \neq b$, entonces existe

$$[a, b; F]^{-1} \in \mathcal{L}(\mathbb{R}^m, \mathbb{R}^m)$$

y

$$\|[a, b; F]^{-1}\| \leq \frac{\beta}{1 - \beta \Phi_0(\|a - w^*\|, \|b - \widetilde{w}\|)} \leq \frac{\beta}{1 - \beta \Phi_0(R, \delta + R)}, \tag{3.4.6}$$

donde $\Phi_0 : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$ es una función continua no decreciente en sus dos argumentos que verifica que:

$$\|[a, b; F] - [w^*, \widetilde{w}; F]\| \leq \Phi_0(\|a - w^*\|, \|b - \widetilde{w}\|) \tag{3.4.7}$$

(II) Si

$$w_{n-1}, w_{n-2} \in B(w^*, R),$$

con $w_{n-1} \neq w_{n-2}$*, y* $a_{n-1}, b_{n-1} \in \Omega$*, entonces* w_n *está bien definido y se verifica que*

$$\|w_n - w^*\| < \Delta(R)\|w_{n-1} - w^*\|,$$

donde

$$\Delta(R) = \frac{\beta\Phi(2\texttt{tol}(1-\lambda)R, (1+2\texttt{tol}(1-\lambda))R)}{1-\beta\Phi_0(R, \delta+R)},$$

para $n \geq 1$*.*

Demostración. Primero, observamos que la condición (3.4.7) no es adicional a (3.4.3), ya que una diferencia dividida que satisface la condición (3.4.3), también satisface (3.4.7) con $w^*, \widetilde{x} \in \Omega$ para cada par de puntos distintos $u, v \in \Omega$.

Segundo, de (3.4.7), obtenemos

$$\begin{aligned}\|I - [w^*, \widetilde{w}; F]^{-1}[a, b; F]\| &\leq \|[w^*, \widetilde{w}; F]^{-1}\|\|[w^*, \widetilde{w}; F] - [a, b; F]\| \\ &\leq \beta\Phi_0(\|w^* - a\|, \|\widetilde{w} - w^*\| + \|w^* - b\|) \\ &< \beta\Phi_0(R, \delta + R).\end{aligned}$$

Como

$$\beta\Phi_0(R, \delta + R) < 1,$$

entonces, por el lema de Banach sobre operadores invertibles, existe el operador $[a, b; F]^{-1}$ y además satisface (3.4.6).

Para probar (II), observamos que existe

$$[a_{n-1}, b_{n-1}; F],$$

ya que

$$a_{n-1} - b_{n-1} = -2\texttt{tol}(w_{n-2} - w_{n-1}) \neq 0$$

y

$$a_{n-1} \neq b_{n-1}.$$

Ahora, del punto (I), se sigue que

$$[a_{n-1}, b_{n-1}; F]$$

es invertible y

$$\| [a_{n-1}, b_{n-1}; F]^{-1} \| \leq \frac{\beta}{1-\beta\Phi_0(R, \delta+R)},$$

por lo que w_n está bien definido.

Por otro lado, de (3.4.4), se sigue que

$$\begin{aligned} w_n - w^* &= w_{n-1} - [a_{n-1}, b_{n-1}; F]^{-1} F(w_{n-1}) - w^* \\ &= [a_{n-1}, b_{n-1}; F]^{-1} \left([a_{n-1}, b_{n-1}; F] (w_{n-1} - w^*) - F(w_{n-1}) + F(w^*) \right) \\ &= [a_{n-1}, b_{n-1}; F]^{-1} \left([a_{n-1}, b_{n-1}; F] - [w_{n-1}, w^*; F] \right)(w_{n-1} - w^*). \end{aligned}$$

Ahora, aplicando las condiciones (3.4.3) y (3.4.7), obtenemos

$$\begin{aligned} \|w_n - w^*\| &\leq \| [a_{n-1}, b_{n-1}; F]^{-1} \| \Phi(\|a_{n-1} - w_{n-1}\|, \|b_{n-1} - w^*\|) \|w_{n-1} - w^*\| \\ &\leq \frac{\beta \|w_{n-1} - w^*\|}{1 - \beta \Phi_0(R, \delta + R)} \\ &\quad \times \Phi \left(\mathtt{tol}(1-\lambda) \|w_{n-1} - w_{n-2}\|, \right. \\ &\quad \left. (1 + \mathtt{tol}(1-\lambda)) \|w_{n-1} - w^*\| + \mathtt{tol}(1-\lambda) \|w_{n-2} - w^*\| \right) \\ &< \frac{\beta \Phi(2\mathtt{tol}(1-\lambda)R, (1 + 2\mathtt{tol}(1-\lambda))R)}{1 - \beta \Phi_0(R, \delta + R)} \|w_{n-1} - w^*\| \\ &= \Delta(R) \|w_{n-1} - w^*\|. \end{aligned}$$

Entonces, la demostración del punto (II) está completa. □

A partir del lema anterior, estudiamos en el siguiente resultado cómo podemos garantizar que $a_n, b_n \in B(w^*, R)$.

Lema 3.2. *Supongamos* (3.4.5),

$$\mathtt{tol} < \frac{1}{1-\lambda}$$

y que la ecuación

$$\alpha\beta(1+\mathtt{tol}(1-\lambda))\Phi(2\mathtt{tol}(1-\lambda)t, (1+2\mathtt{tol}(1-\lambda))t) + (\alpha - t)(1 - \beta\Phi_0(t, \delta + t)) = 0, \tag{3.4.8}$$

donde $\alpha = \|w_0 - w^*\|$, tiene al menos una raíz positiva y denotamos por R la raíz positiva más pequeña. Si $w_k \in B(w^*, R) \subseteq \Omega$, con $k = -1, 0, \ldots, n-1$, y

$$\beta(\Phi(2\mathtt{tol}(1-\lambda)R, (1 + 2\mathtt{tol}(1-\lambda))R) + \Phi_0(R, \delta + R)) < 1, \tag{3.4.9}$$

entonces $a_n, b_n \in B(w^*, R)$.

Demostración. De (3.4.4) y el punto (II) del Lema 3.1, tenemos que

$$\|a_n - w^*\| \leq (1 - \mathtt{tol}(1-\lambda))\|w_n - w^*\| + \mathtt{tol}(1-\lambda)\|w_{n-1} - w^*\| < R,$$

por lo que $a_n \in B(w^*, R)$.

Además, tenemos

$$\begin{aligned}\|b_n - w^*\| &\leq (1 + \texttt{tol}(1-\lambda))\|w_n - w^*\| + \texttt{tol}(1-\lambda)\|w_{n-1} - w^*\| \\ &\leq ((1 + \texttt{tol}(1-\lambda))\Delta(R) + \texttt{tol}(1-\lambda))\|w_{n-1} - w^*\|.\end{aligned}$$

Entonces, del Lema 3.1, se sigue que

$$\|w_{n-1} - w^*\| < \Delta(R)^{n-1}\|w_0 - w^*\| < \|w_0 - w^*\| = \alpha,$$

ya que $\Delta(R) < 1$ como (3.4.9). Entonces, como $\texttt{tol}(1-\lambda) < 1$, tomamos en cuenta (3.4.8) para obtener

$$\|b_n - w^*\| < ((1 + \texttt{tol}(1-\lambda))\Delta(R) + 1)\alpha = R,$$

por lo que $b_n \in B(w^*, R)$. □

Por otro lado, es claro que la condición (3.4.9) del Lema 3.2 implica la condición (3.4.5) del Lema (3.1). Por otro lado, es obvio que $\alpha < R$.

Ahora, consideramos $w_{-1}, w_0 \in B(w^*, R) \subseteq \Omega$, con $w_{-1} \neq w_0$,

$$R > (1 + \texttt{tol}(1-\lambda))\alpha$$

y

$$\|w_{-1} - w^*\| < \frac{1}{\texttt{tol}(1-\lambda)}(R - (1 + \texttt{tol}(1-\lambda))\alpha),$$

donde $\alpha = \|w_0 - w^*\|$. Además,

$$\begin{aligned}\|a_0 - w^*\| &\leq (1 - \texttt{tol}(1-\lambda))\|w_0 - w^*\| + \texttt{tol}(1-\lambda)\|w_{-1} - w^*\| \\ &< (1 - \texttt{tol}(1-\lambda))R + \texttt{tol}(1-\lambda)R \\ &= R\end{aligned}$$

y

$$\|b_0 - w^*\| \leq (1 + \texttt{tol}(1-\lambda))\|w_0 - w^*\| + \texttt{tol}(1-\lambda)\|w_{-1} - w^*\| < R,$$

por lo que $a_0, b_0 \in B(w^*, R) \subseteq \Omega$. Además, como $\lambda \in [0, 1)$, se sigue que

$$a_0 - b_0 = -2\,\texttt{tol}(1-\lambda)(w_0 - w_{-1}) \neq 0.$$

Por lo tanto, a_0 y b_0 son un par de puntos distintos en $B(w_0, R) \subseteq \Omega$ y, del punto (I) del Lema 3.1, existe $[a_0, b_0; F]^{-1}$. Entonces, w_1 está bien definido y, por el Lema 3.1, tenemos que

$$\|w_1 - w^*\| < \Delta(R)\|w_0 - w^*\| < \|w_0 - w^*\| = \alpha < R,$$

por lo que $w_1 \in B(w_0, R) \subseteq \Omega$. Ahora, si $w_1 = w_0$, entonces $w_n = w_0$, para todos $n0$, y $\{w_n\}$ converge a la solución $w^* = w_0$ de $F(x) = 0$. En otro caso, $w_1 \neq w_0$ y, razonando como antes, se sigue que w_2 está bien definido y $w_2 \in B(w_0, R) \subseteq \Omega$.

A continuación, reiterando el razonamiento anterior, es fácil probar, por inducción matemática, el siguiente resultado.

Lema 3.3. *Considerando un método iterativo de la familia* (3.1.5) *para un* λ_0 *fijo en* $[0,1)$ *y* $\texttt{tol} < \frac{1}{1-\lambda_0}$. *Supongamos que se cumplen las siguientes condiciones:*

(C1) *Existen* $w^* \in \Omega$ *con* $F(w^*) = 0$, $\delta, \beta > 0$ *y* $\widetilde{w} \in \Omega$, *con* $\|\widetilde{w} - w^*\| = \delta$, *tal que existe* $[w^*, \widetilde{w}; F]^{-1} \in \mathcal{L}(\mathbb{R}^m, \mathbb{R}^m)$ *con* $\|[w^*, \widetilde{w}; F]^{-1}\| \leq \beta$.

(C2) *Existe una función continua no decreciente en sus dos argumentos* $\Phi : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$ *que satisface la condición* (3.4.3) *con* $x \neq y$ *y* $u \neq v$.

(C3) *Teniendo en cuenta que la condición* (3.4.7) *se satisface con* $a \neq b$, *la ecuación* (3.4.8) *tiene al menos una raíz positiva y denotamos por* R *la raíz positiva más pequeña.*

(C4) $B(w^*, R) \subseteq \Omega$ *y* (3.4.9).

Si $w_{-1}, w_0 \in B(w^*, R) \subseteq \Omega$, *con* $w_{-1} \neq w_0$, $R > (1 + \texttt{tol}(1-\lambda_0))\alpha$ *y*

$$\|w_{-1} - w^*\| < \frac{R - (1 + \texttt{tol}(1-\lambda_0))\alpha}{\texttt{tol}(1-\lambda_0)}, \tag{3.4.10}$$

entonces se sigue para $n \geq 1$*:*

(i) $w_n \in B(w^*, R)$,

(ii) $a_n, b_n \in B(w^*, R) \subseteq \Omega$ *con* $a_n \neq b_n$,

(iii) $\|w_n - w^*\| < \Delta(R)\|w_{n-1} - w^*\| < \Delta(R)^n \|w_1 - w_0\| < \|w_1 - w_0\|$.

A continuación, para cada método iterativo de la familia (3.1.5) con λ_0 en $[0,1)$ y

$$\texttt{tol} < \frac{1}{1-\lambda_0},$$

podemos establecer el siguiente resultado de convergencia local.

Teorema 3.2. *Fijado* $\lambda_0 \in [0,1)$ *y*

$$\texttt{tol} < \frac{1}{1-\lambda_0},$$

suponemos que se cumplen las condiciones (C1)–(C4). Si $w_{-1}, w_0 \in B(w^*, R)$, *con* $w_{-1} \neq w_0$,

$$R > (1 + \texttt{tol}(1-\lambda_0))\alpha$$

y (3.4.10) *se cumple, entonces la sucesión* (3.1.5) *con* $\lambda = \lambda_0$ *está bien definida, permanece en* $B(w^*, R)$ *y converge a* w^*.

Demostración. Del punto (ii) del Lema 3.3, se sigue que la sucesión (3.1.5) está bien definida. Del punto (i) del Lema 3.3, es claro que la sucesión (3.1.5) permanece en $B(w^*, R)$. Finalmente, como $\{\|w_n - w^*\|\}$ es una sucesión estrictamente decreciente de números reales positivos, se sigue del punto (ii) del Lema 3.3 que $\{w_n\}$ converge a w^*. □

A continuación, se presenta el siguiente resultado sobre la unicidad de la solución.

Teorema 3.3. *Bajo las condiciones (C1)–(C4), supóngase que existe* $\widetilde{R} \geq R$ *tal que* $\beta\Phi_0(0, \delta+\widetilde{R}) \leq 1$. *En ese caso,* w^* *es la única solución de la ecuación* $F(x) = 0$ *en* $B(w^*, \widetilde{R}) \cap \Omega$.

Demostración. Sea $\varepsilon^* \in B(w^*, \widetilde{R}) \cap \Omega$ tal que $F(\varepsilon^*) = 0$. Entonces, de (3.4.7), se obtiene que

$$\begin{aligned} \|I - [w^*, \widetilde{w}; F]^{-1}[w^*, \varepsilon^*; F]\| &\leq \|[w^*, \widetilde{w}; F]^{-1}\|\|[w^*, \widetilde{w}; F] - [w^*, \varepsilon^*; F]\| \\ &< \beta\Phi_0(0, \delta + \widetilde{R}) \\ &\leq 1. \end{aligned}$$

Por lo tanto, existe $[w^*, \varepsilon^*; F]^{-1}$. Ahora, de $0 = F(w^*) - F(\varepsilon^*) = [w^*, \varepsilon^*; F](w^* - \varepsilon^*)$, se deduce que $w^* = \varepsilon^*$. □

Si el operador F es diferenciable, es posible aplicar el Teorema 3.2 para $\lambda \in [0, 1)$. Para $\lambda = 1$, el método iterativo (3.1.5) se reduce al método de Newton, obteniéndose así un resultado de convergencia local novedoso, utilizando puntos auxiliares.

Teorema 3.4. *Supóngase que se cumplen las siguientes condiciones:*

(N1) *Existen* $w^* \in \Omega$ *con* $F(w^*) = 0$, $\delta, \beta > 0$ *y* $\widetilde{w} \in \Omega$, *con* $\|\widetilde{w} - w^*\| = \delta$, *tal que existe* $[w^*, \widetilde{w}; F]^{-1} \in \mathcal{L}(\mathbb{R}^m, \mathbb{R}^m)$ *con*

$$[w^*, \widetilde{w}; F]^{-1}\| \leq \beta.$$

(N2) *Existe una función continua creciente en sus dos argumentos* $\Phi : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$ *que satisface la condición* (3.4.3).

(N3) *La ecuación*

$$\beta\left(\Phi(0, t) + \Phi(t, \delta + t)\right) - 1 = 0$$

tiene al menos una raíz real positiva, y se denota por R *la mayor de ellas. Considérese* $r \in \mathbb{R}_+$ *tal que* $r < R$ *con* $B(w^*, r) \subset \Omega$.

Si $w_0 \in B(w^*, r)$, *entonces la sucesión dada por el método de Newton está bien definida,* $w_n \in B(w^*, r)$ *para todos* $n \geq 0$, *y converge a la solución* w^* *de la ecuación* $F(x) = 0$.

Demostración. La demostración es análoga a la del Teorema 3.2. Dado que F es diferenciable, se debe tener en cuenta que

$$[x, x; F] = F'(x).$$

Además,

$$\beta\left(\Phi(0, r) + \Phi(r, \delta + r)\right) < \beta\left(\Phi(0, R) + \Phi(R, \delta + R)\right) = 1,$$

ya que la función Φ es creciente en ambos argumentos. □

3.5. Ejemplos

Para probar la convergencia teórica de la sección anterior, en concreto el Teorema 3.2 para la familia (3.1.5), se va a aplicar varios ejemplos:

- En el primer ejemplo, se utilizará el método de la familia tomando $\lambda = 0$, cuyo algoritmo es

$$\begin{cases} w_{-1}, w_0 \text{ dados en } \Omega, \\ w_{n+1} = w_n - [(1-\texttt{tol})w_n + \texttt{tol}\, w_{n-1}, (1+\texttt{tol})w_n - \texttt{tol}\, w_{n-1}; F]^{-1} F(w_n), \ n \geq 0. \end{cases} \tag{3.5.1}$$

  Además, es claro que (3.5.1) se reduce al método de Kurchatov si $\lambda_0 = 0$ y $\texttt{tol} = 1$. Se utilizará una ecuación integral no lineal del tipo Hammerstein mixto, para ilustrar el ejemplo.

- En el segundo ejemplo, se observará la influencia del parámetro `tol` en (3.5.1) cuando se utiliza para aproximar soluciones. Para esto, se utilizará una ecuación integral no lineal del tipo Hammerstein mixto similar a la del ejemplo anterior.

- Por último, en el tercer ejemplo se analiza la influencia del parámetro λ en las aproximaciones de soluciones. Para ello, se utiliza el método (3.1.5) con $\texttt{tol} = 1$ y la ecuación integral (3.5.4) introducida en el ejemplo anterior.

Ejemplo 3.1. *Consideremos la ecuación integral no lineal*

$$w(s) = \frac{1}{3}\int_0^1 \mathcal{K}(s,t)\left(w(t)^2 + |w(t)|\right) dt, \quad s \in [0,1], \tag{3.5.2}$$

donde el núcleo $\mathcal{K}(s,t)$ es la función de Green en $[0,1]\times[0,1]$.

Primero, utilizamos un proceso de discretización para transformar (3.5.2) *en un problema de dimensión finita mediante una fórmula de cuadratura de Gauss-Legendre con* 8 *nodos*

$$\int_0^1 \ell(t)\, dt = \sum_{i=1}^{8} p_i\, \ell(t_i),$$

donde los nodos t_i y los pesos p_i se determinan para $i = 1, 2, \ldots, 8$. Si denotamos las aproximaciones de $w(t_i)$ por w_i, con $i = 1, 2, \ldots, 8$, entonces la ecuación (4.5.3) *es equivalente al siguiente sistema de ecuaciones no lineales*

$$w_i = \frac{1}{3}\sum_{j=1}^{8} a_{ij}\left(w_i^2 + |w_i|\right), \quad j = 1, 2, \ldots, 8, \tag{3.5.3}$$

donde

$$a_{ij} = p_j\, \mathcal{K}(t_i, t_j) = \begin{cases} p_j\,(1-t_i)\,t_j, & si \quad j \leq i, \\ p_j\,(1-t_j)\,t_i, & si \quad j > i. \end{cases}$$

Ahora, el sistema (3.5.3) *se puede escribir como*

$$F(\mathbf{w}) \equiv \mathbf{w} - \frac{1}{3} A\, \bar{\mathbf{w}} = 0, \qquad F : \mathbb{R}^8 \to \mathbb{R}^8, \qquad F = (F_1, F_2, \ldots, F_8),$$

donde

$$\mathbf{w} = (w_1, w_2, \ldots, w_8)^T,$$
$$A = (a_{ij})_{i,j=1}^{8}$$

y

$$\bar{\mathbf{w}} = \left(w_1^2 + |w_1|, w_2^2 + |w_2|, \ldots, w_8^2 + |w_8|\right)^T,$$

por lo que F es no lineal y no diferenciable.

Es obvio que $\mathbf{w}^* = 0$ es una solución de $F(\mathbf{w}) = 0$. Entonces, del Teorema 3.2, podemos obtener diferentes bolas de convergencia para el método iterativo (3.5.1) *dependiendo de los valores de*

$$\texttt{tol} < \frac{1}{1 - \lambda_0} = 1.$$

A continuación, utilizamos la diferencia dividida de primer orden

$$[\mathbf{c}, \mathbf{d}; F] = \left([\mathbf{c}, \mathbf{d}; F]_{ij}\right)_{i,j=1}^{8},$$

donde

$$[\mathbf{c}, \mathbf{d}; F]_{ij} = \frac{1}{c_j - d_j}\left(F_i(c_1, \ldots, c_{j-1}, c_j, d_{j+1}, \ldots, d_8) - F_i(c_1, \ldots, c_{j-1}, d_j, d_{j+1}, \ldots, d_8)\right),$$

$i, j = 1, 2, \ldots, 8$,

$$\mathbf{c} = (c_1, c_2, \ldots, c_8)^T$$

y

$$\mathbf{d} = (d_1, d_2, \ldots, d_8)^T.$$

Así,

$$[\mathbf{c}, \mathbf{d}; F] = I - \frac{1}{3}A\left(diag\left\{c_i + d_i + \frac{|c_i| - |d_i|}{c_i - d_i}\right\}\right)_{i=1}^{8}.$$

y

$$\|[\mathbf{x}, \mathbf{y}; F] - [\mathbf{u}, \mathbf{v}; F]\| \leq \frac{1}{3}\|A\|(\|\mathbf{x} - \mathbf{u}\| + \|\mathbf{y} - \mathbf{v}\| + 2),$$

por lo que

$$\Phi(z_1, z_2) = \frac{1}{3}\|A\|(z_1 + z_2 + 2).$$

Si elegimos

$$\mathbf{w}_0 = \widetilde{\mathbf{w}} = (1, 1, \ldots, 1)^T,$$

entonces

$$\delta = \alpha = 1,$$

ya que

$$\mathbf{w}^* = 0.$$

Además,

$$\beta = 1.0886\ldots,$$
$$\Phi_0(z_1, z_2) = \Phi(z_1, z_2)$$

y la ecuación (3.4.8) *se reduce a*

$$(0.1793\ldots)z\,\texttt{tol}^2 + ((0.2241\ldots)z + (0.0896\ldots))\,\texttt{tol} \\ +(0.0896\ldots)z^2 - (0.9103\ldots)z + (0.9551\ldots) = 0.$$

Como $\texttt{tol} < 1$*, mostramos en la Tabla 3.1 las raíces positivas más pequeñas* R *de la última ecuación dependiendo de* `tol`*, que satisfacen la condición* (3.4.9)*. Si ahora elegimos*

$$\mathbf{w}_{-1} \in B(\mathbf{w}^*, R)$$

tal que

$$\mathbf{w}_{-1} \neq \mathbf{w}_0,$$

$$R > (1 + \texttt{tol})\alpha$$

y satisfaciendo (3.4.10)*, podemos concluir que la bola de convergencia* $B(\mathbf{w}^*, R)$ *es mejor cuanto mayor sea el valor de* `tol`*. Observar que la ecuación* (3.4.8) *no tiene raíces reales si* $\texttt{tol} \geq 0.9$ *y la condición* (3.4.9) *no se satisface si* $\texttt{tol} \geq 0.8$*.*

tol	R
0.0	$1.1883\ldots$
0.1	$1.2448\ldots$
0.2	$1.3141\ldots$
0.3	$1.4002\ldots$
0.4	$1.5093\ldots$
0.5	$1.6522\ldots$
0.6	$1.8501\ldots$
0.7	$2.1557\ldots$

Tabla 3.1: la bola de convergencia $B(\mathbf{w}^*, R)$ dependiendo de `tol`.

Ejemplo 3.2. *Consideremos la ecuación integral no lineal*

$$w(s) = \frac{3}{4} + \frac{1}{2}\int_0^1 \mathcal{K}(s,t)\left(w(t)^2 + |w(t)|\right)dt, \quad s \in [0,1], \tag{3.5.4}$$

donde el núcleo $\mathcal{K}(s,t)$ *es la función de Green en* $[0,1]\times[0,1]$*.*

Siguiendo un proceso de discretización idéntico al llevado a cabo para la ecuación (3.5.2) *en el Ejemplo 3.1, vemos que la ecuación* (3.5.4) *se transforma en el siguiente sistema no lineal*

$$F(\mathbf{w}) \equiv \mathbf{w} - \mathbf{v} - \frac{1}{2}A\,\bar{\mathbf{w}} = 0, F : \mathbb{R}^8 \to \mathbb{R}^8, F = (F_1, F_2, \ldots, F_8), \tag{3.5.5}$$

donde $\mathbf{w} = (w_1, w_2, \ldots, w_8)^T$*,* $\mathbf{v} = \left(\frac{3}{4}, \frac{3}{4}, \ldots, \frac{3}{4}\right)^T$*,* $A = (a_{ij})_{i,j=1}^8$ *y*

$$\bar{\mathbf{w}} = \left(w_1^2 + |w_1|, w_2^2 + |w_2|, \ldots, w_8^2 + |w_8|\right)^T.$$

Después de eso, utilizamos el método iterativo (3.5.1) *comenzando en*

$$\mathbf{w}_{-1} = \left(-\frac{2}{5}, -\frac{2}{5}, \ldots, -\frac{2}{5}\right)^T \ y \ \mathbf{w}_0 = \left(-\frac{1}{2}, -\frac{1}{2}, \ldots, -\frac{1}{2}\right)^T$$

para aproximar una solución de (3.5.5) *para diferentes valores de* `tol` *y ver, a través de los errores* $\|\mathbf{w}_n - \mathbf{w}^*\|$ *obtenidos en la Tabla 3.2 con el criterio de detención* $\|\mathbf{w}_n - \mathbf{w}_{n-1}\| < 10^{-24}$, *que el método* (3.5.1) *proporciona mejores aproximaciones a la solución cuanto menor sea el valor de* `tol`. *Observe que, aunque no podemos considerar el método de Kurchatov en el Teorema 3.2 porque* `tol` $= 1$, *podemos comparar la velocidad de convergencia del método* (3.5.1) *con respecto al método de Kurchatov. Note que el orden de convergencia computacional ([167]) de* (3.5.1) *aproxima el orden de convergencia dos del método de Kurchatov.*

n	`tol` $= 0.5$	`tol` $= 0.75$	`tol` $= 1$
1	$1.9166\ldots \times 10^{-1}$	$1.9166\ldots \times 10^{-1}$	$1.9166\ldots \times 10^{-1}$
2	$2.0124\ldots \times 10^{-3}$	$4.7342\ldots \times 10^{-3}$	$6.8255\ldots \times 10^{-3}$
3	$2.1654\ldots \times 10^{-7}$	$1.2076\ldots \times 10^{-6}$	$2.5173\ldots \times 10^{-6}$
4	$2.4884\ldots \times 10^{-15}$	$7.7497\ldots \times 10^{-14}$	$3.3692\ldots \times 10^{-13}$

Tabla 3.2: Errores absolutos $\|\mathbf{w}_n - \mathbf{w}^*\|$ obtenidos por el método (3.5.1) para diferentes valores de `tol`.

Ejemplo 3.3. *Se considera de nuevo la ecuación integral no lineal* (3.5.4), *siguiendo el mismo proceso de discretización que se empleó en el Ejemplo 3.2.*

Luego, se aplica el método (3.1.5) *con* `tol` $= 1$, *comenzando en*

$$\mathbf{w}_{-1} = \left(-\frac{2}{5}, -\frac{2}{5}, \ldots, -\frac{2}{5}\right)^T \ y \ \mathbf{w}_0 = \left(-\frac{1}{2}, -\frac{1}{2}, \ldots, -\frac{1}{2}\right)^T,$$

para aproximar una solución de (3.5.5) *para diferentes valores de* λ. *Los errores* $\|\mathbf{w}_n - \mathbf{w}^*\|$ *obtenidos con el criterio de detención* $\|\mathbf{w}_n - \mathbf{w}_{n-1}\| < 10^{-24}$ *se muestran en la Tabla 3.3, donde se observa que el método* (3.1.5) *con* `tol` $= 1$ *proporciona mejores aproximaciones cuanto mayor es el valor de* $\lambda \in [0, 1]$. *Además, el orden de convergencia computacional de* (3.1.5) *con* `tol` $= 1$ *se aproxima al orden de convergencia dos.*

n	$\lambda = 0$	$\lambda = 0.3$	$\lambda = 0.6$
1	$1.9166\ldots \times 10^{-1}$	$1.9166\ldots \times 10^{-1}$	$1.9166\ldots \times 10^{-1}$
2	$6.8255\ldots \times 10^{-3}$	$4.1281\ldots \times 10^{-3}$	$2.0124\ldots \times 10^{-3}$
3	$2.5173\ldots \times 10^{-6}$	$9.1694\ldots \times 10^{-7}$	$2.1654\ldots \times 10^{-7}$
4	$3.3692\ldots \times 10^{-13}$	$4.4668\ldots \times 10^{-14}$	$2.4884\ldots \times 10^{-15}$

Tabla 3.3: Errores absolutos $\|\mathbf{w}_n - \mathbf{w}^*\|$ obtenidos por el método (3.1.5) con `tol` $= 1$ y diferentes valores de λ.

3.6. Accesibilidad de los métodos iterativos de las familias (3.1.3), (3.1.4) y (3.1.5)

En esta sección, se estudia la accesibilidad de los métodos iterativos de las familias (3.1.3), (3.1.4) y (3.1.5), comparando su comportamiento dinámico. Para ello, se adoptan dos enfoques:

- un estudio dinámico complejo utilizando ténicas similares a las que aparecen en [43, 77]
- un estudio dinámico real utilizando ténicas similares a las que aparecen en [105, 107].

En ambos casos, dada una ecuación compleja $F(z) = 0$, se fijan los puntos iniciales w_{-1} y w_0 de los métodos iterativos de las familias, generando una sucesión $\{w_n\}$. Luego, se establece una tolerancia prefijada de forma que

$$\|w_n - w^*\| < 10^{-\rho},$$

donde $\rho \in \mathbb{N}$ y w^* es una solución de $F(z) = 0$. También se establece un número máximo de iteraciones N, tal que

$$\|w_N - w^*\| < 10^{-\rho}.$$

Posteriormente, se considera el par (w_{-1}, w_0), y se asocia un punto $P_{(w_{-1},w_0)}$ en el plano complejo. Si se cumple el criterio de tolerancia en menos de N iteraciones, el punto $P_{(w_{-1},w_0)}$ se colorea de acuerdo con la solución a la que converge el método iterativo, en caso contrario, se colorea de negro. Las áreas no negras en el plano indican la accesibilidad del método.

En el estudio dinámico complejo, se consideran los casos más favorables para la convergencia de los métodos, donde w_{-1} y w_0 están próximos, tomando

$$w_{-1} = w_0 - \frac{1}{10}.$$

En el segundo enfoque real, se considera que las raíces de la ecuación $F(z) = 0$ son reales y se toman todos los valores de w_{-1} y w_0 en la recta real, tomando para ello un eje asociado a los valores de w_{-1} y el otro eje asociado a w_0, usnado el plano de convergencia que aparece en [105, 107]. La ecuación elegida, es la misma que en el capítulo anterior

$$F(z) = z^3 + z|z| - 2z = 0.$$

Se va a comparar la accesibilidad de los métodos iterativos para dicha ecuación que tiene por raíces $z^* = -1$, $z^{**} = 0$, y $z^{***} = 1$.

Se asignaron los siguientes colores a las convergencias asociadas a las raíces: verde para la convergencia a la raíz z^*, azul para la convergencia a la raíz z^{**}, y rojo para la convergencia a la raíz z^{***}. Además, se establecieron $\rho = 5$ y $N = 10$, y los gráficos fueron generados con Python utilizando algoritmos basados en los trabajos de [105, 106].

3.6.1. Un estudio comparativo dinámico de las familias (3.1.3) y (3.1.4)

Dado que la familia (3.1.5) se reduce a la familia (3.1.4) cuando `tol` $= 1$, primero se compararon las familias (3.1.3) y (3.1.4).

Dinámica compleja

Primero se presenta el comportamiento dinámico complejo de las familias (3.1.3) y (3.1.4). En las figuras 3.2-3.4, pueden verse las cuencas asociadas a la familia (3.1.3), para diferentes valores de λ y tomando $N = 10$.

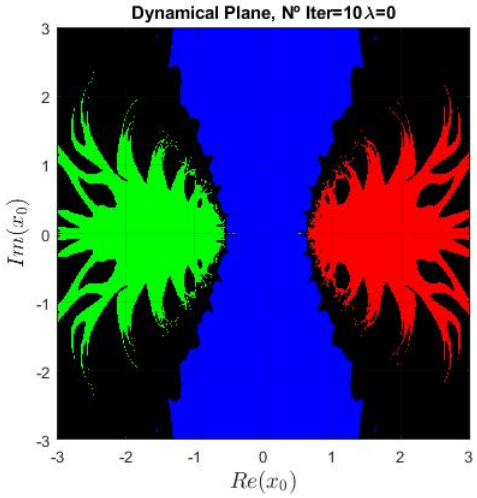


Figura 3.2: Cuencas de atracción asociadas a la iteración de la familia (3.1.3) aplicada al polinomio F para $\lambda = 0$ y $N = 10$.

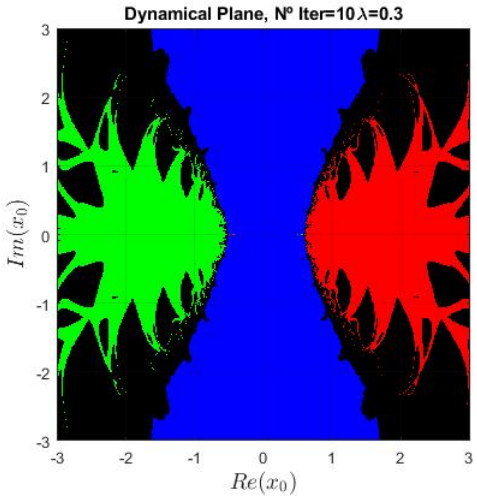


Figura 3.3: Cuencas de atracción asociadas a la iteración de la familia (3.1.3) aplicada al polinomio F para $\lambda = 0.3$ y $N = 10$.

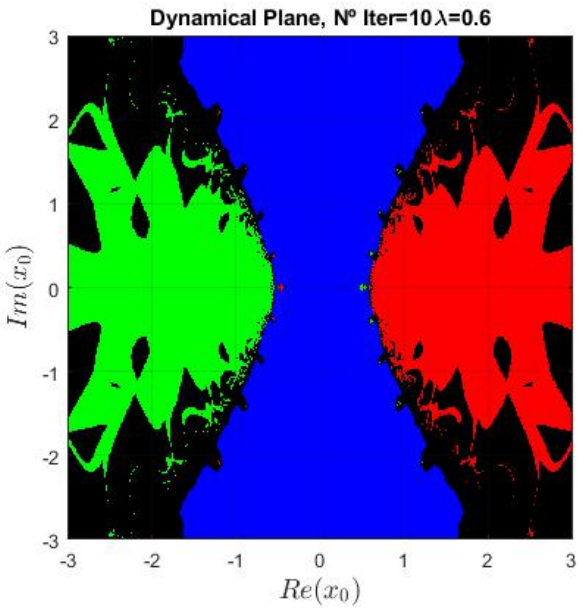


Figura 3.4: Cuencas de atracción asociadas a la iteración de la familia (3.1.3) aplicada al polinomio F para $\lambda = 0.6$ y $N = 10$.

Por otro lado, en las figuras 3.5, 3.6 y 3.7, pueden verse las cuencas de atracción asociadas a la familia (3.1.4), para diferentes valores de $\lambda = 0$, 0.3 y 0.6 y tomando como número de iteraciones $N = 10$.

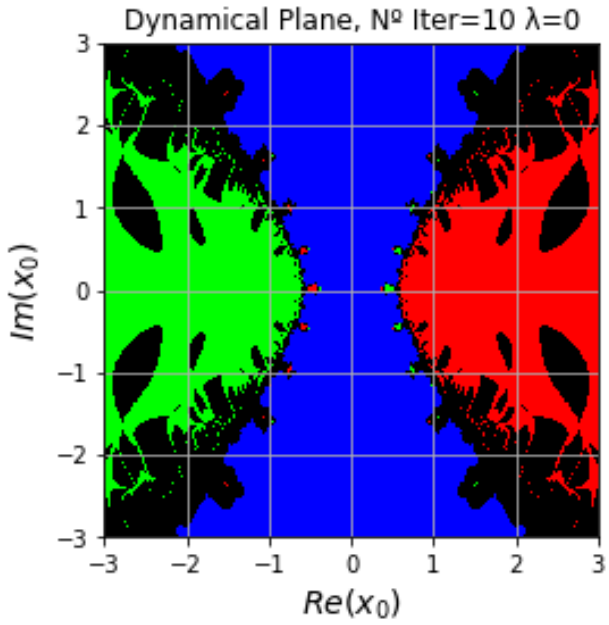


Figura 3.5: Cuencas de atracción asociadas a la iteración de la familia (3.1.4) aplicada al polinomio F para $\lambda = 0$ y $N = 10$.

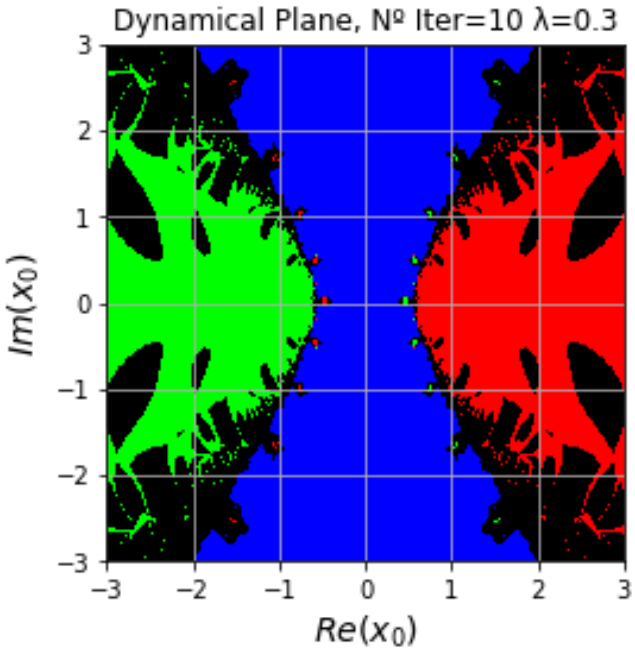


Figura 3.6: Cuencas de atracción asociadas a la iteración de la familia (3.1.4) aplicada al polinomio F para $\lambda = 0.3$ y $N = 10$.

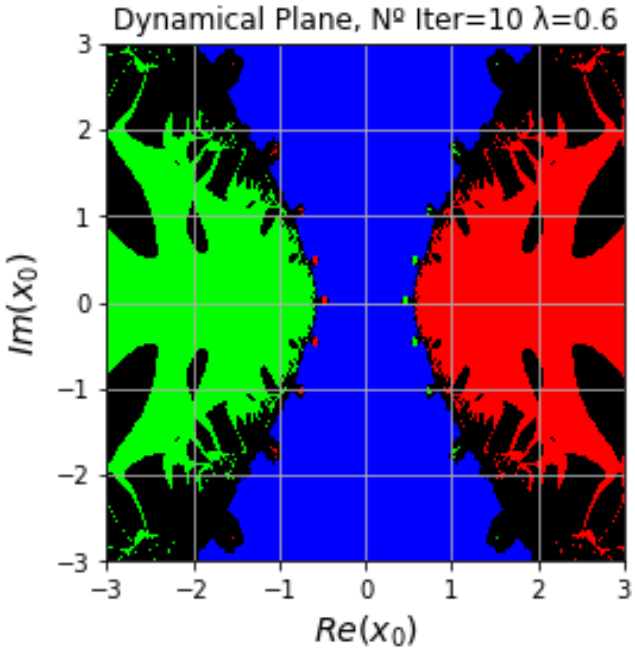
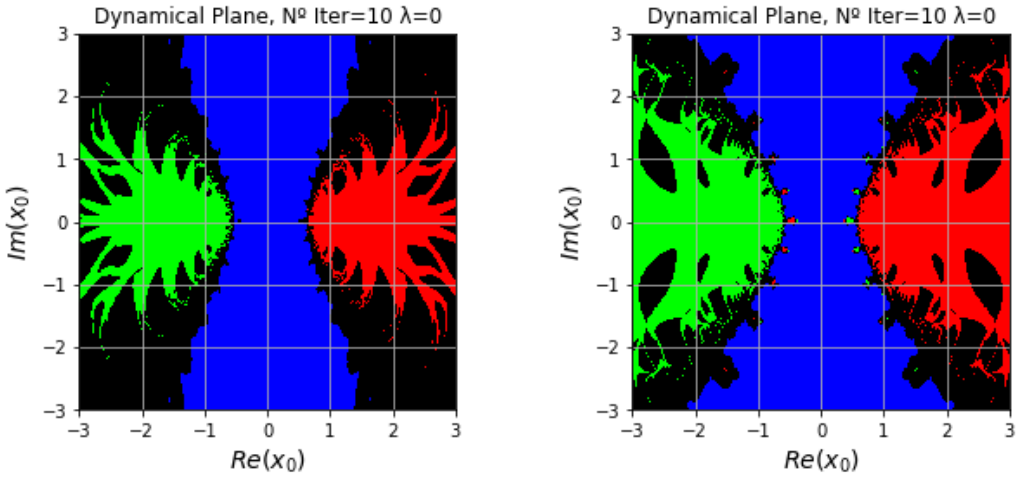


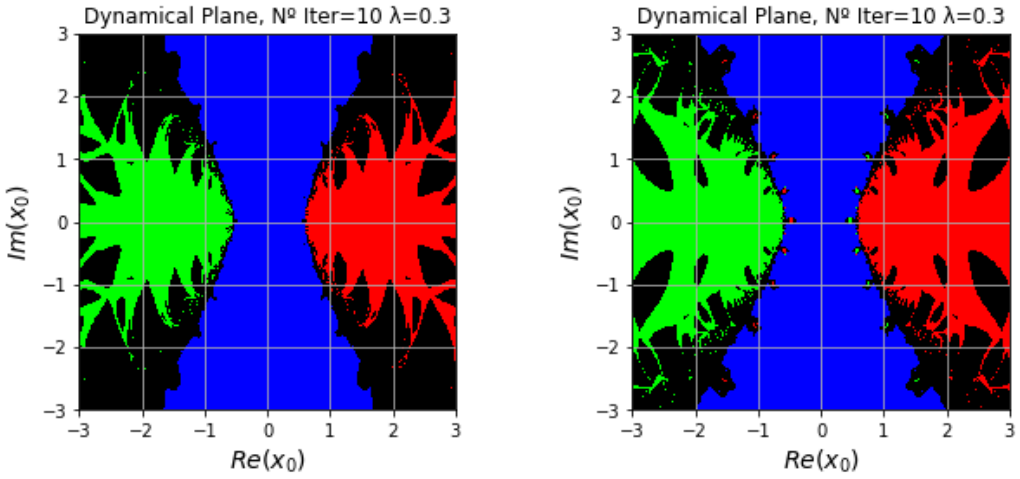
Figura 3.7: Cuencas de atracción asociadas a la iteración de la familia (3.1.4) aplicada al polinomio F para $\lambda = 0.6$ y $N = 10$.

Asimismo, para poder comparar la accesibilidad de ambas familias se van a graficar conjuntamente los elementos de las familias con los mismos valores de λ. Estos gráficos pueden verse en las Figuras 3.8-3.10, donde se observa que la accesibilidad de la familia (3.1.4) es superior a la de (3.1.3).



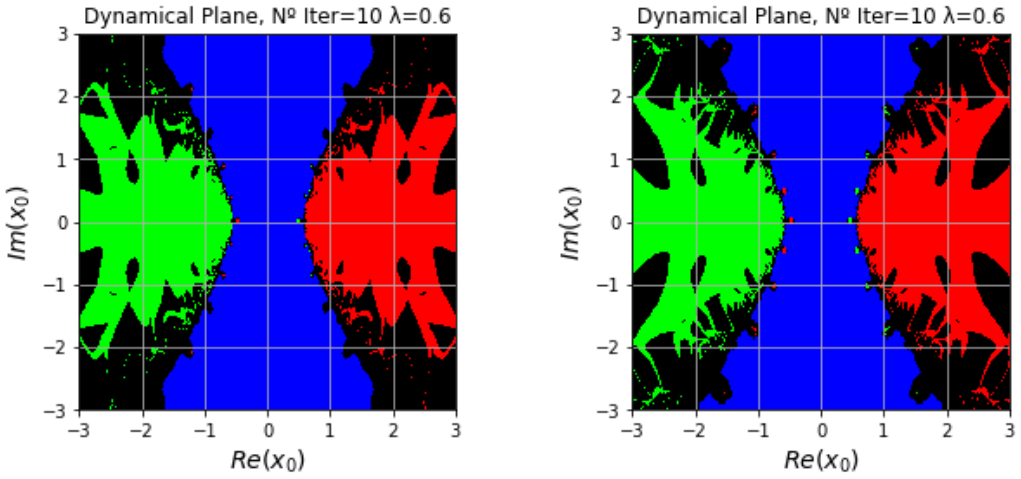
Familia (3.1.3) Familia (3.1.4)

Figura 3.8: Cuencas de atracción asociadas a las diferentes familias para $\lambda = 0$ y $N = 10$.



Familia (3.1.3) Familia (3.1.4)

Figura 3.9: Cuencas de atracción asociadas a las diferentes familias para $\lambda = 0.3$ y $N = 10$.



Familia (3.1.3) Familia (3.1.4)

Figura 3.10: Cuencas de atracción asociadas a las diferentes familias para $\lambda = 0.6$ y $N = 10$.

Además, para poder corroborar de forma numérica este dato se calcularon numéricamente los porcentajes de puntos que convergen a alguna de las raíces, presentados en la Tabla 3.4. De nuevo, en la tabla se observa que la accesibilidad de la familia (3.1.4) es mejor que la de (3.1.3).

λ	Método (3.1.3)	Método (3.1.4)
0	54.98 %	69.42 %
0.3	62.21 %	69.05 %
0.6	68.69 %	69.63 %

Tabla 3.4: Porcentaje de puntos de convergencia en la dinámica compleja para las iteraciones de las familias (3.1.3) y (3.1.4).

A continuación, se adjuntan los códigos de Python utilizados.

3.6.2. Familia (3.1.3) en el caso de dinámica compleja

```
# -*- coding: utf-8 -*-
import numpy as np
import matplotlib.pyplot as plt
import time

# Inicio de tiempo de computacion
start_time = time.time()

# Parametros del metodo
L = 0.6

# Funcion
F = lambda X: X**3 + X * np.abs(X) - 2 * X

# Aproximacion de la derivada, diferencia dividida
U = lambda X, Y: L * X + (1 - L) * Y
DF = lambda X, Y: (F(X) - F(U(X, Y))) / (X - U(X, Y))
m = lambda X, Y: X - F(X) / DF(X, Y)

# Valores iniciales
h=0.01 #Tama o de paso
Rez0 = np.arange(-3, 3+h,h)
Imz0 = np.arange(-3, 3+h, h)
maxiter = 10
tol = 1e-5

puntos = len(Rez0)
ReZ0, ImZ0 = np.meshgrid(Rez0, Imz0)
Z0 = ReZ0 + 1j * ImZ0
Z_1 = Z0 - 0.1
iter = 1
R = np.zeros((puntos, puntos))
G = np.zeros((puntos, puntos))
B = np.zeros((puntos, puntos))
IT = (maxiter + 1) * np.ones_like(Z0)

# Puntos fijos
ZF1 = 1
ZF2 = -1
ZF3 = 0

# Ejecucion del metodo iterativo
while iter <= maxiter:
    Z = m(Z0, Z_1)
    for fil in range(puntos):
        for col in range(puntos):
            Zfc = Z[fil, col]
            if abs(Zfc - ZF1) < tol:
                R[fil, col] = 1
                IT[fil, col] = min(iter, IT[fil, col])
            if abs(Zfc - ZF2) < tol:
                G[fil, col] = 1
                IT[fil, col] = min(iter, IT[fil, col])
            if abs(Zfc - ZF3) < tol:
                B[fil, col] = 1
                IT[fil, col] = min(iter, IT[fil, col])
```

```
    iter += 1
    Z_1 = Z0
    Z0 = Z

# Generacion de la imagen
I = np.zeros((puntos, puntos, 3))
I[:, :, 0] = R
I[:, :, 1] = G
I[:, :, 2] = B

# Representacion de la imagen
plt.imshow(I, extent=[-3, 3, -3, 3])
plt.xlabel("$Re(x_{0})$", fontsize=14)
plt.ylabel("$Im(x_{0})$", fontsize=14)
plt.title(f"Dynamical Plane, N  Iter={maxiter}  ={L}")
plt.grid()
plt.show()

# Calculo de estadisticas
print("% de elementos que convergen:")
print(np.count_nonzero(IT <= maxiter) / np.size(IT) * 100)
print("Valor medio de iteraciones para converger:")
print(np.mean(IT[IT <= maxiter]))

# Tiempo de computacion
end_time = time.time()
print("Tiempo de computo:", end_time - start_time, "segundos")
```

3.6.3. Familia (3.1.4) en el caso de dinámica compleja

```
# -*- coding: utf-8 -*-
import numpy as np
import matplotlib.pyplot as plt
import time

# Inicio de tiempo de computacion
start_time = time.time()

#Parametro del metodo
L=0.6
#Metodo
F= lambda X: X**3+X*abs(X)-2*X
U = lambda X, Y: L*X + (1-L)*Y  # Aqui supongo que L es una constante
     dada
DF = lambda X, Y: (F(2*X - U(X,Y)) - F(U(X,Y))) / (2*X - U(X,Y) - U(X
    ,Y))
M = lambda X, Y: X - F(X) / DF(X, Y)

# Valores iniciales
h=0.01 #Tama o de paso
Rez0 = np.arange(-3, 3+h,h)
Imz0 = np.arange(-3, 3+h, h)
maxiter = 10
tol = 1e-5

puntos = len(Rez0)
ReZ0, ImZ0 = np.meshgrid(Rez0, Imz0)
Z0 = ReZ0 + 1j * ImZ0
Z_1 = Z0 - 0.1
iter = 1
R = np.zeros((puntos, puntos))
G = np.zeros((puntos, puntos))
B = np.zeros((puntos, puntos))
IT = (maxiter + 1) * np.ones_like(Z0)

# Puntos fijos
ZF1 = 1
ZF2 = -1
ZF3 = 0

# Ejecucion del metodo iterativo
while iter <= maxiter:
    Z = M(Z0, Z_1)
    for fil in range(puntos):
        for col in range(puntos):
            Zfc = Z[fil, col]
            if abs(Zfc - ZF1) < tol:
                R[fil, col] = 1
                IT[fil, col] = min(iter, IT[fil, col])
            if abs(Zfc - ZF2) < tol:
                G[fil, col] = 1
                IT[fil, col] = min(iter, IT[fil, col])
            if abs(Zfc - ZF3) < tol:
                B[fil, col] = 1
                IT[fil, col] = min(iter, IT[fil, col])
    iter += 1
```

```
    Z_1 = Z0
    Z0 = Z

# Generacion de la imagen
I = np.zeros((puntos, puntos, 3))
I[:, :, 0] = R
I[:, :, 1] = G
I[:, :, 2] = B

# Representacion de la imagen
plt.imshow(I, extent=[-3, 3, -3, 3])
plt.xlabel("$Re(x_{0})$", fontsize=14)
plt.ylabel("$Im(x_{0})$", fontsize=14)
plt.title(f"Dynamical Plane, N  Iter={maxiter}  ={L}")
plt.grid()
plt.show()

# Calculo de estadisticas
print("% de elementos que convergen:")
print(np.count_nonzero(IT <= maxiter) / np.size(IT) * 100)
print("Valor medio de iteraciones para converger:")
print(np.mean(IT[IT <= maxiter]))

# Tiempo de computacion
end_time = time.time()
print("Tiempo de computo:", end_time - start_time, "segundos")
```

Dinámica real

Asimismo se presenta el comportamiento dinámico real de las familias (3.1.3) y (3.1.4). En las figuras 3.11-3.13, pueden verse las cuencas asociadas a la familia (3.1.3), para diferentes valores de λ y tomando $N = 10$.

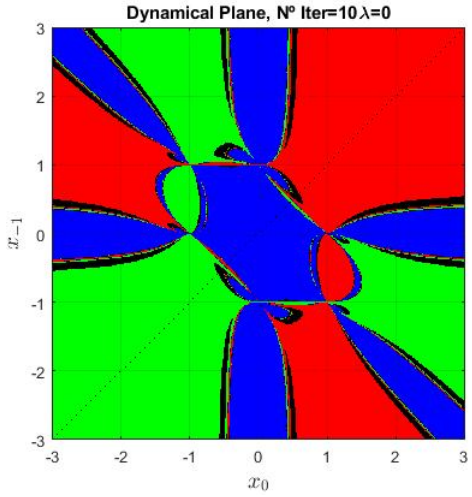


Figura 3.11: Cuencas de atracción asociadas a la iteración de la familia (3.1.3) aplicada al polinomio F para $\lambda = 0$ y $N = 10$.

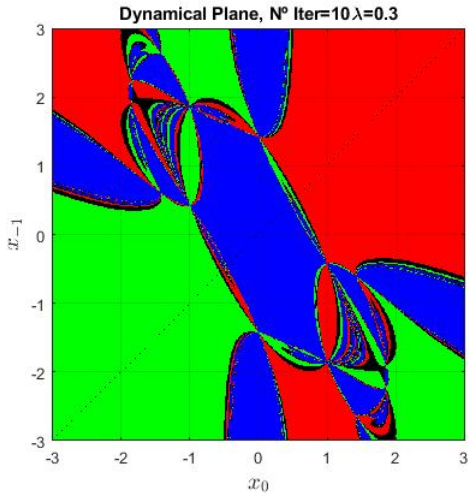


Figura 3.12: Cuencas de atracción asociadas a la iteración de la familia (3.1.3) aplicada al polinomio F para $\lambda = 0.3$ y $N = 10$.

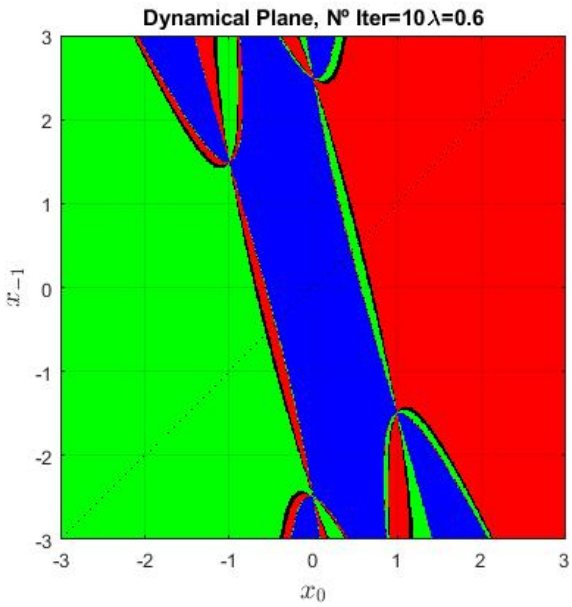


Figura 3.13: Cuencas de atracción asociadas a la iteración de la familia (3.1.3) aplicada al polinomio F para $\lambda = 0.6$ y $N = 10$.

Por otro lado, en las figuras 3.14, 3.15 y 3.16, pueden verse las cuencas de atracción asociadas a la familia (3.1.4), para diferentes valores de $\lambda = 0$, 0.3 y 0.6 y tomando como número de iteraciones $N = 10$.

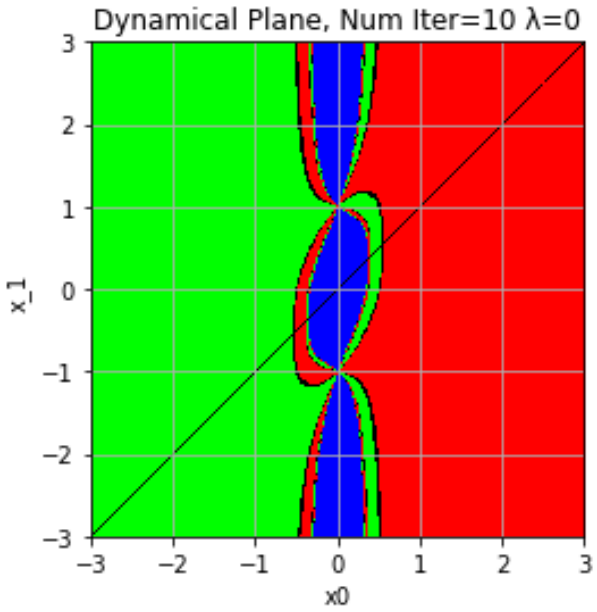


Figura 3.14: Cuencas de atracción asociadas a la iteración de la familia (3.1.4) aplicada al polinomio F para $\lambda = 0$ y $N = 10$.

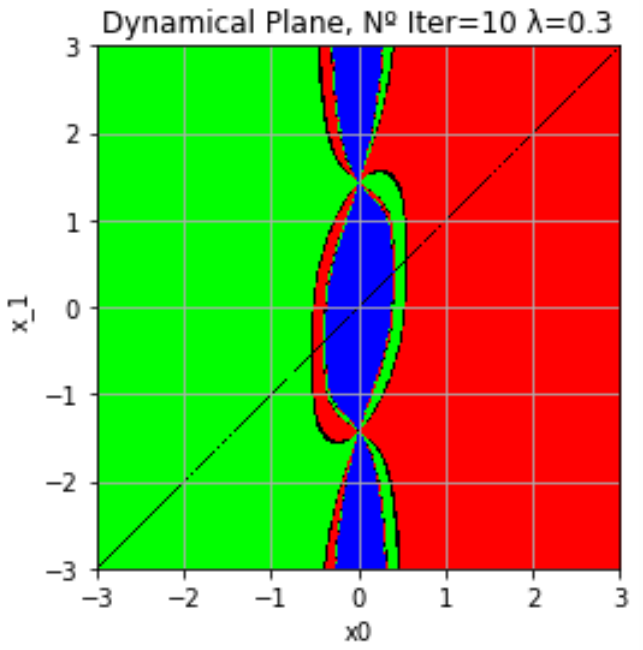


Figura 3.15: Cuencas de atracción asociadas a la iteración de la familia (3.1.4) aplicada al polinomio F para $\lambda = 0.3$ y $N = 10$.

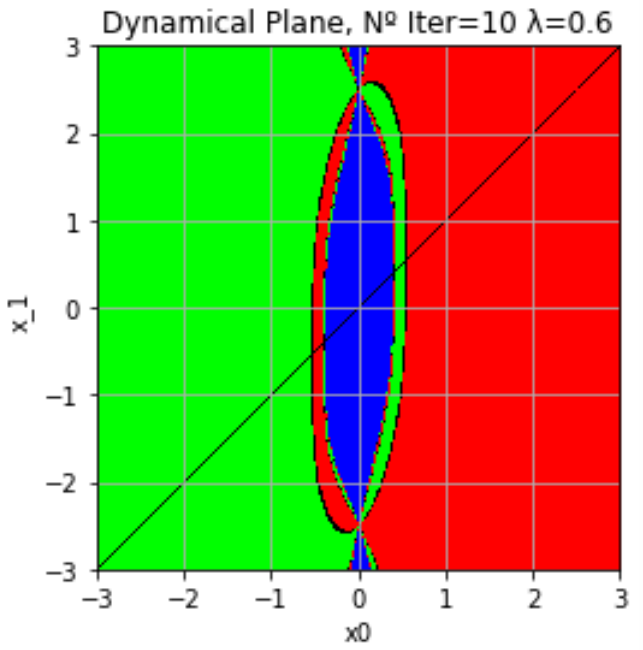
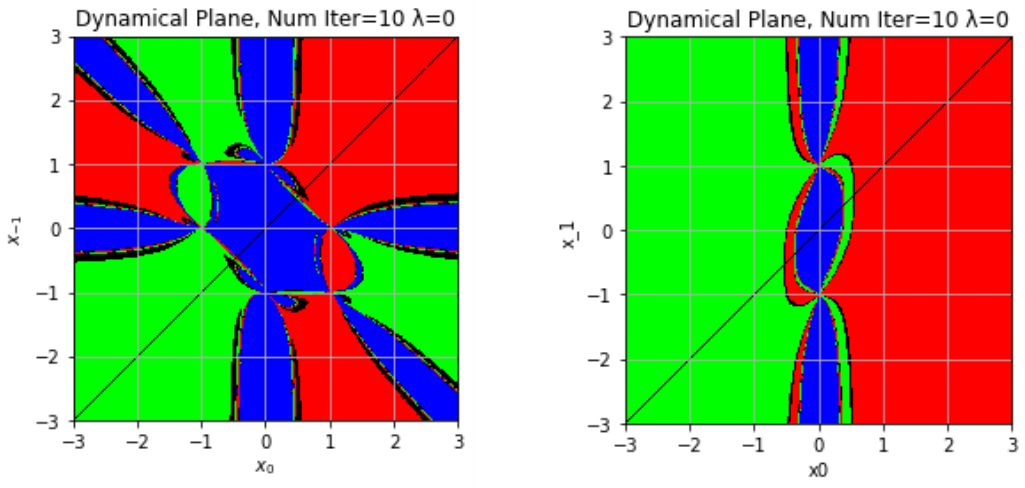


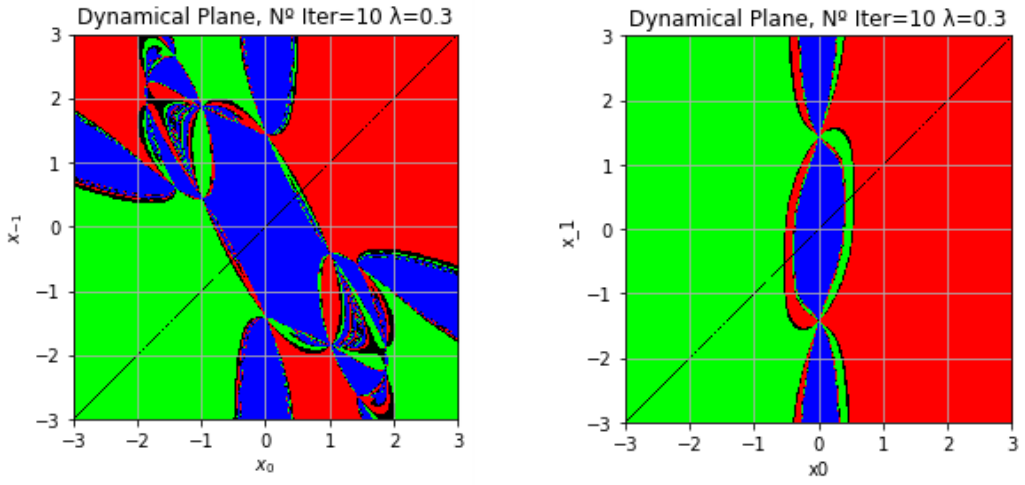
Figura 3.16: Cuencas de atracción asociadas a la iteración de la familia (3.1.4) aplicada al polinomio F para $\lambda = 0.6$ y $N = 10$.

Asimismo, para poder comparar la accesibilidad de ambas familias se van a graficar conjuntamente los elementos de las familias con los mismos valores de λ. Estos gráficos pueden verse en las Figuras 3.17-3.19, donde se observa que la accesibilidad de la familia (3.1.4) es superior a la de (3.1.3).



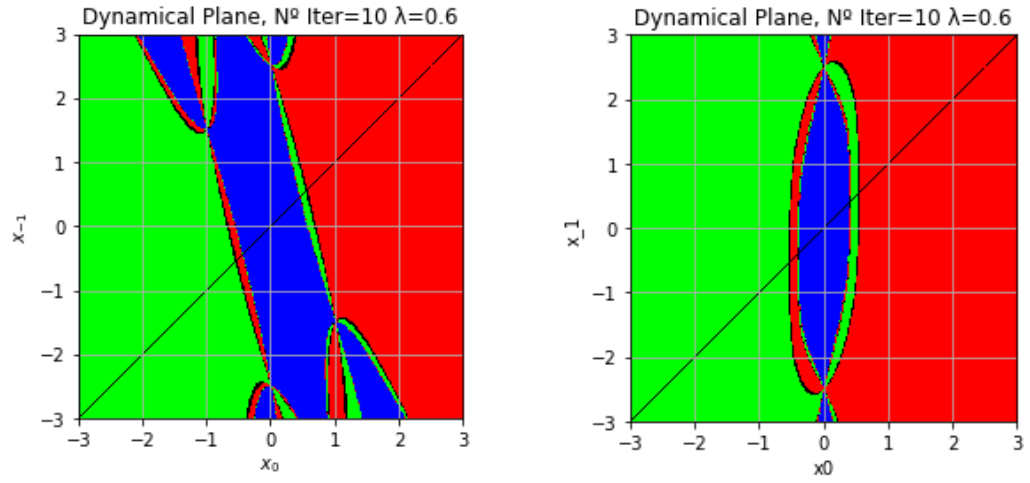
Familia (3.1.3) Familia (3.1.4)

Figura 3.17: Cuencas de atracción asociadas a las diferentes familias para $\lambda = 0$ y $N = 10$.



Familia (3.1.3) Familia (3.1.4)

Figura 3.18: Cuencas de atracción asociadas a las diferentes familias para $\lambda = 0.3$ y $N = 10$.



Familia (3.1.3) Familia (3.1.4)

Figura 3.19: Cuencas de atracción asociadas a las diferentes familias para $\lambda = 0.6$ y $N = 10$.

Tras observar gráficamente las accesibilidades de las familias (3.1.3) y (3.1.4) en las Figuras 3.17-3.19, se calcularon los porcentajes de puntos que garantizan convergencia a las raíces, mostrados en la Tabla 3.5. Como puede comprobarse, en el caso real la accesibilidad de la familia (3.1.4) es superior a la de la familia (3.1.3), al igual que sucedía en el caso complejo.

λ	Método (3.1.3)	Método (3.1.4)
0	94.39 %	98.43 %
0.3	95.08 %	98.73 %
0.6	98.23 %	98.87 %

Tabla 3.5: Porcentaje de puntos de convergencia en la dinámica real para las iteraciones de las familias (3.1.3) y (3.1.4).

A continuación, se adjuntan los códigos de Python utilizados.

3.6.4. Familia (3.1.3) en el caso de dinámica real

```
# -*- coding: utf-8 -*-
#Metodo con memoria secante
# Plano dinamico

import numpy as np
import matplotlib.pyplot as plt
import time

#Parametro del metodo
L=0.3
#Metodo
F= lambda X: X**3+X*abs(X)-2*X
U= lambda X,Y: L*X+(1-L)*Y
DF= lambda X,Y: ((F(X)-F(U(X,Y))))/(X-U(X,Y)) #Aproximacion derivada,
    diferencia dividida
M=lambda X,Y: X-F(X)/DF(X,Y)


#  z0, parte real de los puntos del plano dinamico
#  z_1, parte imaginaria de los puntos del plano dinamico
#  maxiter, numero maximo de iteraciones
#  Z, valor final de la orbita de cada punto
#  I, imagen del plano dinamico
#  tol, valor de tolerancia establecido
#  h, Paso del mallado
tic=time.time() #Inicio tiempo de computacion
h=0.01     #Paso del mallado
tol=1E-5   #Tolerancia de aproximacion
maxiter=10 #Numero maximo iteraciones

z0=np.arange(-3,3+h,h)
z_1=np.arange(-3,3+h,h)
Z0, Z_1=np.meshgrid(z0,z_1, indexing='xy')

itera=1 #Inicializacion parametro para iteraciones
#  R G B para los colores del plano dinamico
#  I, para combinar R G B
R=np.zeros([len(z0),len(z_1)])
G=np.zeros([len(z0),len(z_1)])
B=np.zeros([len(z0),len(z_1)])
I=np.zeros([len(z0),len(z_1),3])
IT=(maxiter+1)*np.ones([len(z0),len(z_1)])
#  Puntos fijos
ZF1=1
ZF2=-1
ZF3=0


while itera<=maxiter:
    Z=M(Z0,Z_1)

    for fil in range(len(z0)):
        for col in range(len(z_1)):
            Zfc=Z[fil,col]
            if np.abs(Zfc-ZF1)<tol:
                R[fil,col]=1
```

```
                IT[fil,col]=min(IT[fil,col],itera)
            if np.abs(Zfc-ZF2)<tol:
                G[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
            if np.abs(Zfc-ZF3)<tol:
                B[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
    #linalg.norm
    #Actualizacion de valores
    itera=itera+1
    Z_1=Z0
    Z0=Z

#Generacion de la imagen del plano
I[...,0]=R
I[...,1]=G
I[...,2]=B

# Se muestra el plano dinamico
plt.figure()
plt.title(f"Dynamical Plane, N  Iter={maxiter}   ={L}")
#plt.plot(np.real(ZF1),np.imag(ZF1),'w*')
#plt.plot(np.real(ZF2),np.imag(ZF2),'w*')
plt.xlabel('$x_{0}$')
plt.ylabel('$x_{-1}$')
plt.grid()
#Se indica origin='lower' para indicar que el origen esta en la
    esquina
#Inferior izquieda
plt.imshow(I,origin='lower', extent=[z0[0],z0[-1],z_1[0],z_1[-1]])

#Numero de iteraciones promedio
print('% de iteraciones que convergen: ', np.size(IT[IT<=maxiter])/np
    .size(IT)*100)
#Valor medio de iteraciones para converger
print('Valor medio de iteraciones para converger: ',np.mean(IT[IT<=
    maxiter]))
#Tiempo de computacion
toc=time.time()
print('Tiempo de computacion:', toc-tic,' segundos')
```

3.6.5. Familia (3.1.4) en el caso de dinámica real

```
# -*- coding: utf-8 -*-
import numpy as np
import matplotlib.pyplot as plt
import time
#Parametro del metodo
L=0.6
#Metodo
F= lambda X: X**3+X*abs(X)-2*X
U = lambda X, Y: L*X + (1-L)*Y  # Aqui supongo que L es una constante
     dada
DF = lambda X, Y: (F(2*X - U(X,Y)) - F(U(X,Y))) / (2*X - U(X,Y) - U(X
    ,Y))
M = lambda X, Y: X - F(X) / DF(X, Y)

#  z0, parte real de los puntos del plano dinamico
#  z_1, parte imaginaria de los puntos del plano dinamico
#  maxiter, numero maximo de iteraciones
#  Z, valor final de la orbita de cada punto
#  I, imagen del plano dinamico
#  tol, valor de tolerancia establecido
#  h, Paso del mallado
tic=time.time() #Inicio tiempo de computacion
h=0.01      #Paso del mallado
tol=1E-5    #Tolerancia de aproximacion
maxiter=10 #Numero maximo iteraciones

z0=np.arange(-3,3+h,h)
z_1=np.arange(-3,3+h,h)
Z0, Z_1=np.meshgrid(z0,z_1, indexing='xy')

itera=1 #Inicializacion parametro para iteraciones
#  R G B para los colores del plano dinamico
#  I, para combinar R G B
R=np.zeros([len(z0),len(z_1)])
G=np.zeros([len(z0),len(z_1)])
B=np.zeros([len(z0),len(z_1)])
I=np.zeros([len(z0),len(z_1),3])
IT=(maxiter+1)*np.ones([len(z0),len(z_1)])
#  Puntos fijos
ZF1=1
ZF2=-1
ZF3=0


while itera<=maxiter:
    Z=M(Z0,Z_1)

    for fil in range(len(z0)):
        for col in range(len(z_1)):
            Zfc=Z[fil,col]
            if np.abs(Zfc-ZF1)<tol:
                R[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
            if np.abs(Zfc-ZF2)<tol:
                G[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
```

```
            if np.abs(Zfc-ZF3)<tol:
                B[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
    #linalg.norm
    #Actualizacion de valores
    itera=itera+1
    Z_1=Z0
    Z0=Z

#Generacion de la imagen del plano
I[...,0]=R
I[...,1]=G
I[...,2]=B

# Se muestra el plano dinamico
plt.figure()
plt.title(f"Dynamical Plane, N   Iter={maxiter}   ={L}")
#plt.plot(np.real(ZF1),np.imag(ZF1),'w*')
#plt.plot(np.real(ZF2),np.imag(ZF2),'w*')
plt.xlabel('x0')
plt.ylabel('x_1')
plt.grid()
#Se indica origin='lower' para indicar que el origen esta en la
    esquina
#Inferior izquieda
plt.imshow(I,origin='lower', extent=[z0[0],z0[-1],z_1[0],z_1[-1]])

#Numero de iteraciones promedio
print('% de iteraciones que convergen: ', np.size(IT[IT<=maxiter])/np
    .size(IT)*100)
#Valor medio de iteraciones para converger
print('Valor medio de iteraciones para converger: ',np.mean(IT[IT<=
    maxiter]))
#Tiempo de computacion
toc=time.time()
print('Tiempo de computacion:', toc-tic,' segundos')
```

3.6.6. Un estudio comparativo dinámico de las familias (3.1.4) y (3.1.5)

Dado que la familia (3.1.5) se reduce a (3.1.4) si `tol` $= 1$, en esta sección se analiza cómo la accesibilidad varía en función del parámetro `tol`. Se concluye que la accesibilidad de (3.1.5) mejora cuando `tol` < 1, superando a la familia (3.1.4).

Dinámica compleja

Observamos gráficamente la accesibilidad según el valor de `tol`. Primero, fijando una tolerancia de 4, es decir, mayor que la supondría para que se redujese a la familia(3.1.4), ver Figuras 3.20- 3.22.

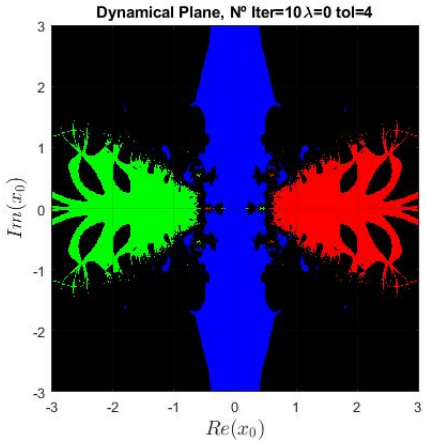


Figura 3.20: Cuencas de atracción asociadas a la iteración de la familia (3.1.5) aplicada al polinomio F con `tol` $= 4$ para $\lambda = 0$ y $N = 10$.

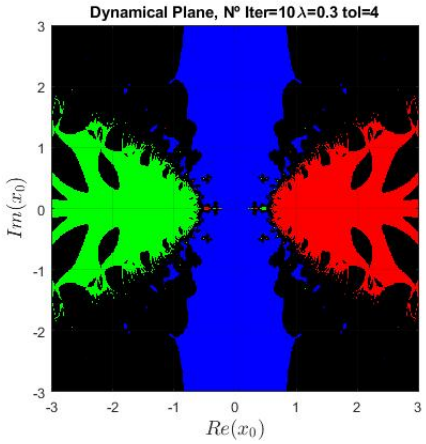


Figura 3.21: Cuencas de atracción asociadas a la iteración de la familia (3.1.5) aplicada al polinomio F con `tol` $= 4$ para $\lambda = 0$ y $N = 10$.

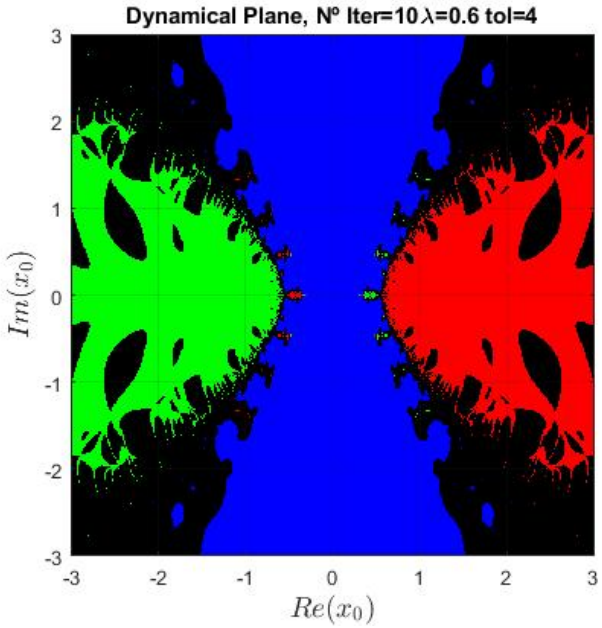


Figura 3.22: Cuencas de atracción asociadas a la iteración de la familia (3.1.5) aplicada al polinomio F con $\texttt{tol} = 4$ para $\lambda = 0.3$ y $N = 10$.

Después, fijando una tolerancia de 0.01, es decir, menor que la supondría para que se redujese a la familia(3.1.4), ver Figuras 3.23- 3.25.

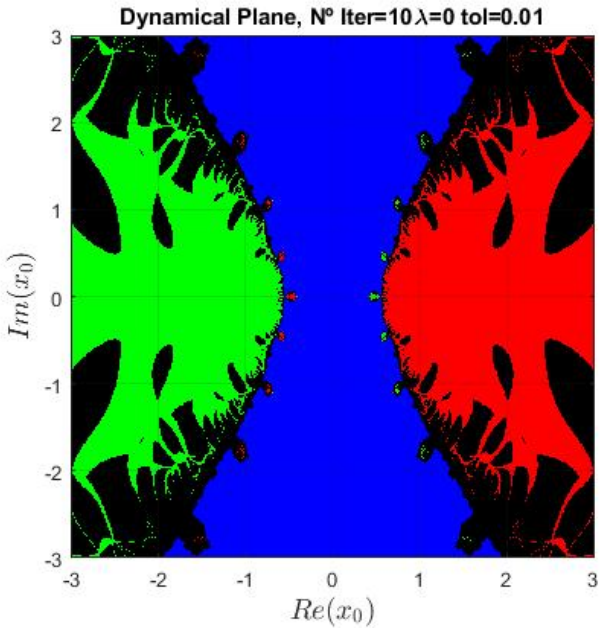


Figura 3.23: Cuencas de atracción asociadas a la iteración de la familia (3.1.5) aplicada al polinomio F con $\texttt{tol} = 0.01$ para $\lambda = 0$ y $N = 10$.

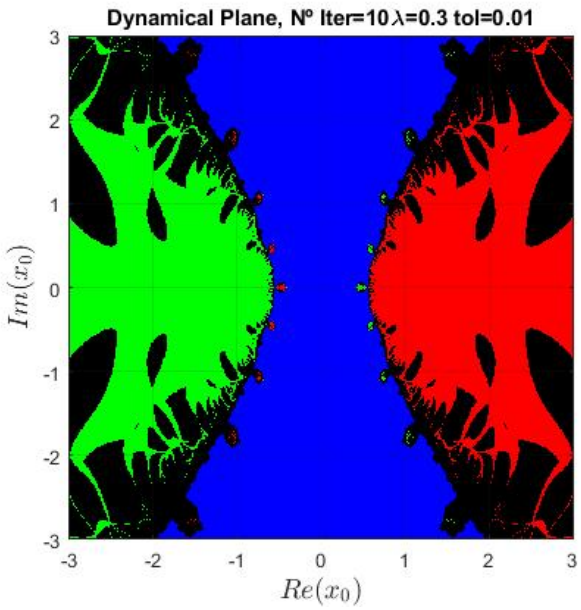


Figura 3.24: Cuencas de atracción asociadas a la iteración de la familia (3.1.5) aplicada al polinomio F con `tol` $= 0.01$ para $\lambda = 0$ y $N = 10$.

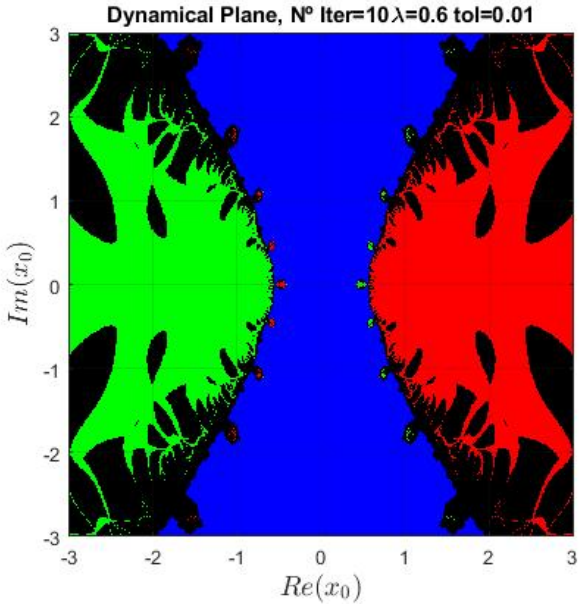
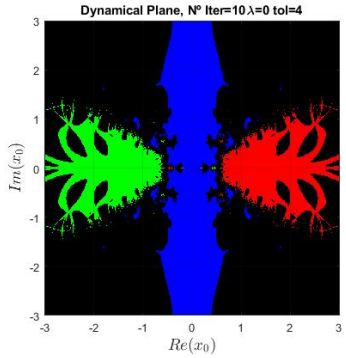
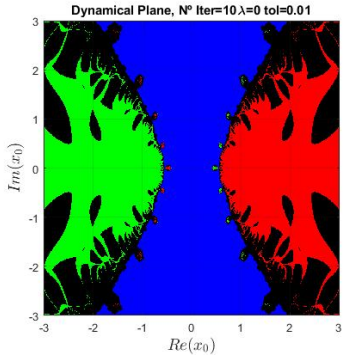


Figura 3.25: Cuencas de atracción asociadas a la iteración de la familia (3.1.5) aplicada al polinomio F con `tol` $= 0.01$ para $\lambda = 0.3$ y $N = 10$.

Asimismo, para poder comparar la accesibilidad de la familia para diferentes valores de `tol` se van a graficar conjuntamente los elementos de la familia con los mismos valores de λ y diferentes de `tol` (Figuras 3.26-3.28), donde se observa que la accesibilidad de la familia (3.1.5) es superior si el valor de `tol` es menor.

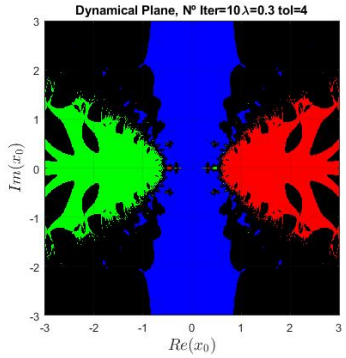


$tol = 4$

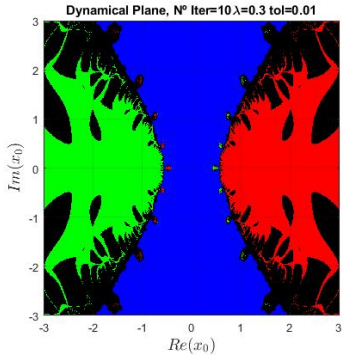


$tol = 0.01$

Figura 3.26: Comparativa de las cuencas de atracción asociadas a la iteración de la familia (3.1.5) aplicada al polinomio F con diferentes valores de `tol` para $\lambda = 0$ y $N = 10$.

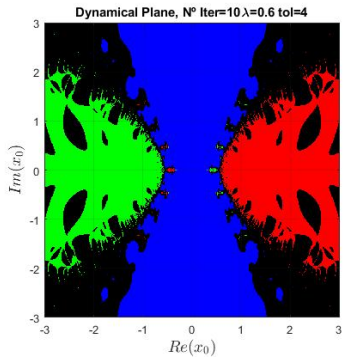


$tol = 4$

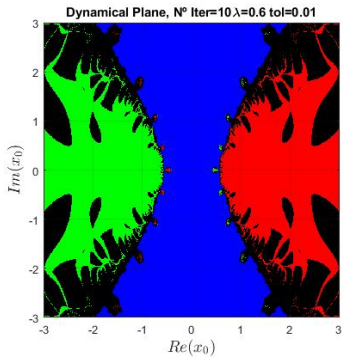


$tol = 0.01$

Figura 3.27: Comparativa de las cuencas de atracción asociadas a la iteración de la familia (3.1.5) aplicada al polinomio F con diferentes valores de `tol` para $\lambda = 0.3$ y $N = 10$.



$tol = 4$



$tol = 0.01$

Figura 3.28: Comparativa de las cuencas de atracción asociadas a la iteración de la familia (3.1.5) aplicada al polinomio F con diferentes valores de `tol` para $\lambda = 0.6$ y $N = 10$.

Una vez que la accesibilidad de la familia (3.1.5) se ha visto gráficamente en las Figuras 3.20 y 3.25, analizamos su comportamiento numérico. Para esto, calculamos el porcentaje de puntos que garantizan la convergencia a cualquiera de las raíces y lo mostramos en la Tabla 3.6, donde observamos que la accesibilidad es mejor cuanto menor es el valor del parámetro `tol`.

λ	`tol`	Método (3.1.5)
0	0.01	70.18 %
	1	69.42 %
	4	31.31 %
0.3	0.01	70.18 %
	1	69.05 %
	4	45.04 %
0.6	0.01	70.19 %
	1	69.63 %
	4	63.40 %

Tabla 3.6: Porcentaje de puntos de convergencia en la dinámica compleja para iteraciones de la familia (3.1.5) con `tol` $= 0.01, 1, 4$.

A continuación, se adjuntan el código de Python utilizado.

3.6.7. Familia (3.1.5) en el caso de dinámica compleja

```
# -*- coding: utf-8 -*-
import numpy as np
import matplotlib.pyplot as plt
import time

# Inicio de tiempo de computacion
start_time = time.time()

# Parametros del metodo
L = 0.6  # Parametro lambda
T = 0.01  # Parametro tol

#Metodo
F = lambda X: X**3 + X*abs(X) - 2*X
U = lambda X, Y: L*X + (1-L)*Y  # X representa x0 Y representa x_1
DF = lambda X, Y: (F(X + T*(X - U(X, Y))) - F(X - T*(X - U(X, Y)))) /
     (2 * T * (X - U(X, Y)))
M = lambda X, Y: X - F(X) / DF(X, Y)

# Valores iniciales
h=0.01 #Tama o de paso
Rez0 = np.arange(-3, 3+h,h)
Imz0 = np.arange(-3, 3+h, h)
maxiter = 10
tol = 1e-5

puntos = len(Rez0)
ReZ0, ImZ0 = np.meshgrid(Rez0, Imz0)
Z0 = ReZ0 + 1j * ImZ0
Z_1 = Z0 - 0.1
iter = 1
R = np.zeros((puntos, puntos))
G = np.zeros((puntos, puntos))
B = np.zeros((puntos, puntos))
IT = (maxiter + 1) * np.ones_like(Z0)

# Puntos fijos
ZF1 = 1
ZF2 = -1
ZF3 = 0

# Ejecucion del metodo iterativo
while iter <= maxiter:
    Z = M(Z0, Z_1)
    for fil in range(puntos):
        for col in range(puntos):
            Zfc = Z[fil, col]
            if abs(Zfc - ZF1) < tol:
                R[fil, col] = 1
                IT[fil, col] = min(iter, IT[fil, col])
            if abs(Zfc - ZF2) < tol:
                G[fil, col] = 1
                IT[fil, col] = min(iter, IT[fil, col])
            if abs(Zfc - ZF3) < tol:
                B[fil, col] = 1
                IT[fil, col] = min(iter, IT[fil, col])
```

```
        iter += 1
        Z_1 = Z0
        Z0 = Z

# Generacion de la imagen
I = np.zeros((puntos, puntos, 3))
I[:, :, 0] = R
I[:, :, 1] = G
I[:, :, 2] = B

# Representacion de la imagen
plt.imshow(I, extent=[-3, 3, -3, 3])
plt.xlabel("$Re(x_{0})$", fontsize=14)
plt.ylabel("$Im(x_{0})$", fontsize=14)
plt.title(f"Dynamical Plane, N  Iter={maxiter}  ={L}, tol={T}")
plt.grid()
plt.show()

# Calculo de estadisticas
print("% de elementos que convergen:")
print(np.count_nonzero(IT <= maxiter) / np.size(IT) * 100)
print("Valor medio de iteraciones para converger:")
print(np.mean(IT[IT <= maxiter]))

# Tiempo de computacion
end_time = time.time()
print("Tiempo de computo:", end_time - start_time, "segundos")
```

Dinámica real

Al igual que en el caso complejo, se va a observar gráficamente la accesibilidad según el valor de `tol`. Primero, fijando una tolerancia de 4, es decir, mayor que la supondría para que se redujese a la familia(3.1.4), ver Figuras 3.29- 3.31.

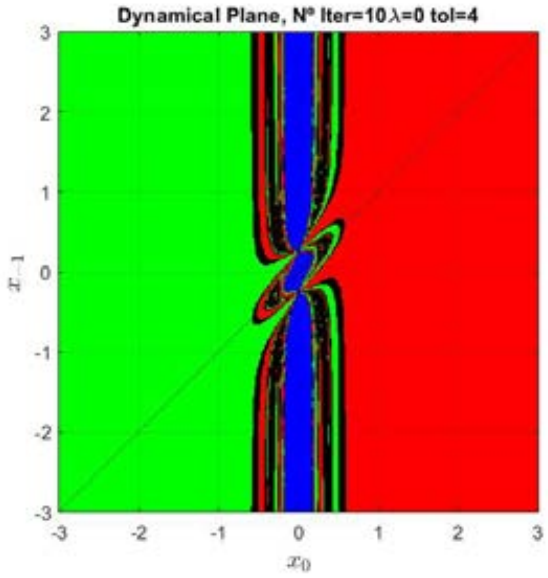


Figura 3.29: Cuencas de atracción asociadas a la iteración de la familia (3.1.5) aplicada al polinomio F con `tol` $= 4$ para $\lambda = 0$ y $N = 10$.

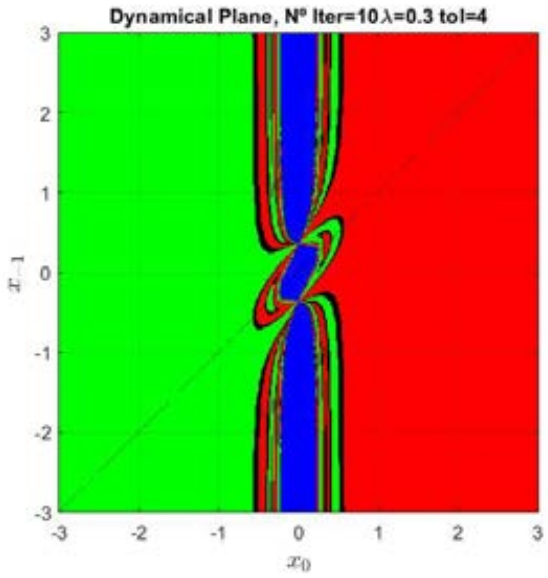


Figura 3.30: Cuencas de atracción asociadas a la iteración de la familia (3.1.5) aplicada al polinomio F con `tol` $= 4$ para $\lambda = 0.3$ y $N = 10$.

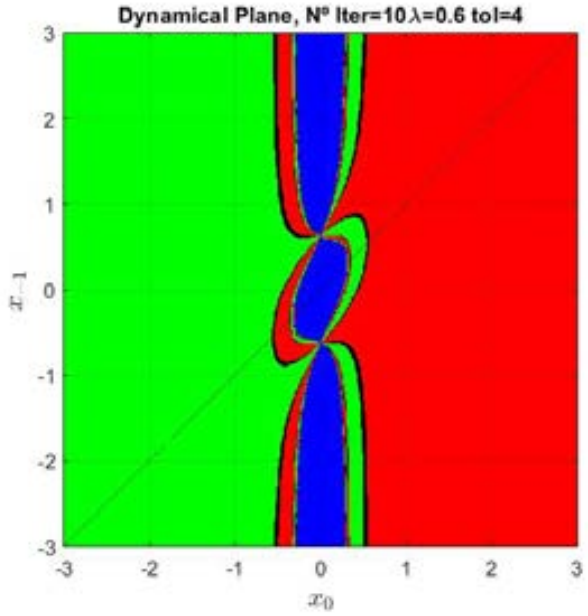


Figura 3.31: Cuencas de atracción asociadas a la iteración de la familia (3.1.5) aplicada al polinomio F con $\texttt{tol} = 4$ para $\lambda = 0.6$ y $N = 10$.

Después, fijando una tolerancia de 0.01, es decir, menor que la supondría para que se redujese a la familia(3.1.4), ver Figuras 3.32- 3.34.

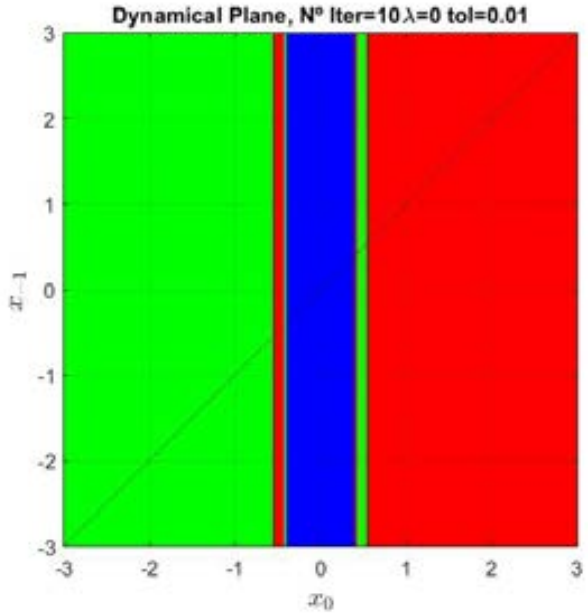


Figura 3.32: Cuencas de atracción asociadas a la iteración de la familia (3.1.5) aplicada al polinomio F con $\texttt{tol} = 0.01$ para $\lambda = 0$ y $N = 10$.

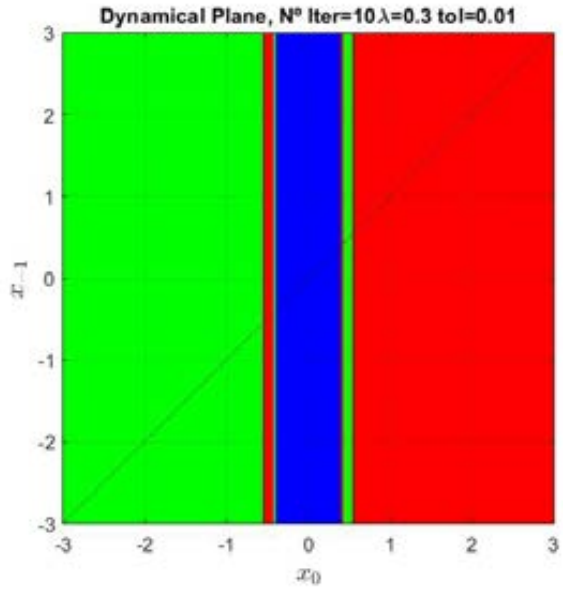


Figura 3.33: Cuencas de atracción asociadas a la iteración de la familia (3.1.5) aplicada al polinomio F con `tol` $= 0.01$ para $\lambda = 0.3$ y $N = 10$.

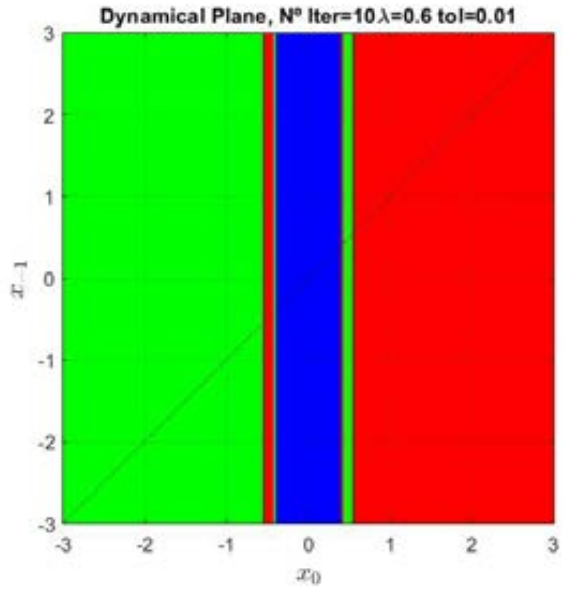
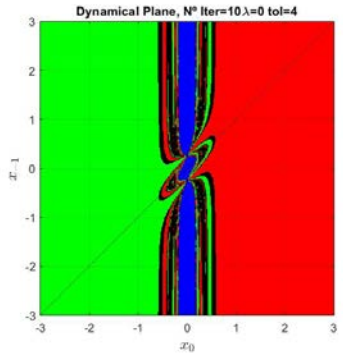
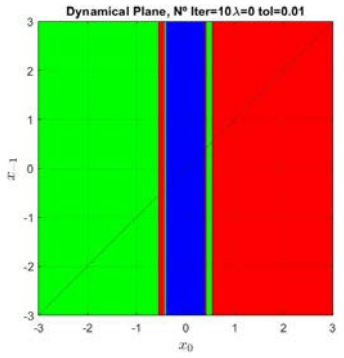


Figura 3.34: Cuencas de atracción asociadas a la iteración de la familia (3.1.5) aplicada al polinomio F con `tol` $= 0.01$ para $\lambda = 0.6$ y $N = 10$.

Asimismo, para poder comparar la accesibilidad de la familia para diferentes valores de `tol` se van a graficar conjuntamente los elementos de la familia con los mismos valores de λ y diferentes de `tol` (Figuras 3.35-3.37), donde se observa que la accesibilidad de la familia (3.1.5) es superior si el valor de `tol` es menor.

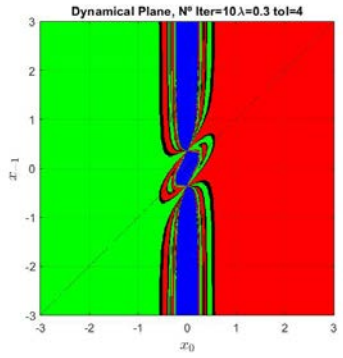


$tol = 4$

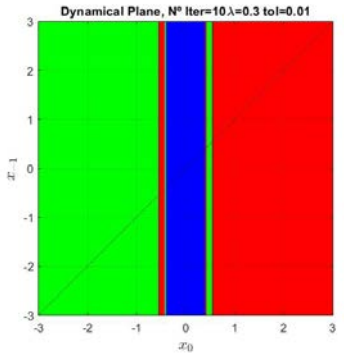


$tol = 0.01$

Figura 3.35: Comparativa de las cuencas de atracción asociadas a la iteración de la familia (3.1.5) aplicada al polinomio F con diferentes valores de `tol` para $\lambda = 0$ y $N = 10$.

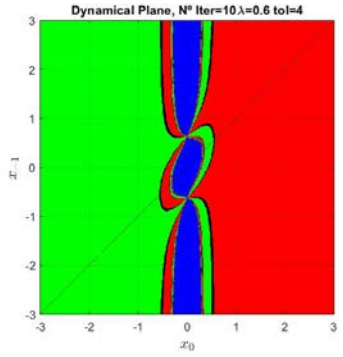


$tol = 4$

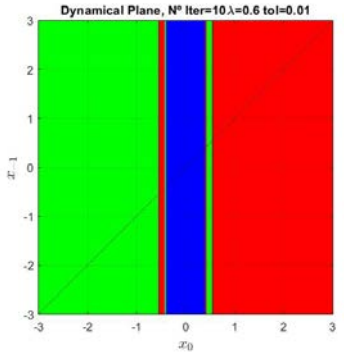


$tol = 0.01$

Figura 3.36: Comparativa de las cuencas de atracción asociadas a la iteración de la familia (3.1.5) aplicada al polinomio F con diferentes valores de `tol` para $\lambda = 0.3$ y $N = 10$.



$tol = 4$



$tol = 0.01$

Figura 3.37: Comparativa de las cuencas de atracción asociadas a la iteración de la familia (3.1.5) aplicada al polinomio F con diferentes valores de `tol` para $\lambda = 0.6$ y $N = 10$.

Una vez que la accesibilidad de la familia (3.1.5) se ha visto gráficamente en las Figuras 3.29 y 3.34, analizamos su comportamiento numérico. Para esto, calculamos el porcentaje de puntos que garantizan la convergencia a cualquiera de las raíces y lo mostramos en la Tabla 3.7, donde observamos que la accesibilidad es mejor cuanto menor es el valor del parámetro `tol`.

λ	`tol`	Método (4.2.4)
0	0.01	99.01 %
	1	98.42 %
	4	92.08 %
0.3	0.01	98.98 %
	1	98.73 %
	4	95.66 %
0.6	0.01	98.94 %
	1	98.87 %
	4	97.95 %

Tabla 3.7: Porcentaje de puntos de convergencia en la dinámica real para las iteraciones de la familia (3.1.5) con $\texttt{tol} = 0.01, 1, 4$.

A continuación, se adjuntan el código de Python utilizado.

3.6.8. Familia (3.1.5) en el caso de dinámica real

```
# -*- coding: utf-8 -*-
import numpy as np
import matplotlib.pyplot as plt
import time

# Parametros del metodo
L = 0.6  # Parametro lambda
T = 0.01  # Parametro tol

#Metodo
F = lambda X: X**3 + X*abs(X) - 2*X
U = lambda X, Y: L*X + (1-L)*Y  # X representa x0 Y representa x_1
DF = lambda X, Y: (F(X + T*(X - U(X, Y))) - F(X - T*(X - U(X, Y)))) /
     (2 * T * (X - U(X, Y)))
M = lambda X, Y: X - F(X) / DF(X, Y)

#  z0, parte real de los puntos del plano dinamico
#  z_1, parte imaginaria de los puntos del plano dinamico
#  maxiter, numero maximo de iteraciones
#  Z, valor final de la orbita de cada punto
#  I, imagen del plano dinamico
#  tol, valor de tolerancia establecido
#  h, tama o de paso del mallado
tic=time.time() #Inicio tiempo de computacion
h=0.01      #Tama o de paso del mallado
tol=1E-5    #Tolerancia de aproximacion
maxiter=10 #Numero maximo iteraciones

z0=np.arange(-3,3+h,h)
z_1=np.arange(-3,3+h,h)
Z0, Z_1=np.meshgrid(z0,z_1, indexing='xy')

itera=1 #Inicializacion parametro para iteraciones
#  R G B para los colores del plano dinamico
#  I, para combinar R G B
R=np.zeros([len(z0),len(z_1)])
G=np.zeros([len(z0),len(z_1)])
B=np.zeros([len(z0),len(z_1)])
I=np.zeros([len(z0),len(z_1),3])
IT=(maxiter+1)*np.ones([len(z0),len(z_1)])
#  Puntos fijos
ZF1=1
ZF2=-1
ZF3=0


while itera<=maxiter:
    Z=M(Z0,Z_1)

    for fil in range(len(z0)):
        for col in range(len(z_1)):
            Zfc=Z[fil,col]
            if np.abs(Zfc-ZF1)<tol:
                R[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
            if np.abs(Zfc-ZF2)<tol:
```

```
                G[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
            if np.abs(Zfc-ZF3)<tol:
                B[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
    #linalg.norm
    #Actualizacion de valores
    itera=itera+1
    Z_1=Z0
    Z0=Z

#Generacion de la imagen del plano
I[...,0]=R
I[...,1]=G
I[...,2]=B

# Se muestra el plano dinamico
plt.figure()
#plt.title('Dynamical Plane, Num Iter=%s' %(str(maxiter)))
plt.title(f"Dynamical Plane, N  Iter={maxiter}  ={L}, tol={T}")
#plt.plot(np.real(ZF1),np.imag(ZF1),'w*')
#plt.plot(np.real(ZF2),np.imag(ZF2),'w*')
plt.xlabel('$x_{0}$')
plt.ylabel('$x_{-1}$')
plt.grid()
#Se indica origin='lower' para indicar que el origen esta en la
    esquina
#Inferior izquieda
plt.imshow(I,origin='lower', extent=[z0[0],z0[-1],z_1[0],z_1[-1]])

#Numero de iteraciones promedio
print('% de iteraciones que convergen: ', np.size(IT[IT<=maxiter])/np
    .size(IT)*100)
#Valor medio de iteraciones para converger
print('Valor medio de iteraciones para converger: ',np.mean(IT[IT<=
    maxiter]))
#Tiempo de computacion
toc=time.time()
print('Tiempo de computacion:', toc-tic,' segundos')
```

3.7. Conclusiones

En este capítulo se considera una función no lineal $F : \Omega \subset \mathbb{R}^m \to \mathbb{R}^m$, donde Ω es un dominio convexo abierto en $\mathbb{R}^m$, con

$$F(x) = (F_1(x), F_2(x), \ldots, F_m(x))$$

y

$$F_i : \Omega \subseteq \mathbb{R}^m \to \mathbb{R}$$

para cada $i = 1, 2, \ldots, m$, con

$$x = (x_1, x_2, \ldots, x_m).$$

Uno de los métodos más utilizados es el método de Newton, debido a su convergencia cuadrática y su alta eficiencia computacional. Sin embargo, uno de los principales inconvenientes de este método es que requiere la evaluación de la derivada del operador involucrado en cada iteración. Esto no solo lo limita a situaciones donde el operador es diferenciable, sino que también puede ser ineficiente cuando la evaluación de la derivada es costosa o compleja. En tales casos, es habitual recurrir a métodos iterativos que, en lugar de derivadas, utilizan diferencias divididas para aproximar las soluciones. Partiendo de lafamilia presentada en [85]:

$$\begin{cases} w_{-1}, w_0 \text{ dados en } \Omega, \quad \lambda \in [0,1], \\ u_n = \lambda w_n + (1-\lambda) w_{n-1}, \quad n \geq 0, \\ w_{n+1} = w_n - [u_n, w_n; F]^{-1} F(w_n), \end{cases}$$

donde el parámetro $\lambda \in [0, 1]$. El orden de convergencia R de esta familia es, al menos, $\frac{1+\sqrt{5}}{2}$ para $\lambda \in [0, 1)$, de manera similar al método de la secante.

Partiendo de dicha familia, se construyó una familia biparamétrica de métodos iterativos que presenta convergencia cuadrática, basada en el uso de diferencias divididas simétricas y un parámetro adicional:

$$\begin{cases} w_{-1}, w_0 \text{ dados en } \Omega, \quad \lambda \in [0,1], \\ u_n = \lambda w_n + (1-\lambda) w_{n-1}, \quad n \geq 0, \\ w_{n+1} = w_n - [u_n, 2w_n - u_n; F]^{-1} F(w_n). \end{cases}$$

El orden de convergencia R de esta familia es al menos cuadrático para $\lambda \in [0, 1]$, equivalente al del método de Newton.

Por otro lado, mediante la introducción de un parámetro $\texttt{tol} > 0$ y usando la aproximación:

$$[w_n - \texttt{tol}(w_n - u_n), w_n + \texttt{tol}(w_n - u_n); F] \approx F'(w_n)$$

y se obtiene la siguiente familia biparamétrica de métodos iterativos:

$$\begin{cases} w_{-1}, w_0 \text{ dados en } \Omega, \quad \lambda \in [0,1], \\ u_n = \lambda w_n + (1-\lambda) w_{n-1}, \quad n \geq 0, \\ w_{n+1} = w_n - [w_n - \texttt{tol}(w_n - u_n), w_n + \texttt{tol}(w_n - u_n); F]^{-1} F(w_n). \end{cases}$$

Una vez construída la familia, se ha presentado el orden de el orden de convergencia del método iterativo.Posteriormente, se ha presentado una comparativa de las eficiencias de las 3 familias, de donde se desprende que en la que se ha introducido el parámetro `tol` es la más eficiente, tanto para problemas pequeños como para problemas grandes. Posteriormente se han presentado las condiciones que van a agarantizar la convergencia local de la familia que depende del parámetro `tol` y se presentan dos resultados de convergencia y uno de unicidad.

Después de garantizar la convergencia de forma teórica, se aplican a 3 ejemplos diferentes: En el primer ejemplo, se utiliza una ecuación integral no lineal del tipo Hammerstein mixto, para ilustrar el ejemplo. En el segundo ejemplo, se observará la influencia del parámetro `tol` cuando se utiliza para aproximar soluciones. Para esto, se utilizará una ecuación integral no lineal del tipo Hammerstein mixto similar a la del ejemplo anterior. Por último, en el tercer ejemplo se analiza la influencia del parámetro λ en las aproximaciones de soluciones.

Por otro lado, se lleva a cabo un estudio dinámico comparativo entre las familias. Asimismo, se analiza la accesibilidad de la familia desde dos perspectivas. Primero, de forma experimental, mediante un estudio dinámico, tanto real como complejo, de los métodos iterativos de la familia biparamétrica. Esto permite concluir que la familia biparamétrica intrdoducida tiene mejor accesibilidad que las otras dos, en términos de porcentajes de puntos iniciales convergentes a las soluciones.

Por último, para ambos enfoques se han facilitado códigos para poder replicarlos de forma fácil en Python.

Capítulo 4

Un procedimiento para obtener convergencia cuadrática a partir del método de la secante

Los resultados de este capítulo han sido publicados en [60]:

- "A procedure to obtain quadratic convergence from the secant methods".
- DOI: `https://doi.org/10.1016/j.cam.2024.115912`.
- Revista "Journal of Computational and Applied Mathematics".

4.1. Introducción

En este capítulo se va a introducir una familia uniparamétrica de métodos iterativos que no requieren el uso de derivadas en su algoritmo para aproximar una solución x^* de una ecuación no lineal $F(x) = 0$, donde $F : \Omega \subseteq X \to Y$ es un operador no lineal definido en un dominio convexo, abierto y no vacío Ω, perteneciente a un espacio de Banach X, cuyos valores están en otro espacio de Banach Y.

Como ya se ha comentado anteriormente, el método de Newton es uno de los procedimientos más utilizados para obtener una solución x^* de la ecuación $F(x) = 0$, con el algoritmo dado por

$$\begin{cases} x_0 \text{ dado en } \Omega, \\ x_{n+1} = x_n - [F'(x_n)]^{-1} F(x_n), \quad n \geq 0. \end{cases} \tag{4.1.1}$$

Este método ofrece convergencia cuadrática y un costo computacional moderado, lo que asegura su buena eficiencia. Sin embargo, una de las principales limitaciones del método de Newton radica en la necesidad de calcular la derivada $F'(x)$ en cada iteración, lo que impide su aplicación en ecuaciones con operadores no diferenciables o cuando la evaluación de la derivada es computacionalmente costosa.

Como a lo largo de toda la tesis, este capítulo se enfoca en métodos iterativos que eviten el cálculo de derivadas del operador F en cada paso del algoritmo. De

nuevo, aparece el método de la secante como el más conocido de estos métodos ([6, 138]) y viene formulado sustituyendo la derivada por el operador diferencia dividida de primer orden $[u, v; F]$ de F en u y v con $u \neq v$ ([22, 75]) donde $[u, v; F]$ es un operador lineal y acotado de X a Y que cumple la relación

$$[u, v; F](u - v) = F(u) - F(v), \quad u, v \in \Omega.$$

Así, el método de la secante viene definido por:

$$\begin{cases} x_0, x_{-1} \text{ dados en } \Omega, \\ x_{n+1} = x_n - [x_{n-1}, x_n; F]^{-1} F(x_n), \quad n \geq 0, \end{cases} \tag{4.1.2}$$

y presenta una convergencia superlineal con un orden de convergencia R de al menos $\dfrac{1+\sqrt{5}}{2}$ ([54]).

A la hora de seleccionar un método iterativo para resolver $F(x) = 0$ se considera su eficiencia, que relaciona la velocidad de convergencia con el costo computacional del método. Para garantizar la convergencia cuadrática, que posee el método de Newton, es necesario aproximar adecuadamente la primera derivada del operador F mediante una buena diferencia dividida. Las diferencias divididas simétricas suelen ofrecer mejores resultados en general. Partiendo de esta premisa, se va a modificar el método de la secante para construir una familia uniparamétrica de métodos iterativos con convergencia cuadrática.

El propósito de este capítulo es presentar una familia uniparamétrica de métodos iterativos que, además de poseer convergencia cuadrática, ofrece una eficiencia computacional superior a la del método de la secante (4.1.2). Asimismo, se realiza un análisis experimental sobre la accesibilidad de dicha familia desde un enfoque dinámico de los métodos iterativos. También se da un resultado que garantiza la convergencia local de esta familia bajo unas condiciones.

La estructura del capítulo es la siguiente:

- En la Sección 4.2, se construye una familia uniparamétrica de métodos iterativos con memoria y segundo orden de convergencia, partiendo de una aproximación adecuada de la primera derivada del operador que interviene en el método de Newton.
- En la Sección 4.3, se presenta un análisis comparativo de la eficiencia de esta familia de métodos iterativos en relación con el método de la secante, demostrando que la familia tiene una mejor eficiencia computacional.
- En la Sección 4.4, se estudia la convergencia local de la familia bajo ciertas condiciones sobre la diferencia dividida utilizada en el algoritmo.
- En la Sección 4.5, se proporcionan tres ejemplos que ilustran los objetivos del capítulo.
- En la Sección 4.6, se lleva a cabo un estudio dinámico comparativo entre la familia y el método de la secante, mostrando que la accesibilidad de este último es inferior.
- Por último, en la Sección 4.7, se presentan las conclusiones que se pueden extraer de este capítulo.

4.2. Desarrollo de una familia uniparamétrica de métodos iterativos

Para aproximar la solución x^* de la ecuación $F(x) = 0$, en [85] Hernández y Rubio proponen una familia uniparamétrica de métodos iterativos tipo secante, dada por:

$$\begin{cases} x_{-1}, x_0 \text{ iniciales en } \Omega, \quad \theta \in [0,1], \\ y_n = \theta x_n + (1-\theta)x_{n-1}, \quad n \geq 0, \\ x_{n+1} = x_n - [y_n, x_n; F]^{-1} F(x_n), \end{cases} \tag{4.2.1}$$

donde $[z, w; F]$, con $z, w \in \Omega$, representa una diferencia dividida de primer orden.

Cabe destacar que los métodos descritos en (4.2.1) generalizan el método de la secante. En el contexto real, se observa que cuanto más próximos se encuentren x_n y y_n, mayor será la velocidad de convergencia. En efecto, en [85] se demuestra que un valor elevado de $\theta \in [0,1]$ incrementa la velocidad de convergencia, acercándose al comportamiento del método de Newton cuando θ se aproxima a 1.

Para deducir (4.2.1), Hernández y Rubio primero exploran el caso real. A partir de las aproximaciones x_{n-1} y x_n, intentan mejorar el método de la secante mediante la aproximación

$$\widehat{x}_{n+1} = x_n - [x_{n-1}, x_n; F]^{-1} F(x_n).$$

Para ello, definen un punto intermedio y_n, ubicado entre x_{n-1} y x_n, como $y_n = \theta x_n + (1-\theta)x_{n-1}$, con $\theta \in [0,1)$, y formulan la aproximación

$$\widetilde{x}_{n+1} = x_n - [y_n, x_n; F]^{-1} F(x_n).$$

A través de una interpretación geométrica, es evidente que $\widetilde{x}_{n+1}$ es una mejor estimación de x^* que $\widehat{x}_{n+1}$.

En vista de lo anterior, se puede afirmar que (4.2.1) constituye una combinación entre el método de la secante y el método de Newton. Si $\theta = 0$, este esquema se reduce al método de la secante, y cuando $\theta = 1$ y F es diferenciable, se obtiene el método de Newton, pues en ese caso $x_n = y_n$ y $[y_n, x_n; F] = F'(x_n)$. Según [62, 87], el orden de convergencia R de (4.2.1) es al menos $\dfrac{1+\sqrt{5}}{2}$ cuando $\theta \in [0,1)$, siendo el mismo que el del método de la secante.

En el estudio presentado en [53], se propone otra mejora del método de la secante en el caso real. Este enfoque fija la aproximación x_{n-1} y define $y_n = \theta x_n + (1-\theta)x_{n-1}$ con $\theta \geq 1$, acercando y_n a x^* más que x_n. Así, la nueva estimación

$$\widetilde{x}_{n+1} = x_n - [x_{n-1}, y_n; F]^{-1} F(x_n)$$

ofrece una mejor aproximación a x^* que $\widehat{x}_{n+1}$. Este enfoque se extiende a espacios de Banach y genera la siguiente familia uniparamétrica de métodos tipo secante:

$$\begin{cases} x_{-1}, x_0 \text{ iniciales en } \Omega, \quad \theta \geq 1, \\ y_n = \theta x_n + (1-\theta)x_{n-1}, \quad n \geq 0, x_{n+1} = x_n - [x_{n-1}, y_n; F]^{-1} F(x_n). \end{cases} \tag{4.2.2}$$

Existe un valor $\theta_0 \in \mathbb{R}$, $\theta_0 \geq 2$, tal que la familia (4.2.2) presenta un orden de convergencia R de al menos $\frac{1+\sqrt{5}}{2}$ cuando $\theta \in [1, \theta_0]$ y $\theta \neq 2$. En particular, cuando $\theta = 2$, se obtiene un método iterativo de orden dos, conocido como el método de Kurchatov ([8, 9, 100, 147]). También se observa que la familia (4.2.2) se reduce al método de la secante cuando $\theta = 1$.

Como se mencionó en la introducción, el objetivo central de este capítulo es adaptar el método de la secante para construir una familia uniparamétrica de métodos iterativos con convergencia cuadrática. Al observar el algoritmo del método de Kurchatov,

$$\begin{cases} x_{-1}, x_0 \text{ iniciales en } \Omega, \\ x_{n+1} = x_n - [x_{n-1}, 2x_n - x_{n-1}; F]^{-1} F(x_n), \end{cases} \tag{4.2.3}$$

que se deriva de (4.2.2) con $\theta = 2$ y presenta convergencia cuadrática, puede argumentarse que dicha convergencia cuadrática se alcanza debido a la aproximación de la derivada

$$[x_{n-1}, 2x_n - x_{n-1}; F] = [x_n - (x_n - x_{n-1}), x_n + (x_n - x_{n-1}); F] \approx F'(x_n),$$

la cual se utiliza en el método de Newton. Esta es una diferencia dividida simétrica de primer orden. Para lograr convergencia cuadrática en la práctica, es necesario obtener una buena aproximación de la primera derivada de F en el método de Newton. Por tanto, en lugar de considerar la diferencia dividida

$$[y_n, x_n; F] = [x_n - (1-\theta)(x_n - x_{n-1}), x_n; F]$$

en (4.2.1), se propone utilizar la diferencia dividida

$$[x_n - (1-\theta)(x_n - x_{n-1}), x_n + (1-\theta)(x_n - x_{n-1}); F],$$

lo que lleva a la siguiente familia uniparamétrica de métodos iterativos:

$$\begin{cases} x_{-1}, x_0 \text{ iniciales en } \Omega, \quad \theta \in [0, 1], \\ y_n = x_n - (1-\theta)(x_n - x_{n-1}), \quad n \geq 0, \\ z_n = x_n + (1-\theta)(x_n - x_{n-1}), \quad n \geq 0, \\ x_{n+1} = x_n - [y_n, z_n; F]^{-1} F(x_n). \end{cases} \tag{4.2.4}$$

La familia (4.2.4) incluye tanto el método de Kurchatov como el método de Newton, ya que (4.2.4) se reduce al método de Kurchatov cuando $\theta = 0$ y al método de Newton cuando $\theta = 1$ y F es diferenciable.

Si se reescribe la familia de métodos iterativos (4.2.4) como

$$\begin{cases} x_{-1}, x_0 \text{ iniciales en } \Omega, \quad \theta \in [0, 1], \\ y_n = \theta x_n + (1-\theta) x_{n-1}, \quad n \geq 0, \\ z_n = (2-\theta) x_n - (1-\theta) x_{n-1}, \quad n \geq 0, \\ x_{n+1} = x_n - [y_n, z_n; F]^{-1} F(x_n), \end{cases} \tag{4.2.5}$$

es posible demostrar (ver [54], p. 204, Teorema 1) que el orden de convergencia R de (4.2.4) es cuadrático si $\theta \in [0, 1]$, igual al del método de Newton.

4.3. Análisis comparativo de eficiencia entre el método de la secante y la familia uniparamétrica (4.2.4)

Para evaluar la eficiencia del método de la secante (4.1.2) en comparación con la familia uniparamétrica de métodos iterativos (4.2.4), es necesario conocer tanto los órdenes de convergencia R como el número de operaciones (productos y divisiones) requeridas en cada iteración de los algoritmos.

Consideramos que $F : \Omega \subseteq \mathbb{R}^m \to \mathbb{R}^m$, y utilizamos la diferencia dividida ([83]), la cual está representada por la matriz

$$[u, v; F] = ([u, v; F]_{ij})_{i,j=1}^{m} \in \mathcal{L}(\mathbb{R}^m, \mathbb{R}^m),$$

donde

$$[u, v; F]_{ij} = \frac{1}{u_j - v_j} \left(F_i(u_1, \ldots, u_{j-1}, u_j, v_{j+1}, \ldots, v_m) - F_i(u_1, \ldots, u_{j-1}, v_j, v_{j+1}, \ldots, v_m)\right),$$

si $u_j \neq v_j$, para $i, j = 1, 2, \ldots, m$, con

$$u = (u_1, u_2, \ldots, u_m)$$

y

$$v = (v_1, v_2, \ldots, v_m).$$

Para calcular x_{n+1} utilizando el método de la secante (4.1.2), se requieren

$$m^2 + 2m$$

operaciones para obtener la diferencia dividida $[u, v; F]$,

$$(m^3 - m)/3$$

para realizar la descomposición LU, y

$$m^2$$

operaciones para resolver dos sistemas triangulares. De este modo, el número total de operaciones es

$$p(m) = \frac{m}{3}\left(m^2 + 6m + 5\right),$$

y la eficiencia computacional, que se define como el cociente entre el orden de convergencia del método y el número de operaciones necesarias por iteración (ver [124, 155]), es

$$CE_{(4.1.2)} = \left(\frac{1 + \sqrt{5}}{2}\right)^{1/p(m)}.$$

Por otro lado, la familia uniparamétrica (4.2.4) requiere el mismo número de operaciones que el método de la secante (4.1.2), además de

$$4m$$

operaciones adicionales para calcular y_n y z_n. Por lo tanto, la eficiencia computacional de la familia es

$$CE_{(4.2.4)} = 2^{\frac{1}{p(m)+4m}}.$$

A continuación, se presenta una comparación de la eficiencia para diferentes valores de m en la Figura 4.1, donde se observa que la familia (4.2.4) es más eficiente que (4.1.2) y tienen una eficiencia computacional similar. Para este análisis, se emplea una escala logarítmica.

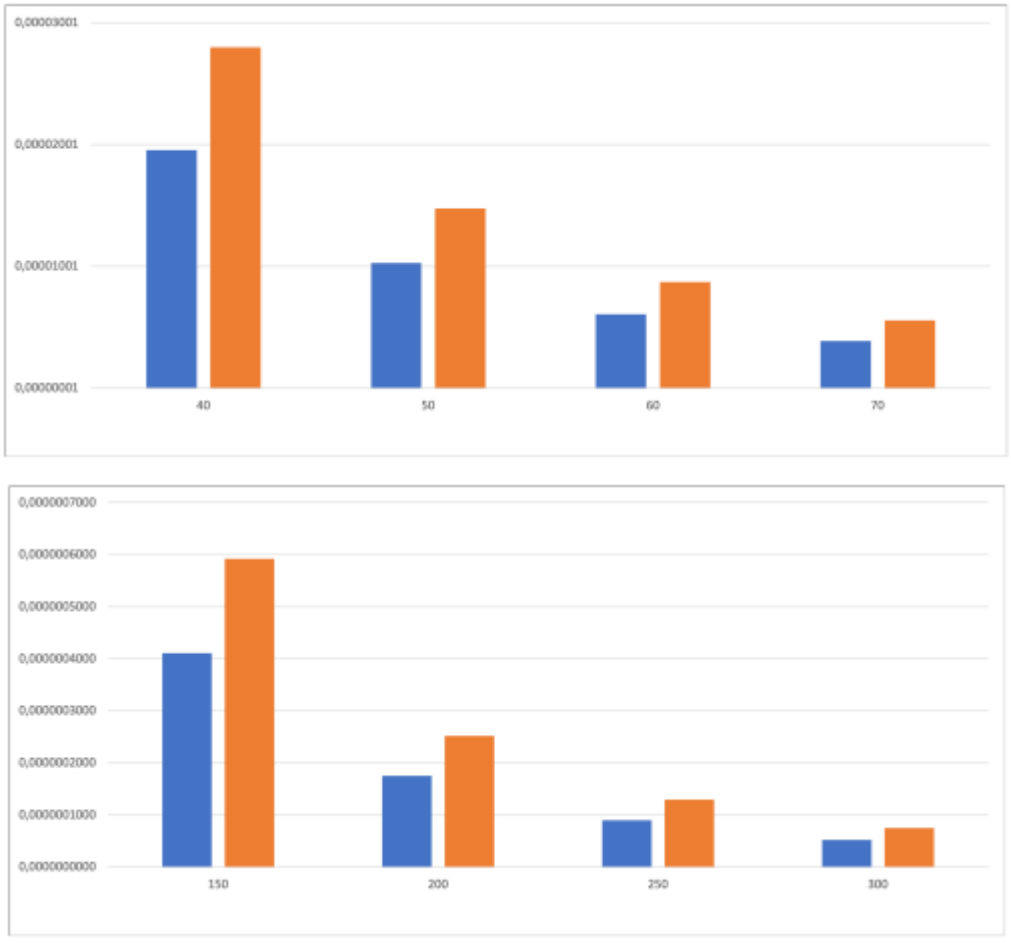


Figura 4.1: En la parte superior, eficiencias computacionales del método de la secante (4.1.2) (color azul) y la familia uniparamétrica (4.2.4) (color naranja) para diferentes problemas de tamaño pequeño, $m = 40$, 50, 60 y 70, (eje de abscisas) utilizando una escala logarítmica. En la parte inferior, eficiencias computacionales del método de la secante (4.1.2) (color azul) y la familia uniparamétrica (4.2.4) (color naranja) para diferentes problemas de tamaño grande, $m = 150$, 200, 250 y 300, (eje de abscisas) utilizando una escala logarítmica.

4.4. Análisis de la convergencia local

Nos centramos ahora en el estudio de la convergencia local de la familia (4.2.4). Como ya se ha comentado con anterioridad, el análisis de la convergencia local de un método iterativo se basa en establecer ciertas condiciones sobre la solución x^* y sobre el operador F, lo que permite definir una bola de convergencia. Esta bola indica la región en la cual las aproximaciones iniciales conducen a la solución x^*.

Para obtener resultados de convergencia local de la familia (4.2.4), es necesario imponer condiciones sobre la diferencia dividida de primer orden del operador F. En primer lugar, se asume que existe una diferencia dividida de primer orden $[u, v; F]$ de F en Ω.

En segundo lugar, se consideran las siguientes condiciones:

(LC1) Existen $x^* \in \Omega$ tal que $F(x^*) = 0$, junto con $\mu, \beta > 0$ y $\widetilde{x} \in \Omega$, donde

$$\|\widetilde{x} - x^*\| = \mu,$$

tales que $[x^*, \widetilde{x}; F]^{-1} \in \mathcal{L}(X, X)$ con

$$\|[x^*, \widetilde{x}; F]^{-1}\| \leq \beta.$$

(LC2) La diferencia dividida $[x, y; F]$ cumple

$$\|[x, y; F] - [u, v; F]\| \leq g(\|x - u\|, \|y - v\|), \quad x, y, u, v \in \Omega, \tag{4.4.1}$$

donde

$$g : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$$

es una función continua y no decreciente en ambos argumentos.

(LC3) La ecuación escalar

$$\beta\left((2-\theta)g(2(1-\theta)t, (3-2\theta)t) + \theta g_0(t, \mu + t)\right) = \theta, \tag{4.4.2}$$

donde g_0 está dada por

$$\|[p, q; F] - [x^*, \widetilde{x}; F]\| \leq g_0(\|p - x^*\|, \|q - \widetilde{x}\|) \tag{4.4.3}$$

y $g_0 : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$ es una función continua y no decreciente en sus dos argumentos, tiene al menos una raíz positiva, denotada por R, siendo R la menor de estas raíces.

(LC4) $B(x^*, R) \subseteq \Omega$.

Cabe señalar que si $g(0, 0) = 0$, el operador F es diferenciable (ver [85]). Sin embargo, si $g(0, 0) \neq 0$, entonces F es un operador no diferenciable. Por lo tanto, bajo la condición (4.4.1), se puede estudiar la convergencia local de la familia (4.2.4) tanto para operadores diferenciables como no diferenciables.

Considerando ahora F como un operador no diferenciable y tomando $\theta \in [0, 1)$, se observa que la condición (4.4.3) no es adicional a (4.4.1), ya que una diferencia

dividida que satisface (4.4.1) también satisface (4.4.3) para $x^*, \widetilde{x} \in \Omega$ y cualquier par de puntos distintos $z, w \in \Omega$.

El siguiente resultado se puede obtener fácilmente mediante el lema de Banach sobre operadores invertibles, dado que

$$\beta g_0(R, \mu + R) < 1,$$

lo cual es una consecuencia de (4.4.2).

Lema 4.1. *Bajo las condiciones (LC1)-(LC2)-(LC3)-(LC4), si $y, z \in B(x^*, R)$, con $y \neq z$, entonces existe $[y, z; F]^{-1}$ y cumple*

$$\|[y, z; F]^{-1}\| \leq \frac{\beta}{1 - \beta g_0(\|y - x^*\|, \|z - \widetilde{x}\|)} \leq \frac{\beta}{1 - \beta g_0(R, \mu + R)}. \tag{4.4.4}$$

A continuación, se escogen $x_{-1} \neq x_0$ y se considera que $x_n \neq x_{n+1}$, ya que, en caso contrario, $x^* = x_n = x_{n+1}$ y la sucesión (4.2.4) sería claramente convergente. Así, a partir de la formulación de la familia de métodos iterativos (4.2.5), se obtiene el siguiente resultado.

Lema 4.2. *Bajo las condiciones (LC1)-(LC2)-(LC3)-(LC4), si $x_{n-1}, x_{n-2} \in B(x^*, R)$, con $x_{n-1} \neq x_{n-2}$, entonces x_n está bien definido y*

$$\|x_n - x^*\| < \Psi(R)\|x_{n-1} - x^*\|,$$

para $n \geq 1$, donde

$$\Psi(R) = \frac{\beta g(2(1-\theta)R, (3-2\theta)R)}{1 - \beta g_0(R, \mu + R)}.$$

Demostración. Primero, existe

$$[y_{n-1}, z_{n-1}; F],$$

ya que

$$y_{n-1} - z_{n-1} = 2(1-\theta)(x_{n-2} - x_{n-1}) \neq 0$$

y $y_{n-1} \neq z_{n-1}$. A partir del Lema 4.1,

$$[y_{n-1}, z_{n-1}; F]$$

es invertible, cumpliendo

$$\| [y_{n-1}, z_{n-1}; F]^{-1} \| \leq \frac{\beta}{1 - \beta g_0(R, \mu + R)},$$

lo que garantiza que x_n está bien definido.

A partir de la ecuación (4.2.5) para la familia, se deduce que

$$\begin{aligned} x_n - x^* &= x_{n-1} - [y_{n-1}, z_{n-1}; F]^{-1} F(x_{n-1}) - x^* \\ &= [y_{n-1}, z_{n-1}; F]^{-1} \left([y_{n-1}, z_{n-1}; F] (x_{n-1} - x^*) - F(x_{n-1}) + F(x^*) \right) \\ &= [y_{n-1}, z_{n-1}; F]^{-1} \left([y_{n-1}, z_{n-1}; F] - [x_{n-1}, x^*; F] \right)(x_{n-1} - x^*). \end{aligned}$$

Tomando normas en la última igualdad y aplicando las condiciones (4.4.1) y (4.4.3), se obtiene que

$$\|x_n - x^*\| < \Psi(R)\|x_{n-1} - x^*\|.$$

□

Teorema 4.1. *Para $\theta \in [0,1)$, si se cumplen las condiciones (LC1)-(LC2)-(LC3)-(LC4), y $x_{-1}, x_0 \in B(x^*, R)$, con $x_{-1} \neq x_0$ y*

$$\|x_{-1} - x^*\| < \frac{R - (2-\theta)\gamma}{1-\theta},$$

donde $\gamma = \|x_0 - x^*\|$, entonces la sucesión (4.2.4) converge a x^* con $x_n \in B(x^*, R)$ y

$$\|x_n - x^*\| \leq \left(\frac{\theta}{2-\theta}\right)^n \|x_0 - x^*\|,$$

para $n \geq 1$.

Demostración. Primero, consideremos las siguientes estimaciones:

$$\begin{aligned}
\|y_0 - x^*\| &\leq \theta\|x_0 - x^*\| + (1-\theta)\|x_{-1} - x^*\| < R,\\
\|z_0 - x^*\| &\leq (2-\theta)\|x_0 - x^*\| + (1-\theta)\|x_{-1} - x^*\| < R,
\end{aligned}$$

lo que implica que $y_0, z_0 \in B(x^*, R) \subseteq \Omega$.

Además, dado que $\theta \in [0,1)$, se tiene

$$y_0 - z_0 = -2(1-\theta)(x_0 - x_{-1}) \neq 0.$$

Por lo tanto, y_0 y z_0 son puntos distintos en $B(x^*, R) \subseteq \Omega$, y, por el Lema 4.1, existe el inverso del operador$[y_0, z_0; F]^{-1}$.

En consecuencia, x_1 está bien definido y, por el Lema 4.2,

$$\|x_1 - x^*\| < \Psi(R)\|x_0 - x^*\| < R,$$

lo que implica que $x_1 \in B(x^*, R) \subseteq \Omega$.

Utilizando nuevamente (4.2.5) y el Lema 4.2, se deduce que

$$\|y_1 - x^*\| \leq \theta\|x_1 - x^*\| + (1-\theta)\|x_0 - x^*\| < R,$$

por lo que $y_1 \in B(x^*, R)$. Además,

$$\begin{aligned}
\|z_1 - x^*\| &\leq (2-\theta)\|x_1 - x^*\| + (1-\theta)\|x_0 - x^*\|\\
&\leq ((2-\theta)\Psi(R) + 1 - \theta)\|x_0 - x^*\|\\
&< ((2-\theta)\Psi(R) + 1 - \theta)R\\
&< R,
\end{aligned}$$

dado que $\Psi(R) < 1$. Así, $z_1 \in B(x^*, R)$.

Repitiendo el proceso para valores mayores de n, es posible demostrar por inducción que $x_n \in B(x^*, R)$ y que

$$\|x_n - x^*\| \leq \Psi(R)^n\|x_0 - x^*\| = \left(\frac{\theta}{2-\theta}\right)^n \|x_0 - x^*\|,$$

para $n \geq 1$, lo que prueba que la sucesión (4.2.4) converge a x^*, ya que $\Psi(R) < 1$. □

4.5. Ejemplos

En esta sección, se presentan tres ejemplos que complementan el estudio anterior de forma que se prueban los resultados teóricos presentados.

- En el primer ejemplo, se ilustra la aplicación del Teorema 4.1 y para ello se va a considerar una ecuación integral no lineal, en la que se calcularán los parámetros necesarios para poder aplicar el teorema.
- En el segundo ejemplo, se analiza la influencia del parámetro θ en las aproximaciones a las soluciones y se computarán los radios de convergencia asociados para diferentes valores del parámetro θ donde se observará que cuanto mayor es este valor mayor es el radio de la bola de convergencia.
- Finalmente, en el tercer ejemplo, se compara la familia (4.2.4) con otros métodos cuadráticos que también son aplicables a problemas no suaves. En particular, se considera el conocido método de Steffensen [7, 61], cuyo algoritmo es:

$$x_{n+1} = x_n - [x_n, x_n + F(x_n); F]^{-1} F(x_n), \quad n \geq 0, \quad \text{con } x_0 \text{ dado.} \quad (4.5.1)$$

y el método tipo Steffensen propuesto en [4], dado por:

$$x_{n+1} = x_n - \left[x_n - \frac{1}{2}F(x_n), x_n + \frac{1}{2}F(x_n); F\right]^{-1} F(x_n), \quad n \geq 0 \quad \text{con } x_0 \text{ dado.} \quad (4.5.2)$$

Ejemplo 4.1. *Consideremos la siguiente ecuación integral no lineal:*

$$x(s) = \int_0^1 \mathcal{G}(s,t)\left(\frac{1}{3}x(t)^2 + \frac{1}{2}|x(t)|\right) dt, \quad s \in [0,1], \quad (4.5.3)$$

donde el núcleo $\mathcal{G}(s,t)$ es la función de Green en $[0,1] \times [0,1]$.

Para resolver esta ecuación, primero se emplea un proceso de discretización que transforma (4.5.3) *en un problema de dimensión finita utilizando una cuadratura de Gauss-Legendre con* 8 *nodos:*

$$\int_0^1 d(t)\, dt = \sum_{i=1}^{8} w_i\, d(t_i),$$

donde los nodos t_i y los pesos w_i se determinan para $i = 1, 2, \ldots, 8$. Las aproximaciones de $x(t_i)$ se denotan por x_i, con $i = 1, 2, \ldots, 8$, de modo que la ecuación (4.5.3) *se convierte en el sistema de ecuaciones no lineales:*

$$x_i = \sum_{j=1}^{8} a_{ij}\left(\frac{1}{3}x_i^2 + \frac{1}{2}|x_i|\right), \quad j = 1, 2, \ldots, 8, \quad (4.5.4)$$

donde

$$a_{ij} = w_j\, \mathcal{G}(t_i, t_j) = \begin{cases} w_j\,(1 - t_i)\, t_j, & si\ j \leq i, \\ w_j\,(1 - t_j)\, t_i, & si\ j > i. \end{cases}$$

El sistema (4.5.4) *se puede escribir como:*

$$F(\mathbf{x}) \equiv \mathbf{x} - A\left(\frac{1}{3}\mathbf{u} + \frac{1}{2}\mathbf{v}\right) = 0, \qquad F : \mathbb{R}^8 \to \mathbb{R}^8, \qquad F = (F_1, F_2, \ldots, F_8),$$

donde

$$\mathbf{x} = (x_1, x_2, \ldots, x_8)^T,$$
$$A = (a_{ij})_{i,j=1}^{8},$$
$$\mathbf{u} = (x_1^2, x_2^2, \ldots, x_8^2)^T$$

y

$$\mathbf{v} = (|x_1|, |x_2|, \ldots, |x_8|)^T .$$

Así, F es no lineal y no diferenciable, y es evidente que $\mathbf{x}^* = \mathbf{0}$ es una solución del sistema (4.5.4).

A continuación, se calculan las bolas de convergencia para la familia de métodos iterativos (4.2.4) *según los valores de $\theta \in [0, 1]$.*

Utilizamos la diferencia dividida de primer orden

$$[\mathbf{c}, \mathbf{d}; F] = ([\mathbf{c}, \mathbf{d}; F]_{ij})_{i,j=1}^{8} ,$$

donde

$$[\mathbf{c}, \mathbf{d}; F]_{ij} = \frac{1}{c_j - d_j} \left(F_i(c_1, \ldots, c_{j-1}, c_j, d_{j+1}, \ldots, d_8) - F_i(c_1, \ldots, c_{j-1}, d_j, d_{j+1}, \ldots, d_8)\right),$$

para $i, j = 1, 2, \ldots, 8$, con

$$\mathbf{c} = (c_1, c_2, \ldots, c_8)^T$$

y

$$\mathbf{d} = (d_1, d_2, \ldots, d_8)^T.$$

Así,

$$[\mathbf{c}, \mathbf{d}; F] = I - A\left(diag\left\{\frac{1}{3}(c_i + d_i) + \frac{1}{2}\frac{|c_i| - |d_i|}{c_i - d_i}\right\}\right)_{i=1}^{8}.$$

y se tiene

$$\|[\mathbf{x}, \mathbf{y}; F] - [\mathbf{u}, \mathbf{v}; F]\| \leq \frac{1}{3}\|A\|(\|\mathbf{x} - \mathbf{u}\| + \|\mathbf{y} - \mathbf{v}\| + 3),$$

lo que permite obtener

$$g(y, z) = \frac{1}{3}\|A\|(y + z + 3).$$

A continuación, se toma

$$\widetilde{\mathbf{x}} = (1, 1, \ldots, 1)^T,$$

lo que implica que $\mu = 1$, ya que $\mathbf{x}^* = \mathbf{0}$. Además, se obtiene

$$\beta = 1.1129\ldots,$$

$$g_0(y, z) = g(y, z)$$

y la ecuación (4.4.2) *se simplifica a:*

$$\Big((0.1833\ldots)\theta^2 - (0.5042\ldots)\theta + (0.4583\ldots)\Big)\, t - (0.9541\ldots)\theta + (0.2750\ldots) = 0.$$

En la Tabla 4.1 se muestra la raíz positiva más pequeña R de la ecuación anterior para distintos valores de θ. Asimismo, se observa que la bola de convergencia $B(\mathbf{x}^*, R)$ mejora conforme aumenta θ. Cabe señalar que la ecuación (4.4.2) *no tiene raíces positivas si $\theta \leq 0.2$.*

θ	R
0.3	$0.0346\ldots$
0.4	$0.3728\ldots$
0.5	$0.8014\ldots$
0.6	$1.3408\ldots$
0.7	$2.0119\ldots$
0.8	$2.8331\ldots$
0.9	$3.8126\ldots$
1	$4.9386\ldots$

Tabla 4.1: Radio de la bola de convergencia $B(\mathbf{x}^*, R)$ en función de θ.

Para aplicar el Teorema 4.1, es necesario seleccionar convenientemente $\mathbf{x}_0$ y $\mathbf{x}_{-1}$ dentro de $B(\mathbf{x}^*, R)$. Por ejemplo, si se elige

$$\mathbf{x}_0 = (1, 1, \ldots, 1)^T,$$

entonces $\gamma = 1$ y $\mathbf{x}_{-1}$ debe cumplir

$$\|x_{-1} - x^*\| < \frac{R - (2-\theta)\gamma}{1-\theta}.$$

Si se toma $\theta = 0.7$, se tiene

$$R = 2.0119\ldots,$$

y la convergencia del método (4.2.4) *con $\theta = 0.7$ comenzando en $\mathbf{x}_0$ y cualquier $\mathbf{x}_{-1} \in B(\mathbf{x}^*, R)$ está garantizada.*

Ejemplo 4.2. *Consideremos la ecuación integral no lineal:*

$$x(s) = \frac{1}{2} + \int_0^1 \mathcal{G}(s,t)\left(x(t)^2 + \frac{3}{4}|x(t)|\right) dt, \quad s \in [0,1], \tag{4.5.5}$$

donde $\mathcal{G}(s,t)$ es la función de Green en $[0,1] \times [0,1]$.

Aplicando el mismo proceso de discretización que en el ejemplo anterior, la ecuación integral (4.5.5) *se transforma en el sistema no lineal:*

$$F(\mathbf{x}) \equiv \mathbf{x} - \mathbf{u} - A\left(\mathbf{v} + \frac{3}{4}\mathbf{w}\right) = 0, F : \mathbb{R}^8 \to \mathbb{R}^8, F = (F_1, F_2, \ldots, F_8),$$

donde

$$\mathbf{x} = (x_1, x_2, \ldots, x_8)^T,$$

$$\mathbf{u} = \left(\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2}\right)^T,$$
$$A = (a_{ij})_{i,j=1}^{8},$$
$$\mathbf{v} = (x_1^2, x_2^2, \ldots, x_8^2)^T$$

y

$$\mathbf{w} = (|x_1|, |x_2|, \ldots, |x_8|)^T.$$

Luego, se aplica el método (4.2.4) *comenzando con*

$$\mathbf{x}_{-1} = \left(-\frac{3}{10}, -\frac{3}{10}, \ldots, -\frac{3}{10}\right)^T$$

y

$$\mathbf{x}_0 = \left(-\frac{2}{5}, -\frac{2}{5}, \ldots, -\frac{2}{5}\right)^T$$

para aproximar una solución del sistema para diferentes valores de θ*. Los errores* $\|\mathbf{x}^* - \mathbf{x}_n\|$ *obtenidos en la Tabla 4.2, con el criterio de parada* $\|\mathbf{x}_n - \mathbf{x}_{n-1}\| < 10^{-24}$*, muestran que el método* (4.2.4) *ofrece mejores aproximaciones cuanto mayor es el valor de* θ*. Además, el orden de convergencia computacional de* (4.2.4) *es dos.*

n	$\theta = 0$	$\theta = 1/4$	$\theta = 1/2$
1	$1.9500\ldots \times 10^{-1}$	$1.9500\ldots \times 10^{-1}$	$1.9500\ldots \times 10^{-1}$
2	$1.2953\ldots \times 10^{-2}$	$1.0110\ldots \times 10^{-2}$	$4.3747\ldots \times 10^{-3}$
3	$1.9442\ldots \times 10^{-5}$	$1.1825\ldots \times 10^{-5}$	$2.2005\ldots \times 10^{-6}$
4	$4.3257\ldots \times 10^{-11}$	$1.5996\ldots \times 10^{-11}$	$5.5327\ldots \times 10^{-13}$
5	$2.1463\ldots \times 10^{-22}$	$2.9423\ldots \times 10^{-23}$	$--$

Tabla 4.2: Errores absolutos $\|\mathbf{x}^* - \mathbf{x}_n\|$ obtenidos con (4.2.4) y diferentes valores de θ.

Ejemplo 4.3. *Consideremos la siguiente ecuación integral no lineal:*

$$x(s) = \frac{7}{8}s - \frac{1}{2} + \int_0^1 st|x(t)|\,dt, \quad s \in [0,1]. \tag{4.5.6}$$

Se observa que la solución de esta ecuación es

$$x^*(s) = s - \frac{1}{2}.$$

Para aproximar la solución de (4.5.6) *utilizando* (4.2.4)*, primero es necesario verificar que el operador* $[y_n, z_n; F]^{-1}$ *se calcula en cada paso. Para ello, se considera:*

$$[y, z; F]\,\varphi(s) = \varphi(s) - s\int_0^1 th[y,z](t)\varphi(t)\,dt,$$

donde $F : \mathcal{C}[0,1] \to \mathcal{C}[0,1]$ *está dado por:*

$$[F(x)](s) = x_n(s) - \frac{7}{8}s + \frac{1}{2} - s\int_0^1 t|x_n(t)|\,dt$$

y

$$h[y,z](t)=\begin{cases}\dfrac{|y(t)|-|z(t)|}{y(t)-z(t)} & si\ t\in[0,1]\ y\ y(t)\neq z(t),\\ 0 & si\ t\in[0,1]\ y\ y(t)=z(t).\end{cases}$$

Así, se define $[y,z;F]:\mathcal{C}[0,1]\to\mathcal{C}[0,1]$ *mediante:*

$$[y,z;F](\varphi)(s)=\varphi(s)-s\int_0^1 th[y,z](t)\varphi(t)\,dt. \tag{4.5.7}$$

Si $y,z\in\mathcal{C}[0,1]$ *con* $y\neq z$*, entonces* $[y,z;F]\in\mathcal{L}\left(\mathcal{C}[0,1],\mathcal{C}[0,1]\right)$ *y*

$$[y,z;F](y-z)=F(y)-F(z).$$

Por lo tanto, $[y,z;F]$ *es una diferencia dividida de primer orden para el operador*

$$F:\mathcal{C}[0,1]\to\mathcal{C}[0,1].$$

A partir de esta diferencia dividida (4.5.7)*, se puede aplicar* (4.2.4) *para aproximar una solución de* (4.5.6)*.*

Si

$$[y_n,z_n;F]\,\varphi(s)=\varphi(s)-s\int_0^1 th[y_n,z_n](t)\varphi(t)\,dt=\nu(s),$$

entonces

$$\varphi(s)=[y_n,z_n;T]^{-1}\,\nu(s)=\nu(s)+s\,\mathcal{I}, \tag{4.5.8}$$

donde

$$\mathcal{I}=\int_0^1 th[y_n,z_n](t)\varphi(t)\,dt$$

es un número real. Para calcular $\mathcal{I}$*, se multiplica* (4.5.8) *por* $sh[y_n,z_n](s)$*, se integra entre* 0 *y* 1*, y se obtiene:*

$$\mathcal{I}=\int_0^1 sh[y_n,z_n](s)\nu(s)\,ds+\left(\int_0^1 s^2h[y_n,z_n](s)\,ds\right)\mathcal{I},$$

lo que resulta en:

$$\mathcal{I}=\frac{\int_0^1 sh[y_n,z_n](s)\nu(s)\,ds}{1-\int_0^1 s^2h[y_n,z_n](s)\,ds}.$$

Finalmente, se obtiene:

$$[y_n,z_n;F]^{-1}\,\nu(s)=\nu(s)+s\,\frac{\int_0^1 sh[y_n,z_n](s)\nu(s)\,ds}{1-\int_0^1 s^2h[y_n,z_n](s)\,ds}.$$

El algoritmo para aplicar (4.2.4) *a partir de*

$$x_{-1}(s),x_0(s)\in\mathcal{C}[0,1]$$

es el siguiente:

1. *Calcular*

$$F(x_n)(s)=x_n(s)-\frac{7}{8}s+\frac{1}{2}-s\int_0^1 t|x_n(t)|\,dt.$$

2. Calcular

$$y_n = x_n - (1-\theta)(x_n - x_{n-1})z_n = x_n + (1-\theta)(x_n - x_{n-1}).$$

3. Calcular $h[y_n, z_n](s)$,

$$\int_0^1 s^2 h[y_n, z_n](s)\, ds \int_0^1 sh[y_n, z_n](s)T(x_n)(s)\, ds.$$

4. Calcular

$$P_n = \frac{\int_0^1 sh[y_n, z_n](s)T(x_n)(s)\, ds}{1 - \int_0^1 s^2 h[y_n, z_n](s)\, ds}.$$

5. Calcular

$$x_{n+1}(s) = x_n(s) - T(x_n)(s) - sP_n.$$

Los métodos iterativos (4.5.1) *y* (4.5.2) *se aplican de manera similar al método* (4.2.4)*, ya que sus algoritmos son comparables.*

En la Tabla 4.3, se observa que la solución

$$x^*(s) = s - \frac{1}{2}$$

se aproxima después de cuatro iteraciones con los tres métodos cuadráticos, utilizando el criterio de parada

$$\|x^*(s) - x_n(s)\| \leq 10^{-32},$$

$n \geq 0$*, y comenzando en*

$$x_{-1}(s) = s + \frac{9}{10}$$

y

$$x_0(s) = s + 1.$$

n	Método (4.5.1)	Método (4.5.2)	Método (4.2.4) con $\theta = \frac{9}{10}$
0	1.5	1.5	1.5
1	$4.7395\ldots \times 10^{-2}$	$4.7395\ldots \times 10^{-2}$	$4.3747\ldots \times 10^{-2}$
2	$8.5089\ldots \times 10^{-4}$	$5.1825\ldots \times 10^{-4}$	$1.2617\ldots \times 10^{-4}$
3	$3.3368\ldots \times 10^{-21}$	$2.7806\ldots \times 10^{-21}$	$3.4538\ldots \times 10^{-22}$

Tabla 4.3: Errores absolutos $\|x^*(s) - x_n(s)\|$ del Ejemplo 4.3.

Se observa que los errores obtenidos con los tres métodos son del mismo orden, lo cual es esperado dado que los tres métodos presentan el mismo orden de convergencia, dos, aunque los errores obtenidos con el método (4.2.4) *son ligeramente mejores.*

4.6. Análisis dinámico comparativo entre el método de la secante y la familia uniparamétrica (4.2.4)

En esta sección, se estudian las accesibilidades y el comportamiento dinámico tanto del método de la secante como de la familia uniparamétrica (4.2.4). Para este fin, se analiza la dinámica compleja en 4.6.1 y 4.6.3, utilizando un enfoque similar al de estudios previos como [43, 77, 105, 106]. En estos experimentos, se libera el parámetro x_0, fijando $x_{-1} = x_0 - 0.1$, con una tolerancia de 10^{-15} y un máximo de 10 iteraciones en 4.6.1 y de 10 o 20 en 4.6.3. Por otro lado, también se va a realizar un análisis dinámico real en las secciones 4.6.5 y 4.6.7, para realizarlo se van a dejar libres tanto x_0 como x_{-1}.

4.6.1. Análisis dinámico complejo de los métodos aplicados a $F(z) = z^3 + z|z| - 2z = 0$.

En esta subsección se comparan las accesibilidades del método de la secante y la familia (4.2.4) al resolver la ecuación que se lleva estudiando durante toda la tesis

$$F(z) = z^3 + z|z| - 2z = 0,$$

cuyas soluciones son $z^* = -1$, $z^{**} = 0$ y $z^{***} = 1$.

Las regiones de convergencia a $z^* = -1$, $z^{**} = 0$ y $z^{***} = 1$ se representan en verde, azul y rojo, respectivamente.

En la Figura 4.2 se muestra el comportamiento dinámico del método de la secante, mientras que las Figuras 4.3 y 4.6 presentan el comportamiento de la familia (4.2.4) para diferentes valores del parámetro θ.

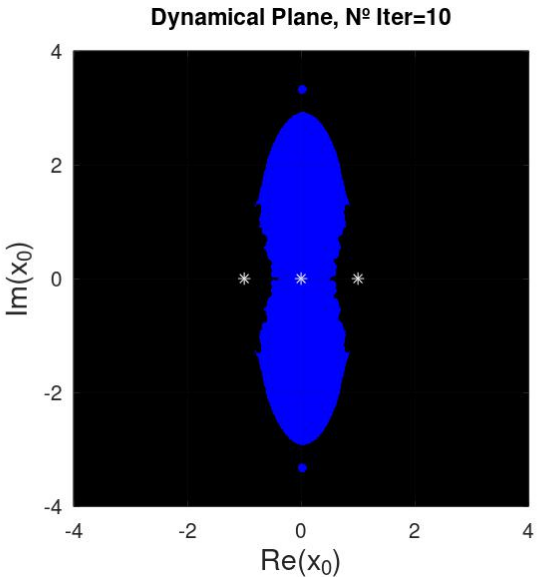


Figura 4.2: Cuenca de atracción obtenida con el método de la secante

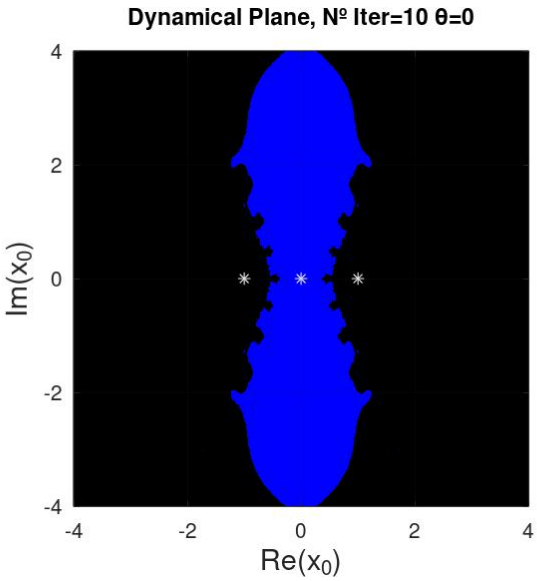


Figura 4.3: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0$ (Método de Kurchatov)

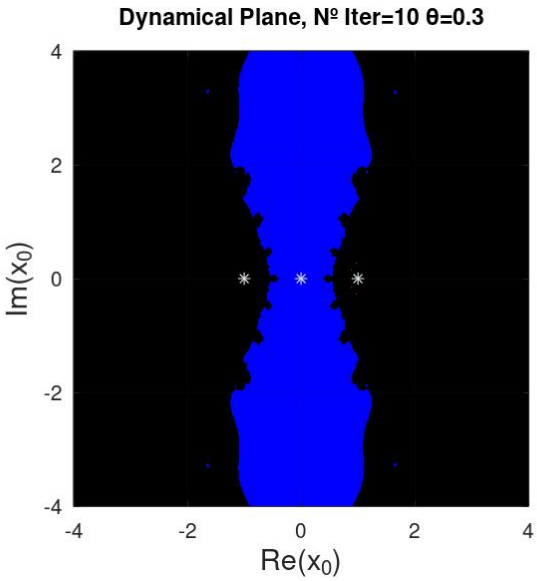


Figura 4.4: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0.3$

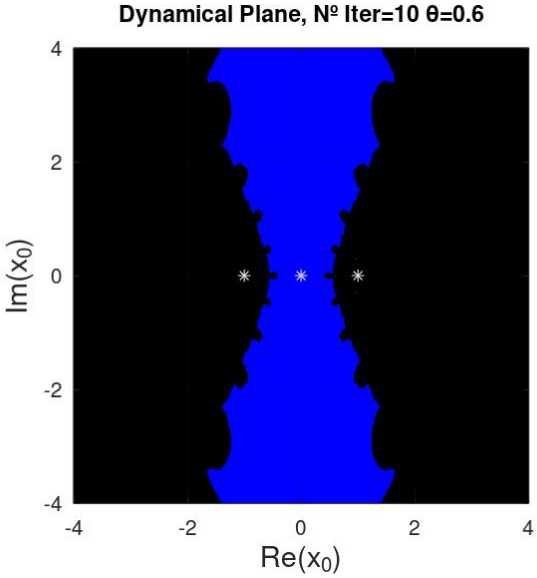


Figura 4.5: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0.6$

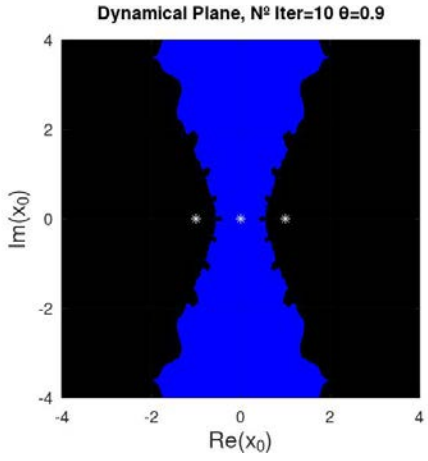


Figura 4.6: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0.9$

A partir de los gráficos anteriores donde se puede ver la comparativa de las tres figuras para diferentes valores del parámetro θ, se intuye que cuanto mayor sea el valor de dicho parámetro, mayor es el porcentaje de puntos que converge a la raíz 0, tomando

$$x_0 = a + b * I$$

libre, y fijando

$$x_{-1} = x_0 - 0.1,$$

con

$$tolerancia = 10^{-15}$$

y un máximo de

$$N = 10$$

iteraciones.

Para poder corroborar estos datos, se procede a un análisis numérico del comportamiento de las accesibilidades.El porcentaje de puntos que garantizan la convergencia hacia alguna de las raíces se resume en la Tabla 4.4, donde se observa que la accesibilidad del método de la secante es inferior a la de la familia uniparamétrica (4.2.4). Asimismo, se observa que el método de Kurchatov presenta una accesibilidad menor en comparación con la familia uniparamétrica.

Método o valor de θ	Porcentaje de puntos de convergencia
Método de la secante	11.21 %
Método de Kurchatov ($\theta = 0$)	20.18 %
$\theta = 0.3$	23.69 %
$\theta = 0.6$	26.98 %
$\theta = 0.9$	29.28 %

Tabla 4.4: Porcentaje de puntos que garantizan la convergencia del método de la secante y algunos métodos de la familia (4.2.4) tras 10 iteraciones.

A continuación, se adjuntan los códigos de Python para que puedan ser fácilmente replicados.

4.6.2. Familia (4.2.4) en el caso de dinámica compleja

```
# -*- coding: utf-8 -*-

#Metodo secante
# Plano dinamico  x^3+x*abs(x)-2*x


import numpy as np
import matplotlib.pyplot as plt
import time

#Metodo
F= lambda X: X**3-X*np.abs(X)-2*X
DF= lambda X,Y: ((F(X)-F(Y)))/(X-Y) #Aproximacion derivada,
    diferencia dividida
M=lambda X,Y: X-F(X)/DF(X,Y)


#  Rez0, parte real de los puntos del plano dinamico
#  Imz0, parte imaginaria de los puntos del plano dinamico
#  maxiter, numero maximo de iteraciones
#  Z, valor final de la orbita de cada punto
#  I, imagen del plano dinamico
#  tol, valor de tolerancia establecido
#  h, tamano de paso del mallado
tic=time.time() #Inicio tiempo de computacion
h=0.01      #Tamano de paso del mallado
tol=1E-15    #Tolerancia de aproximacion
maxiter=20 #Numero maximo iteraciones

Rez0=np.arange(-5,5+h,h)
Imz0=np.arange(-5,5+h,h)
ReZ0, ImZ0=np.meshgrid(Rez0,Imz0, indexing='xy')
Z0=ReZ0+1j*ImZ0
Z_1=Z0-0.1

itera=1 #Inicializacion parametro para iteraciones
#  R G B para los colores del plano dinamico
#  I, para combinar R G B
R=np.zeros([len(Rez0),len(Imz0)])
G=np.zeros([len(Rez0),len(Imz0)])
B=np.zeros([len(Rez0),len(Imz0)])
I=np.zeros([len(Rez0),len(Imz0),3])
IT=(maxiter+1)*np.ones([len(Rez0),len(Imz0)])
#  Puntos fijos
ZF1=-1
ZF2=0
ZF3=1


while itera<=maxiter:
    Z=M(Z0,Z_1)

    for fil in range(len(Rez0)):
        for col in range(len(Imz0)):
            Zfc=Z[fil,col]
            if np.abs(Zfc-ZF1)<tol:
```

```
                R[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
            if np.abs(Zfc-ZF2)<tol:
                G[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
            if np.abs(Zfc-ZF3)<tol:
                B[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
    #linalg.norm
    #Actualizacion de valores
    itera=itera+1
    Z_1=Z0
    Z0=Z

#Generacion de la imagen del plano
I[...,0]=R
I[...,1]=G
I[...,2]=B

# Se muestra el plano dinamico
plt.figure()
plt.title('Dynamical Plane, Num Iter=%s' %(str(maxiter)))
plt.xlabel('Re(x0)')
plt.ylabel('Im(x0)')
plt.grid()
#Representacion puntos fijos
plt.plot(np.real(ZF1),np.imag(ZF1),'w*')
plt.plot(np.real(ZF2),np.imag(ZF2),'w*')
plt.plot(np.real(ZF3),np.imag(ZF3),'w*')
#Se indica origin='lower' para indicar que el origen esta en la
    esquina
#Inferior izquieda
plt.imshow(I,origin='lower', extent=[Rez0[0],Rez0[-1],Imz0[0],Imz0
    [-1]])

#Numero de iteraciones promedio
print('% de iteraciones que convergen: ', np.size(IT[IT<=maxiter])/np
    .size(IT)*100)
#Valor medio de iteraciones para converger
print('Valor medio de iteraciones para converger: ',np.mean(IT[IT<=
    maxiter]))
#Tiempo de computacion
toc=time.time()
print('Tiempo de computacion:', toc-tic,' segundos')
```

4.6.3. Análisis dinámico complejo de los métodos aplicados a $F(z) = z^3 - z|z| - 2z = 0$.

En esta subsección se comparan las accesibilidades del método de la secante y la familia (4.2.4) al resolver la ecuación que se lleva estudiando durante toda la tesis

$$F(z) = z^3 - z|z| - 2z = 0,$$

cuyas soluciones son $z^* = -2$, $z^{**} = 0$ y $z^{***} = 2$. Las regiones de convergencia a $z^* = -2$, $z^{**} = 0$ y $z^{***} = 2$ se representan en verde, azul y rojo, respectivamente. En las Figuras 4.7- 4.8 se muestra el comportamiento dinámico del método de la secante, mientras que las Figuras 4.7– 4.16 presentan el comportamiento de la familia (4.2.4) para diferentes valores del parámetro θ.

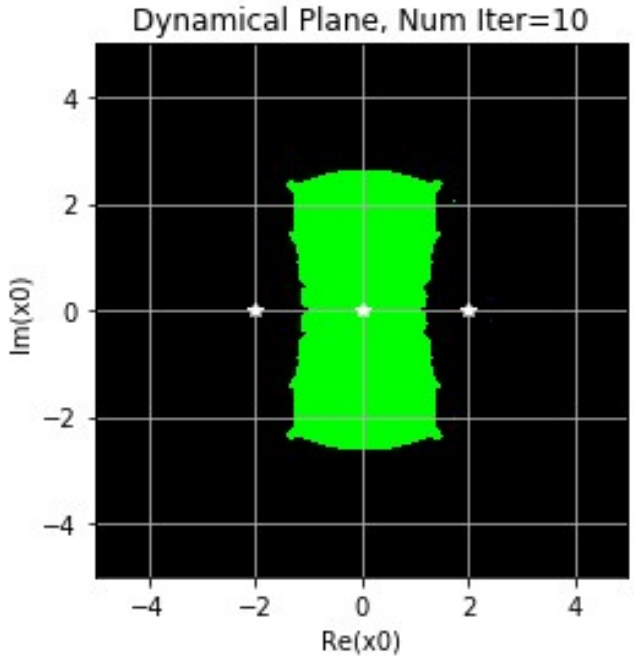


Figura 4.7: Cuenca de atracción obtenida con el método de la secante con 10 iteraciones

A partir de los gráficos anteriores donde se puede ver la comparativa de las tres figuras para diferentes valores del parámetro θ, se intuye que cuanto mayor sea el valor de dicho parámetro, mayor es el porcentaje de puntos que converge a la raíz 0.

Para poder corroborar estos datos, se procede a un análisis numérico del comportamiento de las accesibilidades.El porcentaje de puntos que garantizan la convergencia hacia alguna de las raíces se resume en las Tablas 4.5– 4.6, donde se observa que la accesibilidad del método de la secante es inferior a la de la familia uniparamétrica (4.2.4). Asimismo, se observa que el método de Kurchatov presenta una accesibilidad menor en comparación con la familia uniparamétrica.

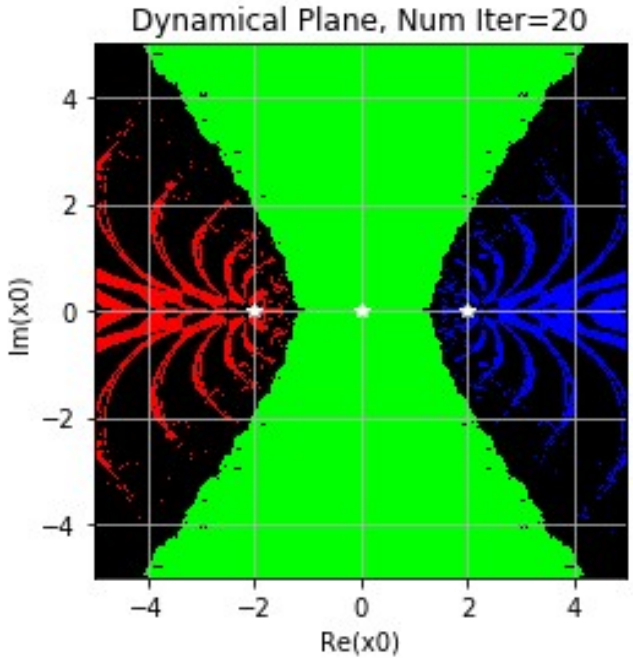


Figura 4.8: Cuenca de atracción obtenida con el método de la secante con 20 iteraciones

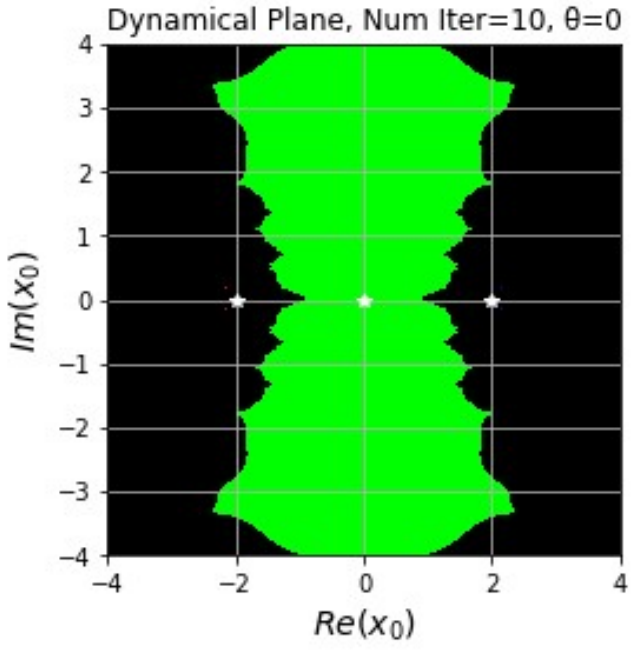


Figura 4.9: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0$ (Método de Kurchatov) con un máximo de 10 iteraciones.

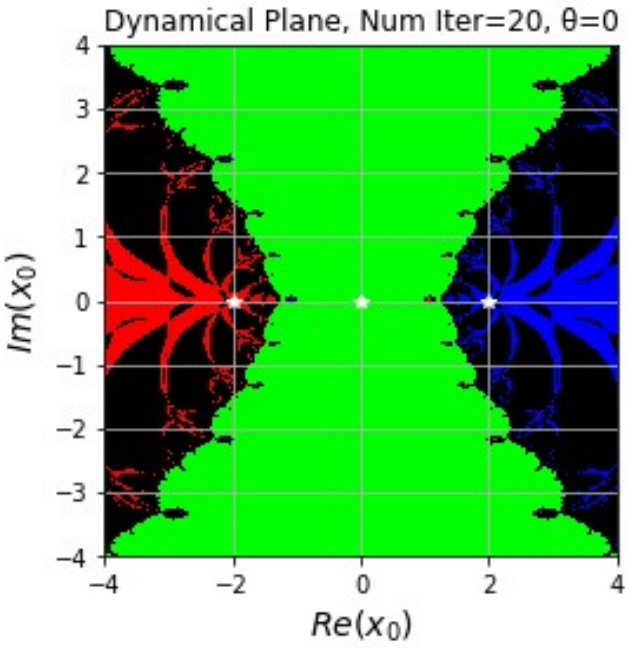


Figura 4.10: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0$ (Método de Kurchatov) con un máximo de 20 iteraciones.

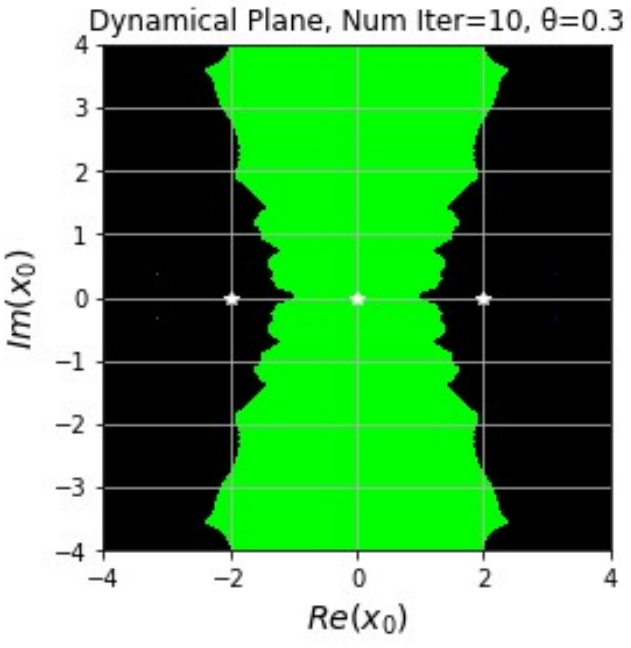


Figura 4.11: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0.3$ con un máximo de 10 iteraciones.

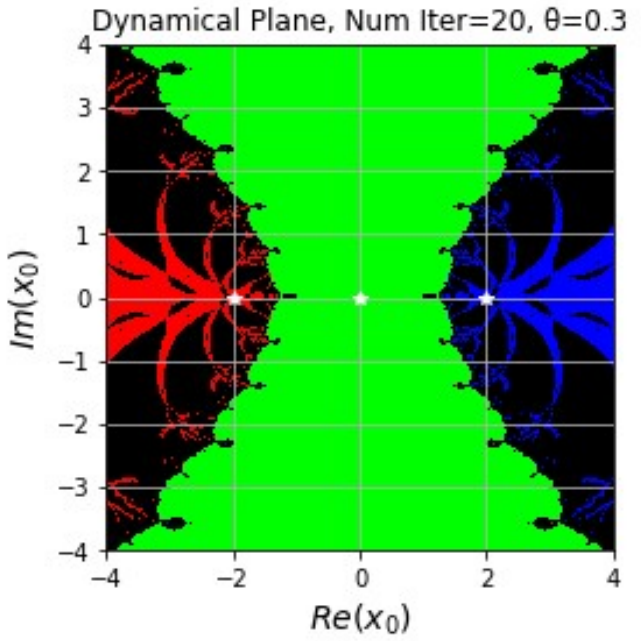


Figura 4.12: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0.3$ con un máximo de 20 iteraciones.

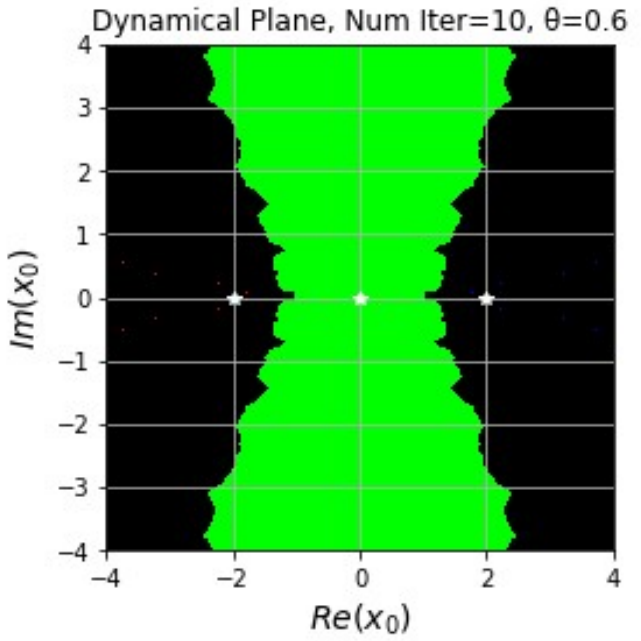


Figura 4.13: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0.6$ con un máximo de 10 iteraciones.

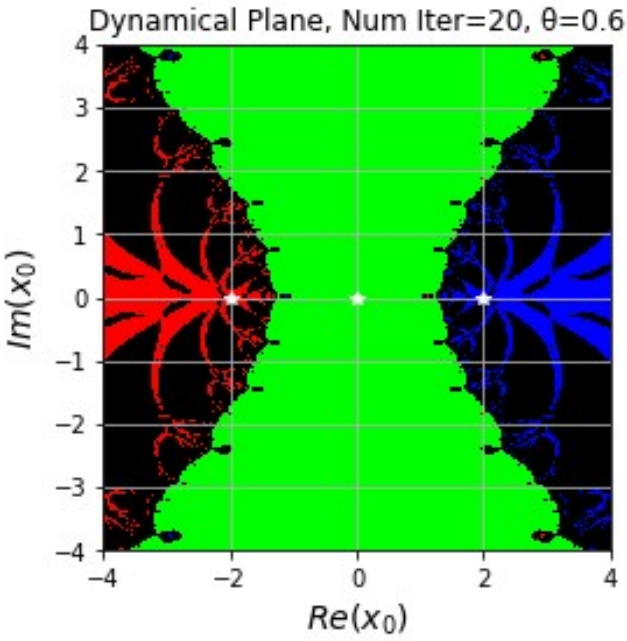


Figura 4.14: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0.6$ con un máximo de 20 iteraciones.

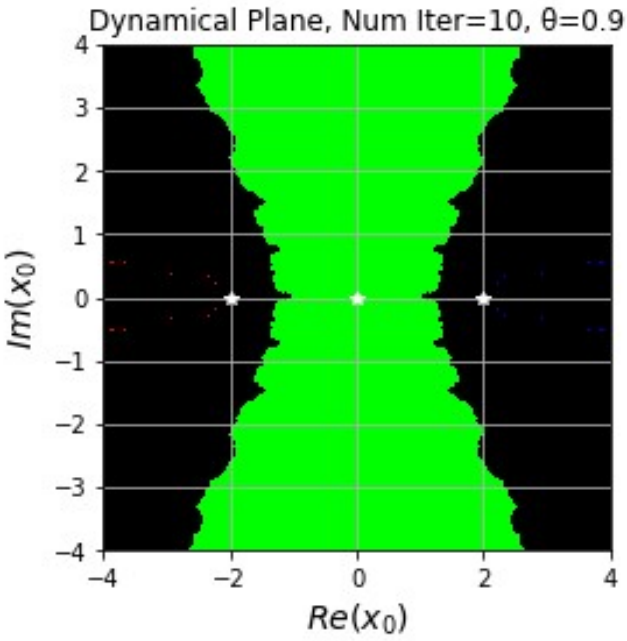


Figura 4.15: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0.9$ con un máximo de 10 iteraciones.

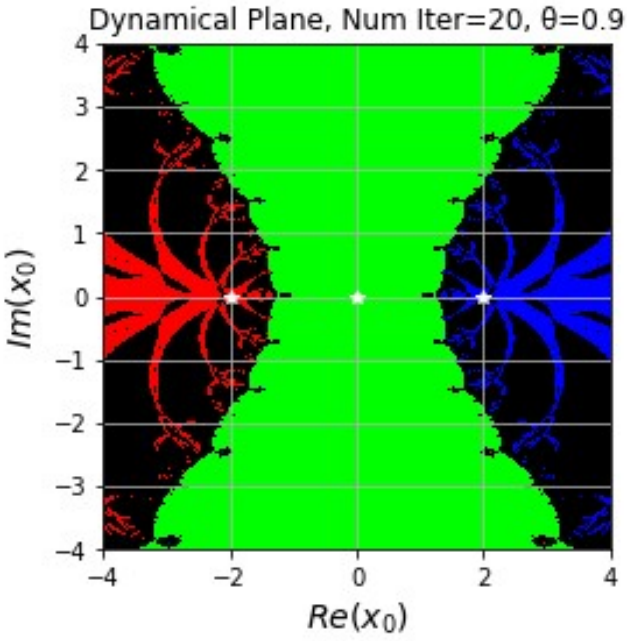


Figura 4.16: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0.9$ con un máximo de 20 iteraciones.

Método o valor de θ	Porcentaje de puntos de convergencia
Método de la secante	13.11 %
Método de Kurchatov ($\theta = 0$)	42.31 %
$\theta = 0.3$	44.41 %
$\theta = 0.6$	45.40 %
$\theta = 0.9$	46.70 %

Tabla 4.5: Porcentaje de puntos que garantizan la convergencia del método de la secante y algunos métodos de la familia (4.2.4) tras 10 iteraciones.

Método o valor de θ	Porcentaje de puntos de convergencia
Método de la secante	58.07 %
Método de Kurchatov ($\theta = 0$)	68.89 %
$\theta = 0.3$	66.38 %
$\theta = 0.6$	64.46 %
$\theta = 0.9$	63.65 %

Tabla 4.6: Porcentaje de puntos que garantizan la convergencia del método de la secante y algunos métodos de la familia (4.2.4) tras 20 iteraciones.

A continuación, se adjuntan los códigos de Python para que puedan ser fácilmente replicados.

4.6.4. Familia (4.2.4) en el caso de dinámica compleja

```
# -*- coding: utf-8 -*-

#Metodo secante
# Plano dinamico  x^3+x*abs(x)-2*x


import numpy as np
import matplotlib.pyplot as plt
import time

#Metodo
F= lambda X: X**3-X*np.abs(X)-2*X
DF= lambda X,Y: ((F(X)-F(Y)))/(X-Y) #Aproximacion derivada,
    diferencia dividida
M=lambda X,Y: X-F(X)/DF(X,Y)


#  Rez0, parte real de los puntos del plano dinamico
#  Imz0, parte imaginaria de los puntos del plano dinamico
#  maxiter, numero maximo de iteraciones
#  Z, valor final de la orbita de cada punto
#  I, imagen del plano dinamico
#  tol, valor de tolerancia establecido
#  h, tamano de paso del mallado
tic=time.time() #Inicio tiempo de computacion
h=0.01      #Tamano de paso del mallado
tol=1E-15    #Tolerancia de aproximacion
maxiter=20 #Numero maximo iteraciones

Rez0=np.arange(-5,5+h,h)
Imz0=np.arange(-5,5+h,h)
ReZ0, ImZ0=np.meshgrid(Rez0,Imz0, indexing='xy')
Z0=ReZ0+1j*ImZ0
Z_1=Z0-0.1

itera=1 #Inicializacion parametro para iteraciones
#  R G B para los colores del plano dinamico
#  I, para combinar R G B
R=np.zeros([len(Rez0),len(Imz0)])
G=np.zeros([len(Rez0),len(Imz0)])
B=np.zeros([len(Rez0),len(Imz0)])
I=np.zeros([len(Rez0),len(Imz0),3])
IT=(maxiter+1)*np.ones([len(Rez0),len(Imz0)])
#  Puntos fijos
ZF1=-1
ZF2=0
ZF3=1


while itera<=maxiter:
    Z=M(Z0,Z_1)

    for fil in range(len(Rez0)):
        for col in range(len(Imz0)):
            Zfc=Z[fil,col]
            if np.abs(Zfc-ZF1)<tol:
```

```
                R[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
            if np.abs(Zfc-ZF2)<tol:
                G[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
            if np.abs(Zfc-ZF3)<tol:
                B[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
    #linalg.norm
    #Actualizacion de valores
    itera=itera+1
    Z_1=Z0
    Z0=Z

#Generacion de la imagen del plano
I[...,0]=R
I[...,1]=G
I[...,2]=B

# Se muestra el plano dinamico
plt.figure()
plt.title('Dynamical Plane, Num Iter=%s' %(str(maxiter)))
plt.xlabel('Re(x0)')
plt.ylabel('Im(x0)')
plt.grid()
#Representacion puntos fijos
plt.plot(np.real(ZF1),np.imag(ZF1),'w*')
plt.plot(np.real(ZF2),np.imag(ZF2),'w*')
plt.plot(np.real(ZF3),np.imag(ZF3),'w*')
#Se indica origin='lower' para indicar que el origen esta en la
    esquina
#Inferior izquieda
plt.imshow(I,origin='lower', extent=[Rez0[0],Rez0[-1],Imz0[0],Imz0
    [-1]])

#Numero de iteraciones promedio
print('% de iteraciones que convergen: ', np.size(IT[IT<=maxiter])/np
    .size(IT)*100)
#Valor medio de iteraciones para converger
print('Valor medio de iteraciones para converger: ',np.mean(IT[IT<=
    maxiter]))
#Tiempo de computacion
toc=time.time()
print('Tiempo de computacion:', toc-tic,' segundos')
```

4.6.5. Análisis dinámico real de los métodos aplicados a $F(x) = x^3 + x|x| - 2x = 0$.

En esta subsección se comparan las accesibilidades del método de la secante y la familia (4.2.4) al resolver la ecuación que se lleva estudiando durante toda la tesis

$$F(x) = x^3 + x|z| - 2x = 0,$$

cuyas soluciones son

$$x^* = -1,$$

$$x^{**} = 0$$

y

$$x^{***} = 1.$$

Las regiones de convergencia a $x^* = -1$, $x^{**} = 0$ y $x^{***} = 1$ se representan en verde, azul y rojo, respectivamente.

En las Figuras 4.17- 4.18 se muestra el comportamiento dinámico del método de la secante, mientras que las Figuras 4.7– 4.16 presentan el comportamiento de la familia (4.2.4) para diferentes valores del parámetro θ.

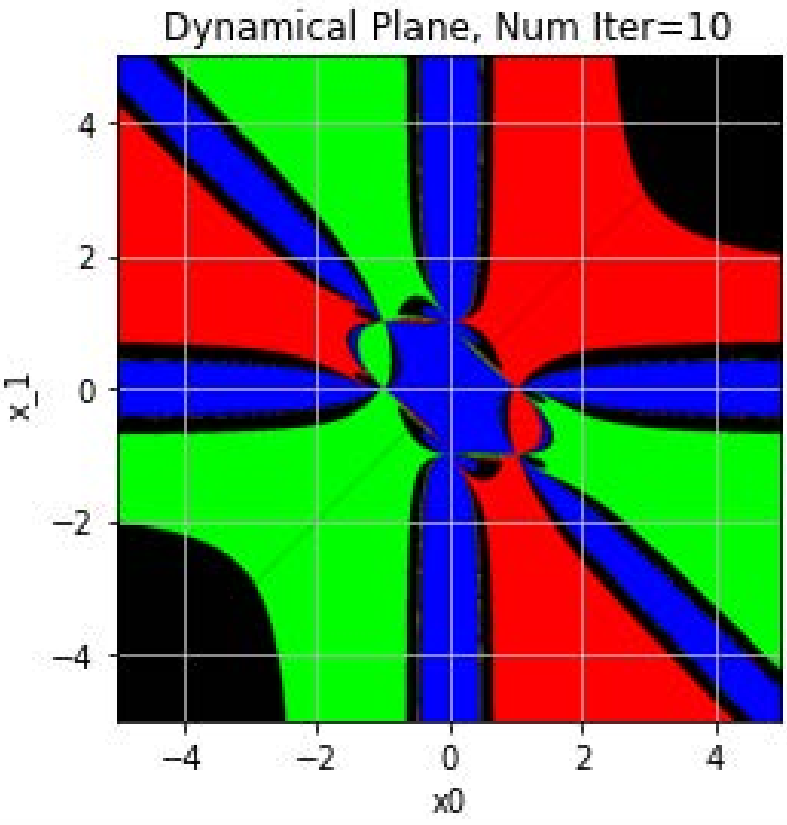


Figura 4.17: Cuenca de atracción obtenida con el método de la secante con 10 iteraciones

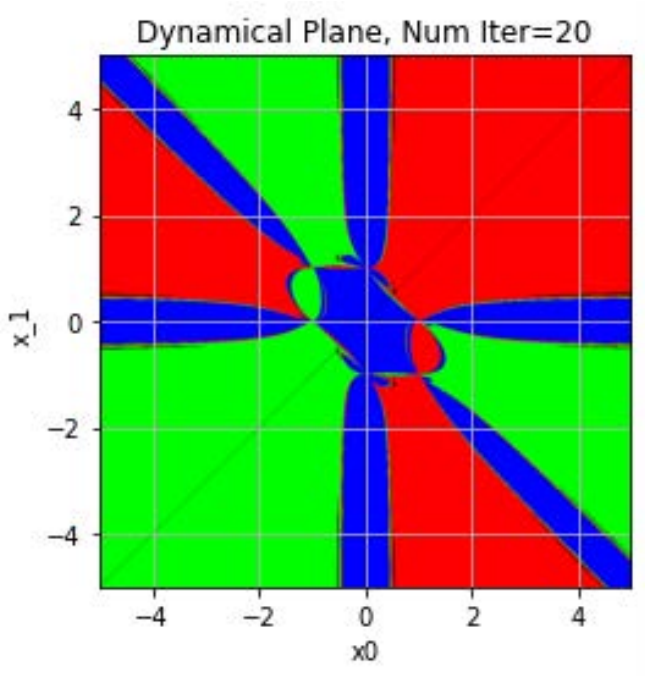


Figura 4.18: Cuenca de atracción obtenida con el método de la secante con 20 iteraciones

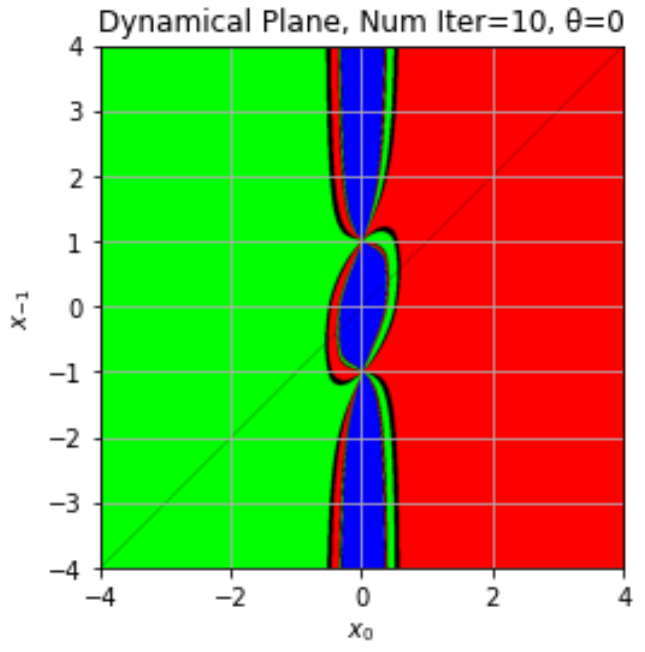


Figura 4.19: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0$ (Método de Kurchatov) con un máximo de 10 iteraciones.

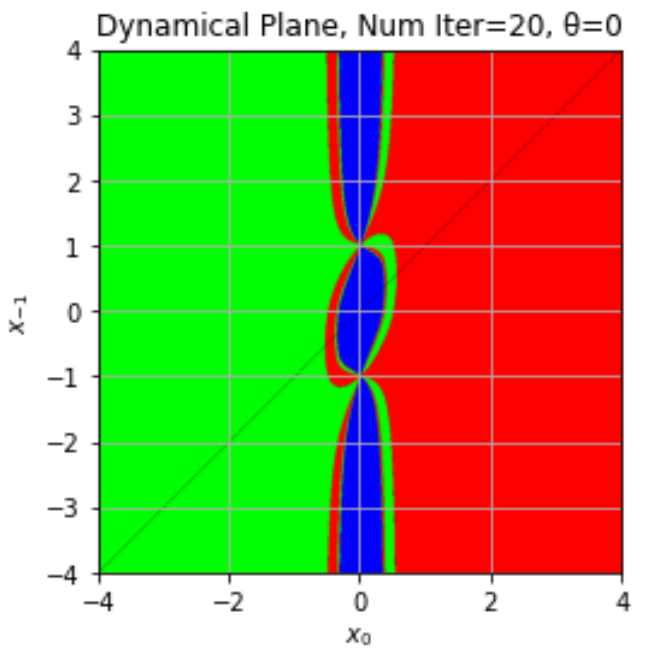


Figura 4.20: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0$ (Método de Kurchatov) con un máximo de 20 iteraciones.

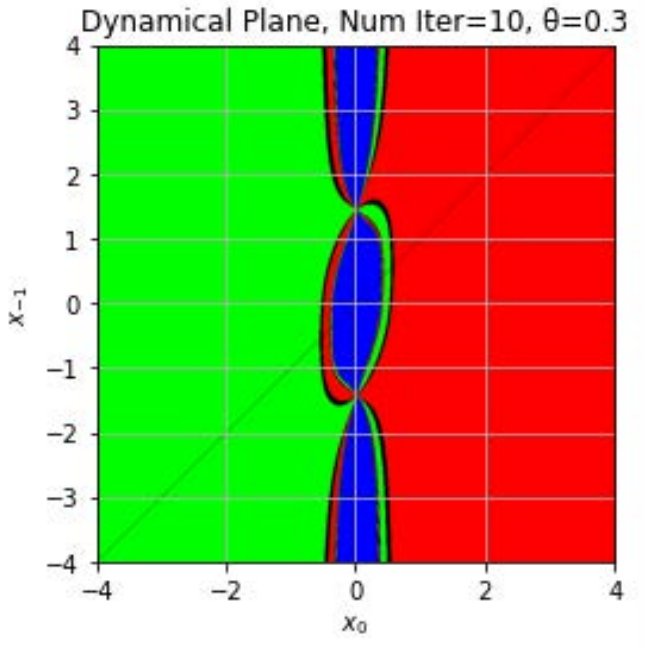


Figura 4.21: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0.3$ con un máximo de 10 iteraciones.

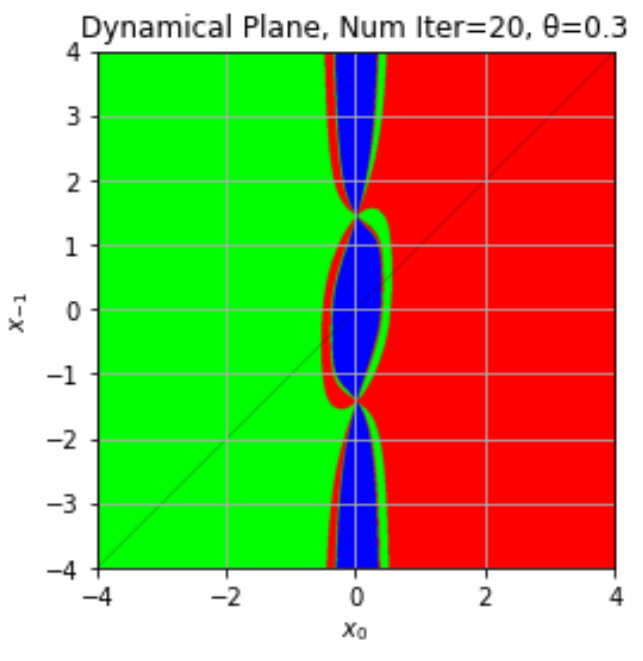


Figura 4.22: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0.3$ con un máximo de 20 iteraciones.

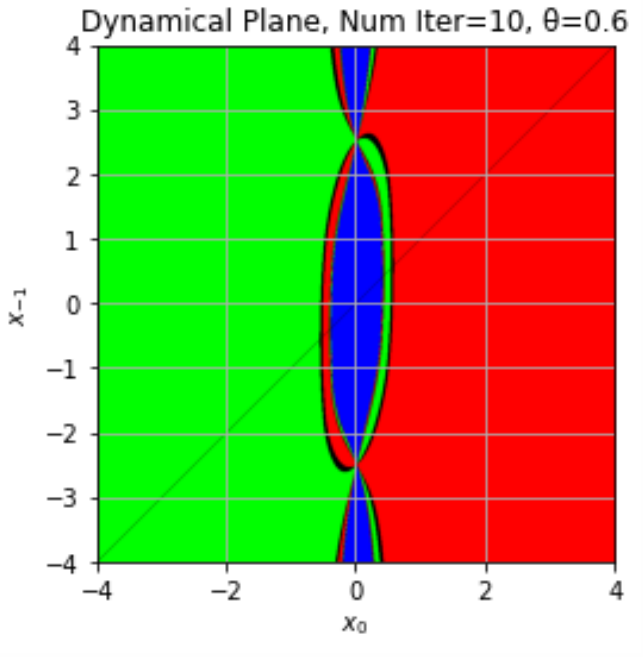


Figura 4.23: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0.6$ con un máximo de 10 iteraciones.

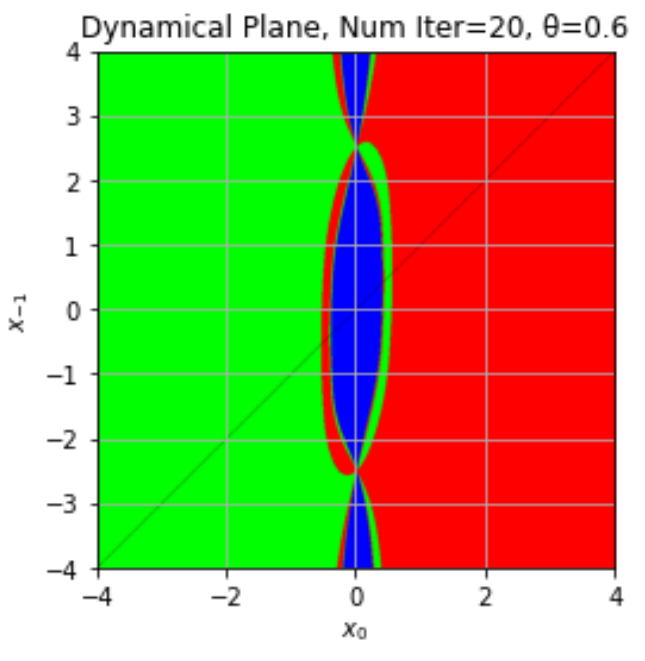


Figura 4.24: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0.6$ con un máximo de 20 iteraciones.

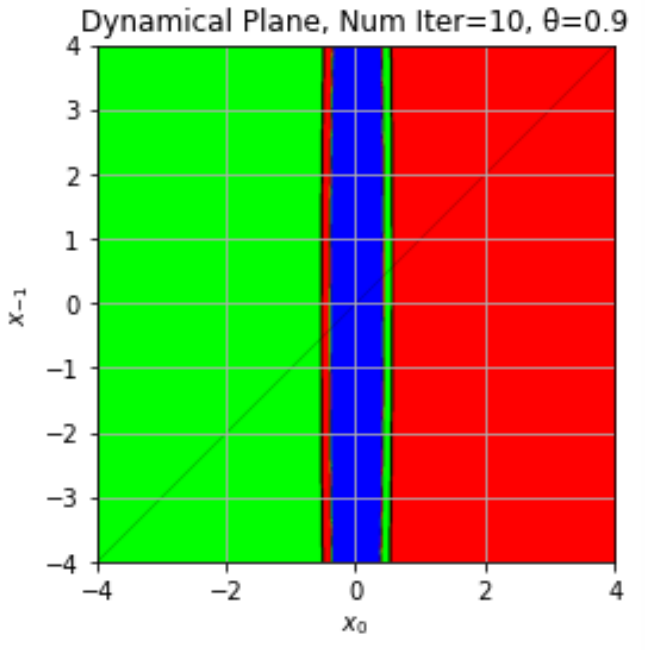


Figura 4.25: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0.9$ con un máximo de 10 iteraciones.

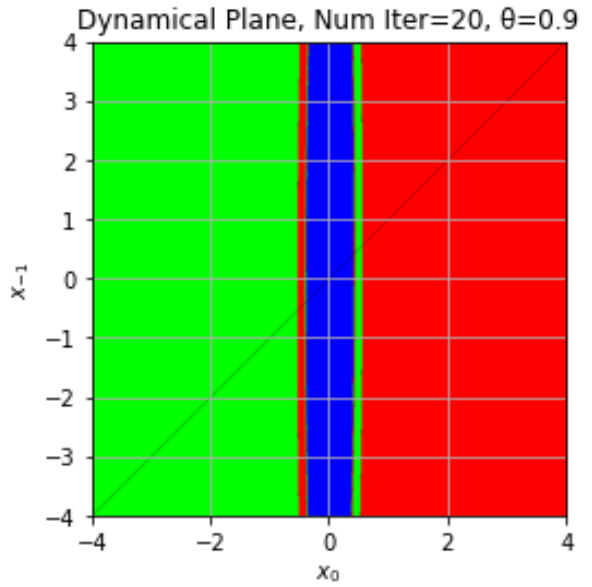


Figura 4.26: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0.9$ con un máximo de 20 iteraciones.

De nuevo, el porcentaje de puntos que garantizan la convergencia hacia alguna de las raíces se resume en las Tablas 4.7– 4.8, donde se observa que la accesibilidad del método de la secante es inferior a la de la familia uniparamétrica (4.2.4).

Método o valor de θ	Porcentaje de puntos de convergencia
Método de la secante	78.88 %
Método de Kurchatov ($\theta = 0$)	97.72 %
$\theta = 0.3$	98.11 %
$\theta = 0.6$	98.48 %
$\theta = 0.9$	98.66 %

Tabla 4.7: Porcentaje de puntos que garantizan la convergencia del método de la secante y algunos métodos de la familia (4.2.4) tras 10 iteraciones.

Método o valor de θ	Porcentaje de puntos de convergencia
Método de la secante	99.53 %
Método de Kurchatov ($\theta = 0$)	99.83 %
$\theta = 0.3$	99.85 %
$\theta = 0.6$	99.85 %
$\theta = 0.9$	99.86 %

Tabla 4.8: Porcentaje de puntos que garantizan la convergencia del método de la secante y algunos métodos de la familia (4.2.4) tras 20 iteraciones.

4.6.6. Familia (4.2.4) en el caso de dinámica real

```
# -*- coding: utf-8 -*-

#Metodo con memoria tipo secante
# Plano dinamico  x^3+x*abs(x)-2*x

import numpy as np
import matplotlib.pyplot as plt
import time


#Metodo
F= lambda X: X**3+X*np.abs(X)-2*X
U= lambda X, Y: X-(1-Sigma)*(X-Y) # X representa Xn e Y Xn-1
V= lambda X, Y: X+(1-Sigma)*(X-Y)  # U representa yn V zn, las letras
    ya estaban cogidas en el codigo
DF= lambda X,Y: ((F(U(X,Y))-F(V(X,Y))))/(U(X,Y)-V(X,Y)) #Aproximacion
    derivada, diferencia dividida
M=lambda X,Y: X-F(X)/DF(X,Y)


#  z0, parte real de los puntos del plano dinamico
#  z_1, parte imaginaria de los puntos del plano dinamico
#  maxiter, numero maximo de iteraciones
#  Z, valor final de la orbita de cada punto
#  I, imagen del plano dinamico
#  tol, valor de tolerancia establecido
#  h, tamano de paso del mallado
tic=time.time() #Inicio tiempo de computacion
h=0.01     #Tamano de paso del mallado
tol=1E-15   #Tolerancia de aproximacion
maxiter=20 #Numero maximo iteraciones

Sigma=0.9 #Parametro del metodo

z0=np.arange(-4,4+h,h)
z_1=np.arange(-4,4+h,h)
Z0, Z_1=np.meshgrid(z0,z_1, indexing='xy')

itera=1 #Inicializacion parametro para iteraciones
#  R G B para los colores del plano dinamico
#  I, para combinar R G B
R=np.zeros([len(z0),len(z_1)])
G=np.zeros([len(z0),len(z_1)])
B=np.zeros([len(z0),len(z_1)])
I=np.zeros([len(z0),len(z_1),3])
IT=(maxiter+1)*np.ones([len(z0),len(z_1)])
#  Puntos fijos
ZF1=1
ZF2=-1
ZF3=0


while itera<=maxiter:
    Z=M(Z0,Z_1)

    for fil in range(len(z0)):
```

```
        for col in range(len(z_1)):
            Zfc=Z[fil,col]
            if np.abs(Zfc-ZF1)<tol:
                R[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
            if np.abs(Zfc-ZF2)<tol:
                G[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
            if np.abs(Zfc-ZF3)<tol:
                B[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
    #linalg.norm
    #Actualizacion de valores
    itera=itera+1
    Z_1=Z0
    Z0=Z

#Generacion de la imagen del plano
I[...,0]=R
I[...,1]=G
I[...,2]=B

# Se muestra el plano dinamico
plt.figure()
#plt.title('Dynamical Plane, Num Iter=%s' %(str(maxiter)))
plt.title('Dynamical Plane, Num Iter={}, \u03B8={}'.format(maxiter,
    Sigma))
#plt.plot(np.real(ZF1),np.imag(ZF1),'w*')
#plt.plot(np.real(ZF2),np.imag(ZF2),'w*')
plt.xlabel('$x_{0}$')
plt.ylabel('$x_{-1}$')
plt.grid()
#Se indica origin='lower' para indicar que el origen esta en la
    esquina
#Inferior izquieda
plt.imshow(I,origin='lower', extent=[z0[0],z0[-1],z_1[0],z_1[-1]])

#Numero de iteraciones promedio
print('% de iteraciones que convergen: ', np.size(IT[IT<=maxiter])/np
    .size(IT)*100)
#Valor medio de iteraciones para converger
print('Valor medio de iteraciones para converger: ',np.mean(IT[IT<=
    maxiter]))
#Tiempo de computacion
toc=time.time()
print('Tiempo de computacion:', toc-tic,' segundos')
```

4.6.7. Análisis dinámico real de los métodos aplicados a $F(x) = x^3 - x|x| - 2x = 0$.

En esta subsección se comparan las accesibilidades del método de la secante y la familia (4.2.4) al resolver la ecuación

$$F(x) = x^3 - x|x| - 2x = 0,$$

cuyas soluciones son

$$x^* = -2,$$

$$x^{**} = 0$$

y

$$x^{***} = 2.$$

Las regiones de convergencia a $x^* = -2$, $x^{**} = 0$ y $x^{***} = 2$ se representan en verde, azul y rojo, respectivamente. En las Figuras 4.27- 4.28 se muestra el comportamiento dinámico del método de la secante, mientras que las Figuras 4.27– 4.36 presentan el comportamiento de la familia (4.2.4) para diferentes valores del parámetro θ.

El porcentaje de puntos que garantizan la convergencia hacia alguna de las raíces se resume en las Tablas 4.9– 4.10.

Método o valor de θ	Porcentaje de puntos de convergencia
Método de la secante	87.16 %
Método de Kurchatov ($\theta = 0$)	95.85 %
$\theta = 0.3$	96.42 %
$\theta = 0.6$	97.04 %
$\theta = 0.9$	97.38 %

Tabla 4.9: Porcentaje de puntos que garantizan la convergencia del método de la secante y algunos métodos de la familia (4.2.4) tras 10 iteraciones.

Método o valor de θ	Porcentaje de puntos de convergencia
Método de la secante	99.68 %
Método de Kurchatov ($\theta = 0$)	99.78 %
$\theta = 0.3$	99.80 %
$\theta = 0.6$	99.82 %
$\theta = 0.9$	99.84 %

Tabla 4.10: Porcentaje de puntos que garantizan la convergencia del método de la secante y algunos métodos de la familia (4.2.4) tras 20 iteraciones.

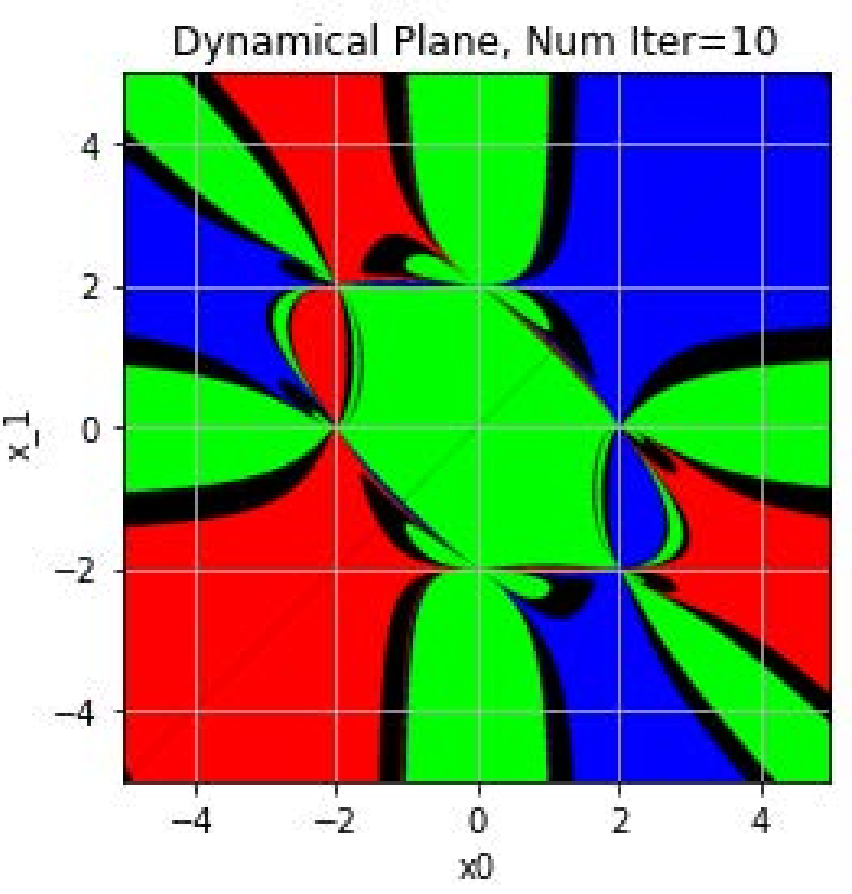


Figura 4.27: Cuenca de atracción obtenida con el método de la secante con 10 iteraciones

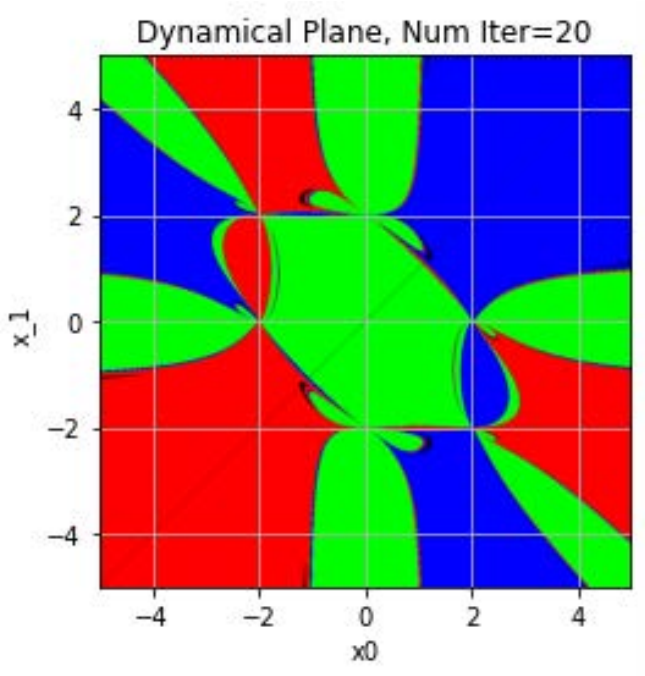


Figura 4.28: Cuenca de atracción obtenida con el método de la secante con 20 iteraciones

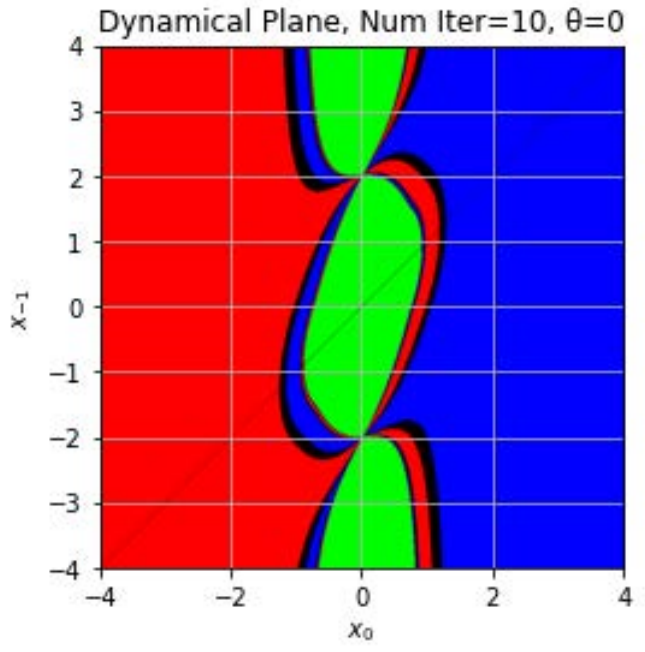


Figura 4.29: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0$ (Método de Kurchatov) con un máximo de 10 iteraciones.

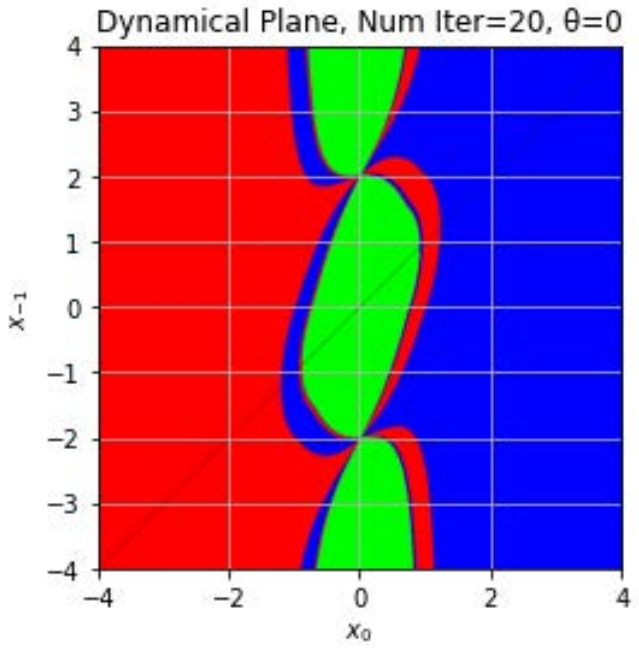


Figura 4.30: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0$ (Método de Kurchatov) con un máximo de 20 iteraciones.

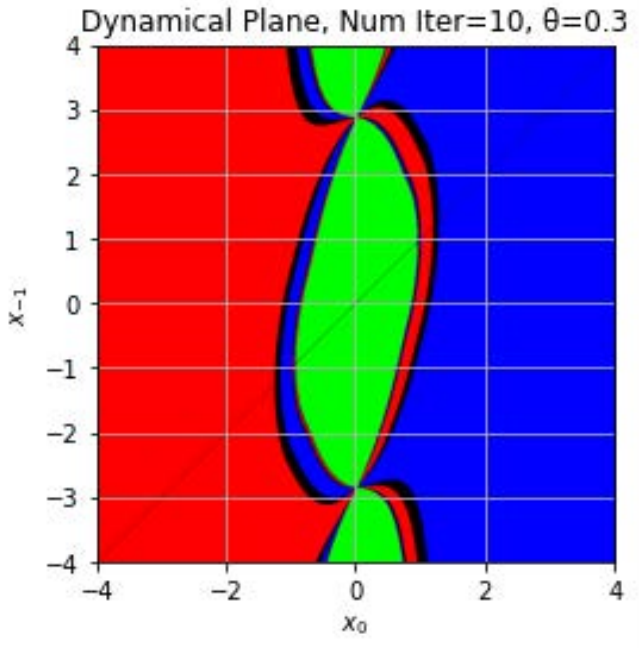


Figura 4.31: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0.3$ con un máximo de 10 iteraciones.

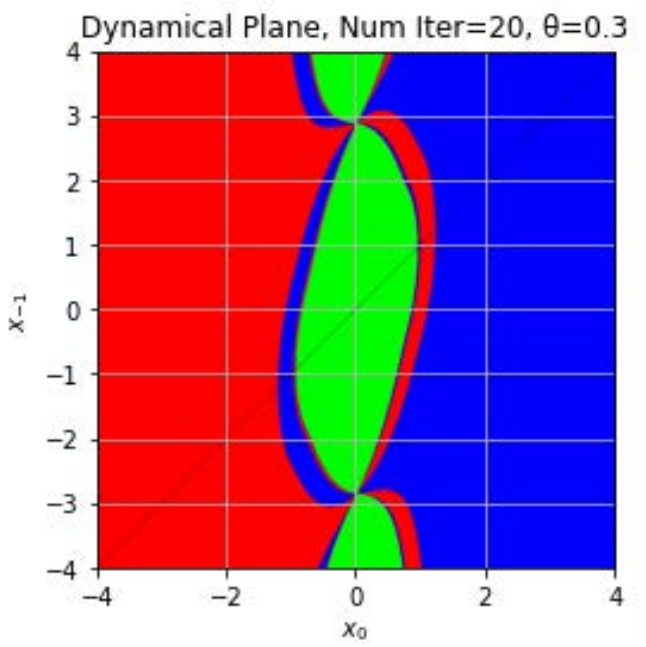


Figura 4.32: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0.3$ con un máximo de 20 iteraciones.

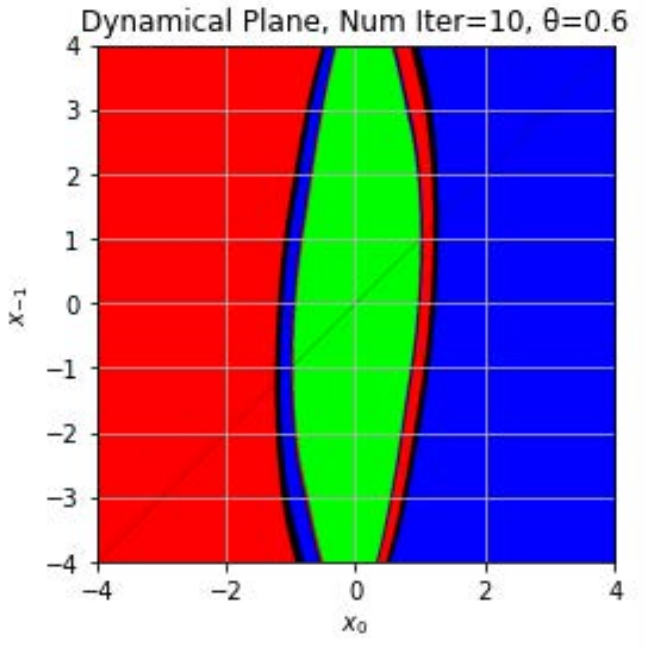


Figura 4.33: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0.6$ con un máximo de 10 iteraciones.

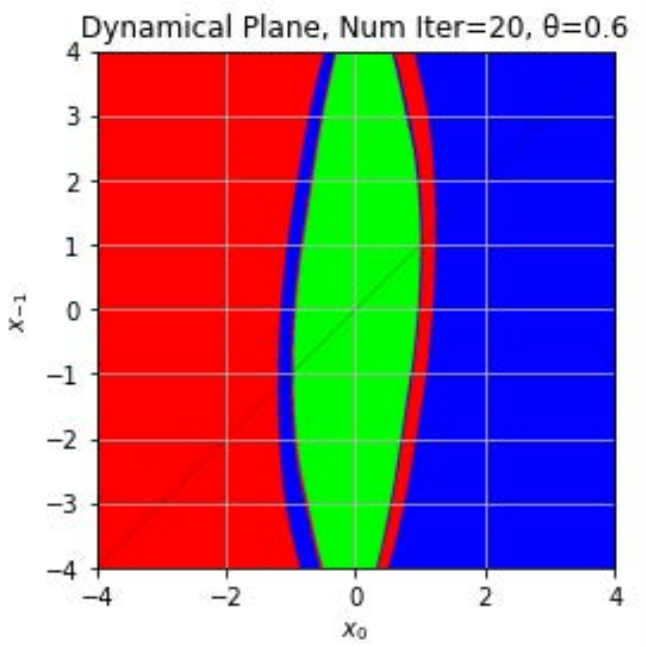


Figura 4.34: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0.6$ con un máximo de 20 iteraciones.

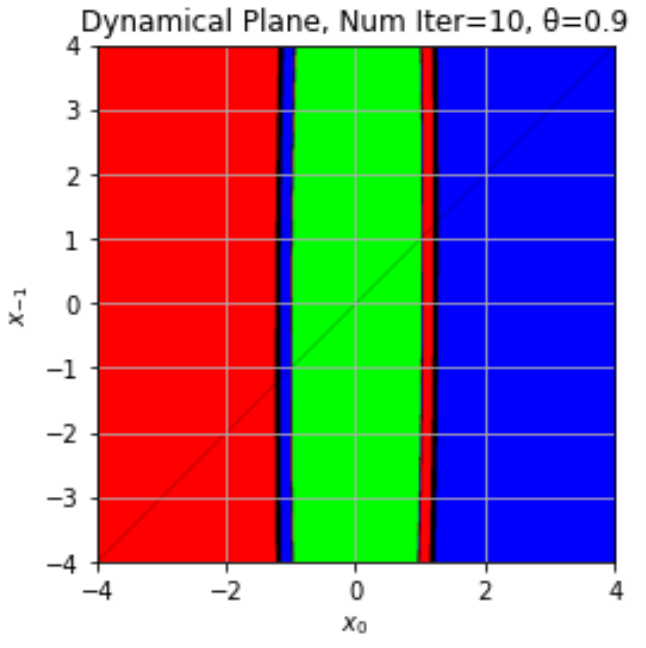


Figura 4.35: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0.9$ con un máximo de 10 iteraciones.

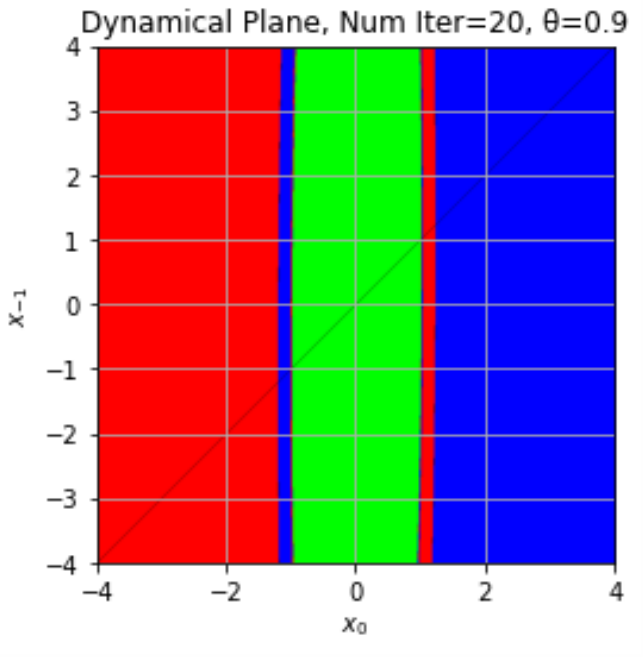


Figura 4.36: Cuencas de atracción obtenidas para algunos métodos de la familia (4.2.4) para $\theta = 0.9$ con un máximo de 20 iteraciones.

4.6.8. Familia (4.2.4) en el caso de dinámica real

```
# -*- coding: utf-8 -*-

#Metodo con memoria tipo secante
# Plano dinamico  x^3-x*abs(x)-2*x

import numpy as np
import matplotlib.pyplot as plt
import time


#Metodo
F= lambda X: X**3-X*np.abs(X)-2*X
U= lambda X, Y: X-(1-Sigma)*(X-Y) # X representa Xn e Y Xn-1
V= lambda X, Y: X+(1-Sigma)*(X-Y)  # U representa yn V zn, las letras
    ya estaban cogidas en el codigo
DF= lambda X,Y: ((F(U(X,Y))-F(V(X,Y))))/(U(X,Y)-V(X,Y)) #Aproximacion
    derivada, diferencia dividida
M=lambda X,Y: X-F(X)/DF(X,Y)


#  z0, parte real de los puntos del plano dinamico
#  z_1, parte imaginaria de los puntos del plano dinamico
#  maxiter, numero maximo de iteraciones
#  Z, valor final de la orbita de cada punto
#  I, imagen del plano dinamico
#  tol, valor de tolerancia establecido
#  h, tamano de paso del mallado
```

```
tic=time.time() #Inicio tiempo de computacion
h=0.01      #Tamano de paso del mallado
tol=1E-15   #Tolerancia de aproximacion
maxiter=20 #Numero maximo iteraciones

Sigma=0.9 #Parametro del metodo

z0=np.arange(-4,4+h,h)
z_1=np.arange(-4,4+h,h)
Z0, Z_1=np.meshgrid(z0,z_1, indexing='xy')

itera=1 #Inicializacion parametro para iteraciones
#  R G B para los colores del plano dinamico
#  I, para combinar R G B
R=np.zeros([len(z0),len(z_1)])
G=np.zeros([len(z0),len(z_1)])
B=np.zeros([len(z0),len(z_1)])
I=np.zeros([len(z0),len(z_1),3])
IT=(maxiter+1)*np.ones([len(z0),len(z_1)])
#  Puntos fijos
ZF1=-2
ZF2=0
ZF3=2


while itera<=maxiter:
    Z=M(Z0,Z_1)

    for fil in range(len(z0)):
        for col in range(len(z_1)):
            Zfc=Z[fil,col]
            if np.abs(Zfc-ZF1)<tol:
                R[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
            if np.abs(Zfc-ZF2)<tol:
                G[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
            if np.abs(Zfc-ZF3)<tol:
                B[fil,col]=1
                IT[fil,col]=min(IT[fil,col],itera)
    #linalg.norm
    #Actualizacion de valores
    itera=itera+1
    Z_1=Z0
    Z0=Z

#Generacion de la imagen del plano
I[...,0]=R
I[...,1]=G
I[...,2]=B

# Se muestra el plano dinamico
plt.figure()
#plt.title('Dynamical Plane, Num Iter=%s' %(str(maxiter)))
plt.title('Dynamical Plane, Num Iter={}, \u03B8={}'.format(maxiter,
    Sigma))
#plt.plot(np.real(ZF1),np.imag(ZF1),'w*')
#plt.plot(np.real(ZF2),np.imag(ZF2),'w*')
```

```
plt.xlabel('$x_{0}$')
plt.ylabel('$x_{-1}$')
plt.grid()
#Se indica origin='lower' para indicar que el origen esta en la
    esquina
#Inferior izquieda
plt.imshow(I,origin='lower', extent=[z0[0],z0[-1],z_1[0],z_1[-1]])

#Numero de iteraciones promedio
print('% de iteraciones que convergen: ', np.size(IT[IT<=maxiter])/np
    .size(IT)*100)
#Valor medio de iteraciones para converger
print('Valor medio de iteraciones para converger: ',np.mean(IT[IT<=
    maxiter]))
#Tiempo de computacion
toc=time.time()
print('Tiempo de computacion:', toc-tic,' segundos')
```

4.7. Conclusiones

El uso de diferencias divididas simétricas ha permitido desarrollar una familia uniparamétrica de métodos iterativos tipo secante, tomando como base el método de la secante clásico. Observando el método de Kurchatov que tiene convergencia cuadrática, es conocido que esta se consigue gracias a la aproximación de la derivada

$$[x_{n-1}, 2x_n - x_{n-1}; F] = [x_n - (x_n - x_{n-1}), x_n + (x_n - x_{n-1}); F] \approx F'(x_n),$$

la cual se utiliza en el método de Newton. Esta es una diferencia dividida simétrica de primer orden. Para lograr convergencia cuadrática en la práctica, es necesario obtener una buena aproximación de la primera derivada de F en el método de Newton. Por tanto, usando la diferencia dividida

$$[x_n - (1-\theta)(x_n - x_{n-1}), x_n + (1-\theta)(x_n - x_{n-1}); F],$$

lo que lleva a la siguiente familia uniparamétrica de métodos iterativos:

$$\begin{cases} x_{-1}, x_0 \text{ iniciales en } \Omega, \quad \theta \in [0,1], \\ y_n = x_n - (1-\theta)(x_n - x_{n-1}), \quad n \geq 0, \\ z_n = x_n + (1-\theta)(x_n - x_{n-1}), \quad n \geq 0, \\ x_{n+1} = x_n - [y_n, z_n; F]^{-1} F(x_n). \end{cases}$$

Esta familia, se ha reescrito

$$\begin{cases} x_{-1}, x_0 \text{ iniciales en } \Omega, \quad \theta \in [0,1], \\ y_n = \theta x_n + (1-\theta) x_{n-1}, \quad n \geq 0, \\ z_n = (2-\theta) x_n - (1-\theta) x_{n-1}, \quad n \geq 0, \\ x_{n+1} = x_n - [y_n, z_n; F]^{-1} F(x_n), \end{cases}$$

para, mediante el teorema 1 que aparece en [54], probar que el orden de convergencia R es cuadrático si $\theta \in [0,1]$.

Una vez construida la familia y probado el orden de convergencia se ha estudiado su eficiencia, mostrando que el de la nueva familia es mejor que la secante, tanto para problemas pequeños como para grandes. Además, los métodos iterativos de la familia presentan una mayor velocidad de convergencia que el método de la secante.

Por otro lado, se presentan las condiciones para garantizar la convergencia local de los miembros de la familia y se ha presentado un resultado doden la bola de convergencia (que determina la accesibilidad del método iterativo a partir de aproximaciones iniciales dentro de la misma) mejora a medida que aumenta el valor del parámetro que define la familia. Además, se observa de forma experimental que se obtienen mejores aproximaciones a la solución cuanto mayor es dicho valor del parámetro.

Después de garantizar la convergencia de forma teórica, se aplican a 3 ejemplos diferentes:En el primer ejemplo, se va a considerar una ecuación integral no lineal.

En el segundo ejemplo, se analiza la influencia del parámetro θ en las aproximaciones a las soluciones y se computaron los radios de convergencia asociados para diferentes valores del parámetro θ donde se observó que cuanto mayor es este valor mayor es el radio de la bola de convergencia. En el tercer ejemplo, se compara la familia con otros métodos cuadráticos que también son aplicables.

Por otro lado, se lleva a cabo un estudio dinámico comparativo para diferentes valores del parámetro θ. Para ello, se realizan estudios dinámicos reales y complejos de los métodos de la familia, aplicado a dos polinomios diferentes

$$F(z) = z^3 - z|z| - 2z$$

y

$$F(z) = z^3 - z|z| + 2z,$$

con diferente número máximo de iteraciones y donde puede observarse que la accesibilidad depende del valor del parámetro.

Por último, para ambos enfoques se han facilitado códigos para poder replicarlos de forma fácil en Python.

Capítulo 5

Artículos derivados de la tesis

A lo largo del periodo que ha durado la realización de esta tesis doctoral se han obtenido resultados vinculados con los estudios realizados y fruto de estos resultados se han realizado publicaciones en revistas de reconocido prestigio. Así, cada uno de los capítulos 2, 3 y 4 son las adaptaciones de los artículos publicados que se recogen y realacionan a continuación.

5.1. Artículo "On the set of initial guesses for the secant method"

El Capitulo 2 ha dado lugar a la publicación de un Artículo en la revista "Mathematical Methods in the Applied Science". La cual es una interdisciplinaria que aborda problemas lineales y no lineales, directos e inversos, relacionados con procesos físicos. Su objetivo es facilitar la comunicación y colaboración entre matemáticos y científicos, publicando artículos que, aunque matemáticamente avanzados, sean relevantes y accesibles para diversas disciplinas. El artículo publicado lleva por título "On the set of initial guesses for the secant method", con el número de doi: `https://doi.org/10.1002/mma.9052` A continuación, se recoge la primera página del mismo:

Received: 30 April 2022 | Revised: 4 November 2022 | Accepted: 14 December 2022

DOI: 10.1002/mma.9052

SPECIAL ISSUE PAPER

WILEY

On the set of initial guesses for the secant method

Alejandro Moysi | **José Antonio Ezquerro** | **Miguel Ángel Hernández-Verón** | **Ángel Alberto Magreñán**

Department of Mathematics and Computation, University of La Rioja, Logroño, Spain

Correspondence
Alejandro Moysi, Department of Mathematics and Computation, University of La Rioja, Calle Luis de Ulloa s/n, 26004 Logroño, Spain.
Email: alejandro.moysi@alum.unirioja.es

Communicated by: J. R. Torregrosa

Funding information
This research was partially supported by Ministerio de Ciencia, Innovación y Universidades under grant PGC2018-095896-B-C21.

The secant method is the most used iterative method to solve an operator equation where the operator involved is nondifferentiable. A known problem that arises when applying this method is its accessibility. Then, we try to improve it by using the technique of decomposition of the operator involved, so that the operator is in turn the sum of two operators, one differentiable and the other continuous but not differentiable. For this, we use a family of Newton-secant-type iterative methods that arises from Newton's method and from a well-known family of secant-type iterative methods. We study the accessibility of the new methods in two different ways: From the convergence balls of the methods, obtained from a local study of the convergence, and from a dynamic study of the methods. Some examples related to chemistry are also presented to prove the theoretical results.

KEYWORDS

accessibility, convergence ball, dynamics, Hammerstein integral equation, local convergence, Newton's method, the secant method

MSC CLASSIFICATION

45G10, 47H99, 65J15

1 | INTRODUCTION

Many problems of Mathematics, Mathematical Chemistry, or Engineering can be written as a nonlinear equation,[1,2] as for example, boundary value problems for differential equations, nonlinear integral equations arising in many contexts, such as the theory of elasticity, electrostatics, the potential theory, and radiative heat transfer problems, and systems of nonlinear equations resulting from the discretization of numerous problems. Since we can write all these problems as the operator equation $F(x) = 0$, we need to give the operator F some generality. So, we consider the operator $F : \Omega \subseteq X \to Y$, where X and Y are Banach spaces and Ω is a nonempty open convex subset of X.

In general, the roots of $F(x) = 0$ cannot be expressed in a closed form, so this problem is commonly solved by applying iterative methods. If the operator F is differentiable, Newton's method[3] is the most used iteration to solve $F(x) = 0$, due to its computational efficiency, which is given by

$$\begin{cases} x_0 \text{ given in } \Omega, \\ x_{n+1} = x_n - [F'(x_n)]^{-1} F(x_n), \ n \geq 0, \end{cases} \tag{1}$$

5.2. Artículo "A significant improvement of a family of secant-type methods"

El Capítulo 3 ha dado lugar a la publicación de un Artículo en la revista "Journal of Computational and Applied Mathematics", que publica artículos originales de alto valor científico en todas las áreas de las matemáticas computacionales y aplicadas. El principal interés de la revista está en artículos que describan y analicen nuevas técnicas computacionales para resolver problemas científicos o de ingeniería. También es de importancia el análisis mejorado, incluyendo la efectividad y aplicabilidad, de métodos y algoritmos existentes. La eficiencia computacional (por ejemplo, la convergencia, estabilidad, precisión, etc.) debe ser demostrada e ilustrada con ejemplos numéricos no triviales.

El artículo lleva por título "A significant improvement of a family of secant-type methods", con número de doi: `https://doi.org/10.1016/j.cam.2022.115002`

A continuación, se recoge la primera página del mismo:

Journal of Computational and Applied Mathematics 424 (2023) 115002

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A significant improvement of a family of secant-type methods☆

J.A. Ezquerro, M.A. Hernández-Verón, Á.A. Magreñán, A. Moysi *

Department of Mathematics and Computation, University of La Rioja., Calle Luis de Ulloa s/n, 26004 Logroño, Spain

ARTICLE INFO

Article history:
Received 14 September 2022
Received in revised form 30 October 2022

MSC:
45G10
47H99
65J15

Keywords:
Secant-type methods
Local convergence
Accessibility
Dynamics
Efficiency
Hammerstein integral equation

ABSTRACT

Secant-type iterative methods are usually used to solve nonlinear systems of equations where the operator involved is nondifferentiable or its derivative is costly. From a uniparametric family of secant-type methods, that is reduced to the secant method and Newton's method for some particular values of the parameter involved, with operational cost similar to that of the secant and Newton methods and superlinear convergence, we construct a biparametric family of iterative methods with quadratic convergence, study their local convergence, their efficiency and, based on the dynamics of the methods, their accessibility.

1. Introduction

Newton's method and its variants are used to solve nonlinear operator equations of the form $H(x) = 0$, where this equation can represent a differential equation, an integral equation or, by following a process of discretization, a system of nonlinear equations. Throughout this work, we consider the nonlinear function $H : \Omega \subset \mathbb{R}^m \to \mathbb{R}^m$, where Ω is a nonempty open convex domain in $\mathbb{R}^m$, such that $H(x) = (H_1(x), H_2(x), \ldots, H_m(x))$ with $H_i : \Omega \subseteq \mathbb{R}^m \to \mathbb{R}$, for $i = 1, 2, \ldots, m$ and $x = (x_1, x_2, \ldots, x_m)$.

One of the most used iterative methods to approximate a solution w^* of the equation $H(x) = 0$ is Newton's method, whose algorithm is

$$\begin{cases} w_0 \text{ given in } \Omega, \\ w_{n+1} = w_n - [H'(w_n)]^{-1} H(w_n), \quad n \geq 0. \end{cases} \tag{1}$$

The quadratic convergence and the low operational cost of the method guarantees its good computational efficiency. But this method has a known limitation: the derivative $H'(x)$ has to be evaluated at each iteration, which makes the method not applicable to equations with nondifferentiable operators and in situations where the evaluation of the derivative is costly. In these cases, it is common to approximate derivatives by divided differences and apply iterative methods that use divided differences instead of derivatives. So, in this work, we are interested in iterative methods that avoid the

☆ This research was partially supported by Ministerio de Ciencia, Innovación y Universidades under grant PGC2018-095896-B-C21.

* Corresponding author.
E-mail addresses: jezquer@unirioja.es (J.A. Ezquerro), mahernan@unirioja.es (M.A. Hernández-Verón), angel-alberto.magrenan@unirioja.es (Á.A. Magreñán), alejandro.moysi@alum.unirioja.es (A. Moysi).

https://doi.org/10.1016/j.cam.2022.115002

5.3. Artículo "A procedure to obtain quadratic convergence from the secant method"

El Capítulo 4 ha dado lugar a la publicación de un Artículo en la revista "Journal of Computational and Applied Mathematics". El artículo lleva por título "A procedure to obtain quadratic convergence from the secant method", con número de doi: `https://doi.org/10.1016/j.cam.2024.115912`

A continuación, se recoge la primera página del mismo:

Journal of Computational and Applied Mathematics 448 (2024) 115912

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A procedure to obtain quadratic convergence from the secant method

J.A. Ezquerro, M.A. Hernández-Verón, Á.A. Magreñán *, A. Moysi

Department of Mathematics and Computation, University of La Rioja, Calle Luis de Ulloa s/n, 26004 Logroño, Spain

ARTICLE INFO

MSC:
45G10
47H99
65J15

Keywords:
Secant-type methods
Local convergence
Accessibility
Dynamics
Efficiency
Integral equation

ABSTRACT

From the secant method and using symmetric divided differences, we construct a uniparametric family of secant-type iterative methods with quadratic convergence and better computational efficiency than the secant method. In addition, we experimentally analyze the accessibility of the family from a dynamic point of view and study the local convergence.

1. Introduction

We present a new uniparametric family of iterative methods that do not use derivatives in the algorithm to approximate a solution x^* of a nonlinear equation $T(x) = 0$, where $T : \Omega \subseteq X \to Y$ is a nonlinear operator defined on a non-empty open convex domain Ω of a Banach space X with values in a Banach space Y.

One of the most used iterative methods to approximate a solution x^* of the equation $T(x) = 0$ is Newton's method, whose algorithm is

$$\begin{cases} x_0 \text{ given in } \Omega, \\ x_{n+1} = x_n - [T'(x_n)]^{-1} T(x_n), \quad n \geq 0. \end{cases} \tag{1}$$

The quadratic convergence and the low operational cost of the method guarantee a good computational efficiency. But this method has a known limitation: the derivative $T'(x)$ has to be evaluated at each iteration, which makes the method not applicable to equations with nondifferentiable operators and in situations where the evaluation of the derivative has a high computational cost.

Thus, in this work, we are interested in studying iterative methods that avoid the computation of derivatives of the operator T at every step of their algorithms. Among these, the secant method is possibly the most known [1,2], is given by

$$\begin{cases} x_0, x_{-1} \text{ given in } \Omega, \\ x_{n+1} = x_n - [x_{n-1}, x_n; T]^{-1} T(x_n), \quad n \geq 0, \end{cases} \tag{2}$$

* Corresponding author.
E-mail addresses: jezquer@unirioja.es (J.A. Ezquerro), mahernan@unirioja.es (M.A. Hernández-Verón), angel-alberto.magrenan@unirioja.es (Á.A. Magreñán), alejandro.moysi@alum.unirioja.es (A. Moysi).

https://doi.org/10.1016/j.cam.2024.115912
Received 10 August 2023; Received in revised form 18 January 2024
Available online 5 April 2024

Capítulo 6

Otros artículos relacionados indirectamente con la tesis

A lo largo del periodo del doctorado, se han realizado otros trabajos de investigación, que si bien no están directamente relacionados con la línea de la tesis relativo a mejoras en el método de la secante como lo está el trabajo anterior, si se considera que tienen relevancia e interés en el campo de los métodos numéricos y por tanto recogerlos en un capítulo a parte.

6.1. Artículo "Ball comparison between frozen Potra and Schmidt–Schwetlick schemes with dynamical analysis"

En primer lugar, un artículo publicado en la revista "Computational and Mathematics Methods", con referencia doi: https://doi.org/10.1002/cmm4.1186

En este artículo se propone una nueva investigación relacionada con la convergencia de los esquemas congelados de Potra y Schmidt–Schwetlick cuando se aplican a ecuaciones. El objetivo del estudio es establecer una comparación entre dos soluciones de ecuaciones bajo las mismas condiciones. En particular, se analiza el radio de convergencia y la coincidencia de la bola de unicidad, mientras que las estimaciones de error presentan, en general, diferencias.

Se extiende la convergencia local para operadores con valores en espacios de Banach utilizando únicamente la diferencia dividida de primer orden y la primera derivada de los esquemas. Esto representa una ventaja significativa, ya que permite mejorar la convergencia sin necesidad de calcular derivadas de orden superior, cuyo cálculo puede resultar complejo o, en algunos casos, no ser posible.

Asimismo, se presenta un estudio dinámico del comportamiento de un método en comparación con su alternativa no congelada, con el fin de analizar el comportamiento de ambos. Se estudian las cuencas de atracción de los dos métodos en tres polinomios distintos, que incluyen casos con dos soluciones reales, tres soluciones reales y una combinación de dos soluciones reales y dos soluciones complejas diferentes. A continuación, se recoge la primera página del artículo:

Received: 5 October 2020 | Accepted: 15 March 2021

DOI: 10.1002/cmm4.1186

SPECIAL ISSUE PAPER

WILEY

Ball comparison between frozen Potra and Schmidt–Schwetlick schemes with dynamical analysis

Michael Argyros[1] | **Ioannis K. Argyros**[1] | **Daniel González**[2] | **Ángel Alberto Magreñán**[3] | **Alejandro Moysi**[3] | **Íñigo Sarría**[4]

[1]Department of Mathematical Sciences, Cameron University, Lawton, Oklahoma, USA

[2]Escuela de Ciencias Físicas y Matemáticas, Universidad de Las Américas, Quito, Ecuador

[3]Department of Mathematics and Computation, Universidad de La Rioja, La Rioja, Spain

[4]Escuela Superior de Ingeniería y Tecnología, Universidad Internacional de La Rioja, La Rioja, Spain

Correspondence
Ángel Alberto Magreñán, Department of Mathematics and Computation, Universidad de La Rioja, La Rioja, Spain.
Email: angel-alberto.magrenan@unirioja.es

Abstract

In this article, we propose a new research related to the convergence of the frozen Potra and Schmidt–Schwetlick schemes when we apply to equations. The purpose of this study is to introduce a comparison between two solutions to equations under the same conditions. In particular, we show the convergence radius and the uniqueness ball coincidence, while the error estimates are generally different. In this work, we extended the local convergence for Banach space valued operators using only the divided difference of order one and the first derivative of the schemes. This is a great advantage since we improve convergence by avoiding calculating higher-order derivatives that can either be difficult or not even exist. On the other hand, we also present a dynamical study of the behavior of a method compared with its no frozen alternative in order to see the behavior of both. We will study the basins of attraction of the two methods to three different polynomials involving two real, three real, and two real and two complex different solutions.

KEYWORDS

ball convergence, Banach space, Potra scheme, Schmidt–Schwetlick scheme, secant scheme

1 | INTRODUCTION

Consider X_1, X_2 to stand for Banach spaces, $\vartheta \subset X_1$ be a convex set, and $H : \vartheta \to X_2$ is differentiable in the Fréchet sense. We are concerned with approximating a solution λ of equation

$$H(y) = 0 \tag{1}$$

by the Potra (PS) and the Schmidt–Schwetlick-type algorithm (SSTA) given for each $n = 0, 1, 2, \ldots, 0 \leq j \leq k-1$, by

$$\text{(PS)} \quad \begin{cases} y_n^{(j+1)} = y_n^{(j)} - [y_{n-1}, y_n; H]^{-1} H(y_n^{(j)}), \\ y_n^{(0)} = y_n, \quad y_{n+1} = x_n^{(k)}, \end{cases} \tag{2}$$

$$\text{(SSTA)} \quad \begin{cases} y_n^{(j+1)} = y_n^{(j)} - [y_n, z_n; H]^{-1} H(y_n^{(j)}), \\ y_n^{(0)} = z_n, \quad y_{n+1} = y_n^{(k-1)}, \quad z_{n+1} = y_n^{(k)}, \end{cases} \tag{3}$$

6.2. Artículo "On the existence and the approximation of solutions of Volterra integral equations of the second kind"

En segundo lugar, un artículo publicado en la revista "Applied Mathematics and Computation", con referencia doi: https://doi.org/10.1016/j.amc.2024.128829

En este artículo Se estudia la existencia de soluciones de las ecuaciones integrales de Volterra de segunda especie y su aproximación. Para ello, se analiza el papel del Principio de Contracción de Banach en la localización de una solución y su aproximación mediante el método de aproximaciones sucesivas.

El análisis previo se mejora proporcionando una ubicación de la solución y, posteriormente, aproximándola mediante un método de colocación que emplea polinomios de Lagrange. Para evitar el fenómeno de Runge, que puede surgir al incrementar el número de nodos, se eligen los ceros de Chebyshev como puntos de colocación. Sin embargo, al aumentar el número de nodos para mejorar la precisión, puede presentarse el problema del mal condicionamiento del sistema lineal involucrado en el método de colocación. Este último inconveniente se soluciona aplicando un método iterativo libre de inversión, el cual utiliza únicamente productos de matrices

A continuación, se recoge la primera página del artículo:

Applied Mathematics and Computation 478 (2024) 128829

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Full Length Article

On the existence and the approximation of solutions of Volterra integral equations of the second kind ☆

J.A. Ezquerro, M.A. Hernández-Verón, Á.A. Magreñán, A. Moysi

Department of Mathematics and Computation, University of La Rioja, Calle Luis de Ulloa s/n, 26004 Logroño, Spain

ARTICLE INFO

MSC:
45D05
65F10
65J10
65R20

Keywords:
Volterra integral equation
Chebyshev's zeros
Collocation method
Ulm-type iterative method

ABSTRACT

We study the existence of solution of Volterra integral equations of the second kind and how we can approximate it. For this, we analyse the role played by the Banach Contraction Principle to locate a solution and its approximation by the method of successive approximations. We improve the previous analysis by giving a location of the solution and then approximate it by a collocation method that uses Lagrange polynomials. To avoid the Runge phenomenon that may appear when the number of nodes increases, we choose the Chebyshev zeros as the collocation points. But, when increasing the number of nodes to improve the precision, the problem of possible ill-conditioning of the linear system involved in the collocation method may arise. This last problem is solved by applying an inverse-free iterative method that uses only matrix products.

1. Introduction

In this paper we study the Volterra integral equations of the second kind of the form [4]

$$u(x) = f(x) + \lambda \int_a^x K(x,t)\,u(t)\,dt, \quad \lambda \in \mathbb{R}, \quad x \in [a,b], \tag{1}$$

where $-\infty < a < x \leq b < +\infty$, $f(x) \in C[a,b]$ is given, $K(x,t)$ is the kernel of the integral equation such that $K : [a,b] \times [a,b] \to \mathbb{R}$ is a continuous function in both arguments and $u(x) \in C[a,b]$ is the unknown function to determine. In particular, there are two main aims that we consider: to study the existence of a solution and, when it exists, to obtain an effective procedure to approximate it.

For the first aim, the study of the existence of a solution of (1), we start with a usual procedure that consists in applying the well-known Banach Contraction Principle [7]. For this, we consider the operator $T : C[a,b] \to C[a,b]$ such that

$$[T(u)](x) = f(x) + \lambda \int_a^x K(x,t)u(t)\,dt, \quad \lambda \in \mathbb{R}, \quad x \in [a,b], \tag{2}$$

so that to guarantee a solution of the equation (1), we apply the Banach Contraction Principle to the operator (2), since a fixed point of (2) is a solution of (1).

☆ This research was partially supported by Ministerio de Ciencia, Innovación y Universidades under grant PGC2018-095896-B-C21.

E-mail addresses: jezquer@unirioja.es (J.A. Ezquerro), mahernan@unirioja.es (M.A. Hernández-Verón), angel-alberto.magrenan@unirioja.es (Á.A. Magreñán), alejandromoysi@yahoo.es (A. Moysi).

https://doi.org/10.1016/j.amc.2024.128829
Received 15 December 2023; Accepted 12 May 2024

Available online 27 May 2024

Capítulo 7

Conclusiones y trabajo futuro

7.1. Conclusiones

El presente trabajo ha abordado el estudio y desarrollo de métodos iterativos tipo secante, analizando su convergencia, accesibilidad y estabilidad en la resolución de ecuaciones no lineales. Se ha demostrado que, aunque el método de la secante presenta ventajas en términos computacionales al evitar el cálculo de derivadas, su accesibilidad y tasa de convergencia pueden ser mejoradas mediante modificaciones adecuadas. En este sentido, se han introducido nuevas familias de métodos iterativos que interpolan entre la secante y el método de Newton, obteniendo esquemas híbridos que optimizan la velocidad de convergencia sin incrementar significativamente el costo computacional.

Uno de los aspectos más relevantes de este trabajo ha sido el análisis de la accesibilidad de los métodos iterativos, es decir, la cantidad de aproximaciones iniciales que garantizan la convergencia hacia una solución. A través del estudio de la convergencia local y del análisis dinámico de cuencas de atracción en el plano complejo, se ha evidenciado que los métodos propuestos amplían la región de convergencia en comparación con la secante tradicional. Esto representa un avance significativo en la aplicabilidad de estos algoritmos a problemas donde la elección de condiciones iniciales es crítica.

Otro punto destacado ha sido la incorporación de Python como herramienta principal de desarrollo numérico. A diferencia de entornos tradicionalmente utilizados en este campo, como MATLAB o Mathematica, Python ha permitido una implementación más flexible y escalable, integrando librerías especializadas en análisis numérico, visualización y dinámica compleja. La capacidad de este lenguaje para representar gráficamente los resultados, en particular a través del análisis dinámico de los métodos iterativos, ha sido fundamental para validar empíricamente los modelos teóricos desarrollados.

Además del análisis teórico, se han realizado estudios numéricos aplicados a problemas concretos, incluyendo ecuaciones integrales de Hammerstein y ecuaciones no diferenciables. Los resultados obtenidos han confirmado que los métodos propuestos no solo presentan una mayor velocidad de convergencia en muchos casos, sino que también son más robustos ante variaciones en los parámetros iniciales.

En particular, la combinación de diferencias divididas y descomposición del operador ha permitido extender el uso de estos métodos a un espectro más amplio de problemas matemáticos y computacionales.

En resumen, este trabajo ha demostrado que la optimización de los métodos tipo secante mediante la inclusión de parámetros de control y diferencias divididas mejora sustancialmente su desempeño. La convergencia acelerada y la mayor accesibilidad de estas variantes representan una contribución relevante en el campo de los métodos iterativos para ecuaciones no lineales, con aplicaciones potenciales en diversas disciplinas científicas e ingenieriles.

7.2. Trabajo futuro

Los resultados obtenidos en esta investigación abren diversas posibilidades para futuros desarrollos y aplicaciones de los métodos iterativos tipo secante. Algunas de las líneas de trabajo que podrían explorarse incluyen:

1. Extensión a problemas en espacios de Banach más generales. Si bien este estudio ha abordado ecuaciones no lineales en espacios de Banach con ciertas restricciones, sería interesante analizar el comportamiento de estos métodos en espacios más generales, incluyendo problemas en espacios de Hilbert y otros entornos funcionales más complejos.

2. Desarrollo de métodos iterativos adaptativos. La implementación de estrategias adaptativas que ajusten dinámicamente los parámetros del método en función del comportamiento local de la iteración podría mejorar la estabilidad y eficiencia de los algoritmos. Esto permitiría, por ejemplo, que el método se aproxime más al comportamiento de Newton en ciertas regiones y conserve la flexibilidad de la secante en otras.

3. Optimización en términos de estabilidad computacional. Aunque los métodos desarrollados han demostrado mejoras en la convergencia y accesibilidad, aún se pueden explorar estrategias para reducir errores de acumulación numérica y mejorar la estabilidad computacional en iteraciones de alto orden.

4. Aplicaciones en ecuaciones diferenciales y sistemas dinámicos. Los métodos iterativos tipo secante podrían extenderse a la resolución de ecuaciones diferenciales no lineales y sistemas dinámicos, explorando su viabilidad en la modelización de fenómenos físicos y biológicos donde el cálculo de derivadas sea costoso o no posible.

5. Implementación y optimización en entornos de cómputo paralelo. Dado que Python ofrece herramientas para la computación en paralelo y distribuida, una posible extensión de este trabajo sería la optimización de los métodos iterativos en arquitecturas de alto rendimiento. Esto permitiría aplicar estas técnicas en problemas de gran escala en la ciencia computacional e ingeniería.

6. Exploración del impacto de los métodos en inteligencia artificial y machine learning. Los métodos iterativos juegan un papel clave en el entrenamiento de

modelos de aprendizaje automático. Explorar el uso de métodos tipo secante optimizados en algoritmos de optimización para deep learning podría ser una línea de trabajo con aplicaciones significativas en la inteligencia artificial.

En conclusión, la investigación realizada no solo ha permitido profundizar en el análisis de los métodos tipo secante, sino que también ha abierto múltiples líneas de desarrollo y aplicación en distintos campos matemáticos y computacionales. La combinación de enfoques teóricos, experimentales y computacionales sugiere que la optimización de estos métodos continuará siendo un área de interés en la resolución de problemas no lineales.

Referencias bibliográficas

[1] S. Amat, S. Busquier, C. Bermúdez and Á. A. Magreñán. On the election of the damped parameter of a two-step relaxed Newton-type method. *Nonlinear Dynamics*, **84** (1) (2016), 9–18.

[2] S. Amat, S. Busquier and Á. A. Magreñán. Reducing chaos and bifurcations in Newton-type methods, *Abstract and Applied Analysis*, **2013** (1) nº. 726701 (2013), 1–10.

[3] S. Amat, S. Busquier and S. Plaza. Review of some iterative root-finding methods from a dynamical point of view, *Scientia, Series A: Math. Sci.*, **10** (2004), 3–35.

[4] Amat, S., Ezquerro, J. A. and Hernández-Verón. On a Steffensen-like method for solving nonlinear equations, *Calcolo*, **53** (2) (2016), 171–188.

[5] S. Amat, Á. A. Magreñán and N. Romero. On a family of two step Newton-type methods, *Applied Mathematics and Computation* **219** (4) (2013), 11341–11347.

[6] I. K. Argyros. On the secant method, *Publ. Math. Debrecen*, **43** (3-4) (1993), 223–238.

[7] I. K. Argyros. A new convergence theorem for Steffensen's method on Banach spaces and applications, *Southwest J. Pure Appl. Math.*, **1** (1997), 23–29.

[8] I. K. Argyros. On a two–point Newton–like method of convergent order two, *Int. J. Comput. Math.*, **82** (2005), 219–233.

[9] I. K. Argyros. A Kantorovich-type analysis for a fast iterative method for solving nonlinear equations, *J. Math. Anal. Appl.*, **332** (2007), 97–108.

[10] I. K. Argyros. *Convergence and applications of Newton-type iterations*. Springer, New York, 2008.

[11] I. K. Argyros, Y. J. Cho and S. Hilout. *Numerical methods for equations and its applications*. CRC Press/Taylor and Francis, New York, 2012.

[12] I. K. Argyros and J. M. Gutiérrez. A unified approach for enlarging the radius of convergence for Newton's method and applications, *Nonlinear Functional Analysis and Applications*, **10** (2005), 555–563.

[13] I. K. Argyros and J. M. Gutiérrez. A unifying local and semilocal convergence analysis of Newton-like methods, *Advances in nonlinear variational inequalities*, **10** 1 (2007), 1–11.

[14] I. K. Argyros and S. Hilout. Weaker conditions for the convergence of Newton's method, *Journal of Complexity*, **28** (2012), 364–387.

[15] I. K. Argyros and S. Hilout. On the semilocal convergence of damped Newton's method, *Applied Mathematics and Computation*, **219** 5 (2012), 2808–2824.

[16] I. K. Argyros and S. Hilout. Estimating upper bounds on the limit points of majorizing sequences for Newton's method, *Numerical Algorithm*, **62** 1 (2013), 115–132.

[17] I. K. Argyros and Á. A. Magreñán. General convergence conditions of Newton's method for m-Fréchet differentiable operators, *Journal of Applied Mathematics and Computing* **43** (2013), 491–506.

[18] I. K. Argyros and S. Regmi. *Undergraduate research at Cameron University on iterative procedures in Banach and other spaces.* Nova Science, New York, 2019.

[19] I. K. Argyros and H. Ren. On an improved local convergence analysis for the Secant method, *Numerical Algorithm*, **52** (2009), 257–271.

[20] I.K. Argyros and F. Szidarovszky. *The theory and application of iterative methods.* CRC Press, Boca Ratón, FL 1993.

[21] I. N. Baker. An entire function which has wondering domains, *Journal of the Australian Mathematical Society*, **22** Ser. A (1976), 173–176.

[22] M. Balazs and G. Goldner. On existence of divided differences in linear spaces. *Revue d'analyse numérique et de la théorie de l'approximation* **2** (1973), 3–6.

[23] F. Balibrea and M. Zamora. Propiedades dinámicas de las funciones de Newton asociadas a polinomios. Conjetura de Holmgren, *XXI Congreso de Ecuaciones Diferenciales y Aplicaciones, XI Congreso de Matemática Aplicada*, (2009), 1–10.

[24] S. Banach. *Théorie des opérations linéaires*, Monografie Matematyczne, Varsovia, 1932.

[25] A. F. Beardon. *Iteration of rational functions*, Springer-Verlag, New York, 1991.

[26] L. Billings, J. H. Curry and V. Robins. Chaos in Relaxed Newton's Method: The Quadratic Case, *Academic Press*, University of Colorado at Boulder, 1999.

[27] A. K. Bisoi. Newton raphson method, scaling at fractal boundaries and mathematica, *Mathematical and Computer Modelling*, **21** 10 (1995), 91–102.

[28] P. Blanchard. Complex analytic dynamics on the Riemann sphere. *Bull. of AMS* (new series)**11** 1 (1984), 85–141.

[29] L. Blum, F. Cucker, M. Shub and S. Smale. *Complexity and real computation*, Springer-Verlag, New York, 1997.

[30] A. D. Bryuno. Convergence of transformations of differential equations to normal forms, *Dock. Akad. Nauk URSS*, **165** (1965), 987–989.

[31] B. Campos, A. Cordero, J.R. Torregrosa and P. Vindel. A multidimensional dynamical approach to iterative methods with memory. *Applied Mathematics and Computation*, **271** (2015), 701–715.

[32] B. Campos, A. Cordero, J.R. Torregrosa and P. Vindel. Stability of King's family of iterative methods with memory. *Journal of Computational and. Applied Mathematics*, **318** (2017), 504–514.

[33] L. Carleson and T. Gamelin. *Complex Dynamics*, Springer-Verlag, New York, 1993.

[34] H. Cartan. *Calcul différentiel*, Hermann, Paris, 1971.

[35] A. Cayley. The Newton-Fourier imaginary problem, *American Journal of Mathematics*, **2** (1879), 97–97.

[36] J. Chen and Z. Shen. Convergence analysis of the secant type methods, *Applied Mathematics and Computation*, **188** (2007), 514–524.

[37] F. I. Chicharro. *Proyecto docente e investigador.* Universidad Politécnica de Valencia, 2023.

[38] F. Chicharro, A. Cordero, J. M. Gutiérrez and J. R. Torregrosa; Complex dynamics of derivative-free methods for nonlinear equations, *Applied Mathematics and Computation*, **19** 12 (2013), 7023–7035.

[39] A. Cordero, L. Feng, Á. A. Magreñán and J. R. Torregrosa. A new fourth-order family for solving nonlinear problems and its dynamics. *Journal of Mathematical Chemistry*, **53** (3) (2015), 893–910.

[40] A. Cordero, T. Lotfi, P. Bakhtiari and J. R. Torregrosa. An efficient two-parametric family with memory for nonlinear equations. *Numerical Algorithms*, **68** 2 (2015), 323–335.

[41] A. Cordero, J. R. Torregrosa and P. Vindel. Dynamics of a family of Chebyshev-Halley type methods, *Applied Mathematics and Computation*, **219** 16 (2013), 8568–8583.

[42] H. Cremer. Zum Zentrumproblem, *Mathematische Annalen*, **98** (1927), 151–163.

[43] B. Campos, A. Cordero, J.R. Torregrosa and P. Vindel. Stability of King's family of iterative methods with memory. *Journal of Computational and. Applied Mathematics*, **318** (2017), 504–514.

[44] J. P. Dedieu. *Points fixes, zéros et la méthode de Newton*, Mathématiques et Applications, Springer Verlag, Berlin, **54**, 2006.

[45] J. E. Dennis and R. B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations.* SIAM, Philadelphia, 1996.

[46] R. Devaney. *An introduction to Chaotic Dynamical Systems,* 2^{nd} *edition*, Addison-Wesley, Redwood City, California, 1989.

[47] M. A. Diloné. *Métodos iterativos aplicados a la ecuación de Kepler.* Universidad de La Rioja, 2013

[48] B. I. Epureanu and H. S. Greenside. Fractal basins on the attraction associated with a damped Newton's method, *SIAM Rev.*, **40** (1998), 102–109.

[49] J.A. Ezquerro. *Construcción de procesos iterativos mediante aceleraciones del método de Newton.* Universidad de La Rioja, 1996

[50] J.A. Ezquerro, D. González and M.A. Hernández-verón. A variant of the Newton-Kantorovich theorem for nonlinear integral equations of mixed Hammerstein type, *Applied Mathematics and Computation*, **218** (2012), 9536–9546.

[51] J. A. Ezquerro, J. M. Gutiérrez, M. Á. Hernández, N. Romero and M. J. Rubio: El método de Newton: de Newton a Kantorovich, *La Gaceta de la RSME*, **13** 1 (2010), 53–76.

[52] J. A. Ezquerro, J. M. Gutiérrez, M. Á. Hernández and M. A. Salanova. *El método de Halley: Posiblemente, el método más redescubierto del mundo.* Servicio de Publicaciones de la Universidad de La Rioja, Logroño (Spain), 2001.

[53] J, A, Ezquerro, M. Grau-Sánchez, M. A.Hernández-Verón and M. Noguera, M. Semilocal convergence of secant-like methods for differentiable and nondifferentiable operator equations, *Journal of Mathematical Analysis and Applications*, **398** (2013), 100–112.

[54] J. A. Ezquerro, A. Grau, M. Grau-Sánchez and M.A. Hernández-Verón. A new class of secant-like methods for solving nonlinear systems of equations, *Communications in Applied Mathematics and Computational Science*, **9** 2 (2014), 201–213.

[55] J. A. Ezquerro and M. Á. Hernández. On an application of Newton's method to nonlinear operators with ω-conditioned second derivative, *BIT*, **42** 3 (2002), 519–530.

[56] J. A. Ezquerro and M. Á. Hernández. Generalized differentiability conditions for Newton's method, *IMA J. Numer. Anal*, **22** 1 (2002), 187–205.

[57] J. A. Ezquerro and M. A. Hernández-Verón. *Newton's method: an updated approach of Kantorovich's theory.* Frontiers in Mathematics. Birkhäuser/Springer, Cham, 2017.

[58] J. A. Ezquerro and M. Á. Hernández. Mild differentiability conditions for Newton's method in Banach spaces. Frontiers in Mathematics. Birkhäuser/Springer, Cham, 2020.

[59] J. A. Ezquerro, M. A. Hernández-Verón, Á. A. Magreñán and A. Moysi. A significant improvement of a family of secant-type methods, *Journal of Computational and Applied Mathematics*, **424** (2023), 115002, 1–16.

[60] J. A. Ezquerro, M. A. Hernández-Verón, Á. A. Magreñán and A. Moysi. A procedure to obtain quadratic convergence from the secant method, *Journal of Computational and Applied Mathematics*, **448** (2024), 115912, 1–11.

[61] J.A. Ezquerro, M. A. Hernández, N. Romero and A. I. Velasco. On Steffensen's method on Banach spaces, *Journal of Computational and Applied Mathematics*, **249** (2013), 9–23.

[62] J. A. Ezquerro, M. A.Hernández and M. J. Rubio. Secant-like methods for solving nonlinear integral equations of the Hammerstein type, *Journal of Computational and Applied Mathematics*, **115** 1-2 (2000), 245–254.

[63] N. Fagella. Invariants en dinàmica compleja, *Bulletí de la Societat Catalana de Matemàtiques*, **23** 1 (2007), 29–51.

[64] N. Fagella and X. Jarque. *Iteración compleja y fractales*, Vicens Vives, Barcelona, 2007.

[65] P. Fatou. Sur les équations fonctionelles, *Bulletin de la Société Mathématique de France*, **47** (1919), 161–271.

[66] W. Gander. On Halley's iteration method. The American Mathematical Monthly, **92** 2 (1985), 131–134.

[67] J. Garay and M. Á. Hernández-Verón. *Degree of logarithmic convexity*, Publicaciones del Sem. Mat. García Galdeano. Serie II, **26**, Zaragoza, 1988.

[68] M. García-Olivo. *El método de Chebyshev para el cálculo de las raíces de ecuaciones no lineales.* Universidad de La Rioja, Servicio de Publicaciones, 2013.

[69] W. J. Gilbert. Generalizations of Newton's method, *Fractals*, **9** 3 (2001), 251–262.

[70] W. J. Gilbert. The complex dynamics of Newton's method for a double root, *Computers & Mathematics with Applications*, **22** 10 (1991), 115–119.

[71] A. Giraldo and M. A. Sastre. *Sistemas dinámicos discretos y caos*, Fundación general de la Universidad Politécnica de Madrid, Madrid, 2002.

[72] D. González. *Problemas de valor inicial en la construcción de sucesiones mayorizantes para el método de Newton en espacios de Banach*, Servicio de Publicaciones, Universidad de La Rioja, 2012.

[73] W. B. Gragg and R. A. Tapia. Optimal error bounds for the Newton-Kantorovich theorem, *SIAM Journal on Numerical Analysis*, **11** (1974), 10–13.

[74] M. Grau-Sánchez and J. M. Gutiérrez. Some variants of the Chebyshev-Halley family of methods with fifth-order of convergence, *International Journal of Computer Mathematics*, **87** 4 (2010), 818–833.

[75] M. Grau-Sánchez, M. Noguera and S. Amat. On the approximation of derivatives using divided difference operators preserving the local convergence order of iterative methods, *Journal of Computational and Applied Mathematics*, **237** 1 (2013), 363–372.

[76] J. M. Gutiérrez: *El método de Newton en espacios de Banach*, Servicio de Publicaciones, Universidad de La Rioja, 1995.

[77] J. M. Gutiérrez, Á. A. Magreñán and J. L. Varona. The "Gauss-Seidelization" of iterative methods for solving nonlinear equations in the complex plane, *Applied Mathematics and Computation*, **218** 6 (2011), 2467–2479.

[78] J. M. Gutiérrez and S. Plaza. *Estudio dinámico del método de Newton para resolver ecuaciones no lineales*, Servicio de Publicaciones, Universidad de La Rioja, 2013.

[79] J. M. Gutiérrez, Á. A. Magreñán and J. L. Varona. The "Gauss-Seidelization" of iterative methods for solving nonlinear equations in the complex plane, *Appl. Math. Comp*, **218** (2011), 2467–2479.

[80] J. M. Gutiérrez, Á. A. Magreñán and N. Romero. On the semilocal convergence of Newton-Kantorovich method under center-Lipschitz conditions, *Applied Mathematics and Computation*, **221** (2013), 79–88.

[81] F. von Haeseler and H.-O. Peitgen. Newton's method and complex dynamical systems, *Acta Applicandae Mathematica* , **13** (1988), 3–358.

[82] M. Á. Hernández-Verón. The Newton's method for operators with Hölder continuos first derivative, *Journal of Optimization Theory and Applications*, **109** (2001), 631–648.

[83] M. A. Hernández-Verón, Á. A. Magreñán and M. J. Rubio. Dynamics and local convergence of a family of derivative-free iterative methods, *Journal of Computational and Applied Mathematics*, **354** (2019), 414–430.

[84] M. A. Hernández-Verón and M. J. Rubio. Secant-like methods for solving nonlinear integral equations of the Hammerstein type. *Journal of Computational and Applied Mathematics*, **115** (2000), 245–254.

[85] M. A. Hernández-Verón and M. J. Rubio. A uniparametric family of iterative methods for solving nondifferentiable equations, *Journal of Mathematical Analysis and Applications*, **275** (2002), 821–834.

[86] M. A. Hernández-Verón and M. J. Rubio. On the ball of convergence of secant-like methods for non-differentiable operators, *Applied Mathematics and Computation*, **273** (2016), 506–512.

[87] M. A. Hernández-Verón, M. J. Rubio and J. A. Ezquerro. Solving a special case of conservative problems by secant-like methods, *Applied Mathematics and Computation*, **169** 2 (2005), 926–942.

[88] M. A. Hernández-Verón and M. A. Salanova *La convexidad en la resolución de ecuaciones escalares no lineales.* Universidad de la Rioja, 2011.

[89] M. A. Hernández-Verón, N.Yadav, Á. A. Magreñán and E. Martínez. An improvement of the Kurchatov method by means of a parametric modification, *Mathematical Methods in the Applied Sciences*, **45** 11 (2022), 1–17.

[90] R. A. Homlgren. *A first course in discrete dynamical systems, second edition.* Springer-Verlag, 1996.

[91] R. Hongmin and W. Qingiao. The convergence ball of the Secant method under Hölder continuous divided differences, *Journal of Computational and Applied Mathematics*, **194** (2006), 284–293.

[92] J. Hubbard, D. Schleider and S. Sutherland. How to find all roots of complex polynomials by Newton's method, *Inventiones mathematicae* , **146** (2001), 1–33.

[93] G. Julia. Memoire sur l'iteration des fonctions rationelles, *Journal de Mathématiques pures et appliquées*, **8** 1 (1918), 47–215.

[94] L. V. Kantorovich. On Newton's method for functional eqations, *Dokl. Akad. Nauk SSSR*, **59** (1948), 1237–1240.

[95] L. V. Kantorovich. The majorant principle and Newton's method, *Dokl. Akad. Nauk SSSR*, **76** (1951), 17–20.

[96] L. V. Kantorovich. The method of successive approximations for functional analysis, *Acta Mathematica*, **71** (1939), 63–97.

[97] L. V. Kantorovich and G. P. Akilov. *Functional analysis*, Pergamon Press, Oxford, 1982.

[98] L. Kewei Homocentric convergence ball of the Secant method, *Applied Mathematics-A Journal of Chinese Universities*, **22** 3 (2007), 353–365.

[99] K. Kneisl. Julia sets for the super-Newton method, Cauchy's method and Halley's method, *Chaos*, **11** 2 (2001), 359–370.

[100] K. A. Kurchatov. On a method of linear interpolation for the solution of functional equations, *Dokl. Akad. Nauk SSSR*, **198** 3 (1971), 524–526;

[101] H.-C. Lai and P.-Y. Wu. Error bounds of Newton type process on Banach spaces, *Numerische Mathematik*, **39** (1982), 175–193.

[102] T. Lotfi, Á. A. Magreñán, K. Mahdiani and J. Rainer. A variant of Steffensen-King's type family with accelerated sixth-order convergence and high efficiency index: Dynamic study and approach. *Applied Mathematics and Computation*, **252** (2015), 347–353.

[103] Á. A. Magreñán. *Estudio de la dinámica del método de Newton amortiguado.* Universidad de La Rioja, 2013

[104] Á. A. Magreñán. Different anomalies in a Jarratt family of iterative root-finding methods. *Applied Mathematics and Computation*, **233** (2014), 29–38.

[105] Á. A. Magreñán. A new tool to study real dynamics: The convergence plane. *Applied Mathematics and Computation*, **248** (2014), 215–224.

[106] Á. A. Magreñán and I. K. Argyros. *A Contemporary Study of Iterative Methods.* Academic Press, Elsevier, 2018.

[107] Á. A. Magreñán and J. M. Gutiérrez. Real dynamics for damped Newton's method applied to cubic polynomials. *Journal of Computational and Applied Mathematics*, **275** (2015), 527–538.

[108] B. Mandelbrot. *The fractal geometry of nature*, W. H. Freeman & Co., 1977.

[109] G. J. Miel. The Kantorovich theorem with optimal error bounds, *The American Mathematical Monthly*, **86** (1979), 212–215.

[110] G. J. Miel. Unifid error analysis for Newton-type methods, *Numerische Mathematik*, **33** (1979), 391–396.

[111] G. J. Miel. Majorizing sequences and error bounds for iterative methods, *Mathematics of Computation*, **34** (1980), 185–202.

[112] J. Milnor. *Dynamics in one complex variable: Introductory lectures. Third edition*, Princeton University Press, Princeton, New Jersey, 2006.

[113] F. Mirzaee and S. Alipour. An efficient cubic B-spline and bicubic B-spline collocation method for numerical solutions of multidimensional nonlinear stochastic quadratic integral equations. *Mathematical Methods in the Applied Sciences*, **43** 1 (2020), 384–397.

[114] F. Mirzaee and N. Samadyar. Explicit representation of orthonormal Bernoulli polynomials and its application for solving Volterra–Fredholm–Hammerstein integral equations, *SeMA Journal*, 77 (2020), 81–96.

[115] F. Mirzaee, E. Solhi and N. Samadyar. Moving least squares and spectral collocation method to approximate the solution of stochastic Volterra-Fredholm integral equations, *Applied Numerical Mathematics*, **161** (2021), 275–285.

[116] P. Montel. *Leçons sur les familles normales des fonctions analytiques et leurs applications*, Gauthier-Villars, Paris, 1927.

[117] A. Moysi, J.A. Ezquerro, M. A. Hernández and Á. A. Magreñán. On the set of initial guesses for the secant method. *Mathematical Methods in the Applied Sciences*, **48** 27 (2025), 7398–7411.

[118] I. P. Mysosvskikh. On convergence of L. V. Kantorovich's method for functional equations and its applications, *Dokl. Akad. Nauk SSSR*, **70** (1950), 565–568.

[119] J. W. Neuberger. Continuous Newton's Method for polynomials, *The Mathematical Intelligencer*, **21** 2 (1999), 18–23.

[120] P. Niamsup. Dynamics of Newton's functions of Barna's polynomials, *International Journal of Mathematics and Mathematical Sciences*, **29** 2 (2002), 79–84.

[121] K. Nishizawa and M. Fujimura. Families of rational maps and convergence basins of Newton's method, *Proceedings of the Japan Academy*, **68** Ser. A (1992), 143-147.

[122] J. M. Ortega. The Newton-Kantorovich theorem, *The American Mathematical Monthly*, **75** (1968), 658–660.

[123] J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*, Academic Press, New York, 1970.

[124] A. M. Ostrowski. Solution of equations and systems of equations, *Academic Press*, New York, 1966.

[125] A. M. Ostrowski. *Solution of equations in euclidean and Banach spaces*, Academic Press, New York, 1973.

[126] H.-O. Peitgen, D. Saupe and F. von Haeseler. Cayley's problem and Julia Sets, *The Mathematical Intelligencer*, **6** 2 (1984), 3–20.

[127] H.-O. Peitgen, H. Jürgens and D. Saupe. *Chaos and Fractals. New Frontiers of Science. Second Edition*, Springer-Verlag, New York, 2004.

[128] S. Plaza and N. Romero. Attracting cycles for the relaxed Newton's method, *Journal of Computational and Applied Mathematics*, **235** 10 (2011), 3238–3244.

[129] S. Plaza. Dinámica del método de Newton, *Actas coloquios nacionales de Sistemas Dinámicos (1991)*, (1992), 82–119.

[130] S. Plaza. Conjugacies classes of some numerical methods, *Proyecciones*, **20** 1 (2001), 1–17.

[131] B. T. Polyak. Newton-Kantorovich method and its global convergence, *Journal of Mathematical Sciences*, **133** 4 (2006), 1513–1523.

[132] A. D. Polyanin and A. V. Manzhirov. *Handbook of integral equations.* CRC Press, Boca Raton, 1998.

[133] F. A. Potra and V. Pták. Sharp error bounds for Newton's process, *Numerische Mathematik*, **34** (1980), 63–72.

[134] V. Pták. The rate of convergence of Newton's process, *Numerische Mathematik*, **25** (1976), 279–285.

[135] L. B. Rall. Quadratic equations in Banach spaces, *Rendiconti del Circolo Matematico di Palermo*, **10** (1961), 314–332.

[136] L. B. Rall. *Computational Solution of Nonlinear Operator Equations.* Robert E. Krieger Publishing Company, Inc., California, 1979.

[137] S. Regmi. *Optimized iterative methods with applications in diverse disciplines.* Nova Science, New York, 2021.

[138] S. Regmi. *Improved error estimates for some Newton-type methods.* In: Argyros, M. I., Argyros, I. K. and Regmi, S., (ed.) Hilbert space and its applications, Nova Science, New York, 2021.

[139] H. Ren and I. K. Argyros. Local convergence of efficient Secant-type methods for solving nonlinear equations, *Applied Mathematics and computation*, **218** (2012), 7655–7664.

[140] W. C. Rheinboldt. An adaptative continuation process for solving systems of nonlinear equations. *Banach Center Publications*, vol. 3 (1977), 129–142.

[141] G. H. Roberts and J. Horgan-Kobleski. Newton's versus Halley's method: A dynamical system approach, *International Journal of Bifurcation and Chaos*, **14** 10 (2004), 3459–3475.

[142] N. Romero. *Familias paramétricas de procesos iterativos de alto orden de convergencia*, Servicio de Publicaciones, Universidad de La Rioja, 2006.

[143] M.J. Rubio. *Procesos iterativos definidos mediante diferencias divididas.* Universidad de La Rioja, 1996

[144] R. Russo. *Método de la secante.* Cátedra de Métodos Numéricos. Universidad Nacional del Mar del Plata.

[145] J. Saberi-Nadja and M. Heid. Solving nonlinear integral equations in the Urysohn form by Newton-Kantorovich-quadrature method, *Computers $ Mathematics with Applications*, **60** (2010), 2018–2065.

[146] G. Saunder. *Iteration of rational functions in one complex variable and basins of attractive points*, Ph.D. Thesis, University of California, Berkeley, 1984.

[147] S. M. Shakhno. On a Kurchatov's method of linear interpolation for solving nonlinear equations. *PAMM: Proceedings in Applied Mathematics and Mechanics*, **4** (2004), 650–651.

[148] S. M. Shakhno. On the Secant method under generalized Lipschitz conditions for the divided difference operator, *PAMM: Proceedings in Applied Mathematics and Mechanics*, **7** (2007), 2060083–2060084.

[149] M. Shishikura. The connectivity of the Julia set and fixed points, *Complex dynamics: families and friends (Ed. by D. Schleicher), A. K. Peters*, (2009), 257–276.

[150] C. L. Siegel. Iteration of analytic functions, *Annals of Mathematics*, **43** (1942), 607–612.

[151] S. Smale. Newton's method estimates from data at one point, *R. Ewing, K. Gross, C. Martin (Eds.), The Merging of Disciplines: New Directions in Pure, Applied and Computational Mathematics, Springer, New York*, (1986), 185–196.

[152] S. Sutherland. *An Introduction to Julia and Fatou Sets* Springer Proceedings in Mathematics and Statistics, 2014.

[153] D. Sullivan. Quasi conformal homeomorphisms and dynamics I. Solution of Fatou-Julia problem wandering domains, *Annals of Mathematics*, **122** 2 (1985), 401–418.

[154] J. A. Terán and C. M. Rúa. *El método de Newton para raíces complejas. Fractales en el problema de Cayley.* Universidad de Colombia, 2018.

[155] J. F. Traub. *Iterative methods for the solution of equations*, Prentice Hall, Englewood Cliffs, New Jersey, 1964.

[156] J. L. Varona. Graphic and numerical comparison between iterative methods, *The Mathematical Intelligencer*, **24** 1 (2002), 37–46.

[157] J. L. Varona. Representación gráfica de fractales mediante un programa de cálculo simbólico, *La Gaceta de la RSME*, **6** (2003), 213–230.

[158] J. C. Yakoubsohn. A universal constant for the convergence of Newton's method and an application to the classical homotopy method, *Numerical Algorithm*, **9** 3-4 (1995), 223–244.

[159] J. C. Yakoubsohn. Finding zeros of analytic functions: a theory for secant type methods, *Journal of Complexity*, **15** 2 (1999), 239–281.

[160] T. Yamamoto. Historical developments in convergence analysis for Newton's and Newton-like methods, *Journal of Computational and Applied Mathematics*, **124** (2000), 1–23.

[161] T. Yamamoto. Error bounds for Newton's iterates derived from the Kantorovich theorem, *Numerische Mathematik*, **4** (1986), 91–98.

[162] T. Yamamoto. A method for finding sharp error bounds for Newton's method under the Kantorovich assumptions, *Numerische Mathematik*, **49** (1986), 203–220.

[163] W. Yang. Symmetries of the Julia sets of Newton's method for multiple roots, *Applied Mathematics and Computation*, **217** (2010), 2490–2494.

[164] T. J. Ypma. Historical development of the Newton-Raphson method, *SIAM Rev.*, **37** (1995), 531-551.

[165] J. Walsh. The dynamics of Newton's method for cubic polynomials, *The College Mathematics Journal*, **26** 1 (1995), 22-28.

[166] D. R. Wang and F. G. Zhao. The theory of Smale's point estimation and its applications, *Journal of Computational and Applied Mathematics*, **60** (1995), 253-269.

[167] S. Weerakoon and T. G. I. Fernando. A variant of Newton's method with accelerated third-order convergence. *Applied Mathematics Letters*, **13** (2000), 87–93.

[168] W. Werner. *Some improvements of classical iterative methods for the solution of nonlinear equations.* Numerical Solution of Nonlinear Equations, 2006.

[169] S. Wong. Newton's method and symbolic dynamics, *Proceedings of the American Mathematical Society*, **91** (1984), 245–253.

[170] Zuse Institute Berlin. Software repository for Peter Deuflhards Book "Newton Methods for Nonlinear Problems-Affine Invariance and Adaptative Algorithms", `http://www.zib.de/Numerik/NewtonLib/index.en.html`