

Solución de Ecuaciones y Sistemas de Ecuaciones no Lineales por medio del Cálculo

Fraccionario y Algoritmos Heurísticos

Andrés Camilo Rincón Santana, Juan Manuel Cardona Erazo y Daniel Gustavo Rojas Hernández

Trabajo de Grado para Optar al Título de Ingenieros Electrónicos

Director

Carlos Rodrigo Correa Cely

Doctor en Ciencias e Ingeniería de Polímeros

Universidad Industrial de Santander

Facultad de Fisicomecánicas

Escuela de Ingeniería Eléctrica, Electrónica y de Telecomunicaciones (E3T)

Ingeniería Electrónica

Bucaramanga

2024

### **Dedicatoria**

A Dios por dotarme de entendimiento y perseverancia para cumplir con los objetivos de este arduo camino, y por concederme la sabiduría necesaria para alcanzar este logro. A mi familia, expreso mi profunda gratitud por su apoyo incondicional, su amor fraternal y su infinita paciencia. Valoro enormemente el esfuerzo y el sacrificio que han dedicado a lo largo de este proceso.

Andrés Camilo Rincón Santana

A Dios por bendecirme con salud, sabiduría y brindarme la fortaleza para afrontar los retos que se han venido presentando. A mi familia quienes han estado presentes en cada etapa de mi formación y de los cuales nunca me ha faltado su apoyo incondicional. A mis compañeros y maestros por ser parte de este bello proceso en el que me he forjado superando desafíos, compartiendo éxitos y momentos de alegría.

Juan Manuel Cardona Erazo

A mi Padre Celestial, con infinita gratitud, por concederme la sabiduría, fortaleza y entendimiento, herramientas esenciales para alcanzar esta nueva meta. A mi mamá, Jennyfer Hernández, y a mi hermana, Sara Rojas, por siempre brindarme su amor incondicional y su apoyo constante.

Daniel Gustavo Rojas Hernández

### **Agradecimientos**

Agradezco al todopoderoso por llenarme de fortaleza para afrontar los momentos difíciles, por dotarme de salud y perseverancia para culminar con éxito este proceso. A mis padres Yolanda y Ernesto por ser mis consejeros, por estar siempre presentes con su amor incondicional y por ser un buen ejemplo para mí y mis hermanas. A mi hermana Maritza por ser mi gran amiga, por acudir a mi llamado cada que lo necesitaba y a mi Gabi hermosa por ser mi motivación y el motor de mi vida, cuya sonrisa llena mis días de alegría y felicidad. Adicionalmente a mis amigos y colegas que me acompañaron en las jornadas de estudio, a mis docentes y a cada persona que contribuyo para el cumplimiento de este éxito en mi vida.

Andrés Camilo Rincón Santana

Agradezco profundamente a Dios por brindarme la iluminación y fortaleza necesarias para salir adelante a pesar de las adversidades. A mi familia quienes han brindado su apoyo incondicional en cada momento de mi vida y se han mantenido unidos sin importar las circunstancias, especialmente a mis abuelos Roque Acosta y Gladys Marina Paz quienes me dieron la mejor crianza sin la cual esto no habría sido posible. A mi madre Jackeline Erazo por su amor infinito y comprensión. A mi padre Fredy Cardona por su ejemplo. A mi pareja María Alejandra Alba por su compañía e inspiración para no rendirme. A nuestro director el DR. Carlos Rodrigo Correa por permitirnos realizar este bonito trabajo bajo su excelente supervisión. A mis compañeros y maestros por ser parte de este bello proceso en el que me he forjado compartiendo éxitos y momentos de alegría.

Juan Manuel Cardona Erazo

Agradezco a mi Padre Celestial por brindarme sabiduría y entendimiento, los cuales me sirvieron para alcanzar esta nueva meta. Gracias por darme la fortaleza y por siempre acompañarme en los momentos más difíciles que tuve que enfrentar en este proceso. Toda la gloria y la honra para mi Padre Celestial. A mi mamá, Jennyfer Hernández, y a mi abuelita, Carmenza Hernández, quienes siempre estuvieron apoyándome y brindándome todo su amor para poder seguir adelante. A mi amigo Yonatan Felipe Hernández Moreno, que me acompaña desde el cielo, pero fue mi compañero de lucha en este gran camino; siempre estuvo para brindarme sus consejos y una sonrisa. A mis compañeros y maestros, gracias por brindarme su conocimiento, el cual me ayudó a cumplir este logro en mi vida.

Daniel Gustavo Rojas Hernández

## Tabla de Contenido

	<b>Pág.</b>
Introducción.....	13
1. Objetivos del trabajo de grado.....	14
1.1 Objetivo General.....	14
1.2 Objetivos Específicos.....	14
2. Conceptos previos.....	15
2.1 Particle Swarm Optimization (PSO).....	15
2.2 Cálculo Fraccionario (CF).....	16
2.2.1 Derivada Fraccionaria de Caputo.....	17
2.2.2 Derivada Fraccionaria de Caputo-Fabrizio.....	17
2.3 PSO con velocidad fraccionaria.....	18
2.4 Método de discretización L1.....	20
3. Implementación y resultados.....	21
3.1 Velocidad fraccionaria con Caputo.....	21
3.2 Velocidad fraccionaria con Caputo-Fabrizio.....	23
3.3 Funciones de prueba.....	24
3.3.1 Comparación entre el PSO estándar y el PSO fraccionario (Función de Prueba).....	25
3.4 Sistemas de Ecuaciones.....	35
3.4.1 Comparación entre el PSO estándar y el PSO fraccionario (Sistema de Ecuaciones).....	35

3.5 Refinamiento del algoritmo fraccionario.....	44
3.5.1 Reducción de los coeficientes de aceleración.....	45
3.5.2 Limitación de la velocidad.....	45
3.5.3 Función híbrida.....	46
3.5.4 Comparación procedimiento iterativo.....	47
4. Conclusiones.....	48
5. Recomendaciones.....	50
Referencias.....	52
Apéndices.....	54

### Lista de Tablas

	<b>Pág.</b>
Tabla 1. <i>Función de Prueba Schwefel 2.22</i> .....	25
Tabla 2. <i>Densidad de probabilidad para la función Schwefel 2.22</i> .....	26
Tabla 3. <i>Tiempo de convergencia para la función Schwefel 2.22</i> .....	28
Tabla 4. <i>Boxplot para la función Schwefel 2.22</i> .....	29
Tabla 5. <i>Influencia de h para la función Schwefel 2.22</i> .....	32
Tabla 6. <i>Error porcentual y tiempo de simulación para las funciones de prueba</i> .....	34
Tabla 7. <i>Sistema de ecuaciones 3*3</i> .....	35
Tabla 8. <i>Densidad de probabilidad para Sistema de Ecuaciones 3*3</i> .....	36
Tabla 9. <i>Tiempo de convergencia para Sistema de Ecuaciones 3*3</i> .....	37
Tabla 10. <i>Boxplot para Sistema de Ecuaciones 3*3</i> .....	39
Tabla 11. <i>Evolución de la función objetivo</i> .....	41
Tabla 12. <i>Error porcentual y tiempo de simulación para los sistemas de ecuaciones</i> .....	43

**Lista de Figuras**

	<b>Pág.</b>
Figura 1. <i>Comparación procedimiento iterativo</i> .....	47

**Lista de Apéndices**

	<b>Pág.</b>
Apéndice A. Sistema de Ecuaciones no Lineales 1.....	53
Apéndice B. Funciones de optimización 1.....	53
Apéndice C. Variaciones de parámetros Funciones de Optimización.....	53
Apéndice D. Especificaciones Equipos Utilizado.....	53
Apéndice E. Funciones Sistemas Análisis 2.....	53
Apéndice F. Funciones Sistemas Análisis 3.....	53
Apéndice G. Procedimiento Método L1.....	53
Apéndice H. Funciones de Prueba.....	53

## Glosario

**Adaptación:** capacidad de un algoritmo para ajustarse y encontrar soluciones efectivas en diversos entornos de búsqueda, mejorando su rendimiento frente a cambios en las condiciones del problema.

**Algoritmos Heurísticos:** métodos de búsqueda que emplean estrategias basadas en la experiencia para encontrar soluciones adecuadas a problemas complejos, sin garantizar una solución óptima, pero logrando resultados eficientes en tiempos razonables.

**Aproximación Numérica:** método de resolver problemas matemáticos mediante cálculos numéricos en lugar de soluciones analíticas exactas.

**Cálculo Fraccionario (CF):** generalización del cálculo integral y diferencial que se extiende a ordenes no enteros.

**Exploración:** proceso de buscar nuevas áreas en el espacio de soluciones para identificar posibles mejoras.

**Explotación:** refinamiento y mejora de soluciones ya encontradas para optimizar su calidad.

**Función Objetivo:** función que se optimiza en un problema de optimización, también conocida como función de costo o función de evaluación.

**Optimización:** proceso de encontrar la mejor solución posible dentro de un conjunto de alternativas, esto implica evaluar múltiples opciones y seleccionar la que maximiza o minimiza la función objetivo.

## Resumen

**Título:** Solución de Ecuaciones y Sistemas de Ecuaciones no Lineales por medio del Calculo Fraccionario y Algoritmos Heurísticos\*

**Autor:** Andrés Camilo Rincón Santana, Juan Manuel Cardona Erazo y Daniel Gustavo Rojas Hernández\*\*

**Palabras Clave:** Optimización, calculo fraccionario, PSO, velocidad fraccionaria.

### Descripción:

El tema central de la investigación es la integración del cálculo fraccionario (CF) con el algoritmo de Optimización por Enjambre de Partículas (PSO), con el objetivo de abordar problemas de optimización complejos. La innovación radica en la implementación del CF en el PSO para mejorar su capacidad de búsqueda y precisión en la optimización de funciones objetivo.

El CF ofrece una perspectiva poderosa para modelar fenómenos complejos y capturar características no lineales en las funciones objetivo, mientras que el PSO es efectivo en la optimización de funciones no lineales, pero su rendimiento puede mejorarse mediante la adaptación de la velocidad de las partículas utilizando operadores fraccionarios.

En este proyecto se implementa el CF para definir una nueva expresión para la velocidad de las partículas en el PSO, permitiendo una exploración más eficiente del espacio de soluciones y teniendo en cuenta velocidades del pasado que proporcionan más información para futuras generaciones. El PSO fraccionario resultante combina la capacidad de exploración global del PSO estándar con la flexibilidad y adaptabilidad del CF. Al emplear la velocidad fraccionaria de las partículas, el PSO fraccionario puede navegar de manera más eficiente en paisajes de búsqueda complejos, superando las limitaciones de convergencia prematura y ayudando a encontrar soluciones óptimas o cercanas al óptimo en un tiempo razonable.

---

\* Trabajo de Grado.

\*\* Facultad de ingenierías Fisicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Ingeniería Electrónica. Director: Carlos Rodrigo Correa Cely. Doctor en Ciencias e Ingeniería de Polímeros.

### Abstract

**Title:** Solution of Equations and Systems of Nonlinear Equations through Fractional Calculus and Heuristic Algorithms\*

**Author(s):** Andrés Camilo Rincón Santana, Juan Manuel Cardona Erazo y Daniel Gustavo Rojas Hernández\*\*

**Key Words:** Optimization, fractional calculus, PSO, fractional velocity.

#### Description:

The central topic of the research is the integration of fractional calculus (CF) with the Particle Swarm Optimization (PSO) algorithm, with the aim of addressing complex optimization problems. The innovation lies in the implementation of the CF in the PSO to improve its search capacity and precision in the optimization of objective functions.

CF offers a powerful perspective for modeling complex phenomena and capturing non-linear features in objective functions, while PSO is effective in optimizing non-linear functions, but its performance can be improved by adapting particle velocity using fractional operators.

In this project, CF is implemented to define a new expression for the velocity of particles in the PSO, allowing a more efficient exploration of the solution space and considering past velocities that provide more information for future generations. The resulting fractional PSO combines the global scanning capability of standard PSO with the flexibility and adaptability of CF. By employing the fractional velocity of particles, fractional PSO can more efficiently navigate complex search landscapes, overcoming the limitations of premature convergence and helping to find optimal or near-optimal solutions in a reasonable time.

---

\* Bachelor Thesis.

\*\* Facultad de ingenierías Fisicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Ingeniería Electrónica. Director: Carlos Rodrigo Correa Cely. PhD in Polymer Science and Engineering.

## Introducción

La resolución de sistemas de ecuaciones no lineales es un desafío clave en diversas áreas como la física y la económica, donde los métodos convencionales a menudo quedan atrapados en soluciones locales. Para abordar este problema, se propone combinar el algoritmo PSO con el cálculo fraccionario, que extiende los operadores diferenciales a ordenes fraccionarios, permitiendo una mayor flexibilidad en la exploración del espacio de soluciones.

El presente trabajo sigue la línea de investigación planteada por E.J. Solteiro Pires y colaboradores en su artículo “Particle Swarm Optimization with Fractional-Order Velocity”. En su trabajo, los autores desarrollaron una nueva expresión para la velocidad de las partículas por medio del operador fraccionario de Grünwald-Letnikov, formulación que sirvió de base para nuestra solución con los operadores fraccionarios de Caputo y Caputo-Fabrizio. Se trabajó con estos operadores debido a su amplio uso actual en diversas disciplinas, ya que permiten manejar condiciones iniciales de manera eficiente y aportan una estabilidad y precisión superiores en contextos específicos.

La relevancia de este estudio radica en la creciente necesidad de resolver con precisión sistemas no lineales en áreas como el procesamiento de señales, la ingeniería biomédica y la inteligencia artificial. Los avances logrados pueden tener un impacto significativo en decisiones más informadas y en la creación de diseños más eficientes.

Los resultados obtenidos indican que el PSO fraccionario con Caputo-Fabrizio es una opción prometedora, ya que ofrece una rápida convergencia y soluciones cercanas a las esperadas, no obstante, en algunos casos, el PSO estándar sigue siendo el algoritmo más preciso.

Esta modificación del PSO convencional, que combina las matemáticas avanzadas con técnicas computacionales innovadoras, busca contribuir en el avance del conocimiento y ofrecer soluciones efectivas para problemas prácticos.

## **1. Objetivos del trabajo de grado**

En esta sección se presentan los objetivos generales y específicos propuestos en el plan de trabajo de grado. Se exponen la esencia y el propósito del proyecto, estableciendo de forma clara las metas a cumplir.

### **1.1 Objetivo General**

Modificar los métodos tradicionales de optimización heurística para la solución de sistemas de ecuaciones no lineales, donde se implemente una metodología basada en el cálculo fraccionario, con el fin de que se mejore la eficiencia y precisión de las soluciones.

### **1.2 Objetivos Específicos**

1. Investigar la estructura y funcionamiento de los algoritmos de optimización heurísticos, por ejemplo, Particle Swarm Optimization (PSO), en el cual se trabaja con la posición y velocidad de las partículas.

2. Realizar un estudio sobre el cálculo fraccionario, para después incorporarlo en los modelos de optimización heurísticos, como lo es Particle Swarm Optimization (PSO) donde la velocidad ahora será fraccionaria.

3. Desarrollar un entorno de pruebas para la implementación y evaluación de los modelos modificados, llegando así a un análisis de su eficiencia y precisión en la solución de sistemas de

ecuaciones no lineales, comparando los resultados con los modelos convencionales de optimización heurística.

## 2. Conceptos previos

En este capítulo se presentan las definiciones fundamentales que sustentan los conceptos clave para la ejecución del proyecto, destacando las bases teóricas del cálculo fraccionario y su aplicación en la optimización. Se introduce el algoritmo PSO con velocidad fraccionaria, que implementa las derivadas fraccionarias de Caputo y Caputo-Fabrizio. Esta implementación permite actualizar la velocidad de las partículas considerando sus propiedades únicas. Para manejar de manera eficiente las derivadas fraccionarias, se ha empleado el método de aproximación L1. Al incluir las velocidades pasadas de las partículas en el proceso de cálculo, se busca mejorar tanto la exploración como la explotación del algoritmo, optimizando su rendimiento en la búsqueda de soluciones.

### 2.1 Particle Swarm Optimization (PSO)

El PSO es una técnica de optimización inspirado en el comportamiento colectivo de organismos como aves y peces. Utiliza una población de soluciones potenciales, conocidas como partículas, que se mueven a través del espacio de soluciones en busca de la mejor respuesta al problema que se desea resolver. Cada partícula ajusta su posición basándose en su experiencia y en la de las demás, con el objetivo de encontrar la solución óptima (Wang, 2017, p. 387).

El funcionamiento del PSO se basa en dos ecuaciones que actualizan la velocidad y la posición de cada partícula. Estas ecuaciones son:

$$v_{i+1} = wv_i + c_1r_1 \cdot (pbest - x_i) + c_2r_2 \cdot (gbest - x_i),$$

$$x_{i+1} = x_i + v_{i+1},$$

donde:

$v_i$  es la velocidad actual y  $v_{i+1}$  es la velocidad actualizada de la partícula,

$w$  es el peso de inercia,

$c_1$  y  $c_2$  son los coeficientes de aceleración,

$r_1$  y  $r_2$  son valores aleatorios entre 0 y 1,

$x_i$  es la posición actual de la partícula y  $x_{i+1}$  es la posición actualizada de la partícula,

$pbest$  es la mejor posición personal de una partícula,

$gbest$  es la mejor posición global del enjambre.

La actualización de la velocidad consta de tres componentes clave. La primera parte de la fórmula representa la influencia de la velocidad anterior de la partícula, la segunda parte depende de la distancia entre la posición actual de la partícula y su propia posición óptima, llamada elemento cognitivo y la tercera parte se basa en la distancia entre la posición actual de la partícula y la posición óptima global en el enjambre, conocida como el factor social (Wang, 2017, p. 389-390).

## 2.2 Cálculo Fraccionario (CF)

El Cálculo Fraccionario en sus inicios fue considerado como una teoría matemática sin aplicaciones, pero en las últimas décadas se ha producido una explosión de la actividad investigadora en este campo. Fue propuesto inicialmente por Leibnitz y L'Hopital, quienes plantearon derivadas de orden fraccionario como  $1/2$ . A partir de entonces, se desarrollaron diversas definiciones fraccionarias que permiten calcular derivadas e integrales de orden arbitrario (Ozores, 2014, p. 78-79).

A continuación, se describen las definiciones fraccionarias relevantes para este proyecto.

### 2.2.1 Derivada Fraccionaria de Caputo

La derivada fraccionaria de Caputo es un operador esencial en el CF, ampliamente utilizado por su capacidad de manejar condiciones iniciales de una manera más intuitiva, similar a las derivadas clásicas. La derivada fraccionaria de Caputo de una función  $f(x)$  con respecto a  $x$  de orden fraccional  $\alpha$ , denotada como  ${}^c D^\alpha f(x)$ , se define de la siguiente manera:

$${}^c D^\alpha f(x) = \frac{1}{\Gamma(n - \alpha)} \int_0^x \frac{f^{(n)}(\tau)}{(x - \tau)^{\alpha - n + 1}} d\tau,$$

donde:

$\alpha$  es el orden fraccionario de la derivada,

$n$  es el entero más pequeño mayor que  $\alpha$ ,

$f^{(n)}(\tau)$  es la derivada de orden  $n$ -ésimo de la función  $f$  evaluada en  $\tau$ ,

$\Gamma$  es la función Gamma.

Esta expresión calcula la derivada fraccionaria de Caputo ponderando las derivadas ordinarias de la función mediante un núcleo de potencia. A diferencia de otras definiciones de derivadas fraccionarias, la derivada de Caputo emplea las derivadas clásicas, lo que facilita la incorporación de condiciones iniciales estándar.

### 2.2.2 Derivada Fraccionaria de Caputo-Fabrizio

La derivada fraccionaria de Caputo-Fabrizio es una herramienta reciente en el cálculo fraccionario, destacada por modelar fenómenos físicos y procesos dinámicos con memoria, sin singularidades. Introduce un núcleo exponencial, proporcionando una mayor estabilidad y precisión en ciertos contextos. La derivada fraccionaria de Caputo-Fabrizio de una función  $f(x)$  con respecto a  $x$  de orden fraccional  $\alpha$ , denotada como  ${}^{CF} D^\alpha f(x)$ , se define de la siguiente manera:

$${}^{CF}D^\alpha f(x) = \frac{M(\alpha)}{1-\alpha} \int_0^x f^{(n)}(\tau) e^{\left[\frac{-\alpha(x-\tau)}{1-\alpha}\right]} d\tau,$$

donde:

$\alpha$  es el orden fraccionario de la derivada,

$n$  es el entero más pequeño mayor que  $\alpha$ ,

$f^{(n)}(\tau)$  es la derivada de orden  $n$ -ésimo de la función  $f$  evaluada en  $\tau$ ,

$M(\alpha)$  es una función normalizadora que toma valores de 1 para  $\alpha = 0$  y  $\alpha = 1$  (Da Lin, 2021, p. 1035).

La razón por la que  $M(\alpha)$  se asume igual a 1 para estos casos para garantizar que la definición fraccionaria de Caputo-Fabrizio se reduzca a las formas convencionales en los casos límites. Cuando  $\alpha = 0$ , se espera que la definición fraccionaria de Caputo-Fabrizio devuelva la función original  $f(x)$ , mientras que cuando  $\alpha = 1$ , la definición debe reducirse a la derivada ordinaria de primer orden  $f'(x)$ . Para simplificar su implementación en el PSO, a menudo se asume  $M(\alpha) = 1$ , evitando términos adicionales que puedan complicar los cálculos numéricos (Cuahutenango, 2017, p. 44).

### 2.3 PSO con velocidad fraccionaria

En este apartado, se presenta la integración del CF en el algoritmo PSO, introduciendo una nueva formulación para la velocidad de las partículas. Este enfoque tiene como propósito mejorar la capacidad de exploración y explotación del algoritmo, considerando no solo la velocidad actual de las partículas, sino también sus velocidades pasadas.

Se propone un método innovador que integra una metodología basada en el CF en el algoritmo PSO, modificando la ecuación de actualización de velocidad para incluir una derivada de orden fraccionario.

Este enfoque, tiene su punto de partida en la definición fraccionaria de Grünwald-Letnikov, utilizado frecuentemente en implementaciones de tiempo discreto. La derivada fraccionaria de Grünwald-Letnikov se aproxima de la siguiente manera (Solteiro Pires, 2010, p. 296):

$${}^{GL}D^\alpha f(x) \approx \frac{1}{T^\alpha} \sum_{k=0}^r \frac{(-1)^k \Gamma(\alpha + 1) f(x - kh)}{\Gamma(k + 1) \Gamma(\alpha - k + 1)}.$$

En el PSO estándar, la ecuación de actualización de la velocidad se define como:

$$v_{i+1} = v_i + c_1 r_1 \cdot (pbest - x_i) + c_2 r_2 \cdot (gbest - x_i).$$

Esta ecuación puede ser interpretada como la versión discreta de la derivada, la cual puede ser reescrita de la siguiente manera (Solteiro Pires, 2010, p. 297):

$$D^\alpha [v_{i+1}] = c_1 r_1 \cdot (pbest - x_i) + c_2 r_2 \cdot (gbest - x_i).$$

Para implementar el CF, se sustituye en el lado izquierdo de la ecuación por la aproximación de la derivada fraccionaria de Grünwald-Letnikov:

$$\sum_{k=0}^4 \frac{(-1)^k \Gamma(\alpha + 1) v_{i+1-k}}{\Gamma(k + 1) \Gamma(\alpha - k + 1)} = c_1 r_1 \cdot (pbest - x_i) + c_2 r_2 \cdot (gbest - x_i).$$

Realizando la operación de sumatoria y despejando  $v_{t+1}$ , se obtiene la nueva expresión para la velocidad de las partículas.

$$v_{i+1} = \alpha v_i + \frac{1}{2} \alpha (1 - \alpha) v_{i-1} + \frac{1}{6} \alpha (1 - \alpha) (2 - \alpha) v_{i-2} + \frac{1}{24} \alpha (1 - \alpha) (2 - \alpha) (3 - \alpha) v_{i-3} + c_1 r_1 \cdot (pbest - x_i) + c_2 r_2 \cdot (gbest - x_i).$$

Además de los términos del algoritmo PSO estándar, esta nueva ecuación incluye las velocidades pasadas de las partículas, lo que mejora la capacidad del algoritmo para explorar y explotar el espacio de soluciones basándose en la memoria ampliada del enjambre (Solteiro Pires, 2010, p. 298).

## 2.4 Método de discretización L1

El método L1 es una técnica clave en el análisis numérico para resolver ecuaciones diferenciales fraccionarias, facilitando el manejo de los operadores fraccionarios en problemas con dinámicas de memoria y efectos hereditarios. Su relevancia radica en la capacidad para ofrecer aproximaciones precisas y eficientes en estos contextos.

El nombre “Método L1” se refiere a su uso de la interpolación lineal para aproximar derivadas fraccionarias. A diferencia de métodos de orden superior, como L2 o L3, que utilizan interpolaciones cuadráticas o cúbicas respectivamente, el método L1 se basa en una interpolación lineal, ideal para situaciones donde el orden fraccional  $\alpha$  se encuentra en el rango  $0 < \alpha < 1$ . (Barrios, 2018, p. 1092).

Para aplicar el Método L1, se toma como punto de partida la expresión de la derivada fraccionaria de Caputo:

$${}^c D^\alpha f(x) = \frac{1}{\Gamma(1-\alpha)} \int_0^x (x-\tau)^{-\alpha} f'(\tau) d\tau.$$

Siguiendo una serie de pasos que se encuentra detallado en los apéndices y simplificando la expresión, la aproximación por el método L1 para la derivada fraccionaria de Caputo es la siguiente (Qureshi S., 2022):

$${}^c D^\alpha f(x) = \frac{h^{-\alpha}}{\Gamma(2-\alpha)} \sum_{k=0}^{n-1} [f(x_{k+1}) - f(x_k)] [(n-k)^{1-\alpha} - (n-k-1)^{1-\alpha}].$$

Este enfoque también se aplica a la derivada fraccionaria de Caputo-Fabrizio, cuya definición es:

$${}^{CF} D^\alpha f(x) = \frac{M(\alpha)}{1-\alpha} \int_0^x f^{(n)}(\tau) e^{\lambda(x-\tau)} d\tau,$$

donde  $\lambda = \frac{-\alpha}{1-\alpha}$ .

Siguiendo el mismo procedimiento que se realizó para la derivada fraccionaria de Caputo, la aproximación por el método L1 para la derivada fraccionaria de Caputo-Fabrizio es la siguiente:

$${}^{CF}D^\alpha f(x) = \frac{M(\alpha)}{h \cdot \alpha} \sum_{k=0}^{n-1} [f(x_{k+1}) - f(x_k)] [e^{\lambda h(n-k-1)} - e^{\lambda h(n-k)}].$$

Para ambas aproximaciones se puede reducir el error si el parámetro  $h$  también es reducido.

### 3. Implementación y resultados

En esta sección se presenta la implementación de los conceptos básicos para la formulación de las nuevas expresiones de velocidad de las partículas en el algoritmo PSO, utilizando las derivadas fraccionarias de Caputo y Caputo-Fabrizio. Se desarrollaron dos expresiones basadas en estas derivadas, y mediante diversas pruebas se comparó el rendimiento de ambas aproximaciones fraccionarias frente al PSO tradicional. Se explican los procedimientos experimentales, los parámetros empleados y los resultados obtenidos, ofreciendo una visión clara de las ventajas y limitaciones de estas innovaciones.

#### 3.1 Velocidad fraccionaria con Caputo

En esta sección se describe la implementación de la derivada fraccionaria de Caputo en el algoritmo PSO, utilizando la aproximación obtenida por el método L1 para la definición fraccionaria. La expresión inicial de la derivada fraccionaria de Caputo es:

$${}^cD^\alpha f(x) = \frac{h^{-\alpha}}{\Gamma(2-\alpha)} \sum_{k=0}^{n-1} [f(x_{k+1}) - f(x_k)] [(n-k)^{1-\alpha} - (n-k-1)^{1-\alpha}].$$

Para implementar la derivada fraccionaria de Caputo en el PSO, se utiliza la versión discreta de la derivada de la velocidad de las partículas en la siguiente iteración, la cual se describe por medio de la siguiente expresión:

$$D^\alpha[v_{i+1}] = c_1 r_1 \cdot (pbest - x_i) + c_2 r_2 \cdot (gbest - x_i).$$

Aplicando la aproximación de la derivada fraccionaria de Caputo al lado izquierdo de esta ecuación, se obtiene:

$$\begin{aligned} \frac{h^{-\alpha}}{\Gamma(2-\alpha)} \sum_{k=0}^{n-1} [v_{i+k-n+2} - v_{i+k-n+1}] [(n-k)^{1-\alpha} - (n-k-1)^{1-\alpha}] \\ = c_1 r_1 \cdot (pbest - x_i) + c_2 r_2 \cdot (gbest - x_i). \end{aligned}$$

Para  $n = 7$  y después de simplificar, la expresión resultante es la siguiente:

$$\begin{aligned} \frac{h^{-\alpha}}{\Gamma(2-\alpha)} [v_{i+1} + (2^{1-\alpha} - 2)v_i + (3^{1-\alpha} - 2 * 2^{1-\alpha} + 1)v_{i-1} + (4^{1-\alpha} - 2 * 3^{1-\alpha} + 2^{1-\alpha})v_{i-2} \\ + (5^{1-\alpha} - 2 * 4^{1-\alpha} + 3^{1-\alpha})v_{i-3} + (6^{1-\alpha} - 2 * 5^{1-\alpha} + 4^{1-\alpha})v_{i-4} \\ + (7^{1-\alpha} - 2 * 6^{1-\alpha} + 5^{1-\alpha})v_{i-5} + (6^{1-\alpha} - 7^{1-\alpha})v_{i-6}] \\ = c_1 r_1 \cdot (pbest - x_i) + c_2 r_2 \cdot (gbest - x_i). \end{aligned}$$

Despejando  $v_{i+1}$ , se obtiene la nueva expresión para la velocidad fraccionaria del PSO, basada en la Derivada fraccionaria de Caputo. La expresión final para la velocidad es la siguiente:

$$\begin{aligned} v_{i+1} = \frac{\Gamma(2-\alpha)}{h^{-\alpha}} [c_1 r_1 (pbest - x_i) + c_2 r_2 (gbest - x_i)] - (2^{1-\alpha} - 2)v_i \\ - (3^{1-\alpha} - 2 * 2^{1-\alpha} + 1)v_{i-1} - (4^{1-\alpha} - 2 * 3^{1-\alpha} + 2^{1-\alpha})v_{i-2} \\ - (5^{1-\alpha} - 2 * 4^{1-\alpha} + 3^{1-\alpha})v_{i-3} - (6^{1-\alpha} - 2 * 5^{1-\alpha} + 4^{1-\alpha})v_{i-4} \\ - (7^{1-\alpha} - 2 * 6^{1-\alpha} + 5^{1-\alpha})v_{i-5} - (6^{1-\alpha} - 7^{1-\alpha})v_{i-6}. \end{aligned}$$

Esta nueva expresión de actualización de la velocidad tiene incorpora los términos cognitivo y social del PSO estándar, además de varios términos que incluyen la velocidad actual y las velocidades pasadas de las partículas.

### 3.2 Velocidad fraccionaria con Caputo-Fabrizio

Se implemento la derivada fraccionaria de Caputo-Fabrizio en el algoritmo PSO, utilizando la aproximación por el método L1. La expresión de la derivada fraccionaria de Caputo-Fabrizio se describe como:

$${}^{CF}D^\alpha f(x) = \frac{M(\alpha)}{h \cdot \alpha} \sum_{k=0}^{n-1} [f(x_{k+1}) - f(x_k)] [e^{\lambda h(n-k-1)} - e^{\lambda h(n-k)}].$$

De igual manera se parte de la versión discreta de la velocidad en la siguiente iteración, y al lado izquierdo se aplica la aproximación anterior, quedando la expresión de la siguiente manera:

$$\begin{aligned} & \frac{M(\alpha)}{h \cdot \alpha} \sum_{k=0}^{n-1} [v_{i+k-n+2} - v_{i+k-n+1}] [e^{\lambda h(n-k-1)} - e^{\lambda h(n-k)}] \\ & = c_1 r_1 \cdot (pbest - x_i) + c_2 r_2 \cdot (gbest - x_i). \end{aligned}$$

Para  $n = 7$  y después de simplificar, la expresión resultante es la siguiente:

$$\begin{aligned} & \frac{M(\alpha)}{h \cdot \alpha} [(1 - e^{\lambda h})v_{i+1} + (2e^{\lambda h} - e^{2\lambda h} - 1)v_i + (2e^{2\lambda h} - e^{3\lambda h} - e^{\lambda h})v_{i-1} \\ & + (2e^{3\lambda h} - e^{4\lambda h} - e^{2\lambda h})v_{i-2} + (2e^{4\lambda h} - e^{5\lambda h} - e^{3\lambda h})v_{i-3} \\ & + (2e^{5\lambda h} - e^{6\lambda h} - e^{4\lambda h})v_{i-4} + (2e^{6\lambda h} - e^{7\lambda h} - e^{5\lambda h})v_{i-5} \\ & + (e^{7\lambda h} - e^{6\lambda h})v_{i-6}] = c_1 r_1 \cdot (pbest - x_i) + c_2 r_2 \cdot (gbest - x_i). \end{aligned}$$

Despejando  $v_{i+1}$ , se obtiene la nueva expresión para la velocidad fraccionaria del PSO, basada en la derivada fraccionaria de Caputo-Fabrizio. La expresión final para la velocidad es la siguiente:

$$\begin{aligned}
v_{i+1} = & \frac{h\alpha}{M(\alpha)(1 - e^{\lambda h})} [c_1 r_1 (pbest - x_i) + c_2 r_2 (gbest - x_i)] \\
& - \frac{1}{1 - e^{\lambda h}} [(2e^{\lambda h} - e^{2\lambda h} - 1)v_i + (2e^{2\lambda h} - e^{3\lambda h} - e^{\lambda h})v_{i-1} \\
& + (2e^{3\lambda h} - e^{4\lambda h} - e^{2\lambda h})v_{i-2} + (2e^{4\lambda h} - e^{5\lambda h} - e^{3\lambda h})v_{i-3} \\
& + (2e^{5\lambda h} - e^{6\lambda h} - e^{4\lambda h})v_{i-4} + (2e^{6\lambda h} - e^{7\lambda h} - e^{5\lambda h})v_{i-5} \\
& + (e^{7\lambda h} - e^{6\lambda h})v_{i-6}].
\end{aligned}$$

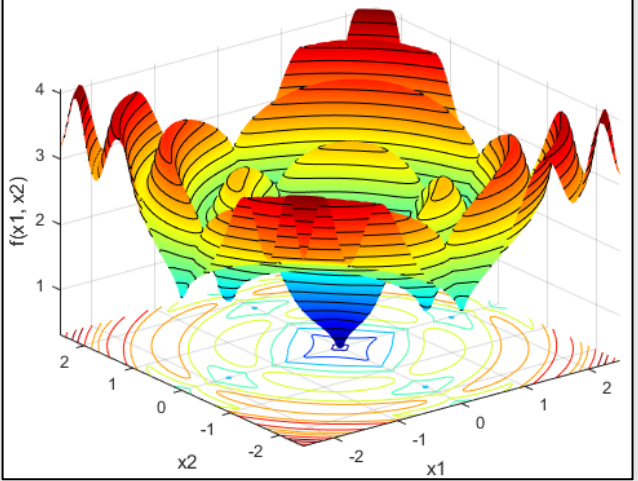
La nueva expresión que actualiza la velocidad de las partículas depende también de la velocidad actual y las velocidades del pasado, así como de los parámetros propios del algoritmo PSO.

### 3.3 Funciones de Prueba

Para evaluar el desempeño de los algoritmos de optimización, se emplearon 15 funciones de prueba conocidas, que presentan diferentes características y niveles de complejidad. Estas funciones permiten analizar la capacidad de los algoritmos para encontrar soluciones óptimas en distintos escenarios. Entre las funciones seleccionadas se encuentran aquellas con múltiples mínimos locales, funciones altamente no lineales y funciones con valles extensos. Esta variedad garantiza una evaluación exhaustiva y comparativa del comportamiento del PSO tradicional y del PSO fraccionario.

En la Tabla 1, se presenta una de las funciones de prueba utilizadas para la realización de las pruebas. Mas funciones junto con sus respectivas graficas se pueden encontrar en los apéndices.

**Tabla 1***Función de Prueba Schwefel 2.22*

Schwefel 2.22	
$f(x, y) = \sqrt{ x } + \sqrt{ y } + (\sin(x^2 + y^2))^2$	
Mínimo global: $(x, y) = (0, 0)$	
Valor en el mínimo global: $f(0, 0) = 0$	
Espacio de búsqueda: $[-10, 10]$	

*Nota:* En la tabla se aprecian las principales características de la función.

### 3.3.1 Comparación entre el PSO Estándar y el PSO Fraccionario (Función de Prueba)

El PSO tradicional utiliza la fórmula estándar para la actualización de la velocidad de las partículas, que corresponde a la primera derivada, es decir, cuando el orden de la derivada  $\alpha$  es igual a 1. En el caso del PSO fraccionario con las definiciones de Caputo y de Caputo-Fabrizio, se introduce el parámetro  $\alpha$ , que varía entre 0 y 1, y permite una exploración más detallada del espacio de búsqueda. Sin embargo, se comprobó que el valor de  $\alpha$  tiene un impacto limitado, predominando la aleatoriedad propia del algoritmo PSO. Por esta razón se optó por trabajar con  $\alpha = 0.7$ , ya que el valor de  $\alpha$  no afecta de manera crítica las soluciones obtenidas por los algoritmos implementados. No se detectó una variación significativa en las soluciones, debido a que el efecto del orden fraccionario es menor en comparación con la influencia de la aleatoriedad del algoritmo PSO.

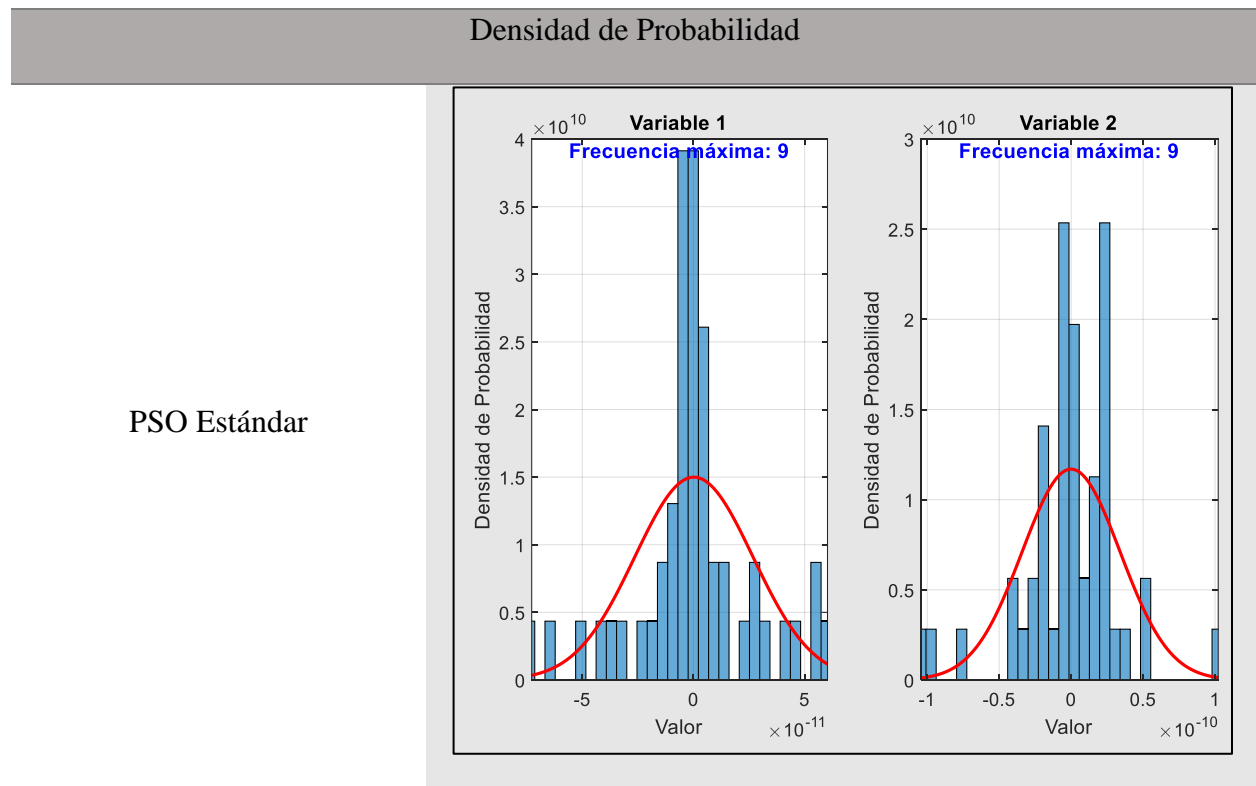
Además, se realizaron las pruebas sobre las funciones con un enjambre con 500 partículas que evolucionaron durante 100 iteraciones y con  $c1 = c2 = 2$ . Estos parámetros fueron definidos con base en la investigación realizada.

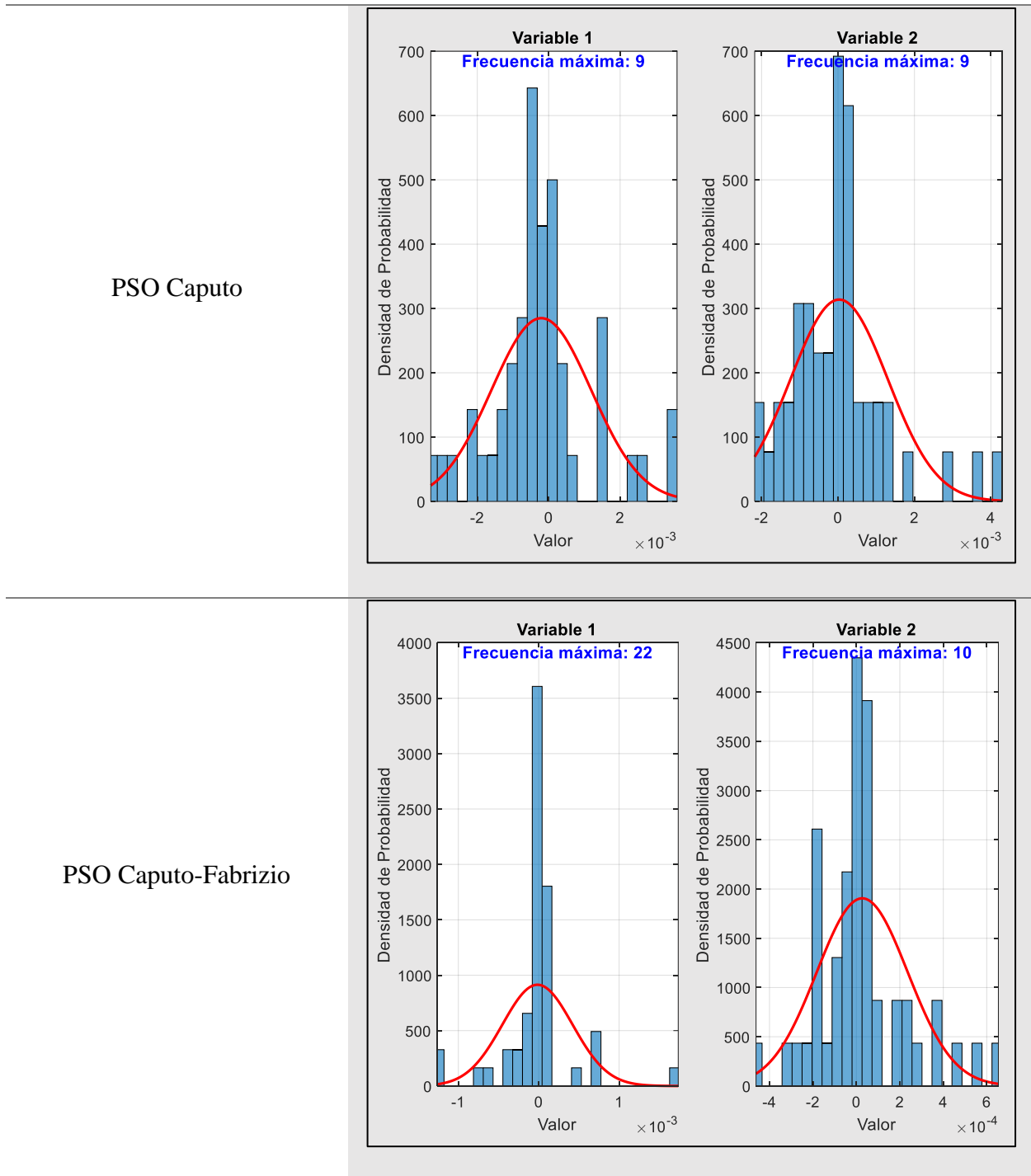
En cada simulación realizada, los resultados obtenidos presentan variaciones, lo que indica la influencia de factores aleatorios, como  $r1$  y  $r2$ , en el desempeño de los algoritmos. Para asegurar la fiabilidad y robustez de los resultados, se decidió realizar un total de 50 simulaciones por cada configuración de algoritmo. Esta estrategia permitió evaluar consistentemente el comportamiento de los algoritmos y seleccionar la mejor opción basada en los resultados de cada simulación.

A continuación, se presentan las pruebas realizadas con el propósito de realizar una comparación entre los tres algoritmos y ver cuál es el comportamiento de cada uno de ellos.

**Tabla 2**

*Densidad de probabilidad para la función Schwefel 2.22*





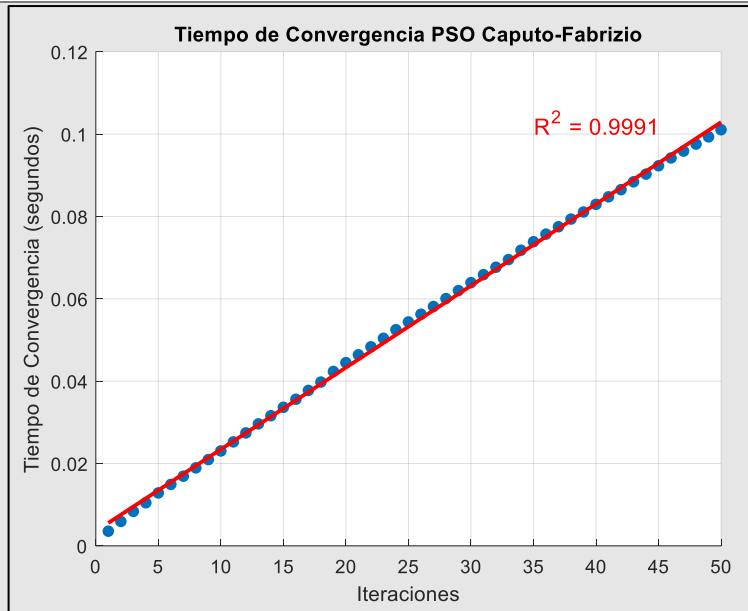
*Nota:* En la tabla se observan las gráficas de densidad de probabilidad de los resultados, para el PSO estándar y sus variantes fraccionarias.

**Tabla 3**

*Tiempo de convergencia para la función Schwefel 2.22*

Tiempo de convergencia	
PSO Estándar	<p style="text-align: center;"><b>Tiempo de Convergencia PSO Estandar</b></p> <p style="text-align: right; color: red;"><math>R^2 = 0.9999</math></p>
PSO Caputo	<p style="text-align: center;"><b>Tiempo de Convergencia PSO Caputo</b></p> <p style="text-align: right; color: red;"><math>R^2 = 1.0000</math></p>

PSO Caputo-Fabrizio



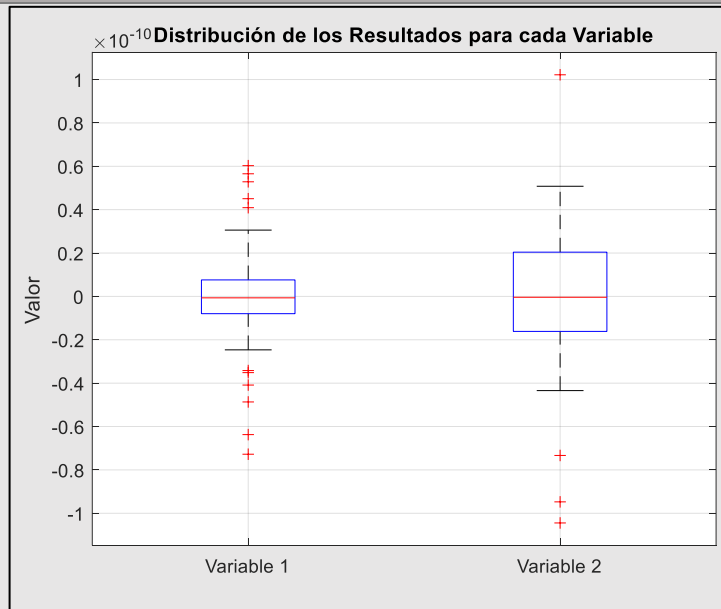
Nota: En la tabla se observan las gráficas de tiempo de convergencia, para el PSO estándar y sus variantes fraccionarias.

**Tabla 4**

Boxplot para la función Schwefel 2.22

PSO Estándar

Boxplot



<p>PSO Caputo</p>	
<p>PSO Caputo-Fabrizio</p>	

*Nota:* En la tabla se observan las gráficas de Boxplot, para el PSO estándar y sus variantes fraccionarias.

Para la función de prueba Schwefel 2.22, el PSO estándar presenta mejor rendimiento frente a los algoritmos fraccionarios, ya que la moda se encuentra en los valores mínimos para

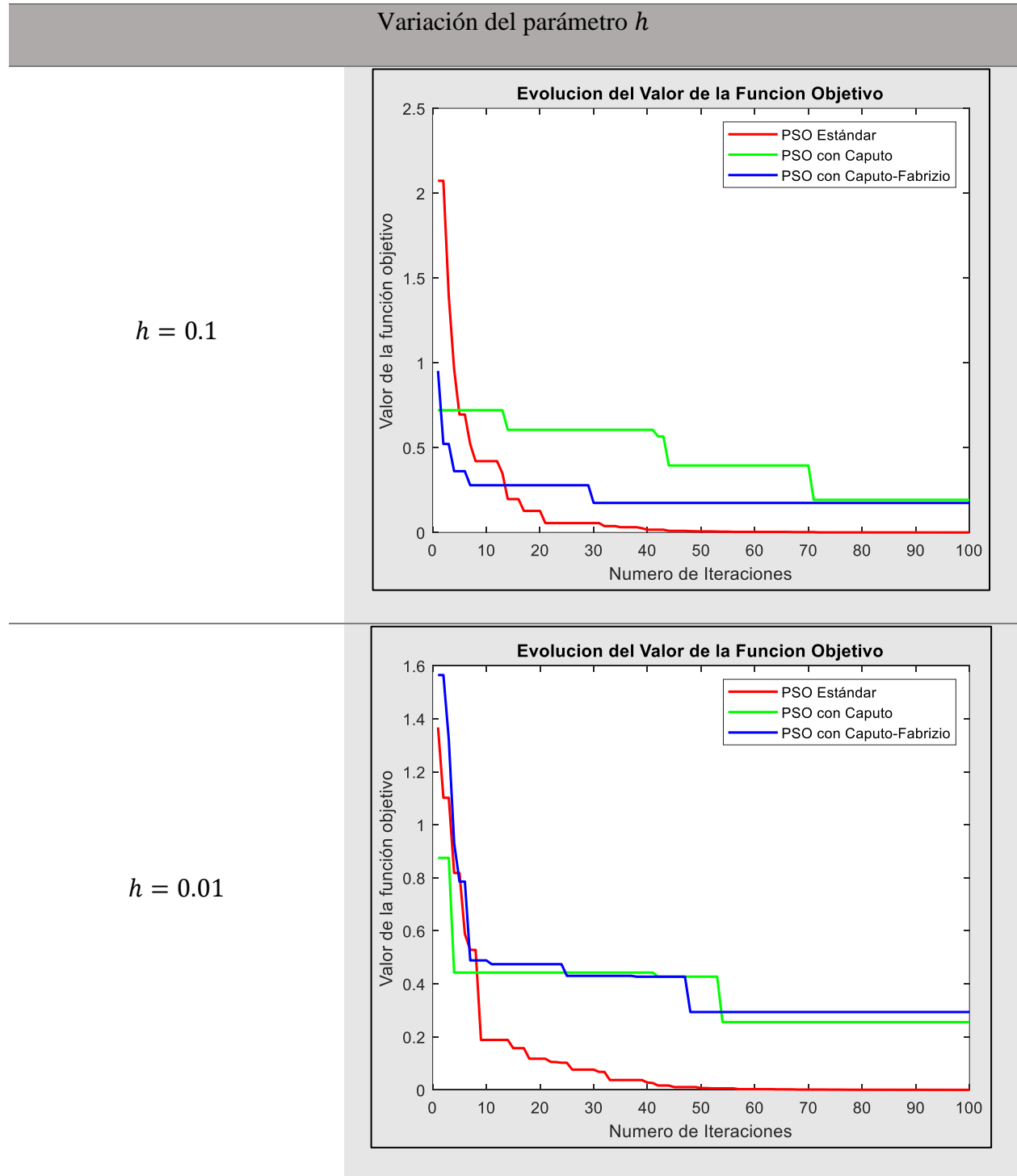
dicha función (0 para ambas variables), lo que predice que la distribución normal concuerda con el histograma, adicionalmente la distribución de los resultados se encuentra en un rango más reducido especialmente para la segunda variable; el rango de valores en el PSO fraccionario es aproximadamente un  $5 \times 10^5\%$  mayor que el rango del PSO estándar, indicando una mayor variabilidad en los resultados obtenidos con el PSO fraccionario. Las variantes fraccionarias del PSO tienen sus soluciones distribuidas en rangos similares, sin embargo, el PSO con Caputo-Fabrizio tiene su moda más centrada sobre el valor esperado (0 para ambas variables) como se aprecia en el Boxplot. En términos de tiempo de convergencia es el PSO con Caputo el algoritmo que presenta mayor velocidad a la hora de converger hacia los valores mínimos de la función, aproximadamente 0.09 segundos, teniendo así que el tiempo de convergencia tanto del PSO estándar como del PSO Caputo-Fabrizio es aproximadamente un 22.22% mayor que el del PSO Caputo, como se aprecia en la Tabla 3.

En los apéndices se pueden encontrar más funciones a las que se le aplicaron las mismas pruebas.

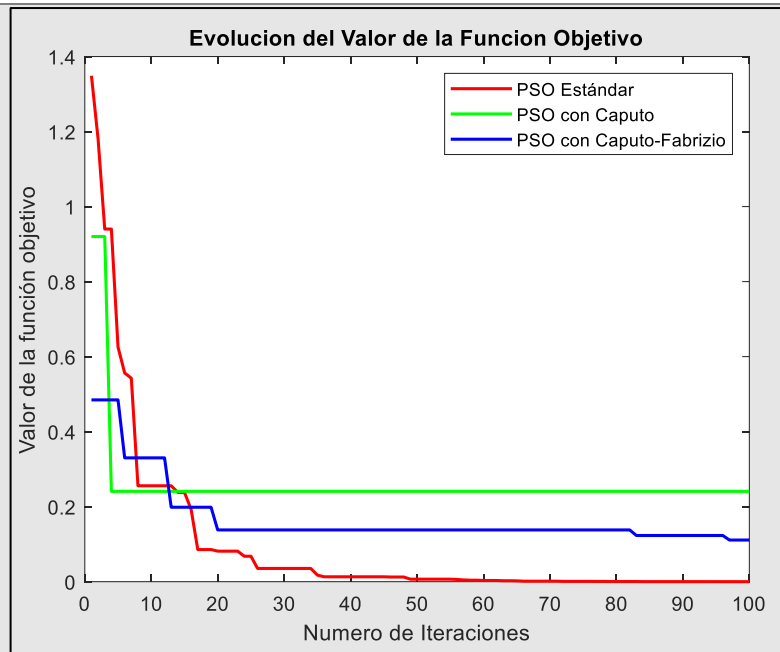
Otro parámetro importante para el PSO fraccionario es el espaciado entre muestras  $h$ , al cual se le asignan distintos valores con el propósito de optimizar su rendimiento. Para las siguientes pruebas se compararon nuevamente los tres algoritmos del PSO. En la Tabla 5 se encuentran consignadas las gráficas resultantes para los siguientes valores  $h = 0.1$ ,  $h = 0.01$  y  $h = 0.001$  y la influencia que tiene sobre las funciones de prueba a la hora de optimizar.

**Tabla 5**

*Influencia de  $h$  para la función Schwefel 2.22*



$$h = 0.001$$



*Nota:* En la tabla se observan las gráficas donde se varía el parámetro  $h$ , para las variantes fraccionarias del PSO.

Con los resultados obtenidos, se puede ver como al reducir el valor del parámetro  $h$ , también se reduce significativamente el número de iteraciones necesarias para la convergencia en el PSO fraccionario. Ambas modificaciones del PSO presentaron una mejoría; después de la reducción del parámetro  $h$ , el algoritmo PSO Caputo mejoro aproximadamente un 92.85% con respecto al número de iteraciones necesarias para converger a la respuesta teóricas y el algoritmo PSO Caputo-Fabrizio presentó un incremento de aproximadamente un 60% con respecto al número de iteraciones necesarias para converger hacia las soluciones esperadas, sin embargo, no logran superar al algoritmo PSO estándar. También se puede observar cómo los PSO fraccionarios convergen alrededor de 0.2, siendo así menos precisos frente al algoritmo convencional. Al realizar estas pruebas se determinó que el valor del espaciado entre muestras sería  $h = 1 * 10^{-6}$ .

También, se determinó el error porcentual de los resultados obtenidos para el valor de la función objetivo y se determinó el tiempo en segundos que le toma a cada algoritmo realizar las 50 simulaciones. Esto lo podemos apreciar en la Tabla 6, para la función mostrada y para las demás funciones que se implementaron:

**Tabla 6**

*Error porcentual y tiempo de simulación para las funciones de prueba*

Algoritmo	PSO Estándar		PSO Caputo		PSO Caputo-Fabrizio		
	Función de prueba	Error P (%)	t (seg)	Error P (%)	t (seg)	Error P (%)	t (seg)
Schwefel 2.22		5.04658E-04	0.4	85.52	1.25	3.001	0.1
Bohachevsky 1	0	7.3E-03	9.2166E-02	5.8E-03	29.60	5.3E-03	
Drop Wave	0	7.7E-04	7.6781E-02	5.5E-03	5.4788E-02	5E-03	
Rosenbrock	0	5.6E-03	1.176329E-02	5.3E-03	3.3483E-02	5.4E-03	
Schaffer N.2	0	5.5E-03	3.79987E-02	4.8E-03	2.1464E-03	4.25E-03	
Schwefel	0	5.7E-03	68.065	4.8E-03	68.065	6.9E-03	
Griewank	0	0.11	14.03	0.093	7.3059	0.087	
Michalewicz	0	0.125	1.9812E-02	0.116	6.57E-05	0.153	
Styblinski-Tang	0	0.135	18.12	0.117	1.468	0.115	
Egg Holder	0	0.1	93.304	0.09	128.1187	0.087	
Beale	0	0.047	1.6598E-05	0.0447	2.225E-03	0.041	
Levy	0	0.0475	1.7115E-03	0.0445	3.350472	0.035	

Three-Hump Camel	0	0.053	2.4953E-08	0.05	3.6597E-07	0.0447
Cross-in-Tray	1.871E-04	0.054	2.8E-06	0.051	1.327E-04	0.045
Bukin N.6	1.1732	0.0445	23.67999	0.0445	14.7404	0.0354

*Nota:* En la tabla se aprecia el error porcentual para cada algoritmo y su tiempo de simulación.

### 3.4 Sistemas de Ecuaciones

Al igual que con las funciones de prueba, se realizaron las mismas verificaciones con un grupo de sistemas de ecuaciones con diferente número de incógnitas, al ser más desafiantes se aumentó el número de partículas a 750, el número de iteraciones a 5000 y el espacio de búsqueda se estableció en un rango de -800 a 800.

En la Tabla 7 se aprecia uno de los sistemas de ecuaciones sobre el que se realizaron las pruebas junto con los valores que se esperan para las incógnitas.

**Tabla 7**

*Sistema de ecuaciones 3\*3*

$6x_1 - 2\cos(x_2x_3) - 1 = 0$ $9x_2 + \sqrt{x_1^2 + \sin(x_3)} + 1.06 + 0.9 = 0$ $60x_3 + 3e^{-x_1x_2} + 10\pi - 3 = 0$	$x_1 \approx 0.4981446$ $x_2 \approx -0.1996059$ $x_3 \approx -0.5288259$
--	---

*Nota:* En la tabla se aprecia el sistema de ecuaciones 3\*3 junto con los valores de las incógnitas esperados del libro de Análisis Numérico 6 Ed.

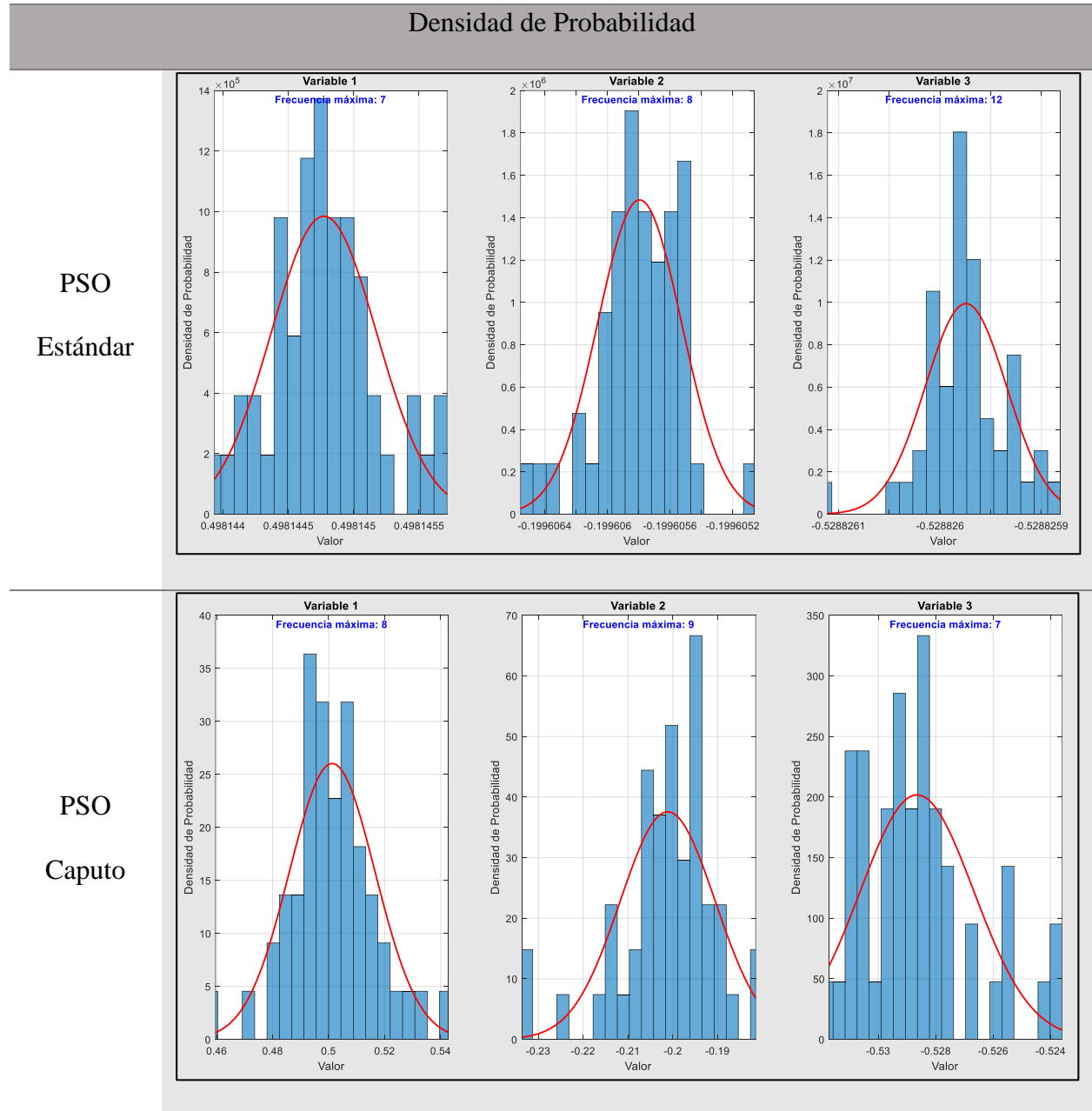
#### 3.4.1 Comparación entre el PSO Estándar y el PSO Fraccionario (Sistema de Ecuaciones)

Para los sistemas de ecuaciones, los algoritmos fraccionarios se trabajan con los parámetros ya dichos anteriormente,  $h = 1 * 10^{-6}$  y  $\alpha = 0.7$ , y se realizaron nuevamente 50 simulaciones.

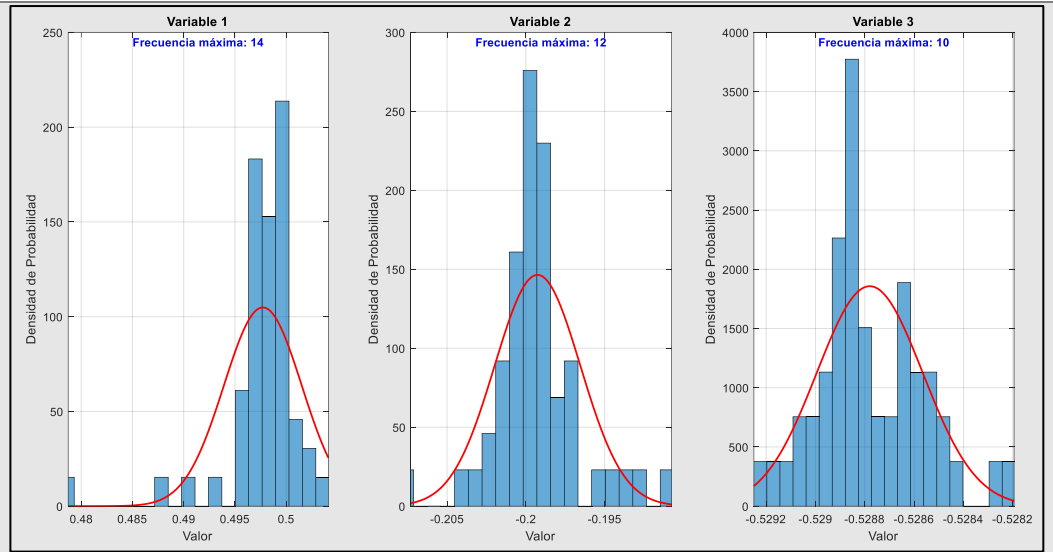
A continuación, se presentan las pruebas realizadas con el propósito de realizar una comparación entre los tres algoritmos y ver cuál es el comportamiento de cada uno de ellos.

**Tabla 8**

*Densidad de probabilidad para Sistema de Ecuaciones 3\*3*



PSO  
Caputo-  
Fabrizio



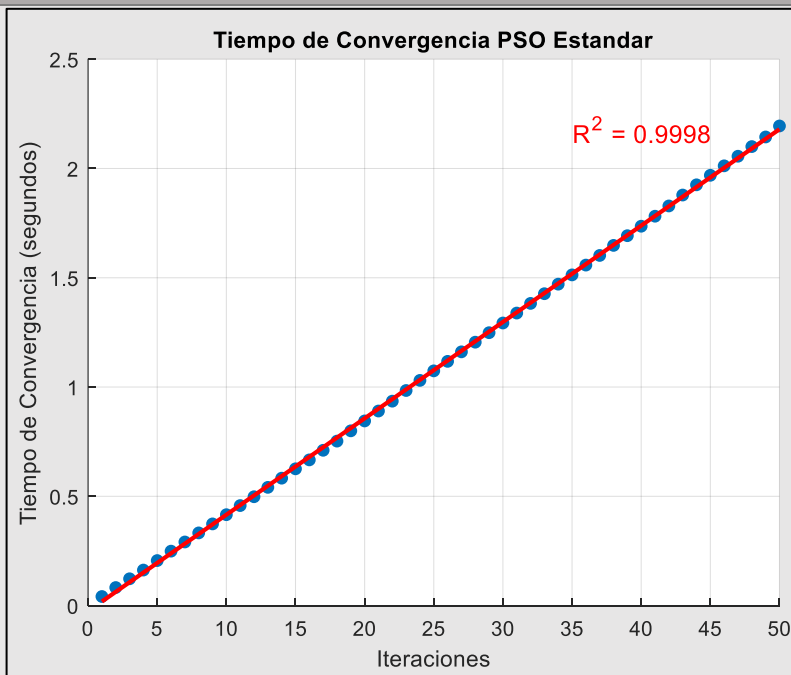
Nota: En la tabla se observan las gráficas de densidad de probabilidad de los resultados, para el PSO estándar y sus variantes fraccionarias.

**Tabla 9**

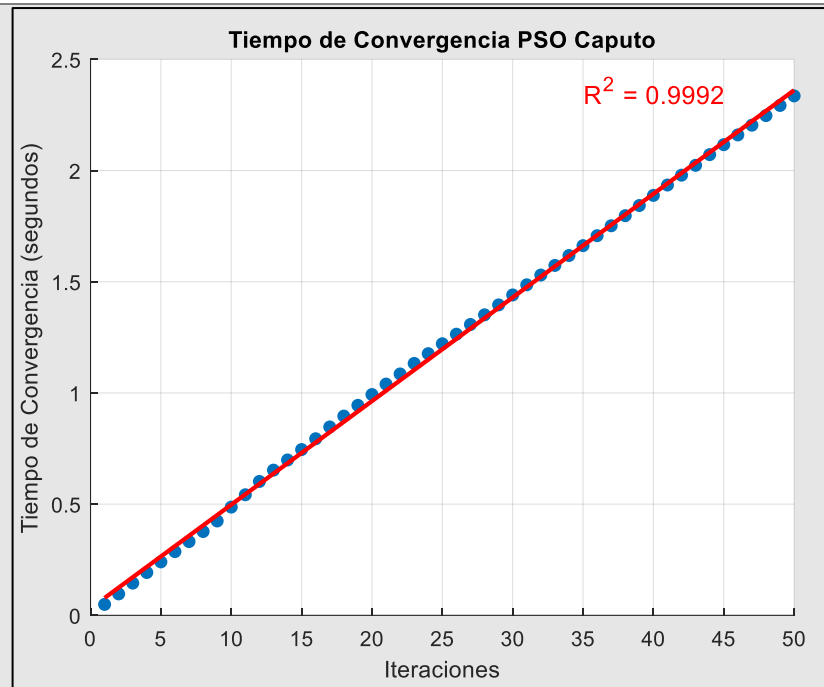
*Tiempo de convergencia para Sistema de Ecuaciones 3\*3*

Tiempo de convergencia

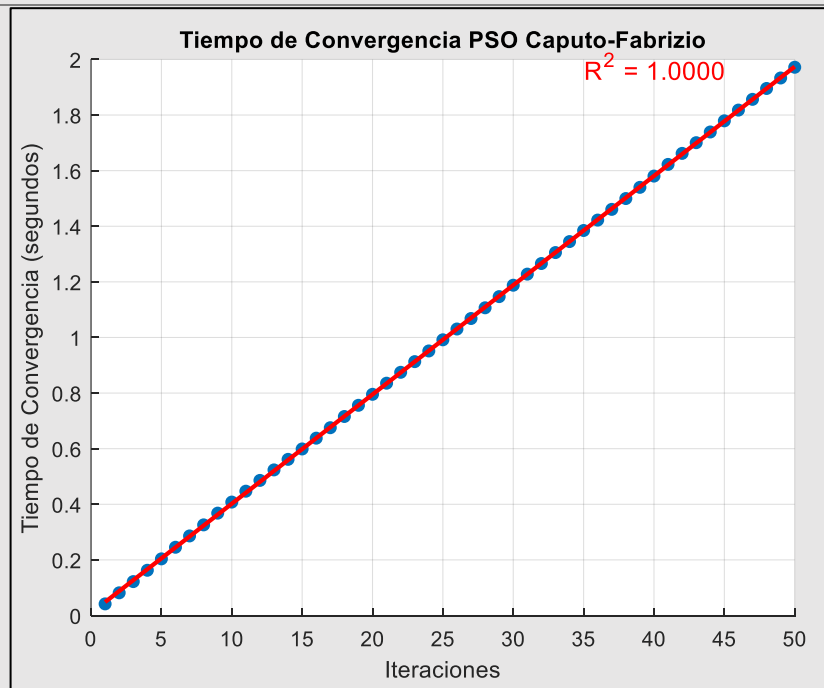
PSO Estándar



PSO Caputo



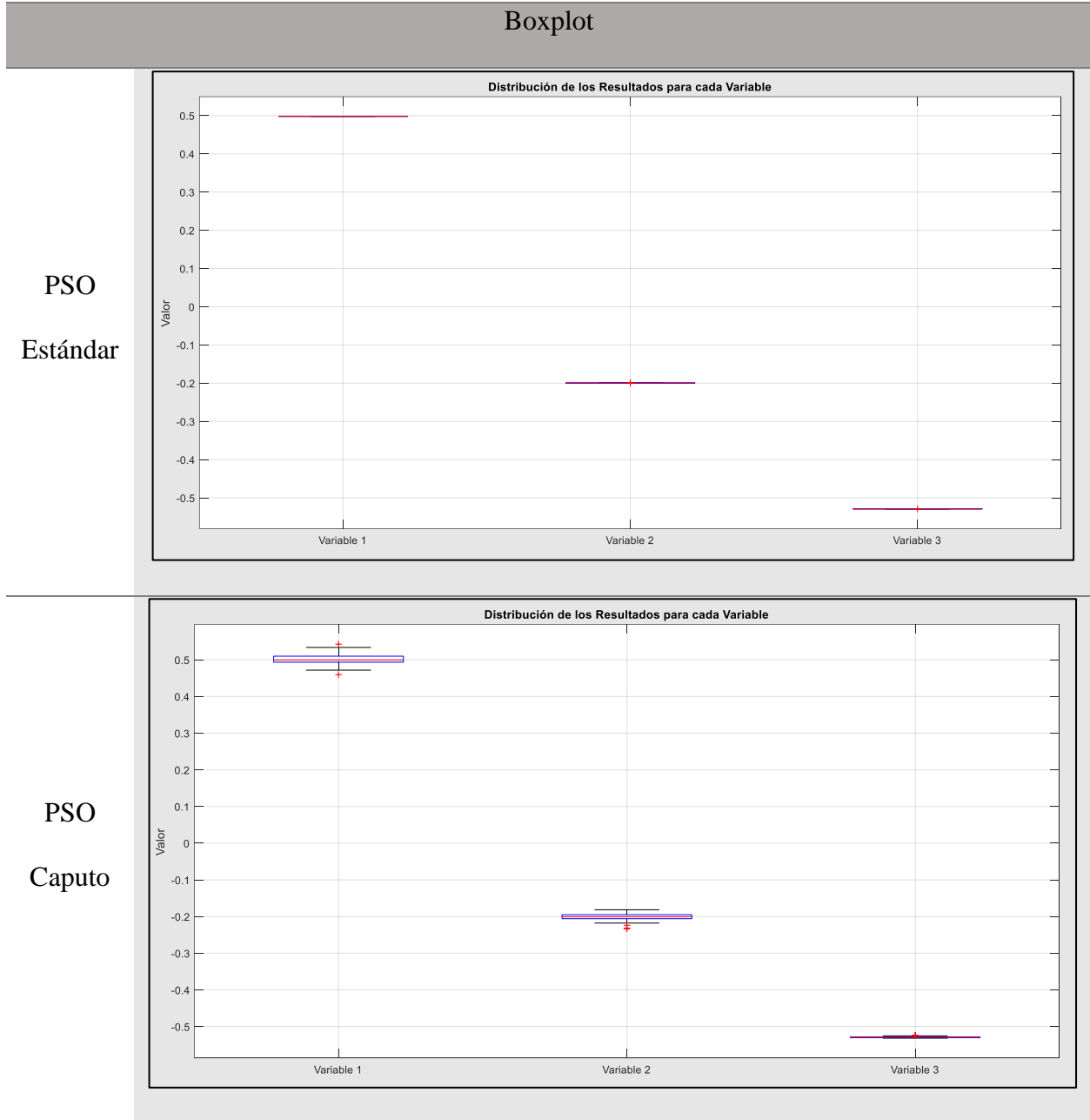
PSO Caputo-Fabrizio

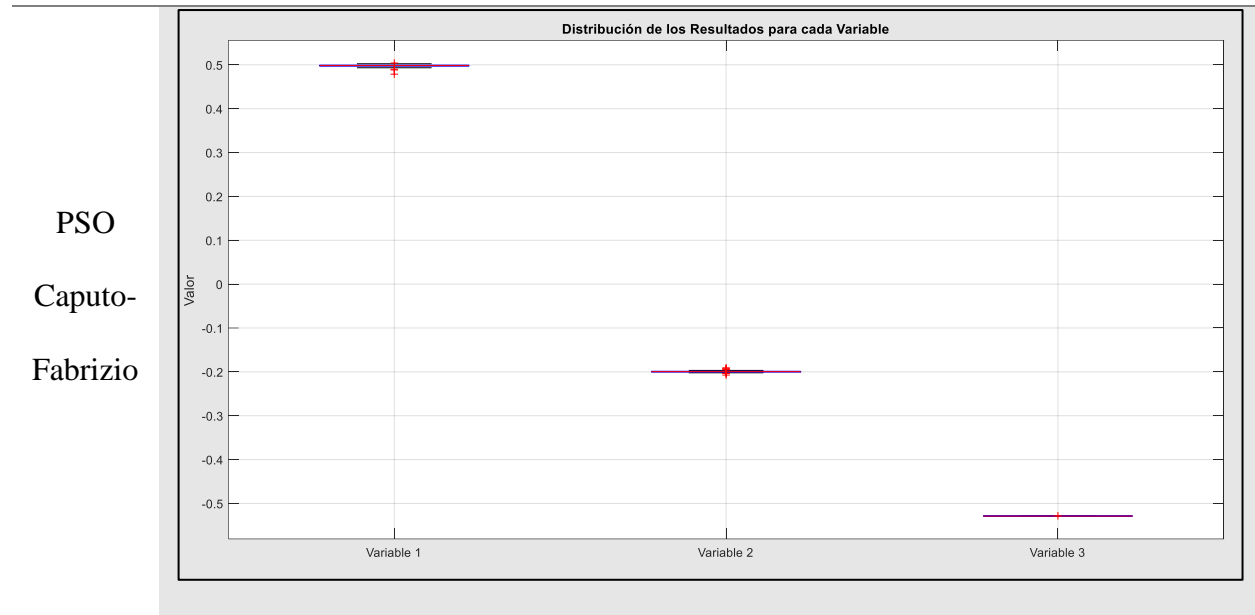


*Nota:* En la tabla se observan las gráficas de tiempo de convergencia, para el PSO estándar y sus variantes fraccionarias.

**Tabla 10**

*Boxplot para Sistema de Ecuaciones 3\*3*





*Nota:* En la tabla se observan las gráficas de Boxplot, para el PSO estándar y sus variantes fraccionarias.

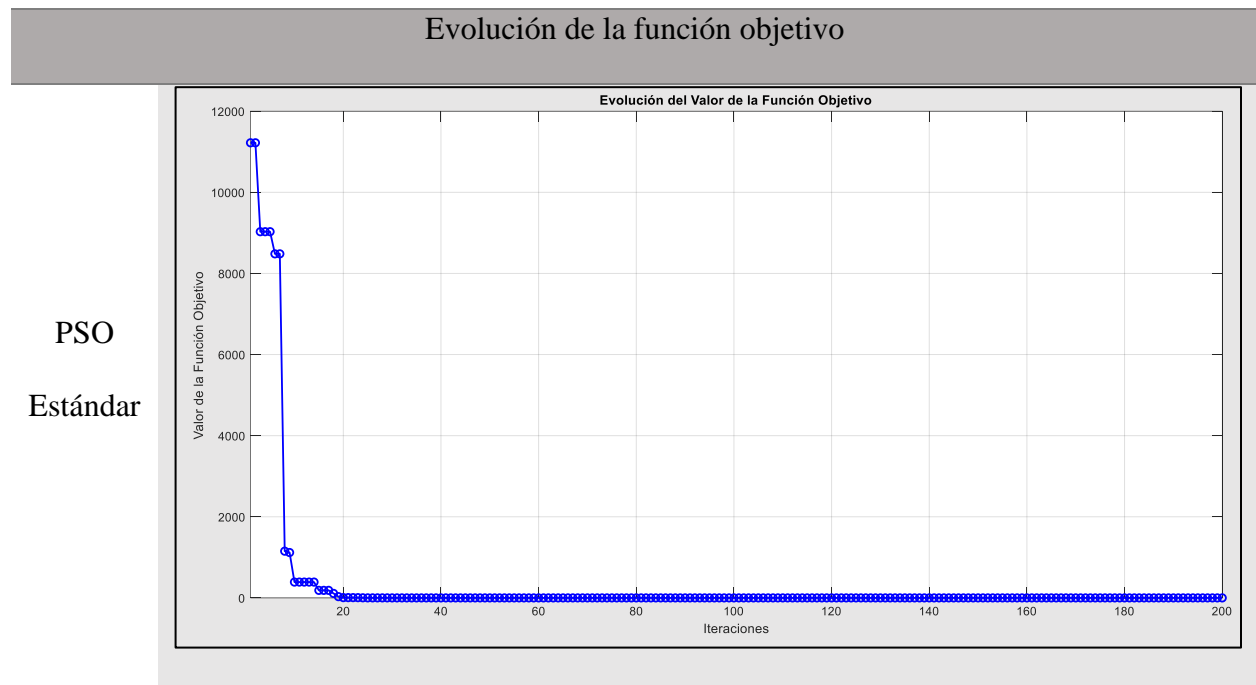
En este caso, se tiene buena variabilidad en los resultados donde la distribución normal predice de buena manera a la distribución de frecuencias y la mediana de las soluciones se encuentra en los valores esperados para cada una de las variables. Para el PSO estándar se tiene una moda pronunciada con una distribución de resultados agrupados en torno a ella, dichos resultados se encuentran en un rango el cual se ajusta perfectamente a los valores esperados. El PSO con Caputo muestra una distribución aceptable de los resultados en torno a la moda la cual está en un rango de valores que se aleja de los resultados esperados, también se observa dispersión en las soluciones, esto debido a la mayor capacidad de exploración con la que cuenta este algoritmo. En el Boxplot se observan algunos valores atípicos para las tres variables, aunque estos se encuentran relativamente próximos a los valores predominantes. En el algoritmo con Caputo-Fabrizio los resultados muestran cierta dispersión y se distribuyen en un rango 9.9% mayor que PSO estándar y 50% menor que PSO Caputo, indicando un equilibrio entre exploración y

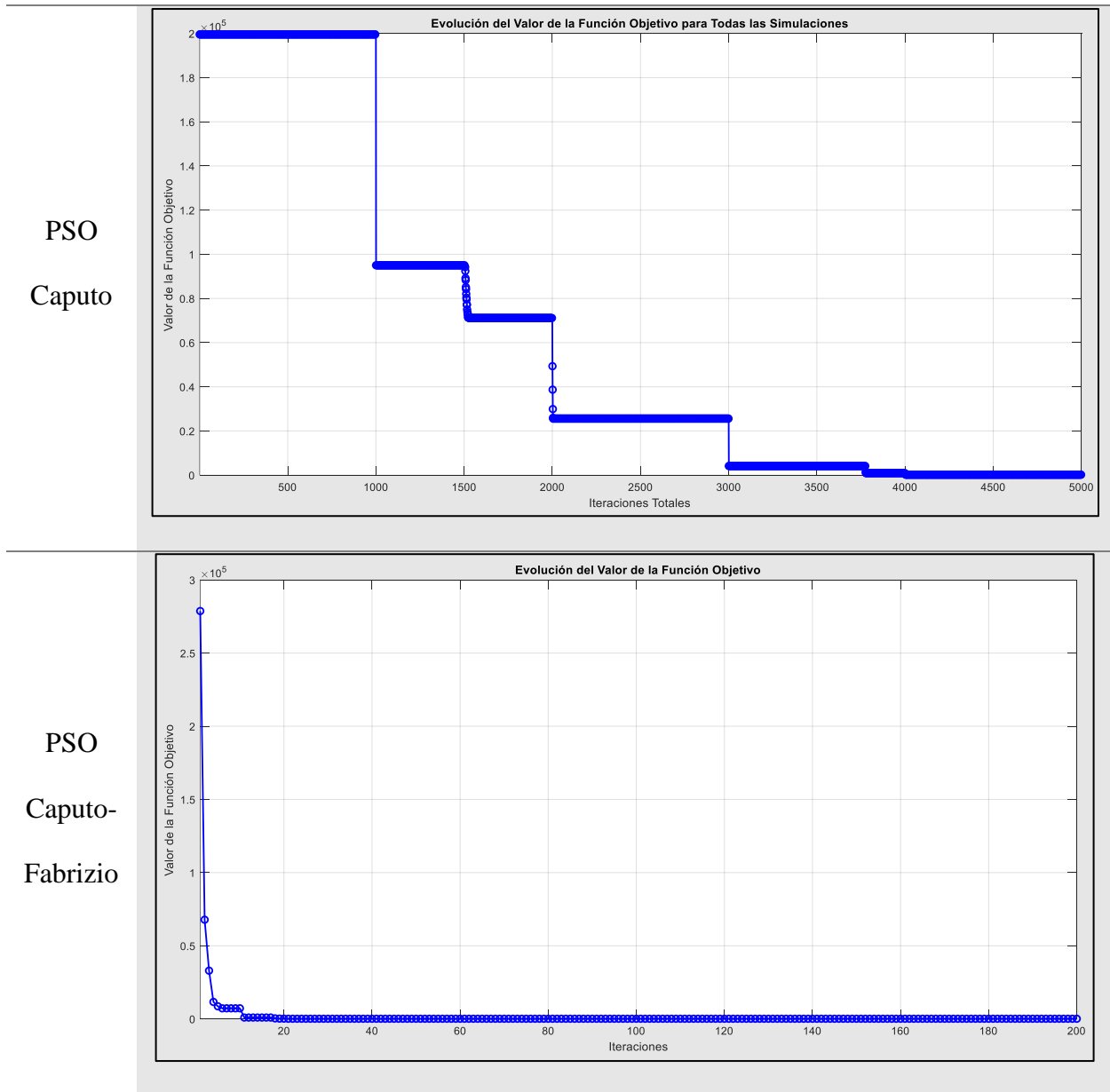
explotación del espacio de búsqueda, con una distribución de los valores aceptable los cuales se posan en torno a la moda. En términos de tiempo de convergencia, el PSO Caputo-Fabrizio se destaca por su mayor velocidad en comparación con los demás algoritmos, siendo 13.04% más rápido que PSO Estándar y 16.67% más rápido que PSO Caputo.

Para este sistema de ecuaciones, se realizó un procedimiento iterativo con el fin de ver la evolución del valor de las incógnitas y de la función objetivo evaluada en dichos valores, esto se aprecia en la Tabla 11.

**Tabla 11**

*Evolución de la función objetivo*





*Nota:* En la tabla se observan las gráficas de evolución de la función objetivo, para el PSO estándar y sus variantes fraccionarias.

Para este sistema de ecuaciones, la evolución del algoritmo indica que tanto el PSO estándar como el PSO con Caputo-Fabrizio requieren un número similar de iteraciones para lograr la convergencia de la función objetivo, con el PSO Caputo-Fabrizio alcanzando la solución en un número ligeramente menor de iteraciones (con velocidad de convergencia aproximadamente un

25% mayor). Mientras que el algoritmo fraccionario con Caputo presenta deficiencia en cuanto a su velocidad de convergencia, ya que no es hasta la iteración número 4000 en adelante que la función objetivo comienza a acercarse al mínimo.

En el área de anexos se pueden encontrar más sistemas de ecuaciones con su respectivo análisis.

También, se determinó el error porcentual de los resultados obtenidos para el valor de la función objetivo y se determinó el tiempo en segundos que le toma a cada algoritmo realizar las 50 simulaciones. Esto lo podemos apreciar en la siguiente tabla:

**Tabla 12**

*Error porcentual y tiempo de simulación para los sistemas de ecuaciones*

Algoritmo	PSO Estándar		PSO Caputo		PSO Caputo-Fabrizio	
	Sistema de Ecuaciones	Error P (%)	t (seg)	Error P (%)	t (seg)	Error P (%)
Anexo 1 Sistema 1	0	0.74	4.51234E-02	1.13	5.6818E-02	0.35
Anexo 1 Sistema 2	0	1.185	7.0197E-06	0.815	5.4623E-07	0.265
Anexo 1 Sistema 3	0	0.454	5.5636E-05	1.05	8.423534E-05	0.335
Anexo 1 Sistema 4	0	1.23	6.5089E-02	1.21	1.333512E-02	1.5
Anexo 1 Sistema 5	9.984E-29	446.146	64.851	479.246	49.2116	483.792
Anexo 4 Sistema 1	0	0.92	5.5629E-03	2.31	1.54218E-02	2.28

Anexo 4	0	2.24	4.6771E-	2.43	1.59217E-	1.975
Sistema 2			03		04	
Anexo 4	0	0.81	3.72377E-	2	4.11182E-	1.87
Sistema 3			03		03	
Anexo 4	5.2453	452.147	87.8741	447.197	11.744	477.816
Sistema 4						
Anexo 4	61.2834	480.932	90.5127	482.012	11.744	507.984
Sistema 5						
Anexo 5	0	0.68	2.2324E-	0.485	1.61812E-	0.53
Sistema 1			04		05	
Anexo 5	0	0.75	1.97881E-	0.585	4.1124E-06	0.547
Sistema 2			03			
Anexo 5	0	0.77	1.18251E-	0.757	4.93282E-	0.587
Sistema 3			05		06	
Anexo 5	15.3721	433.443	74.514	279.136	13.1988	335.0337
Sistema 4						
Anexo 5	9.183	466.229	65.7048	349.9747	12.7541	328.991
Sistema 5						

*Nota:* En la tabla se aprecia el error porcentual para cada algoritmo y su tiempo de simulación.

### 3.5 Refinamiento del algoritmo fraccionario

Como se evidenció en las pruebas realizadas tanto con las funciones de prueba como con los sistemas de ecuaciones, el algoritmo del PSO estándar demostró ser el más eficiente de los tres enfoques evaluados. Este algoritmo logró minimizar la función objetivo hasta el orden de  $10^{-30}$ , y los resultados obtenidos para las incógnitas y variables coinciden de manera precisa con los valores esperados. En contraste, entre los algoritmos fraccionarios, solo el PSO con Caputo-Fabrizio se acercó parcialmente a los resultados esperados, pero la función objetivo alcanzó un orden de  $10^{-4}$ , lo que representa una diferencia considerable en comparación con el PSO estándar.

Por otro lado, el PSO con Caputo fue el que presentó gran deficiencia en su desempeño, demorando significativamente en converger hacia las soluciones.

Con el propósito de mejorar el rendimiento de los algoritmos fraccionarios, se probaron diversas estrategias, como la modificación de los parámetros del PSO, la implementación de un espacio de búsqueda dinámico y la restricción de posiciones y velocidades, entre otras. De todas las estrategias evaluadas, las siguientes destacaron por generar una mejora notable en el rendimiento del PSO fraccionario.

### ***3.5.1 Reducción de los coeficientes de aceleración***

Inicialmente se establecieron los coeficientes de aceleración en  $c1 = c2 = 2$ , buscando tener un equilibrio entre exploración y explotación, ya que estos son los valores más comúnmente utilizados. En las estrategias implementadas en los algoritmos fraccionarios, se notaron mejores resultados al reducir estos coeficientes por debajo de 1, especialmente en el caso del PSO con Caputo.

Tras múltiples pruebas, se optó por utilizar  $c1 = 0.5$  y  $c2 = 0.75$ , valores que demostraron una mejora significativa tanto en las funciones de prueba como en los sistemas de ecuaciones abordados en el presente trabajo.

### ***3.5.2 Limitación de la velocidad***

Limitar la velocidad de las partículas es importante, ya que permite mantener el balance entre la exploración y explotación del espacio de búsqueda. Si las partículas se desplazan demasiado rápido, podrían saltarse regiones prometedoras, lo que dificultaría una búsqueda detallada cerca de los óptimos locales y afectaría negativamente la convergencia. Por el contrario, si las partículas se mueven demasiado lento, corren el riesgo de quedarse atrapadas en mínimos locales, reduciendo la eficiencia del algoritmo.

Para encontrar este equilibrio, se implementó un límite de velocidad máximo, después de la actualización de la velocidad de las partículas, basado en un porcentaje del rango total del espacio de búsqueda, definido como la diferencia entre el límite superior e inferior ( $ub - lb$ ). El factor de 0.1 se eligió para ajustar el rango de movimiento a un nivel que permita una exploración precisa, sin ralentizar tanto a las partículas como para que queden atrapadas.

Dado que la velocidad de las partículas es aproximada, se restringe dentro de este rango definido por la velocidad máxima. Esto previene que las partículas se desplacen demasiado rápido hacia soluciones no viables o se alejen de posibles mínimos, favoreciendo una búsqueda más controlada y eficiente.

### **3.5.3 Función híbrida**

Una de las estrategias más efectivas fue la implementación de una fase híbrida, la cual se obtuvo en la documentación de MATLAB. Se identificó que, aunque el PSO es un método de búsqueda global, en ocasiones no alcanza un mínimo local lo suficientemente preciso. Para solucionar esto, al finalizar las iteraciones del PSO, se emplea una función híbrida utilizando *fmincon*, un optimizador local.

Esta combinación permite refinar la solución proporcionada por el PSO, mejorando la precisión final. El PSO es excelente para explorar el espacio de búsqueda y evitar mínimos locales, pero su capacidad de explotación cercana a los óptimos puede ser limitada. En este contexto, *fmincon* entra en acción como un método de búsqueda local más preciso y eficiente en la explotación de la región cercana al mínimo hallado. Al aplicar esta estrategia híbrida, se aprovechan las ventajas de ambos enfoques, logrando una solución final más exacta y optimizada.

**3.5.4 Comparación procedimiento iterativo**

Con el propósito de comparar las soluciones antes y después de implementar las estrategias previamente mencionadas, se tomó el sistema de ecuaciones de la Tabla 7. Para ello, se realizaron dos procedimientos iterativos, el primero en el que se realizó inicialmente sin restricciones adicionales, mientras que el segundo incluyó la implementación de la limitación de la velocidad y la fase híbrida de optimización. Estos dos enfoques fueron comparados para evaluar el impacto de las mejoras propuestas sobre la calidad y eficiencia de las soluciones obtenidas. Dicha comparación se puede apreciar en la Figura 1.

**Figura 1**

*Comparación procedimiento iterativo*

X1	X2	X3	ValorFuncionObjetivo	X1	X2	X3	ValorFuncionObjetivo
0.49814	-0.19961	-0.52883	1.2671e-29	0.49814	-0.19961	-0.52883	1.2671e-29
0.49814	-0.19961	-0.52883	1.2671e-29	0.49814	-0.19961	-0.52883	1.2671e-29
0.49814	-0.19961	-0.52883	1.2671e-29	0.49814	-0.19961	-0.52883	1.2671e-29
0.49814	-0.19961	-0.52883	1.2671e-29	0.49814	-0.19961	-0.52883	1.2671e-29
0.49814	-0.19961	-0.52883	1.2671e-29	0.49814	-0.19961	-0.52883	1.2671e-29
X1	X2	X3	ValorFuncionObjetivo	x1	x2	x3	ValorFuncionObjetivo
1.7447	0.54852	-0.56298	134.62	0.49814	-0.19961	-0.52883	2.3346e-13
1.4872	0.41733	-0.52868	78.46	0.49814	-0.19961	-0.52883	7.0003e-14
1.1325	0.37919	-0.52473	48.095	0.49814	-0.19961	-0.52883	2.3295e-13
1.1743	0.039548	-0.53692	24.24	0.49814	-0.19961	-0.52883	2.3127e-13
0.92448	-0.14686	-0.52889	7.1178	0.49814	-0.19961	-0.52883	2.3428e-13
X1	X2	X3	ValorFuncionObjetivo	X1	X2	X3	ValorFuncionObjetivo
-0.064304	-0.78788	-0.66189	111.54	0.49814	-0.19961	-0.52883	1.855e-13
0.49355	-0.20409	-0.52831	0.0038069	0.49814	-0.19961	-0.52883	2.3245e-13
0.49982	-0.1958	-0.52862	0.0013524	0.49814	-0.19961	-0.52883	2.3295e-13
0.49173	-0.19468	-0.5289	0.0034728	0.49814	-0.19961	-0.52883	2.3312e-13
0.49542	-0.20194	-0.52865	0.00091702	0.49814	-0.19961	-0.52883	2.3428e-13

*Nota.* En la figura se observa la comparación de los procedimientos iterativos.

Al implementar estas estrategias en los algoritmos PSO fraccionarios, se logró una mejora notable en su rendimiento. Los algoritmos ahora proporcionan soluciones precisas para las

incógnitas de los sistemas de ecuaciones, cumpliendo con los valores esperados desde un inicio. Además, la minimización de la función objetivo, que anteriormente se obtenía en un rango de órdenes superiores, ahora alcanza niveles de precisión de  $10^{-14}$  y menores. Este avance se debe en gran parte a la fase híbrida implementada, que combina la capacidad global del PSO para explorar el espacio de búsqueda con la precisión local del optimizador *fmincon*, permitiendo un refinamiento adicional de las soluciones y garantizando una mayor exactitud en los resultados finales.

#### 4. Conclusiones

En el presente capítulo se exponen las conclusiones del trabajo de grado las cuales surgieron a partir de los resultados obtenidos durante el desarrollo del proyecto.

1. La implementación del cálculo fraccionario en el algoritmo PSO convencional permitió cumplir con los objetivos planteados en la investigación. Aunque los resultados no superan la eficiencia y precisión del PSO estándar, el enfoque fraccionario, utilizando las derivadas de Caputo y Caputo-Fabrizio, se presenta como una alternativa prometedora. Los resultados obtenidos son consistentes y abren la puerta a futuras mejoras, especialmente con pruebas adicionales y el uso de más funciones de evaluación para lograr identificar las ventajas potenciales, así como las limitaciones de esta metodología.

2. Este trabajo es el primero en su género en aplicar las definiciones de las derivadas fraccionarias de Caputo y Caputo-Fabrizio dentro del contexto del PSO. Aunque los resultados

iniciales son prometedores, esta innovación requiere de un mayor número de pruebas y análisis. La aplicación de estas derivadas fraccionarias en problemas de optimización no solo amplía las posibilidades de investigación en esta área, sino que también ofrece nuevas oportunidades para mejorar los algoritmos de optimización en términos de flexibilidad y ajuste fino.

3. Al comparar el PSO estándar con sus variantes fraccionarias, se concluye que el PSO convencional sigue siendo el más eficiente en términos de velocidad de convergencia y precisión en los resultados. Sin embargo, el PSO con Caputo-Fabrizio emerge como una alternativa viable, logrando aproximarse a las soluciones esperadas, aunque con una precisión menor. Esto subraya la necesidad de seguir investigando el enfoque fraccionario, ya que puede ofrecer ventajas en situaciones específicas o con ajustes adicionales en los parámetros.

4. Las modificaciones en los parámetros clave del PSO fraccionario, tales como  $\alpha$  y  $h$ , permitieron mejorar notablemente la velocidad de convergencia, reduciendo el número de iteraciones necesarias para alcanzar el mínimo global. A pesar de estas mejoras, el PSO estándar sigue destacando en términos de eficiencia y precisión. No obstante, los resultados indican que los algoritmos fraccionarios tienen potencial para ser optimizados aún más, lo que puede hacerlos competitivos en aplicaciones futuras.

5. En sistemas de ecuaciones de pequeña dimensión ( $2 \times 2, 3 \times 3, 4 \times 4$ ), tanto el PSO estándar como el PSO fraccionario con Caputo-Fabrizio entregaron soluciones muy cercanas a las esperadas. Sin embargo, en sistemas de mayor tamaño, como los trabajados en este estudio ( $7 \times 7, 8 \times 8, 9 \times 9, 10 \times 10$ ), los algoritmos fraccionarios presentaron desviaciones más significativas

en los resultados de algunas variables. Esto resalta la necesidad de realizar más estudios y ajustes para garantizar la precisión en sistemas de ecuaciones de mayor complejidad.

## 5. Recomendaciones

En el presente capítulo se exponen las recomendaciones del trabajo de grado para que las personas interesadas en una implementación futura las tengan en cuenta.

1. Todas las pruebas y conclusiones obtenidas en este estudio se basaron en las funciones de prueba y sistemas de ecuaciones mencionados en el documento y sus anexos. Para aplicar estos resultados a otros sistemas, es fundamental ajustar parámetros clave como el espacio de búsqueda, el número de partículas y el número de iteraciones. Estos ajustes deben realizarse en función de la complejidad del sistema a resolver, así como de la cantidad de variables o incógnitas involucradas. Un enfoque adaptativo en el ajuste de estos parámetros permitirá optimizar el rendimiento del algoritmo y mejorar su aplicabilidad en una mayor variedad de problemas.

2. Para evitar que el enjambre se quede estancado en soluciones locales, se recomienda mantener los coeficientes de aceleración  $c_1$  y  $c_2$  en un valor por debajo de 1. Esto favorece una mejor exploración del espacio de búsqueda, permitiendo que el enjambre converja hacia soluciones más cercanas al óptimo global en menos iteraciones. Alternativamente, se puede incrementar el número de partículas involucradas en la búsqueda del óptimo, siempre que no se exceda el tamaño del espacio de búsqueda, lo cual puede mejorar la precisión de las soluciones sin comprometer la eficiencia del algoritmo.

3. Es esencial definir de manera adecuada el espacio de búsqueda para evitar que las partículas del enjambre se pierdan o queden atrapadas en posiciones alejadas del óptimo global. Un espacio demasiado amplio puede dificultar la convergencia y aumentar el riesgo de estancamiento. Para contrarrestar este problema, se sugiere aumentar el número de partículas y el número de iteraciones, lo que permite una exploración más exhaustiva del espacio de búsqueda. Aunque esta estrategia puede requerir más tiempo de ejecución, es una forma efectiva de mejorar la robustez y precisión del algoritmo en la búsqueda de soluciones óptimas.

4. Para mejorar el rendimiento del PSO fraccionario, se recomiendan dos estrategias clave: aumentar el número de términos en la expresión de la velocidad de las partículas y disminuir el valor del parámetro de espaciado  $h$ . Al aumentar el número de términos en la ecuación de velocidad, se capturan con mayor detalle los efectos de la memoria fraccionaria, lo que mejora el comportamiento del enjambre. Asimismo, reducir el valor de  $h$  aumenta la precisión de la aproximación fraccionaria, lo que permite al algoritmo ajustar mejor las trayectorias de las partículas y converger de manera más eficiente hacia el óptimo global.

### Referencias

- Lin D., Liao X., Dong L., Yang R., S. Yu S., Ho-Ching Iu H., Fernando T. & Li Z. (2021). Experimental Study of Fractional-Order RC Circuit Model Using the Caputo and Caputo-Fabrizio Derivatives. IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 68, no. 3, pp. 1034-1044.
- Ozores, A. L. (2014). Cálculo Fraccionario y dinámica newtoniana. Pensamiento Matemático, 4(1), 77-105.
- Cuahutenango Barro, B. (2017). Fricción fraccionaria en ecuaciones de onda no lineales con efecto de memoria. Facultad de Matemáticas, Universidad Autónoma de Guerrero, Guerrero, México.
- Solteiro Pires E.J., Tenreiro Machado J.A., de Moura Oliveira P.B., Boaventura Cunha J. & Mendes L. (2010). Particle swarm optimization with fractional-order velocity. Instituto Politécnico de Porto.
- Wang D., Tan D. & Liu L. (2017). Particle swarm optimization algorithm: an overview. Escuela de Ingeniería Eléctrica, Universidad de Zhengzhou, Henan, China.
- Barrios M., Lombardi A. L. & Reyero G.F. (2018). Método numérico de tipo L1 para problemas variacionales fraccionarios. Departamento de Matemática, Facultad de Ciencias Exactas, Ingeniería y Agrimensura, Universidad Nacional de Rosario, Rosario, Argentina.
- Qureshi S. (2022, junio 25). Cálculo fraccionario | Derivada fraccionaria | Método L1 para Caputo | Código MATLAB | Clase 12 [Video]. YouTube.  
[https://www.youtube.com/watch?v=zwu\\_j5XGFuc](https://www.youtube.com/watch?v=zwu_j5XGFuc)

Burden L. R. & Faires J. D. (1993). Análisis numérico 2 Ed. Royal Statistical Society Serie D.

Cap. 9.

MathWorks. (n.d.). Tune particle swarm optimization process. MathWorks.

<https://la.mathworks.com/help/gads/tune-particleswarm-optimization-process.html>

### **Apéndices**

En este capítulo, se presenta el enlace el cual redireccionará al lector a la carpeta que contiene los anexos correspondientes a este proyecto, los cuales respaldan el desarrollo de este y a su vez contribuyen a la comprensión del trabajo realizado.

[https://drive.google.com/drive/folders/1YvTZ\\_81dWNGE2oa\\_giScbfg0gNt8Prf3](https://drive.google.com/drive/folders/1YvTZ_81dWNGE2oa_giScbfg0gNt8Prf3)