

**SOLUCIÓN DE LAS ECUACIONES QUE MODELAN UN CIRCUITO NO LINEAL  
DE CORRIENTE DIRECTA MEDIANTE EL MÉTODO DE LA ESPIRAL**

**JORGE AUGUSTO PORTILLA GARCÍA**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE INGENIERÍA ELÉCTRICA,  
ELECTRÓNICA Y DE TELECOMUNICACIONES  
BUCARAMANGA**

**2012**

**SOLUCIÓN DE LAS ECUACIONES QUE MODELAN UN CIRCUITO NO LINEAL  
DE CORRIENTE DIRECTA MEDIANTE EL MÉTODO DE LA ESPIRAL**

**JORGE AUGUSTO PORTILLA GARCÍA**

**Trabajo de Grado para optar al título de  
INGENIERO ELECTRÓNICO**

**Director**

**PhD. CARLOS RODRIGO CORREA CELY**

**Co-director**

**PhD. (c) IVÁN MAURICIO AMAYA CONTRERAS**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE INGENIERÍA ELÉCTRICA,  
ELECTRÓNICA Y DE TELECOMUNICACIONES  
BUCARAMANGA**

**2012**

## AGRADECIMIENTOS

A los profesores *Ph.D. Carlos Rodrigo Correa* y *Ph.D.(c) Iván Mauricio Amaya*, por su disposición a compartir conocimientos, y a llevar a cabo la dirección de esta tesis, y porque cada una de sus recomendaciones contribuyó definitivamente a que este trabajo fuera mejor.

Al grupo de investigación *CEMOS* adscrito a la *Escuela de Eléctrica, Electrónica y Telecomunicaciones* de la *Universidad Industrial de Santander*, por haberme permitido sus instalaciones.

En general, agradecimientos a la *Universidad Industrial de Santander*.

## DEDICATORIAS

Al Señor *Jesús, Gran Autor de la Vida.*

A mis padres, *Hernando Portilla Barrios* y *Luz Stella García Angarita*, por su inmenso apoyo.

A mi hermano *Cristhian Portilla*, para que persevere por sus metas y las alcance.

A toda mi familia, porque siempre estuvo presente, y en especial a mi Abuela *Adela Angarita* y a mi tía *Yasmira García*.

# CONTENIDO

	pág.
INTRODUCCIÓN .....	19
1. OBJETIVOS .....	21
1.1 General .....	21
1.2 Específicos .....	21
2. PLANTEAMIENTO DEL PROBLEMA .....	22
2.1 Antecedentes .....	22
2.2 Problema a resolver .....	22
3. MARCO TEÓRICO .....	23
3.1 Representación de un problema de optimización .....	23
3.2 El método metaheurístico de la espiral de optimización .....	23
3.3 Teorema para la utilización de un método de optimización, en la solución a un sistema de ecuaciones .....	31
3.4 Modelos para dispositivos <i>no-lineales</i> utilizados .....	33
4. ANÁLISIS DE LA DIVERSIFICACIÓN E INTENSIFICACIÓN EN LA TRAYECTORIA DE ESPIRAL .....	39
4.1 Gráficas de algunas trayectorias .....	40
4.2 Ejemplo de una trayectoria de espiral de carácter diversa .....	44
4.3 Ejemplo de una trayectoria de espiral de carácter intensa .....	45
5. PRUEBAS DEL ALGORITMO .....	47

5.1	Función <i>De Jong's</i> .....	48
5.2	Función <i>Rosenbrock's Valley</i> .....	58
5.3	Función <i>Rastrigin</i> .....	66
5.4	Función <i>Schwefel</i> .....	74
5.5	Característica de convergencia a una sola solución cuando existe múltiples de ellas. ....	83
6.	SOLUCIÓN DE LAS ECUACIONES QUE MODELAN ALGUNOS CIRCUITOS NO- LINEALES DE C.C. ....	86
6.1	Solución a ecuaciones no lineales de un circuito de corriente directa con un diodo (una variable) .....	86
6.2	Solución de ecuaciones no lineales de un circuito de corriente directa con dos diodos (dos variables).....	90
6.3	Solución de ecuaciones no lineales de un circuito de corriente directa con un <i>MOSFET</i> (dos variables) .....	94
6.4	Solución de ecuaciones no lineales de un circuito de corriente directa con tres <i>BJTs</i> (tres y cinco variables).....	98
6.5	Solución a ecuaciones no lineales de un circuito de corriente directa con cuatro <i>BJTs</i> (cuatro variables).....	107
6.6	Solución de ecuaciones no lineales de un circuito de corriente directa con cinco <i>MOSFET</i> (4 variables).....	112
7.	OBSERVACIONES Y CONCLUSIONES .....	117
8.	RECOMENDACIONES .....	120
	REFERENCIAS .....	122
	ANEXOS.....	124

## LISTA DE TABLAS

pág.

Tabla 1. Trabajos de grado 2011-2012.....	17
Tabla 2. Parámetros y resultados de simulación para solución de la función <i>De Jong's</i> en 1D.....	50
Tabla 3. Parámetros y resultados de simulación para solución de la función <i>De Jong's</i> en 2D.....	53
Tabla 4. Parámetros y resultados de simulación para solución de la función <i>De Jong's</i> en 3D.....	55
Tabla 5. Parámetros y resultados de simulación para solución de la función <i>De Jong's</i> en 4D.....	58
Tabla 6. Parámetros y resultados de simulación para solución de la función <i>Rosenbrock's Valley</i> en 2D.....	61
Tabla 7. Parámetros y resultados de simulación para solución de la función <i>Rosenbrock's Valley</i> en 3D.....	63
Tabla 8. Parámetros y resultados de simulación para solución de la función <i>Rosenbrock's Valley</i> en 4D.....	65
Tabla 9. Parámetros y resultados de simulación para solución de la función <i>Rastrigin</i> en 1D.....	67
Tabla 10. Parámetros y resultados de simulación para la solución a la raíz de la función <i>Rastrigin</i> en 2D.....	70
Tabla 11. Parámetros y resultados de simulación para solución de la función <i>Rastrigin</i> en 3D.....	72
Tabla 12. Parámetros y resultados de simulación para solución de la función <i>Rastrigin</i> en 4D.....	74
Tabla 13. Parámetros y resultados de simulación para solución de la función <i>Schwefel</i> en 1D.....	76
Tabla 14. Parámetros y resultados de simulación para función <i>Schwefel</i> en 2D...	78

Tabla 15. Parámetros y resultados de simulación para solución de la función <i>Schwefel</i> en 3D.....	81
Tabla 16. Parámetros y resultados de simulación para solución de la función <i>Schwefel</i> en 4D.....	83
Tabla 17. Resultados y parámetros de la ejecución para resolver ecuación (78). .	85
Tabla 18. Parámetros y resultados de simulación para la solución del modelo de un circuito <i>no-lineal</i> de c.c. de una malla. ....	87
Tabla 19. Parámetros y resultados de la solución al circuito de un diodo usando el comando <i>fsolve</i> de <i>MATLAB</i> .....	89
Tabla 20. Parámetros y resultados de simulación para un circuito <i>no-lineal</i> de c.c. de dos mallas.....	92
Tabla 21. Parámetros y resultados para la solución del circuito de la Figura 41, usando el comando <i>fsolve</i> de <i>MATLAB</i> . ....	93
Tabla 22. Parámetros y resultados de ejecución para el circuito de la Figura 43. .	96
Tabla 23. Parámetros y resultados de la ejecución del algoritmo para el circuito de la Figura 45. ....	100
Tabla 24. Parámetros y resultados obtenidos con <i>fsolve</i> de <i>MATLAB</i> en la solución al modelo del circuito con 3 <i>BJTs</i> . ....	101
Tabla 25. Parámetros y resultados para la ejecución con el modelo <i>Ebers-Moll</i> del circuito <i>cambiador de nivel</i> de la Figura 46.....	105
Tabla 26. Resultados de la solución al modelo de la Figura 46 usando el comando <i>fsolve</i> de <i>Matlab</i> .....	106
Tabla 27. Parámetros y resultados de la ejecución del algoritmo, para el circuito de la Figura 47. ....	110
Tabla 28. Resultados de ejecución repetida para los mismos parámetros. ....	111
Tabla 29. Parámetros y resultados obtenidos con el comando <i>fsolve</i> de <i>MATLAB</i> para resolver el circuito de 4 <i>BJTs</i> .....	112
Tabla 30. Parámetros y resultados para la solución con el algoritmo del modelo de la Figura 48.....	115

## LISTA DE FIGURAS

pág.

Figura 1. Rotación 2D discreta con el modelo de espiral logarítmica en torno al origen.....	25
Figura 2. Rotación 2D discreta con el modelo de espiral logarítmica en torno a un punto arbitrario.....	27
Figura 3. Gráfica de una ecuación no-lineal y la función objetivo asociada.....	32
Figura 4. Trayectoria de una espiral logarítmica obtenida con los parámetros de referencia.....	41
Figura 5. Trayectoria de una espiral logarítmica obtenida al aumentar los parámetros.....	42
Figura 6. Trayectoria de una espiral logarítmica obtenida al disminuir los parámetros.....	43
Figura 7. Ampliación de la Figura 6. ....	44
Figura 8. Trayectoria de una espiral logarítmica diversa. ....	45
Figura 9. Trayectoria de una espiral logarítmica intensa. ....	46
Figura 10. Función <i>De Jongs</i> 1D. ....	49
Figura 11. Convergencia para la solución de la ecuación de <i>De Jong's</i> en 1D. ....	50
Figura 12. Ampliación de la <i>Figura 11</i> . ....	51
Figura 13. Gráfica de la función <i>De Jong's 2D</i> .....	52
Figura 14. Convergencia para la solución de la ecuación de <i>De Jong's</i> en 2D. ....	53
Figura 15. Ampliación sobre la gráfica de la Figura 14. ....	54
Figura 16. Convergencia para la solución de la ecuación de <i>De Jong's</i> en 3D. ....	56
Figura 17. Ampliación de la Figura 16. ....	57
Figura 18. Gráfica de la función <i>Rosenbrock's Valley</i> en 2D. ....	60
Figura 19. Convergencia para la solución de la ecuación de <i>Rosenbrock's Valley</i> en 2D. ....	61
Figura 20. Ampliación de la gráfica mostrada en la Figura 19. ....	62

Figura 21. Convergencia para la solución de la ecuación de <i>Rosenbrock's Valley</i> en 3D. ....	64
Figura 22. Ampliación de la Figura 17. ....	64
Figura 23. Gráfica de la función <i>Rastrigin</i> 1D. ....	67
Figura 24. Convergencia hacia la solución de la raíz de la función <i>Rastrigin</i> en 1D. ....	68
Figura 25. Ampliación de la gráfica de la Figura 20. ....	69
Figura 26. Gráfica de la función <i>Rastrigin</i> 2D. ....	70
Figura 27. Convergencia hacia la solución de la raíz de la función <i>Rastrigin</i> en 2D. ....	71
Figura 28. Ampliación sobre la gráfica de la Figura 23. ....	71
Figura 29. Convergencia hacia la solución de la raíz de la función <i>Rastrigin</i> en 3D. ....	73
Figura 30. Gráfica de la función <i>Schwefel</i> en 1D. ....	75
Figura 31. Convergencia a la solución óptima de la función <i>Schwefel</i> 1D. ....	76
Figura 32. Ampliación de la gráfica de la <i>Figura 27</i> . ....	77
Figura 33. Gráfica de la función <i>Schwefel</i> en 2D. ....	78
Figura 34. Convergencia a la solución óptima en la función <i>Schwefel</i> 2D. ....	79
Figura 35. Ampliación sobre la gráfica de la <i>Figura 30</i> . ....	80
Figura 36. Convergencia al valor óptimo final mínimo de <i>Schwefel</i> en 3D. ....	82
Figura 37. Convergencia hacia una de las soluciones presentes en una región factible. ....	84
Figura 38. Circuito de un solo diodo. ....	86
Figura 39. Convergencia para la solución del circuito de la <i>Figura 34</i> . ....	88
Figura 40. Ampliación sobre la gráfica de la <i>Figura 35</i> . ....	89
Figura 41. Circuito de dos mallas con dos diodos. ....	90
Figura 42. Convergencia de la solución del circuito de la <i>Figura 37</i> . ....	93
Figura 43. Circuito con un solo <i>MOSFET</i> . ....	95
Figura 44. Circuito de un solo <i>MOSFET</i> y tensiones de polarización obtenidas con la simulación en <i>ORCAD</i> . ....	97

Figura 45. Circuito <i>cambiador de nivel</i> .....	98
Figura 46. Designación de nodos para el circuito <i>cambiador de nivel</i> . ....	102
Figura 47. Circuito con cuatro transistores <i>BJT</i> . ....	107
Figura 48. Circuito con 5 <i>MOSFET</i> . ....	113
Figura 49. Circuito con 5 <i>MOSFET</i> y los valores de tensión obtenidos con la simulación en <i>ORCAD</i> . ....	116
Figura 50. Mensaje de inicio en la ejecución del <i>Algoritmo de La Espiral</i> .....	124
Figura 51. Primera lista de opciones desplegada por el algoritmo.....	125
Figura 52. Mensaje de finalización de la ejecución.....	126
Figura 53. Gráficas a modo de ejemplo para la función <i>Schwefel</i> en 2D.....	126
Figura 54. Resultados numéricos entregados por el algoritmo. ....	127
Figura 55. Mensaje de diálogo para seleccionar opción de introducir los parámetros.....	129
Figura 56. Ejemplo de introducción de parámetros al programa desde el teclado. ....	129
Figura 57. Resultados obtenidos en la ejecución con los parámetros de la <i>Figura 56</i> . ....	130
Figura 58. Ejemplo de escritura de ecuaciones en el programa. ....	131
Figura 59. Ejemplo de establecimiento de parámetros desde el <i>script</i> principal..	131
Figura 60. Resultados obtenidos para el ejemplo de ingreso de un modelo.....	132

## LISTA DE ANEXOS

	pág.
ANEXO A. Ejecución con los parámetros señalados en el informe .....	124
ANEXO B. Ejecución con unos parámetros ingresados por el usuario.....	129
ANEXO C. Ejecución con ecuaciones y parámetros ingresados por el usuario ..	130

## RESUMEN

### TÍTULO

**SOLUCIÓN DE LAS ECUACIONES QUE MODELAN UN CIRCUITO NO LINEAL DE CORRIENTE DIRECTA MEDIANTE EL MÉTODO DE LA ESPIRAL \***

### AUTOR

Jorge Augusto Portilla García \*\*

### PALABRAS CLAVES

SISTEMAS DE ECUACIONES NO-LINEALES, CIRCUITOS DE CORRIENTE CONTINUA, OPTIMIZACIÓN GLOBAL, METAHEURÍSTICA, ESPIRAL DE OPTIMIZACIÓN, DIVERSIFICACIÓN, INTENSIFICACIÓN.

### DESCRIPCIÓN

Basado en un algoritmo desarrollado por *Tamura y Yasuda* [1], y denominado *espiral de optimización*, se implementa un programa en *Matlab*, fundamentado en dicho algoritmo y se utiliza para resolver de forma satisfactoria, las ecuaciones correspondientes a funciones típicas de prueba, como lo son *De Jong's*, *Rosenbrock's Valley*, *Rastrigin* y *Schwefel*. También se describe un análisis propio de la manera para diversificar e intensificar la trayectoria de una espiral de optimización. El presente documento, así mismo expone los principales resultados obtenidos en la aplicación del *algoritmo de la espiral de optimización* para resolver las ecuaciones que modelan algunos circuitos de c.c. *no-lineales* con dispositivos como diodos, *BJTs* y *MOSFETs*. Entre los modelos de circuitos resueltos, se encuentra el correspondiente a un *cambiador de nivel de c.c.* con *BJTs*, y un *convertidor de tensión a corriente* con *MOSFETs*. Para este circuito en particular, se obtienen cuatro variables y con el algoritmo implementado en *Matlab* se resuelven dichas ecuaciones, obteniéndose resultados a conformidad; hecho que se refleja en los valores de los errores obtenidos al ejecutar el programa y evaluar las soluciones en las mismas ecuaciones que se introdujeron al programa, y también al comparar con los resultados de *ORCAD CAPTURE*.

---

\* Trabajo de Grado

\*\* Facultad de ingenierías Físico-mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Director: Carlos Rodrigo Correa Cely. Codirector: Iván Mauricio Amaya Contreras.

## SUMMARY

### TITLE

**SOLUTION TO EQUATIONS MODELING A NON-LINEAR CIRCUIT OF DIRECT CURRENT THROUGH THE SPIRAL METHOD**

### AUTHOR

Jorge Augusto Portilla García\*\*

### KEYWORDS

NON-LINEAR EQUATIONS SYSTEMS, DC CIRCUITS, GLOBAL OPTIMIZATION, METAHEURISTIC, SPIRAL OPTIMIZATION, DIVERSIFICATION, INTENSIFICATION.

### DESCRIPTION

Based on an algorithm developed by *Tamura* and *Yasuda* [1], called spiral optimization, a program on Matlab is implemented and used to satisfactorily solve the equations of typical testing functions, such as *De Jong's*, *Rosenbrock's Valley*, *Rastrigin* and *Schwefel*. A proper analysis of the way to diversify and intensify the path of an optimization spiral was also made. This document also presents the main results obtained in applying the spiral optimization algorithm to solve some equations modeling DC circuits with non-linear devices such as diodes, *BJTs* and *MOSFETs*. Among the resolved circuit models, there it is one corresponding to a *DC level shifter* with *BJTs*, and a *voltage-to-current converter* with *MOSFETs*. For this last one in particular, four variables are obtained and the algorithm implemented in Matlab solved these equations, yielding results in accordance; reflected in the values of the errors obtained by running the program and evaluating solutions in the same equations that were introduced to the program, and also comparing the simulation results with *ORCAD CAPTURE*.

---

\* Degree Work

\*\* Facultad de ingenierías Físico-mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Project Director: Carlos Rodrigo Correa Cely. Project Co-director: Iván Mauricio Amaya Contreras.

## Grupo CEMOS

### Línea de trabajo: *Modelado y Optimización*

La línea de modelado y optimización del grupo CEMOS se encuentra orientada hacia la evaluación de los más importantes algoritmos de optimización y fundamentalmente los basados en la inteligencia artificial y la heurística. Los siguientes son los trabajos de grado que a la fecha tienen como objetivo fortalecer esta línea, dándosele un énfasis a su aplicación en la ingeniería electrónica y eléctrica.

**Tabla 1. Trabajos de grado 2011-2012**

SOLUCION DEL MODELO MATEMATICO DE UN CIRCUITO ELECTRONICO COMPUESTO DE ELEMENTOS NO LINEALES MEDIANTE EL OPTIMIZADOR DE FUERZA CENTRAL	ROA, O; RAMIREZ, F
EVALUACION DEL METODO OPTIMIZADOR DE FUERZA CENTRAL FRENTE AL OPTIMIZADOR POR ENJAMBRE DE PARTÍCULAS UNIFICADO EN LA SOLUCIÓN DE ECUACIONES NO LINEALES	BARRERO, K; NAVARRO, O
ANALISIS DEL COMPORTAMIENTO DINAMICO DE UN CIRCUITO ELECTRONICO NO LINEAL MEDIANTE LA SOLUCIÓN DE SU MODELO CON HARMONY SEARCH	TRASLADINO, O; GONZALES, S
APLICACION DEL METODO DE OPTIMIZACION POR FUERZA CENTRAL (CFO) AL DISEÑO DE UN ABSORBEDOR ELECTROMAGNÉTICO ÓPTIMO	BAYONA, C; GONZALEZ, J
ALGORITMO HIBRIDO DEL SIMPLEX CON EL OPTIMIZADOR DE FUERZA CENTRAL (CFO)	GARCIA, J
PROPUESTA DE ESTRATEGIA PARA LA OBTENCIÓN DE LOS PARAMETROS DEL ALGORITMO DE OPTIMIZACIÓN POR FUERZA CENTRAL (CFO)	BUITRAGO, J
SOLUCION DE UN SISTEMA DE ECUACIONES NO LINEAL UTILIZANDO LA ESTRATEGIA BASADA EN EL ALGORITMO HARMONY SEARCH	GALEANO, Y
IMPLEMENTACION DEL ALGORITMO DE BUSQUEDA GRAVITACIONAL (GSA) PARA EL DISEÑO DE UN ABSORBEDOR ELECTROMAGNÉTICO ÓPTIMO	MIRANDA, A; RUIZ, J
SOLUCION DEL MODELO MATEMATICO DE UN CIRCUITO NO LINEAL EN DC, A TRAVES DE UNA ESTRATEGIA DE OPTIMIZACIÓN	CRUZ, J
USO DEL HIBRIDO UPSO CON EL SIMPLEX PARA OPTIMIZACION DE FUNCIONES CON RESTRICCIONES EN PROBLEMAS DE INGENIERÍA ELECTRÓNICA	CANON, E
INTELIGENCIA COMPUTACIONAL INSPIRADA EN LA CUANTICA, APLICADA AL DISEÑO DE COMPENSADORES	HERNANDEZ, M; CORREDOR, D

OPTIMIZACION PARETO MEDIANTE EL METODO DE ENJAMBRE DE PARTICULAS UNIFICADO EN PROBLEMAS DE INGENIERÍA ELECTRÓNICA	QUIROZ, M;
IDENTIFICACION DE PARAMETROS DE UN MOTOR SINCRONO DE IMAN PERMANENTE MEDIANTE EL ALGORITMO DE LA ESPIRAL	RUTTO, A
EL METODO DE COLONIA ARTIFICIAL DE ABEJAS Y EL CRITERIO DE MINIMA ENTROPIA PARA EL DISEÑO ÓPTIMO DE UN DISIPADOR DE CALOR	AVILA, J
IDENTIFICACION DE SISTEMAS NO LINEALES MEDIANTE EL METODO DE OPTIMIZACION DE GOTA DE AGUA INTELIGENTE	CARVAJAL, L; DÁVILA, D
POTENCIAL APLICACION DEL ALGORITMO DE GOTAS DE AGUA INTELIGENTES (GAI-IWD) EN INGENIERÍA ELECTRÓNICA	ARIAS, J
CONTRASTACION DE LOS METODOS DE ENJAMBRE DE PARTICULAS UNIFICADO FRENTE AL MÉTODO COLONIA ARTIFICIAL DE ABJEAS, EN LA IDENTIFICACIÓN DE SISTEMAS	PETRO, E; FUENTES, R.
EVALUACION Y COMPARACION ENTRE EL METODO NEWTON RAPHSON Y CENTRAL FORCE OPTIMIZATION PARA EL ANÁLISIS DE FLUJO DE CARGAS DE UN SISTEMA DE POTENCIA	VALDERRAMA, M
EVALUACION Y COMPARACION ENTRE LOS METODOS NEWTON RAPHSON Y ARTIFICIAL BEE COLONY (ABC) PARA EL ANÁLISIS DEL FLUJO DE CARGA DE UN SISTEMA DE POTENCIA	CELIS, J; RINCON, F
EVALUACION Y COMPARACION ENTRE LOS METODOS UPSO Y NEWTON-RAPHSON PARA EL ANÁLISIS DE FLUJO DE CARGA EN UN SISTEMA DE POTENCIA	PINZON, C; ARDILA, E
DISEÑO DE UN ABSORBEDOR ELECTROMAGNETICO MULTICAPA MEDIANTE EL METODO DE LA ESPIRAL	ORTIZ, R; GARCIA, E
<b>SOLUCION DE LAS ECUACIONES QUE MODELAN UN CIRCUITO NO LINEAL DE CORRIENTE DIRECTA MEDIANTE EL MÉTODO DE LA ESPIRAL</b>	<b>PORTILLA, J</b>
ALGORITMO DE ENJAMBRE DE PARTICULAS PARA SOLUCIONAR ECUACIONES DIOFÁNTICAS LINEALES COMUNMENTE ENCONTRADAS EN PROBLEMAS DE INGENIERÍA ELECTRÓNICA	RAMIREZ, J; OSORIO, F

*(Fuente: grupo CEMOS)*

## INTRODUCCIÓN

El presente informe recopila los principales resultados obtenidos durante la ejecución de un trabajo de investigación que se basa fundamentalmente, en la aplicación de un nuevo método de optimización desarrollado por *Tamura* y *Yasuda*, el cual fue publicado por primera vez en el año 2011 [1].

El algoritmo de optimización de la espiral logarítmica propuesto por *Tamura* y *Yasuda*, ha sido implementado satisfactoriamente en *Matlab* por el autor del presente informe, y se verifica su efectividad para resolver ecuaciones basadas en funciones típicas de prueba, como lo son la función *De Jong's*, *Rosenbrock's Valley*, *Rastrigin* y *Schwefel*. Se exponen los resultados obtenidos en la solución de las ecuaciones mencionadas para el caso de *una* a *cuatro* variables. Es también documentado un análisis propio de la manera para diversificar e intensificar la trayectoria de una espiral de optimización.

En este informe se citan circuitos con elementos *no-lineales* como *diodos*, *transistores BJT* y *MOSFET*; con el propósito de modelarlos y establecer sistemas de ecuaciones de *1* a *5* incógnitas. Entre los modelos de circuitos analizados, se encuentran los correspondientes a un circuito *cambiador de nivel*, y un *convertidor de tensión a corriente*.

Posterior a la verificación del correcto funcionamiento del programa, se utiliza para resolver las ecuaciones que modelan los circuitos seleccionados. Con el fin de comparar, estos modelos se resolvieron con otras herramientas, como el comando *fsolve* de *MATLAB*, o el simulador de circuitos *ORCAD CAPTURE*.

Se presentan tablas con los parámetros de la ejecución del programa y los resultados encontrados para la solución de las ecuaciones, así como los valores

de los errores en la convergencia del método, los tiempos de ejecución de cada programa, y el número de iteraciones realizadas.

La ejecución del algoritmo para los valores registrados en las tablas, se realizó casi en su totalidad, en un equipo *Dell Vostro 3500* con procesador *Intel core i3 CPU M350 @ 2.27Ghz*, memoria *RAM* de *4GB* y un sistema operativo de 32 bits, en el plan de energía de alto rendimiento.

Un sistema de ecuaciones con 5 variables, correspondientes al modelo de un circuito con *BJT*s, se resolvió con el algoritmo empleando un equipo de cómputo del grupo *CEMOS* que cuenta con características superiores al mencionado anteriormente.

## 1. OBJETIVOS

### 1.1 General

Resolver mediante una nueva estrategia numérica el modelo matemático de un circuito no lineal de corriente directa. (Ver páginas 86-116)

### 1.2 Específicos

1. Implementar el algoritmo de optimización de la espiral en la plataforma *Matlab*. (Ver páginas 39-112)
2. Verificar mediante pruebas con funciones típicas, el correcto funcionamiento del algoritmo de optimización de la espiral. (Ver páginas 47 - 86)
3. Plantear las ecuaciones matemáticas correspondientes al modelo exponencial de un circuito *no-lineal* de corriente directa con un máximo de *cuatro* incógnitas, y resolverlas mediante el método de optimización de la espiral. (Ver página 86 - 112)

## 2. PLANTEAMIENTO DEL PROBLEMA

En electrónica es habitual encontrar circuitos con elementos *no-lineales*, los cuales poseen una característica *voltaje-corriente* que no puede ser expresada con una sencilla ecuación lineal, y por consiguiente, no aplica a estos dispositivos, el principio de superposición, y con ello, la herramienta de análisis de circuito, denominada equivalente de *Thevenin*. Entre los dispositivos de dos terminales más comunes se encuentra el diodo de unión *p-n*, y el diodo *Zener* [2].

### 2.1 Antecedentes

Para la solución de circuitos con elementos *no-lineales* existen diversos métodos, tales como: método matemático directo, método gráfico, método iterativo, y el método de modelado por aproximación de segmentos lineales el cual consiste en establecer un modelo menos aproximado, que sí es lineal, y que se usa para regiones limitadas de operación [2].

Todos los métodos mencionados, excepto el método matemático directo, proporcionan respuestas con cierto grado limitado de exactitud. El método gráfico no resulta posible para más de dos variables, y por ende, cuando el método matemático directo no es aplicable, y la aproximación de segmentos lineales es dificultosa, se debe recurrir a un método de iteración.

### 2.2 Problema a resolver

El problema que se considera como *trabajo de grado*, consiste en hallar el valor de las incógnitas asociadas a las ecuaciones que modelan algunos circuitos *no-lineales* de corriente directa, los cuales pueden tener como elemento *no-lineal*, a un diodo, a un transistor *BJT* ó a un *MOSFET*.

### 3. MARCO TEÓRICO

#### 3.1 Representación de un problema de optimización

De manera general, un enunciado de optimización puede expresarse como [3]:

$$\begin{aligned} & \textit{minimizar}_{\mathbf{x} \in \mathfrak{R}^n} f_i(\mathbf{x}), \quad (i = 1, 2, \dots, M), \\ & \textit{sujeto a:} \quad h_j(\mathbf{x}) = 0, \quad (j = 1, 2, \dots, J), \\ & \quad \quad \quad g_k(\mathbf{x}) \leq 0, \quad (k = 1, 2, \dots, N) \end{aligned} \tag{1}$$

En donde  $f_i(\mathbf{x})$ ,  $h_j(\mathbf{x})$  y  $g_k(\mathbf{x})$ , están en función del vector de diseño:

$$\mathbf{x} = (x_1, x_2, \dots, x_n)^T \tag{2}$$

el cual está conformado por las componentes conocidas como variables de decisión. Las ecuaciones para  $h_j(\mathbf{x})$  y las inecuaciones para  $g_k(\mathbf{x})$  en conjunto son denominadas restricciones.

Cada  $f_i(\mathbf{x})$  es conocida como una función objetivo o función costo, y el espacio comprendido por las variables de decisión es nombrado espacio de búsqueda  $\mathfrak{R}^n$ , mientras que el espacio conformado por los valores de las funciones objetivo es el espacio de solución.

#### 3.2 El método metaheurístico de la espiral de optimización

Son características geométricas interesantes de la espiral logarítmica, el hecho de que iniciando en un punto arbitrario, y dirigiéndose hacia el centro, éste se rodea infinitas veces, y sin embargo, se recorre una distancia finita, y a su vez, se trata

de una curva con un ángulo constante entre el radio y el vector tangente en todo punto [4].

La metaheurística es una rama de los métodos de aproximación heurísticos. Se propone un nuevo método de búsqueda multipunto perteneciente a la metaheurística, el cual está basado en una analogía a la espiral logarítmica presente en diversos patrones de la naturaleza [1]. La optimización en dos dimensiones, y su generalización a *n-dimensiones* hace uso del patrón de la espiral logarítmica que se expondrá a continuación.

Se establece una ecuación de tipo discreto para un patrón logarítmico de convergencia al origen [5]:

$$\begin{bmatrix} x_1^{k+1} \\ x_2^{k+1} \end{bmatrix} = \begin{bmatrix} \alpha & -\beta \\ \beta & \alpha \end{bmatrix} \begin{bmatrix} x_1^k \\ x_2^k \end{bmatrix} = A_{\text{espiral}}^k x(k) \quad , k = 0, 1, 2, \dots \quad (3)$$

Siendo necesario para que converja al origen que [5]:

$$\sqrt{\alpha^2 + \beta^2} < 1 \quad (4)$$

y para que pueda rotar en torno al origen [5]:

$$\beta \neq 0 \quad (5)$$

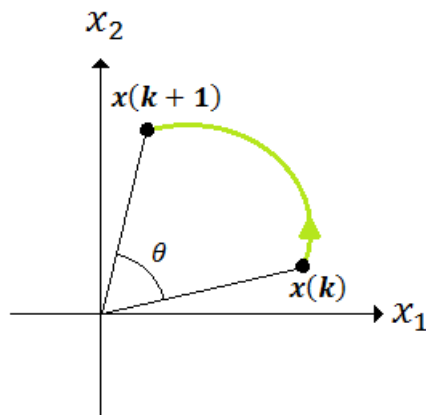
Usando el concepto de matriz de rotación, la matriz de parámetros  $A_{\text{espiral}}^k$  se precisa como [1]:

$$R_{1,2}^{(2)}(\theta) = \begin{bmatrix} \cos\theta & -\text{sen}\theta \\ \text{sen}\theta & \cos\theta \end{bmatrix} \quad (6)$$

en donde  $\theta$  es el ángulo de rotación en torno al origen.

La Figura 1 muestra los elementos de la rotación en torno al origen para cuando se tienen dos variables denominadas  $x_1$  y  $x_2$  [1].

Figura 1. Rotación 2D discreta con el modelo de espiral logarítmica en torno al origen.



(Basado en Tamura y Yasuda [1])

Para lo cual se tiene que [1]:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = rR_{1,2}^{(2)}(\theta) \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \quad (7)$$

En donde  $r$  es la tasa de convergencia de la distancia entre el punto  $x(k)$  y el punto hacia el cual se dirige la trayectoria de la espiral, que en este caso, es el origen.

Asumiendo que:

$$S_2(r, \theta) = rR_{1,2}^{(2)}(\theta) \quad (8)$$

$$x(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \quad (9)$$

La ecuación (7) equivale entonces a:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = S_2(r, \theta)x(k) \quad \parallel \quad x(0) = x_0 \quad (10)$$

en donde  $0 < \theta < 2\pi$  y  $0 < r < 1$ .

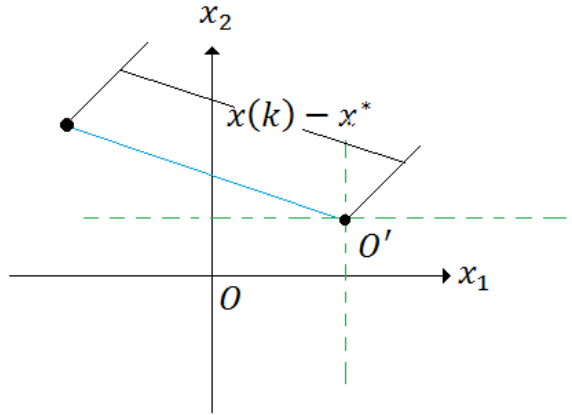
En [1], *Tamura y Yasuda* presenta la ecuación para la convergencia a un punto arbitrario al origen de la matriz de rotación:

$$x(k+1) = S_2(r, \theta)x(k) - (S_2(r, \theta) - I_2)x^* \quad (11)$$

No obstante, en el artículo citado anteriormente, no se presenta la deducción de la ecuación (11). La obtención de dicha ecuación, es expuesta a continuación por parte del autor del presente informe.

Considerando la convergencia a un punto arbitrario denominado  $O'$  como muestra la Figura 2.

Figura 2. Rotación 2D discreta con el modelo de espiral logarítmica en torno a un punto arbitrario.



(Fuente: autor)

en donde  $x^*$  es un punto de convergencia arbitrario, y  $x(k) - x^*$  es la distancia que separa al  $k$  -ésimo punto de este centro arbitrario.

Siendo  $x(k + 1)$  el punto obtenido por la rotación de  $x(k)$  según el modelo discreto mencionado de la espiral, se debe adicionar  $x^*$  para compensar el desplazamiento de los ejes evidenciado en la Figura 2 y con ello se obtiene la siguiente ecuación:

$$x(k + 1) = S_2(r, \theta)[x(k) - x^*] + x^* \quad (12)$$

Aplicando propiedad distributiva del producto de matrices:

$$\Rightarrow x(k + 1) = S_2(r, \theta)x(k) - S_2(r, \theta)x^* + x^* \quad (13)$$

Posfactorizando a  $x^*$  se llega a la ecuación citada:

$$\Rightarrow x(k + 1) = S_2(r, \theta)x(k) - (S_2(r, \theta) - I_2)x^* \quad (14)$$

en donde resulta evidente, que  $I_2$  es una matriz identidad de  $2 \times 2$ .

La ecuación anterior se redefine con el propósito de darle un carácter multipunto, y es ahora [1]:

$$x_i(k + 1) = S_2(r, \theta)x_i(k) - (S_2(r, \theta) - I_2)x^* \quad (15)$$

De modo que  $x^*$  es el centro común establecido como la mejor solución de todos los puntos durante la búsqueda [1].

Se establecen dos estrategias de búsqueda, las cuales se denominan diversificación e intensificación. La diversificación consiste en poder encontrar las mejores soluciones en una región extensa, y la intensificación consiste en encontrar las mejores soluciones en una región más reducida, establecida como resultado de la diversificación. Se entienden como dos técnicas opuestas y a su vez complementarias, porque la diversificación realiza un barrido de búsqueda creyendo encontrar mejores soluciones dentro de una zona relativamente más extensa, mientras que la intensificación hace un barrido de búsqueda más dirigido hacia unas buenas soluciones ya establecidas previamente. El modelo de la espiral descrito por la ecuación (11), realiza naturalmente estas dos técnicas, primero diversificándose, y luego intensificándose, al tiempo que la distancia al centro de rotación se acerca a cero de una forma exponencial [6].

El algoritmo propuesto por los autores *Tamura* y *Yasuda* para realizar la optimización por espiral en dos dimensiones es [1]:

### **Paso 0: Preparación**

Seleccionar el número de puntos de búsqueda  $m$ , de modo que sea al menos 2. Establecer el valor de  $\theta$  y  $r$  pertenecientes a  $S_2(r, \theta)$  de modo que  $0 < \theta < 2\pi$  y  $0 < r < 1$ . A su vez, establecer el número máximo de iteraciones  $k_{max}$ . Iniciar  $k$  en cero.

### **Paso 1: Inicialización**

Fijar los puntos iniciales  $x_i(0) \in \mathcal{R}^2, i = 1, 2, \dots, m$  en la región de factibilidad y el centro  $x^*$  tal como  $x^* = x_{ig}(0), ig = \arg \min_i f(x_i(0)), i = 1, 2, 3, \dots, m$

### **Paso 2: Actualizar $x_i$**

$$x_i(k+1) = S_2(r, \theta)x_i(k) - (S_2(r, \theta) - I_2)x^* \quad , i = 1, 2, 3, \dots \quad (16)$$

### **Paso 3: Actualizar $x^*$**

$$x^* = x_{ig}(k+1), ig = \arg \min_i f(x_i(k+1)), \quad i = 1, 2, 3, \dots, m \quad (17)$$

### **Paso 4: Verificar finalización**

Si  $k = k_{max}$  entonces terminar. En caso contrario  $k = k + 1$  y se retorna al Paso 2.

De forma análoga, *Tamura* y *Yasuda* [1] presentan el esquema de un algoritmo generalizado para  $n$ -dimensiones, en el cual el carácter multidimensional se aprecia en primera instancia, en los superíndices y subíndices  $n$  de las ecuaciones. Para este modelo  $n$ -dimensional, se establece una matriz de rotación  $R_{i,j}^{(n)}$  de  $n$ -dimensión, de modo que, en el caso de rotaciones en torno al origen:

$$x(k+1) = rR^{(n)}(\theta_{1,2}, \theta_{1,3}, \dots, \theta_{n-1,n})x(k) = S_n(r, \theta)x(k) \quad (18)$$

en donde:

$$R^{(n)}(\theta_{1,2}, \theta_{1,3}, \dots, \theta_{n-1,n}) = \prod_{i=1}^{n-1} \left( \prod_{j=1}^i R_{n-i, n+1-j}^{(n)}(\theta_{n-i, n+1-j}) \right) \quad (19)$$

El algoritmo propuesto por *Tamura* y *Yasuda* [1] para el caso de  $n$ -dimensiones es:

### **Paso 0: Preparación**

Seleccionar el número de puntos de búsqueda  $m$ , de modo que sea al menos 2. Establecer el valor de  $\theta$  y  $r$  pertenecientes a  $S_n(r, \theta)$  de modo que  $0 < \theta < 2\pi$  y  $0 < r < 1$ . A su vez, establecer el número máximo de iteraciones  $k_{max}$ . Iniciar  $k$  en cero.

### **Paso 1: Inicialización**

Fijar los puntos iniciales  $x_i(0) \in \mathcal{R}^n, i = 1, 2, \dots, m$  en la región de factibilidad y el centro  $x^*$  tal como  $x^* = x_{ig}(0), ig = \arg \min_i f(x_i(0)), i = 1, 2, 3, \dots, m$

### **Paso 2: Actualizar $x_i$**

$$x_i(k+1) = S_n(r, \theta)x_i(k) - (S_n(r, \theta) - I_n)x^* \quad , i = 1, 2, 3, \dots \quad (20)$$

### **Paso 3: Actualizar $x^*$**

$$x^* = x_{ig}(k+1), ig = \arg \min_i f(x_i(k+1)), i = 1, 2, 3, \dots, m \quad (21)$$

#### Paso 4: Verificar finalización

Si  $k = k_{max}$  entonces terminar. En caso contrario  $k = k + 1$  y se retorna al Paso 2.

### 3.3 Teorema para la utilización de un método de optimización, en la solución a un sistema de ecuaciones

Un sistema de  $n$  ecuaciones no-lineales y  $n$  incógnitas puede ser descrito de la siguiente forma:

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= 0 \\ f_2(x_1, x_2, \dots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, x_2, \dots, x_n) &= 0 \end{aligned} \tag{22}$$

Resolver este sistema de ecuaciones, como un problema de optimización, puede hacerse a través de la búsqueda del valor mínimo de la función objetivo definida como [7]:

$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^n (f_i(x_1, x_2, \dots, x_n))^2 \tag{23}$$

Por ejemplo, si se quiere resolver la ecuación *no-lineal*,

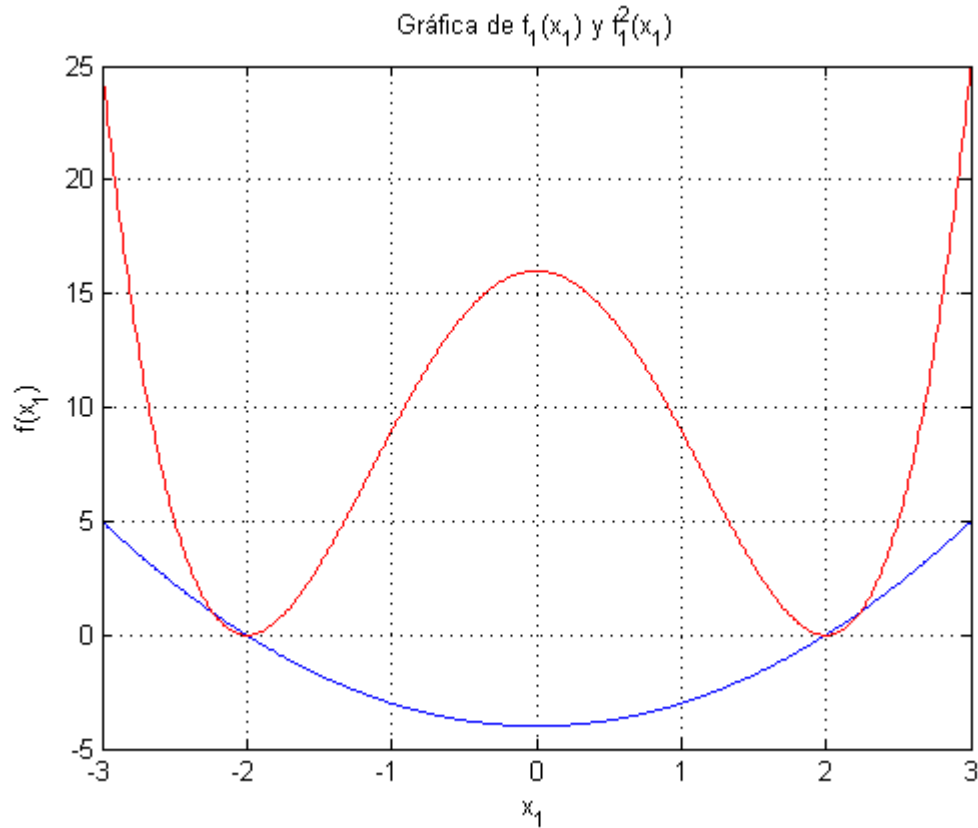
$$f_1 = x_1^2 - 4 = 0 \tag{24}$$

la cual está en términos de la variable  $x_1$ , la función objetivo para el algoritmo de optimización ha de ser:

$$f(x_1) = (x_1^2 - 4)^2 \quad (25)$$

La gráfica de la función *no-lineal* y la función objetivo de las ecuaciones anteriores, se presentan en la Figura 3 en colores azul y rojo respectivamente.

Figura 3. Gráfica de una ecuación no-lineal y la función objetivo asociada.



(Fuente: autor)

En la figura anterior, es posible observar que la función objetivo es *no-negativa*, y que sus valores mínimos corresponden con las raíces de la ecuación *no-lineal* asociada.

### 3.4 Modelos para dispositivos *no-lineales* utilizados

Los modelos utilizados para describir el comportamiento  $I$ - $V$ , tienen un compromiso entre complejidad y precisión, porque al aumentar la complejidad del modelo, se requiere un mayor tiempo de ejecución para conseguir una precisión dada, lo cual no solo ha de aplicar al modelo de un *MOSFET*, sino en general, al modelo matemático de cualquier otro dispositivo o fenómeno de la naturaleza [8]. Para los circuitos *no-lineales* presentados en este documento, se han utilizado los siguientes modelos:

#### 3.4.1 Diodo

Para un diodo se ha establecido que la corriente de dren es [2]:

$$i_D = I_s(e^{\frac{v_D}{nV_T}} - 1) \quad (26)$$

Donde:

$v_D$ : Voltaje del diodo

$n$  : Coeficiente de emisión (valor de 1 para el Germanio)

$I_s$ : Corriente de saturación (usualmente  $10^{-12}$  A)

$V_T$  : Voltaje térmico (usualmente 25.85 mV)

#### 3.4.2 Transistor *BJT*

Para este dispositivo, se presentan algunas consideraciones de interés concernientes a dos modelos distintos del mismo dispositivo:

### **Modelo exponencial básico**

Así se le denomina en este trabajo de grado, al modelo en el cual cuando el transistor opera en la región directa, la corriente en cualquiera de los terminales puede ser expresada en términos de una sola variable, representada por la tensión *base-emisor* del transistor [9], tal como lo expresan las ecuaciones (27) a (29).

$$I_C = \beta I_B = I_S e^{\frac{v_{BE}}{V_T}} \quad (27)$$

$$I_B = \frac{I_S}{\beta} e^{\frac{v_{BE}}{V_T}} \quad (28)$$

$$I_E = (\beta + 1) I_B = \frac{\beta + 1}{\beta} I_S e^{\frac{v_{BE}}{V_T}} \quad (29)$$

Donde:

$I_C$ : Corriente de *colector*.

$I_B$ : Corriente de *base*.

$I_E$ : Corriente de *emisor*.

$\beta$ : Ganancia de corriente de *emisor común* (usualmente 100).

$I_S$ : Corriente de saturación inversa del diodo (del orden de  $10^{-15}$  a  $10^{-12}$  A).

$v_{BE}$ : Tensión de *base-emisor*.

$V_T$ : Voltaje térmico (usualmente 25.85 mV).

### **Modelo Ebers-Moll**

Este es un modelo, que bajo algunas consideraciones especiales se aproxima al modelo exponencial básico. No obstante, representa una concepción más

completa del dispositivo, y con ello, más compleja que la presentada anteriormente, porque en ella se encuentran incluidas todas las regiones de operación del transistor. Propone que dado que un transistor *BJT* está conformado por dos uniones *pn* o *np* entre emisor-base y base-colector, entonces la corriente en cada una de sus terminales puede ser expresada como la superposición de las corrientes en estas dos uniones. Desde esta perspectiva, las corrientes para cada uno de los terminales están cada una en función de dos variables, las cuales son las tensiones *base-emisor* y *base-colector* del transistor, tal como lo muestran las siguientes ecuaciones [10]:

$$i_E = \frac{I_S}{\alpha_F} \left( e^{\frac{v_{BE}}{V_T}} - 1 \right) - I_S \left( e^{\frac{v_{BC}}{V_T}} - 1 \right) \quad (30)$$

$$i_C = I_S \left( e^{\frac{v_{BE}}{V_T}} - 1 \right) - \frac{I_S}{\alpha_R} \left( e^{\frac{v_{BC}}{V_T}} - 1 \right) \quad (31)$$

$$i_B = \frac{I_S}{\beta_F} \left( e^{\frac{v_{BE}}{V_T}} - 1 \right) - \frac{I_S}{\beta_R} \left( e^{\frac{v_{BC}}{V_T}} - 1 \right) \quad (32)$$

Por otra parte, usando la relación entre  $\alpha$  y  $\beta$  se llega a las expresiones:

$$\beta_F = \frac{\alpha_F}{1 - \alpha_F} \quad (33)$$

$$\beta_R = \frac{\alpha_R}{1 - \alpha_R} \quad (34)$$

Donde:

$I_C$ : Corriente de *colector*.

$I_B$ : Corriente de *base*.

$I_E$ : Corriente de *emisor*.

$\beta_F, \beta_R$  : Ganancias de corriente de *emisor común* ( $I_C/I_B$ ) en operación directa e inversa respectivamente.

$\alpha_F, \alpha_R$  : Ganancias de corriente de *base común* ( $I_C/I_E$ ) en operación directa e inversa respectivamente.

$I_S$ : Corriente de saturación inversa del diodo (*del orden de  $10^{-15}$  a  $10^{-12}$  A*).

$v_{BE}$ : Tensión de *base-emisor*.

$V_T$ : Voltaje térmico (*usualmente 25.85 mV*).

### 3.4.3 Transistor MOSFET

Para este transistor, en este trabajo de grado se ha utilizado el modelo propuesto por *Shockley* [11] para dispositivos de largo canal (superiores a los 100nm), en el cual se consideran los efectos de segundo orden de *modulación de canal* y *efecto cuerpo*. La ecuación para la corriente de dren  $i_D(v_{GS}, v_{DS})$  en las regiones de operación de saturación, triodo y corte para el *NMOS* y *PMOS* son presentadas a continuación [12].

#### Transistor *NMOS*

$$i_D = \begin{cases} \frac{1}{2} K_n \frac{W}{L} (V_{GS} - V_t)^2 (1 + \lambda V_{DS}) & V_{DS} \geq V_{GS} - V_t \text{ y } V_{GS} > V_t \\ K_n \frac{W}{L} ((V_{GS} - V_t)V_{DS} - \frac{1}{2} V_{DS}^2) & V_{DS} < V_{GS} - V_t \text{ y } V_{GS} > V_t \\ 0 & V_{GS} < V_t \end{cases} \quad (35)$$

con:

$$K_n = \mu_n C_{ox} \quad (36)$$

## Transistor *PMOS*

$$i_D = \begin{cases} \frac{1}{2} K_p \frac{W}{L} (|V_{GS}| - |V_t|)^2 (1 + \lambda |V_{DS}|) & |V_{DS}| \geq |V_{GS}| - |V_t|, |V_{GS}| > |V_t| \\ K_p \frac{W}{L} ((|V_{GS}| - |V_t|) |V_{DS}| - \frac{1}{2} V_{DS}^2) & |V_{DS}| < |V_{GS}| - |V_t|, |V_{GS}| > |V_t| \\ 0 & |V_{GS}| < |V_t| \end{cases} \quad (37)$$

con:

$$K_p = \mu_p C_{ox} \quad (38)$$

en donde:

$K_n, K_p$ : Parámetros de transconductancia.

$\mu_n, \mu_p$ : Movilidad efectiva de los portadores de carga.

$C_{ox}$ : Capacitancia del óxido por unidad de área.

$W$ : Ancho de la *puerta*.

$L$ : Longitud de la *puerta*.

$V_{GS}$ : Tensión diferencial de *puerta a fuente*.

$V_{DS}$ : Tensión diferencial de *dren a fuente*.

$\lambda$ : Modulación de longitud de canal.

$V_t$ : Tensión de umbral.

Y a su vez:

$$V_t = V_{t0} + \gamma (\sqrt{|2\phi_F + V_{SB}|} - \sqrt{2\phi_F}) \quad (39)$$

en donde:

$V_{t0}$ : Tensión de umbral para *cero-BIAS* ( $V_{SB} = 0$ ).

$V_{SB}$ : Tensión diferencial entre *fuentes* y *cuerpo*.

$\gamma$ : Parámetro del máximo umbral.

$\phi_F$ : Potencial de superficie.

Considerando la dependencia con la temperatura, se tiene una expresión para  $\phi_F$  [13]:

$$\phi_F = \frac{kT}{q} \text{Ln} \left( \frac{N_A e^{\frac{E_g}{2kT}}}{\sqrt{N_c N_v}} \right) \quad (40)$$

donde:

$k$ : Constante de *Boltzmann*.

$T$ : Temperatura en grados *Kelvin*.

$q$ : Magnitud de la carga de un electrón.

$N_A$ : Densidad de dopaje.

$E_g$ : Energía de *band-gap*.

$N_c$ : Densidad de los estados permitidos cerca del borde de la banda de conducción.

$N_v$ : Densidad de los estados permitidos cerca del borde de la banda de conducción.

#### 4. ANÁLISIS DE LA DIVERSIFICACIÓN E INTENSIFICACIÓN EN LA TRAYECTORIA DE ESPIRAL

Dadas las definiciones de diversificación e intensificación presentadas después de la ecuación (15) de este documento; es posible encontrar comparativamente, trayectorias de espiral que se diversifican más que otras. Un punto de vista cualitativo de esto, es que, al establecerse una trayectoria de espiral, hay que considerar que entre más rote en torno a un punto, y más se prorrogue su convergencia hacia él, se convierte en más diversificada su trayectoria, y con ello menos intensificada en torno a dicho punto.

Cuantitativamente, en base a la ecuación (11), la trayectoria de una espiral con este modelo discreto, está determinada por los valores de los parámetros  $r$  y  $\theta$ . Asumiendo un ángulo de rotación, que representa una porción respecto a toda una vuelta, se tiene que:

$$\theta = \frac{2\pi}{n} \quad (41)$$

donde  $0 < \theta < 2\pi$ , y por consiguiente  $n > 1$ . De este modo, cuando suceda una rotación completa, la razón de convergencia habrá hecho que la distancia al punto de convergencia se disminuya por un factor que se ha denominado  $r_v$ , y está dado por:

$$r_v = r^n \quad (42)$$

donde  $0 < r < 1$ , y en consecuencia  $0 < r_v < 1$ .

A medida que  $r_v$  se aproxima a 1, la trayectoria se asemeja a una circunferencia, y si  $r_v$  se aproxima a 0, la trayectoria parecerá una línea recta, tal como se verá más

adelante en la Figura 8 y Figura 9, respectivamente. De las ecuaciones (44) y (45) se llega a qué:

$$r_v = r^{\frac{2\pi}{\theta}} \quad (43)$$

así que, si se quiere diversificar más la trayectoria de la espiral, una condición necesaria, pero no suficiente, es aumentar  $r_v$ , lo cual se consigue aumentando  $r$  y/o aumentando  $\theta$ . Por otra parte, si lo que se quiere es intensificar, se hace necesario disminuir  $r$  y/o disminuir  $\theta$ , lo cual permite una trayectoria más dirigida hacia el centro de rotación. Adicionalmente, la otra condición requerida para diversificar ó intensificar más la trayectoria es hacerla ganar continuidad, es decir, que cada punto esté más cerca de su antecesor, lo cual se consigue, cuando en lo posible  $\theta$  disminuye, y  $r$  aumenta, cada una dentro de su dominio permitido.

Se infiere de lo anterior que, dados los dominios  $0 < r < 1$ , y  $0 < \theta < 2\pi$ ; para diversificar, es necesario aumentar el valor de  $r$ , manteniendo un valor de  $\theta$  pequeño dentro del dominio permitido para  $\theta$ ; y para intensificar, se requiere disminuir el valor de  $\theta$ , manteniendo un valor superior de  $r$ , y que esté dentro del dominio que rige a  $r$ .

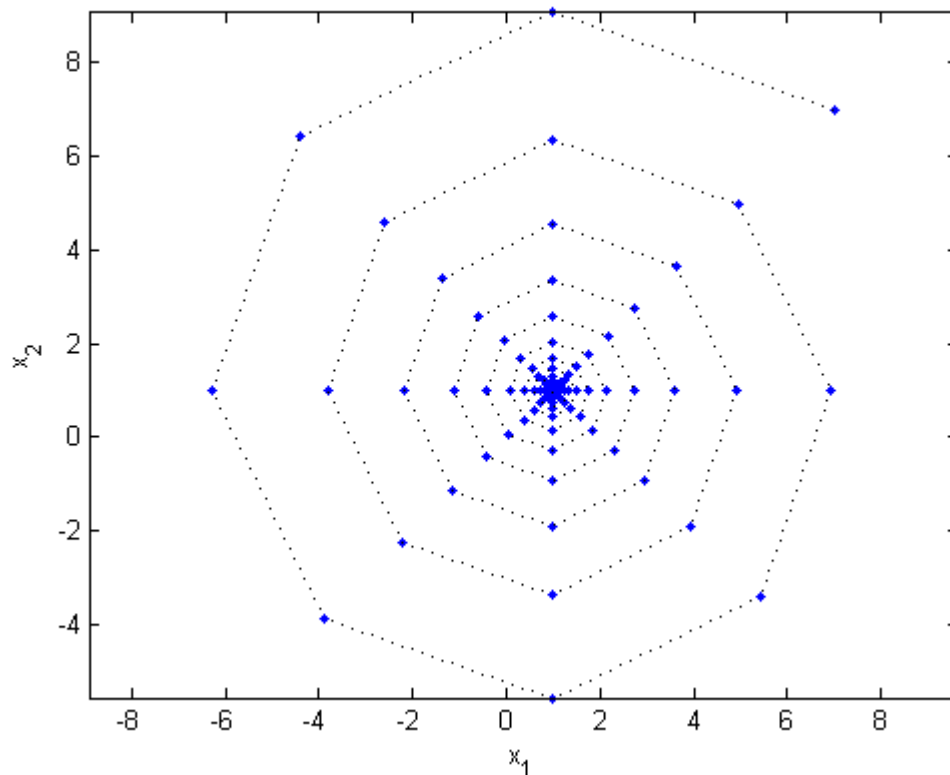
#### 4.1 Gráficas de algunas trayectorias

Con el propósito de evaluar la parte del algoritmo de *Tamura* y *Yasuda*, que implementa la ecuación (11), donde se describe el modelo discreto de la espiral de rotación, y para evidenciar los resultados en torno a las ecuaciones (42) y (43), se presentan diferentes gráficas de una trayectoria de espiral, para un total de 401 puntos, con punto inicial (7, 7) y punto para convergencia de coordenadas (1,1).

#### 4.1.1 Trayectoria de espiral con unos valores de referencia

Son algunos de los valores propuestos por los autores *Tamura y Yasuda* [1], para la ejecución del algoritmo de la espiral,  $r = 0.95$  y  $\theta = \pi/4$ . Con lo cual, de la ecuación (43) da  $r_v = 0.95^8 \cong 0.6634$ . La Figura 4 presenta la trayectoria de una espiral con estos valores de parámetros, y luce como una espiral que diversifica bien.

Figura 4. Trayectoria de una espiral logarítmica obtenida con los parámetros de referencia.



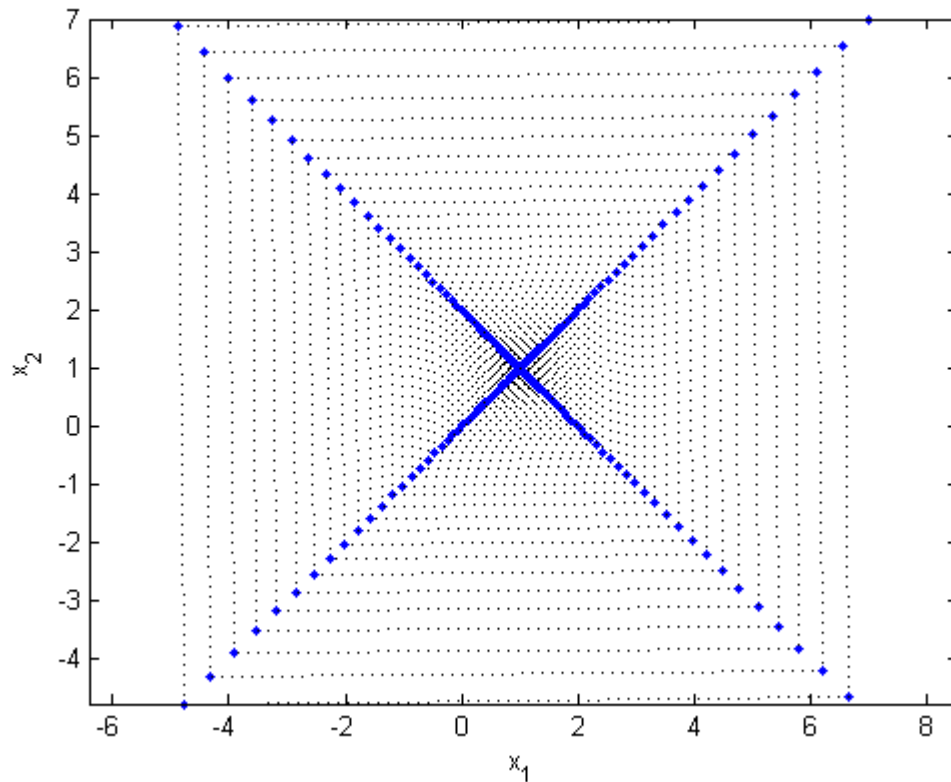
(Fuente: autor)

#### 4.1.2 Trayectoria de espiral aumentando $r$ y $\theta$ .

Considerando los valores  $r = 0.98$  y  $\theta = \pi/2$ ; de la ecuación (43), se tiene que  $r_v = 0.98^4 \cong 0.92237$ , con lo cual efectivamente se consiguió un aumento para el

factor de convergencia por vuelta, comparado con el obtenido para el caso anterior, pero debido a que  $\theta$  es considerablemente grande, cada punto y su antecesor están bien distanciados, y por ello la trayectoria no diversifica muy bien. La Figura 5 muestra la convergencia de la espiral en este caso:

Figura 5. Trayectoria de una espiral logarítmica obtenida al aumentar los parámetros.

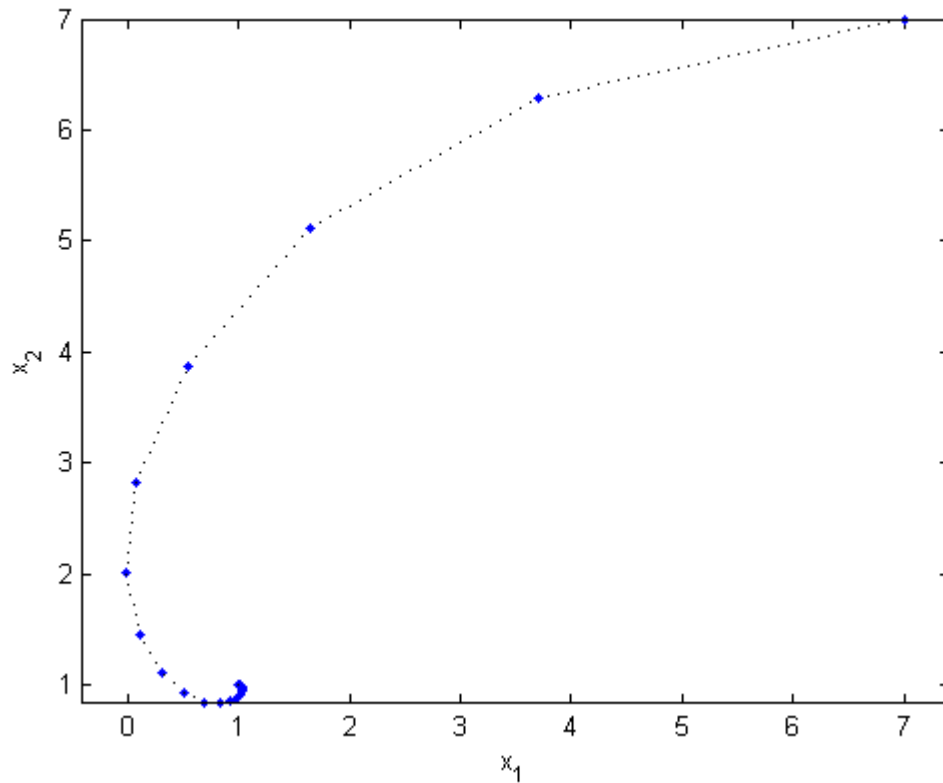


(Fuente: autor)

#### 4.1.3 Trayectoria de espiral disminuyendo $r$ y $\theta$ .

Si ahora los valores de los parámetros son,  $r = 0.7$  y  $\theta = \frac{\pi}{10}$ ; en consecuencia, de la ecuación (43),  $r_v = 0.7^{20} = 7.979226629761191 \times 10^{-4}$ , el cual es menor que los anteriores, pero no obstante, no le permite a la trayectoria intensificar muy bien, debido a que el valor de  $r$  se ha hecho pequeño. La Figura 6 muestra la trayectoria de convergencia en este caso:

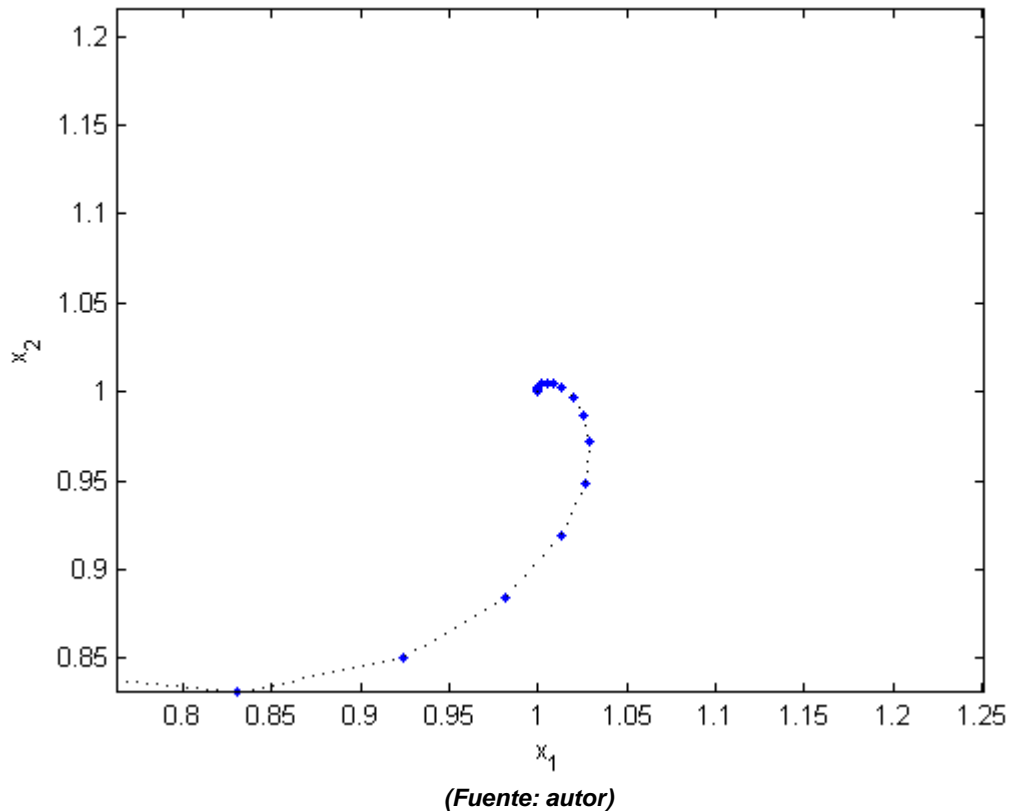
Figura 6. Trayectoria de una espiral logarítmica obtenida al disminuir los parámetros.



(Fuente: autor)

Realizando una ampliación sobre la gráfica anterior, se observa como la trayectoria va rotando en torno al punto de convergencia, tal como lo muestra la Figura 7.

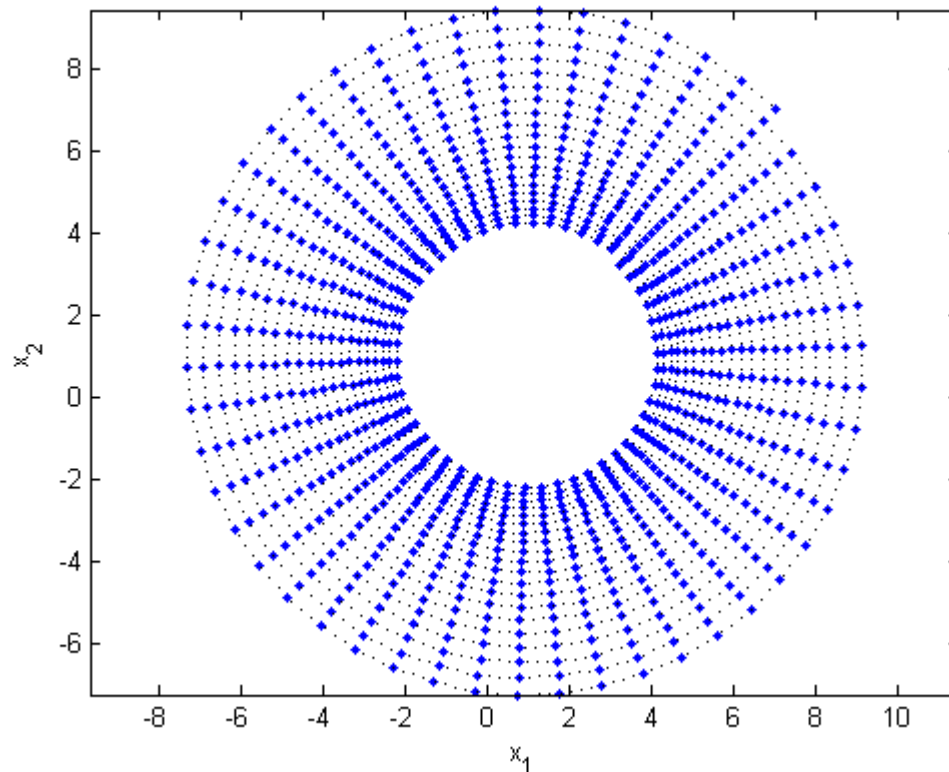
Figura 7. Ampliación de la Figura 6.



#### 4.2 Ejemplo de una trayectoria de espiral de carácter diversa

Si se hace  $r = 0.999$  y  $\theta = \pi/25$ , de la ecuación (43),  $r_v = 0.999^{50} \cong 0.951206$ . La Figura 8 muestra la trayectoria en este caso, la cual luce conformada por lo que parecen circunferencias concéntricas, y como tal, es una trayectoria que se diversifica bastante.

Figura 8. Trayectoria de una espiral logarítmica diversa.

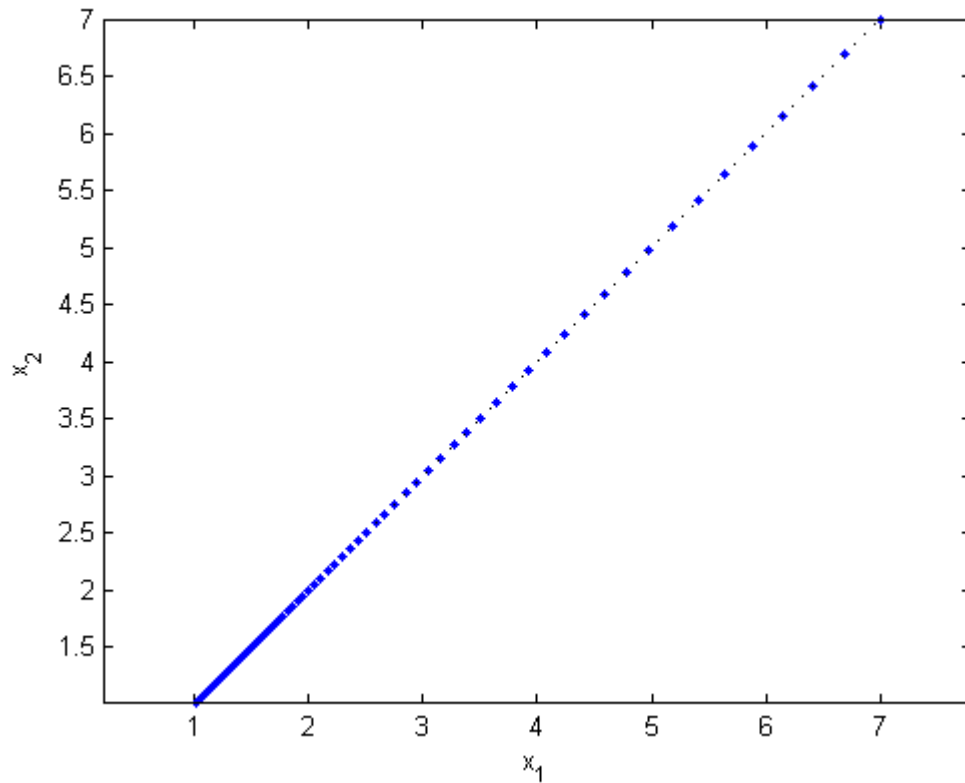


(Fuente: autor)

### 4.3 Ejemplo de una trayectoria de espiral de carácter intensa

Si se hace  $r = 0.95$  y  $\theta = 2\pi/10^{40}$ , usando la ecuación (43), se tiene que  $r_v = 0.95^{(2 \cdot 10^{40})} \rightarrow 0^+$ . La Figura 9 muestra la trayectoria para este caso, que luce como una línea recta, y con ello, tiene un carácter de intensificación.

Figura 9. Trayectoria de una espiral logarítmica intensa.



(Fuente: autor)

Los parámetros que permiten obtener la Figura 9, u otros que también poseen un valor bien pequeño para  $\theta$ , son usados en repetidas ocasiones para resolver sistemas de ecuaciones dentro de esta tesis de investigación, porque arrojan buena precisión en los resultados, siendo que permiten una búsqueda intensificada para cada trayectoria de espiral; y por otra parte, debido al carácter multipunto (y con ello de múltiples trayectorias) de la búsqueda, le conceden también al método, hacerse diverso, dentro de su región factible.

## 5. PRUEBAS DEL ALGORITMO

Se evidencia el cumplimiento del primero y segundo objetivo específico (ver página 21) por medio de utilizar funciones típicas conocidas en la literatura internacional para verificar la convergencia del algoritmo a unos valores esperados. Estas pruebas se realizan y documentan para funciones de 1 a 4 variables. No obstante, el programa elaborado, permite realizarlas de forma satisfactoria hasta un máximo de 10 variables. En el ANEXO se presenta de forma detallada el procedimiento de utilización del algoritmo. Los datos de la ejecución del programa están consignados desde la *Tabla 2* en adelante, con las siguientes convenciones para los parámetros y los resultados:

***Ejec***: Número correspondiente al orden de la ejecución.

***r***: Razón de convergencia citada en la ecuación (7), cuyo valor satisface que  $0 < r < 1$ .

***θ***: Angulo de rotación citado en la ecuación (6), cuyo valor satisface que  $0 < θ < 2π$ .

***kmax***: Número máximo de iteraciones posibles para ejecutar el programa.

***ε***: Error tolerado para la solución final de la función objetivo, el cual se usa como parámetro para detener la ejecución del programa antes de llegar hasta *kmax*.

***m***: Número de puntos usados para la región factible.

***lix<sub>i</sub>, lsx<sub>i</sub>, nxi***: Límite inferior, límite superior y número de divisiones en la retícula de la región factible correspondiente a la variable  $x_i$ .

***x<sub>i</sub>***: Solución óptima final obtenida para la *i*-ésima variable.

***Error***: Es el valor que se obtiene al evaluar la función objetivo en los resultados obtenidos para  $x_i$ .

***k***: Número de iteraciones efectivamente realizadas para converger.

***tejec*** : Tiempo empleado por la ejecución del programa sin generar gráfica.

El programa implementado usa como criterio de parada la comparación con un error tolerado  $\varepsilon$ , en el marco de un ciclo de programación *while*. Cada subprograma de *n-variables* posee una rutina que con base a un límite inferior y superior para las variables, junto con el valor de  $nx_i$ , crea una retícula de puntos para la región factible. Se presentan gráficas de las trayectorias de convergencia para casos de hasta tres dimensiones. No obstante, solo se presenta la tabla de resultados para algunas situaciones de cuatro variables en adelante.

### 5.1 Función De Jong's

Se trata de una de las más sencillas funciones de prueba reportadas, y está caracterizada por ser una función *continua* y *unimodal*. La ecuación para *n-dimensiones* de esta función es [14]:

$$f(x) = \sum_{i=1}^n x_i^2 \quad (44)$$

Esta función posee un mínimo en  $x_i = 0$ ,  $i = 1, 2, \dots, n$  y su valor mínimo es cero. Su región de búsqueda estándar en los algoritmos de optimización está descrita por  $-5.12 \leq x_i \leq 5.12$ .

#### ***Función De Jong's de una variable***

En base a la ecuación (44) se tiene la siguiente función:

$$f(x_1) = (x_1)^2 \quad (45)$$

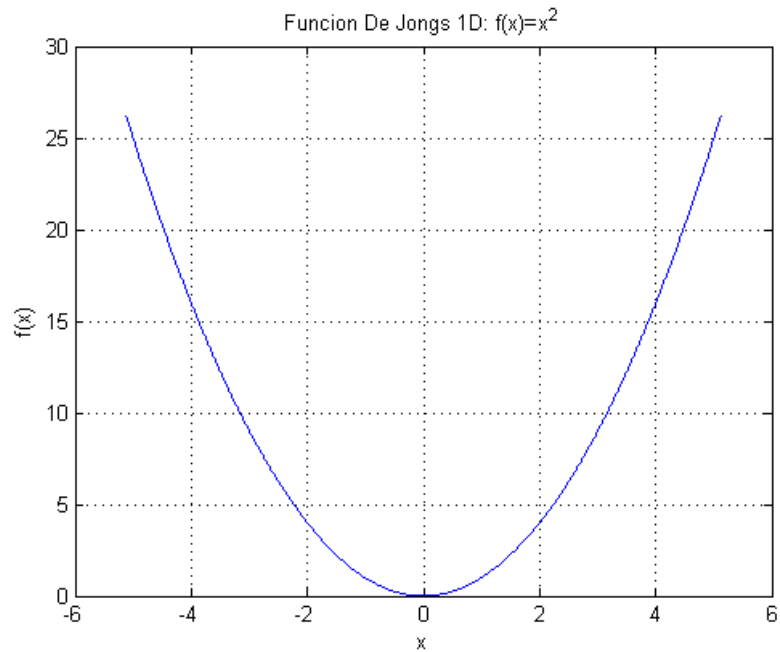
Que tiene su mínimo global en:

$$x_1 = 0$$

(46)

La gráfica de esta función se presenta en la Figura 10.

Figura 10. Función *De Jongs* 1D.



(Fuente: autor)

En la Tabla 2 se presentan los valores de los parámetros de ejecución, así como el resultado obtenido para la variable  $x_1$ , el cual es muy similar al valor teórico expuesto en la ecuación (46).

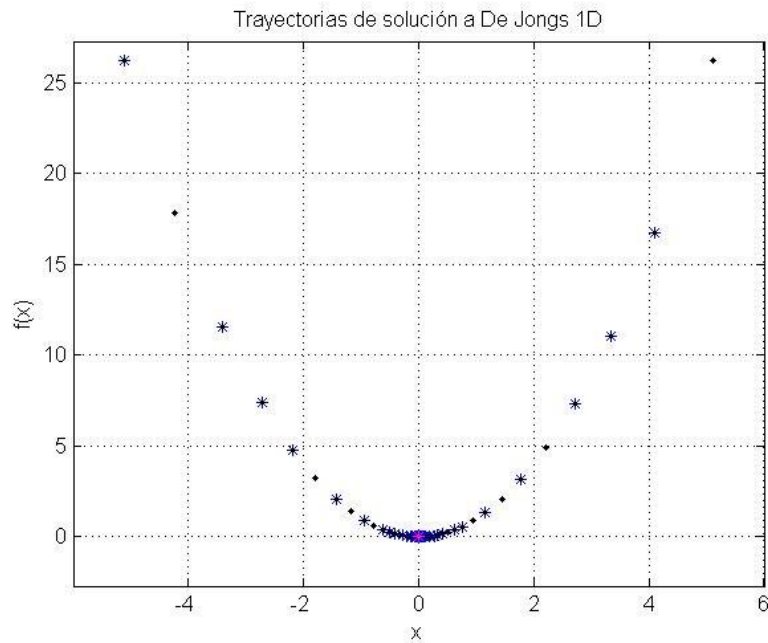
Tabla 2. Parámetros y resultados de simulación para solución de la función *De Jong's* en 1D.

$r$	0.9
$\theta$	$\pi/900$
$k_{max}$	700
$\varepsilon$	$10^{-30}$
$m$	2
$lix1,lsx1,nx1$	-5.12, 5.12, 1
$x1$	$9.271465402276206 \times 10^{-16}$
<i>Error</i>	$8.596007070560470 \times 10^{-31}$
$k$	345
<i>tejec</i>	0.223837 s

(Fuente: autor)

La Figura 11 muestra la trayectoria de convergencia hacia el valor óptimo final  $x1$ , el cual se encuentra destacado en color magenta.

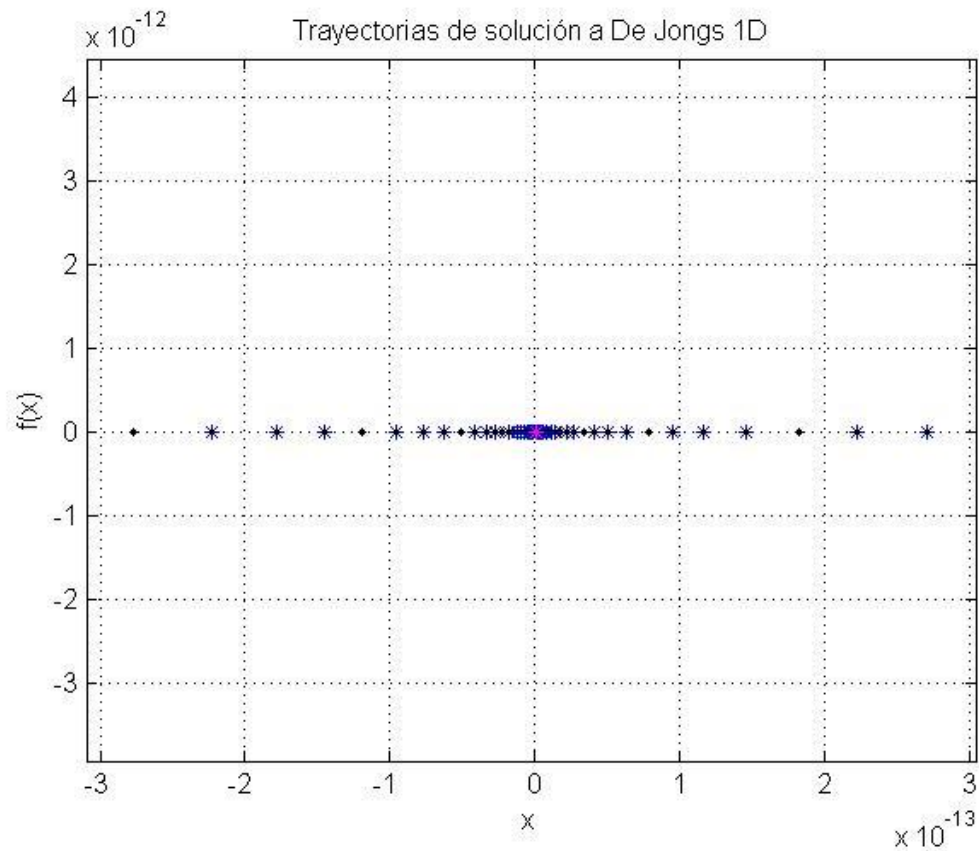
Figura 11. Convergencia para la solución de la ecuación de *De Jong's* en 1D.



(Fuente: autor)

Realizando una ampliación sobre la gráfica presentada anteriormente, se obtiene la Figura 12.

**Figura 12. Ampliación de la Figura 11.**



(Fuente: autor)

### ***Función De Jong's en dos variables***

En base a la ecuación (44), se obtiene la ecuación correspondiente para este caso:

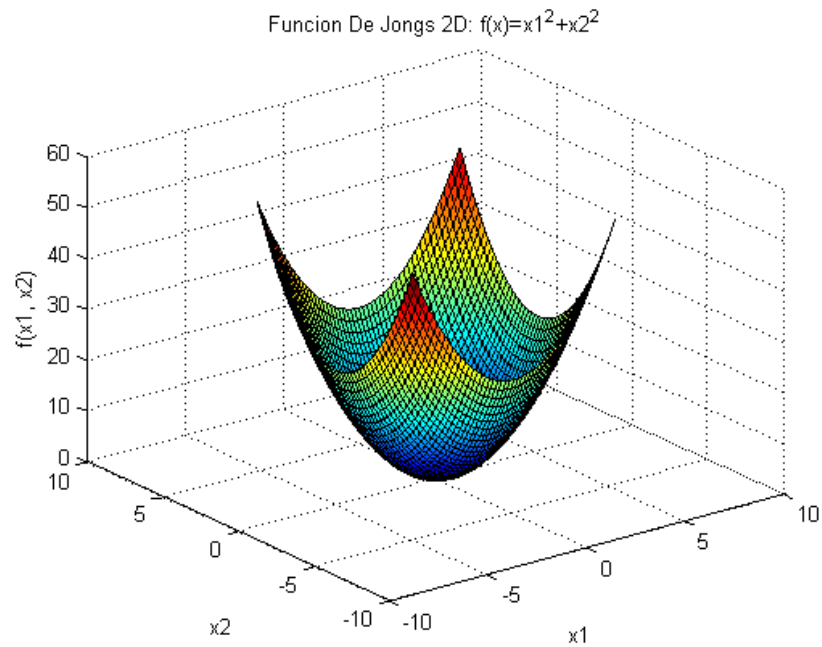
$$f(x_1, x_2) = (x_1)^2 + (x_2)^2 \tag{47}$$

Cuyo valor mínimo teórico se encuentra en:

$$x_1 = 0; \quad x_2 = 0 \quad (48)$$

La gráfica de esta función se presenta en la Figura 13.

Figura 13. Gráfica de la función *De Jong's 2D*.



(Fuente: autor)

En la Tabla 3, para el caso de dos dimensiones, se muestran los resultados del valor óptimo final obtenido con el algoritmo, los cuales tienen buena precisión comparados con los valores de la ecuación (48):

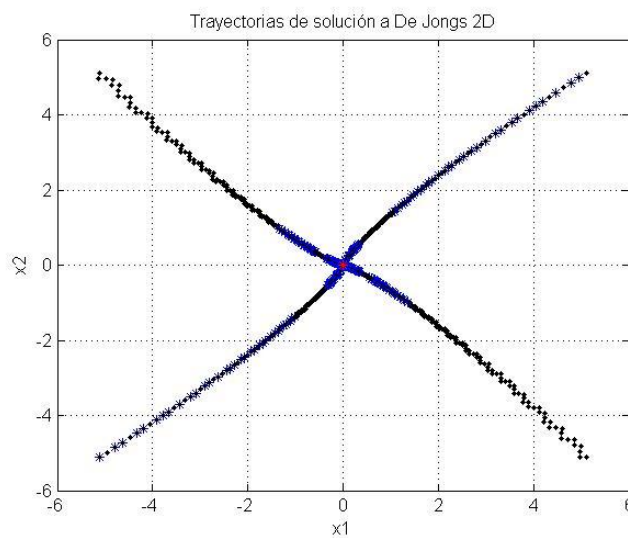
Tabla 3. Parámetros y resultados de simulación para solución de la función *De Jong's* en 2D.

<i>r</i>	0.985
<i>θ</i>	pi/2000
<i>kmax</i>	2000
<i>m</i>	4
<i>ε</i>	10 <sup>-7</sup>
<i>lix1, lisx1, nx1</i>	-5.12, 5.12, 1
<i>lix2 lisx2, nx2</i>	-5.12, 5.12, 1
<i>x1</i>	-7.935770087885020x10 <sup>-5</sup>
<i>x2</i>	2.973547046336231x10 <sup>-4</sup>
<i>Error</i>	9.471746505652629x10 <sup>-8</sup>
<i>k</i>	667
<i>tejec</i>	0.435985 s

(Fuente: autor)

La Figura 14, muestra la trayectoria de convergencia hacia el valor óptimo final, el cual se encuentra destacado en color magenta.

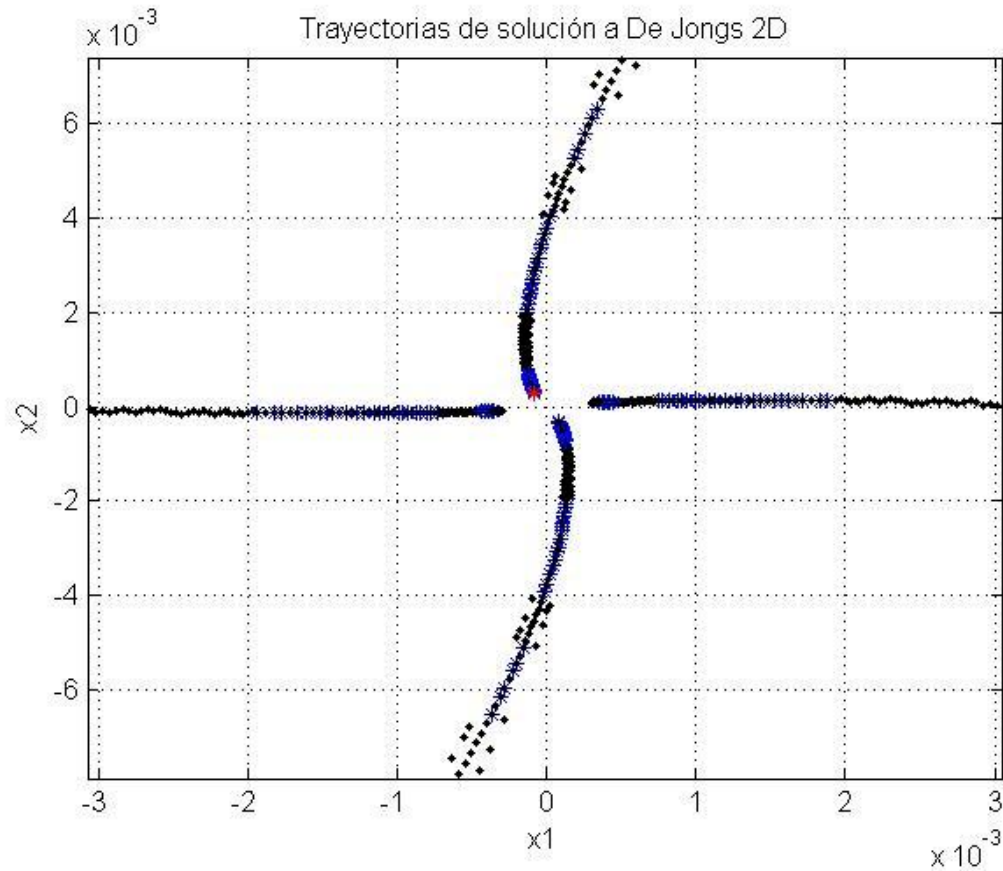
Figura 14. Convergencia para la solución de la ecuación de *De Jong's* en 2D.



(Fuente: autor)

En la figura anterior puede apreciarse que se tienen cuatro puntos iniciales para la región factible, como lo reporta la Tabla 3 ( $m = 4$ ), razón por la cual se observan cuatro brazos de trayectorias de convergencia. Realizando una ampliación sobre la anterior gráfica, se obtiene la Figura 15.

**Figura 15. Ampliación sobre la gráfica de la Figura 14.**



(Fuente: autor)

En la Figura 15, la ampliación evidencia mejor la convergencia hacia el valor de la solución óptima. Las trayectorias de convergencia hacia el valor óptimo final, lucen como espirales degeneradas, debido a la mutación del centro de convergencia de la espiral que sucede en cada iteración, cuando el algoritmo determina el nuevo mejor óptimo.

### ***Función De Jong's para tres variables***

De la ecuación (44), se obtiene la ecuación que corresponde a este caso:

$$f(x_1, x_2, x_3) = (x_1)^2 + (x_2)^2 + (x_3)^2 \quad (49)$$

Cuyo valor mínimo se encuentra en:

$$x_1 = 0, \quad x_2 = 0, \quad x_3 = 0 \quad (50)$$

La Tabla 4 reporta que efectivamente el programa genera con los parámetros allí consignados, valores de solución precisos con los resultados presentados en la ecuación anterior, y con un error menor a  $10^{-30}$ .

**Tabla 4. Parámetros y resultados de simulación para solución de la función *De Jong's* en 3D.**

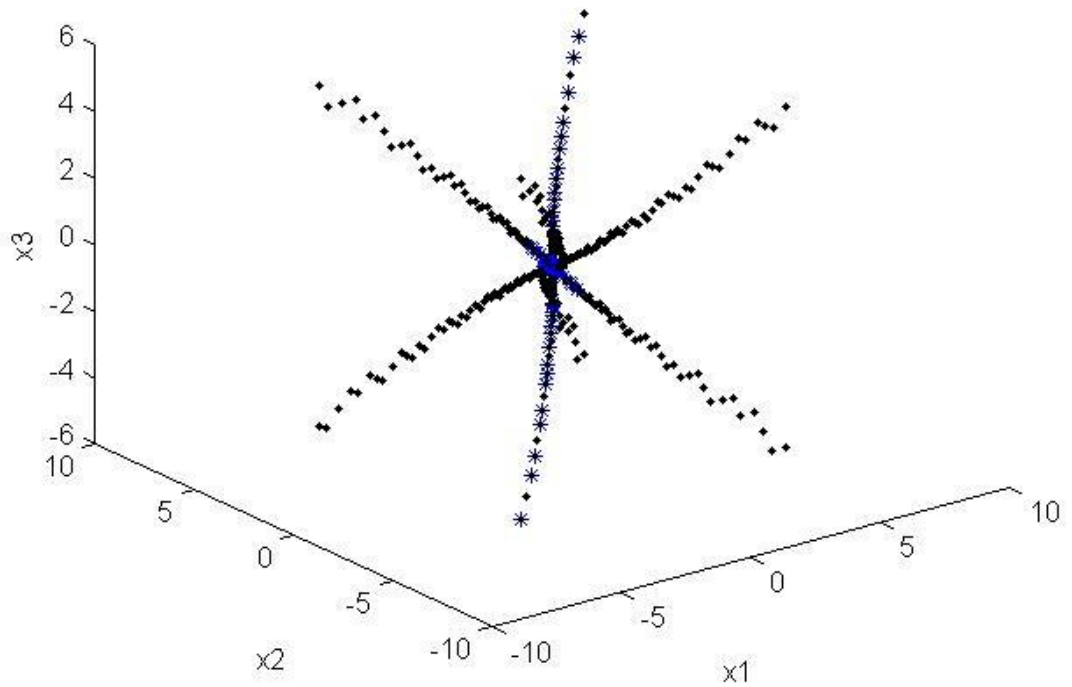
<i>r</i>	0.95
<i>θ</i>	pi/400
<i>Kmax</i>	20000
<i>m</i>	4
<i>ε</i>	$10^{-30}$
<i>lix1,lsx1,nx1</i>	-5.12, 5.12, 1
<i>lix2,lsx2,nx2</i>	-5.12, 5.12, 1
<i>lix3,lsx3,nx3</i>	-5.12, 5.12, 1
<i>x1</i>	$-1.673287767539407 \times 10^{-16}$
<i>x2</i>	$-5.890074141325593 \times 10^{-17}$
<i>x3</i>	$-9.793481678523792 \times 10^{-16}$
<i>k</i>	717
<i>Error</i>	$9.905910507448133 \times 10^{-31}$
<i>tejec</i>	0.673650 s

(Fuente: autor)

Las trayectorias de convergencia hacia el valor óptimo final son presentadas en la Figura 16:

Figura 16. Convergencia para la solución de la ecuación de *De Jong's* en 3D.

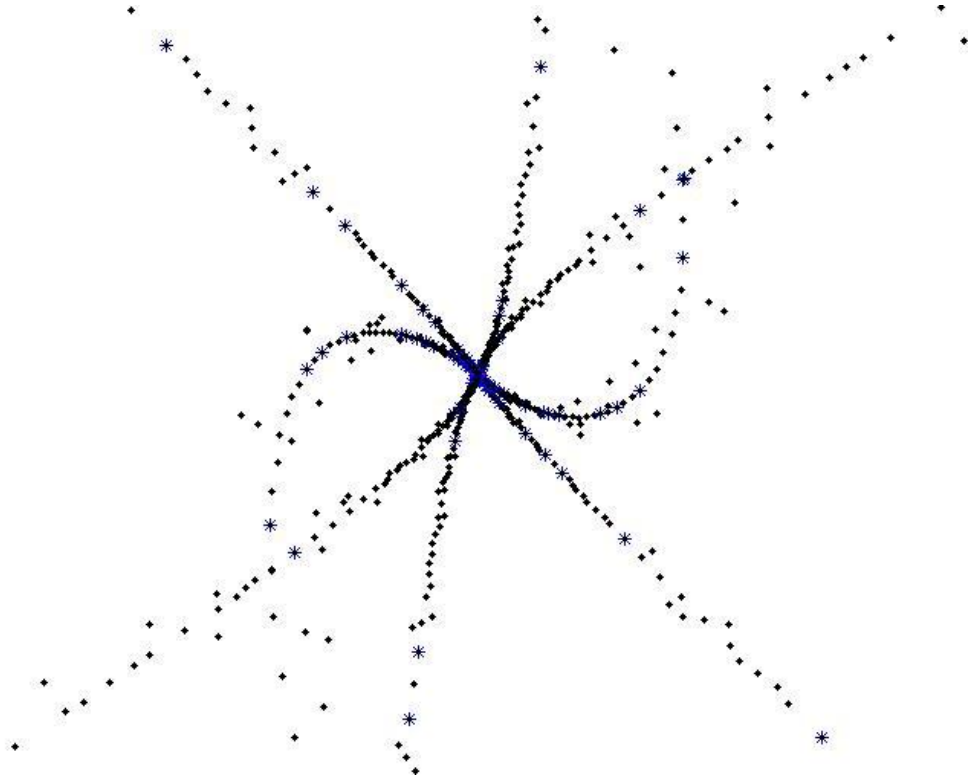
Trayectorias de solución a De Jongs 3D



(Fuente: autor)

Una ampliación sobre la gráfica anterior, se muestra en la Figura 17.

Figura 17. Ampliación de la Figura 16.



(Fuente: autor)

***Función De Jong's para cuatro variables:***

En base a la ecuación (44), se obtiene la ecuación correspondiente para este caso:

$$f(x_1, x_2, x_3, x_4) = (x_1)^2 + (x_2)^2 + (x_3)^2 + (x_4)^2 \quad (51)$$

Cuyo mínimo teórico se encuentra en:

$$x_1 = 0, \quad x_2 = 0, \quad x_3 = 0, \quad x_4 = 0 \quad (52)$$

La Tabla 5 muestra que los valores a los cuales converge finalmente el algoritmo, son cercanos a los teóricos:

Tabla 5. Parámetros y resultados de simulación para solución de la función *De Jong's* en 4D.

<i>r</i>	0.95
<i>θ</i>	$\pi/(10^{27})$
<i>Kmax</i>	40000
<i>m</i>	16
<i>ε</i>	$10^{-30}$
<i>lix1,lsx1,nx1</i>	-5.12, 5.12, 1
<i>lix2,lsx2,nx2</i>	-5.12, 5.12, 1
<i>lix3,lsx3,nx3</i>	-5.12, 5.12, 1
<i>lix4,lsx4,nx4</i>	-5.12, 5.12, 1
<i>x1</i>	$4.680337579261291 \times 10^{-16}$
<i>x2</i>	$4.680337579261291 \times 10^{-16}$
<i>x3</i>	$4.680337579261291 \times 10^{-16}$
<i>x4</i>	$4.680337579261291 \times 10^{-16}$
<i>Error</i>	$8.762223942338177 \times 10^{-31}$
<i>k</i>	721
<i>tejec</i>	1.390463 s

(Fuente: autor)

## 5.2 Función *Rosenbrock's Valley*

Para esta función, la convergencia al óptimo global es difícil porque existe una región extremadamente plana alrededor de él [15], razón por la cual, se constituye en una muy buena función de prueba. La ecuación general para esta función es [14]:

$$f(x) = \sum_{i=1}^{N-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2] \quad (53)$$

de manera que solo está definida para dos o más variables. Esta función posee un mínimo global en  $x_i = 1$ ,  $i = 1, 2, \dots, n$ ; y su valor mínimo es de 0. La región de búsqueda estándar para los algoritmos de optimización es  $-2.048 \leq x_i \leq 2.048$ .

### ***Función Rosenbrock's Valley para dos variables***

La ecuación a la cual se le halla su valor mínimo con el algoritmo es:

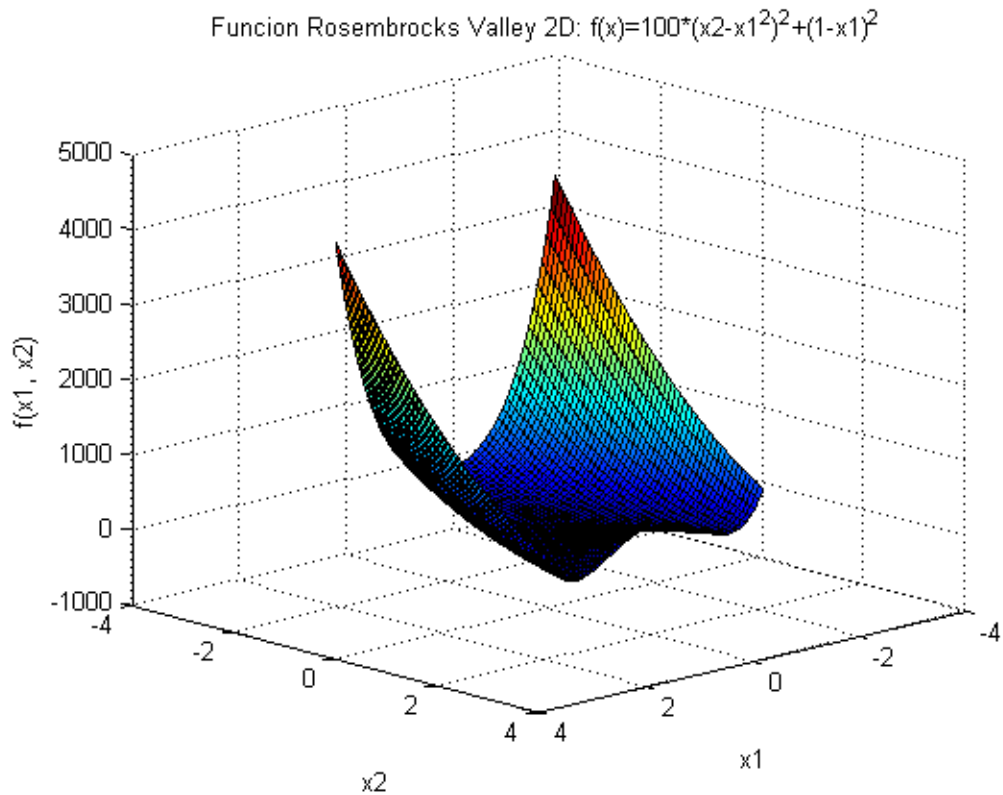
$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \quad (54)$$

Cuyo valor mínimo teóricamente está en:

$$x_1 = x_2 = 1 \quad (55)$$

La gráfica de esta función se presenta en la Figura 18.

Figura 18. Gráfica de la función *Rosenbrock's Valley* en 2D.



La Tabla 6 presenta los parámetros utilizados y los resultados obtenidos con el algoritmo, al contar con una región factible conformada por 9 puntos. En ella se reportan resultados cercanos a la solución teórica señalada en la ecuación anterior.

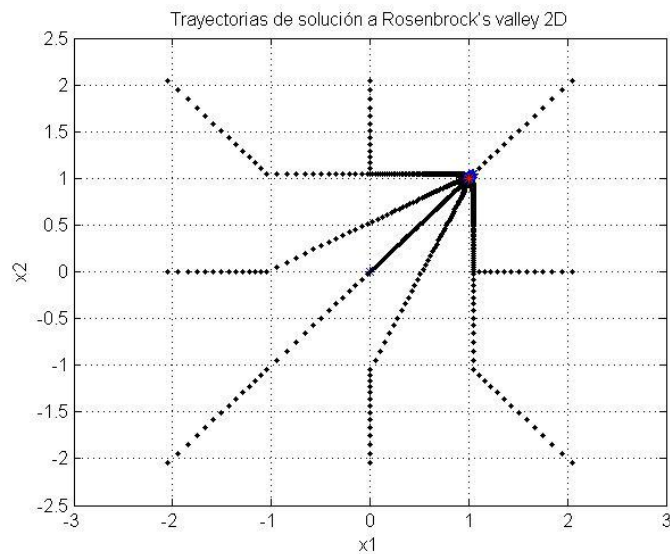
Tabla 6. Parámetros y resultados de simulación para solución de la función *Rosenbrock's Valley* en 2D.

$r$	0.95
$\theta$	$\pi/10^{10}$
$kmax$	20000
$m$	9
$\varepsilon$	$10^{-10}$
$lix1,lsx1,nx1$	-2.048, 2.048, 2.048
$lix2,lsx2,nx2$	-2.048, 2.048, 2.048
$x1$	0.999998465692026
$x2$	0.999995965806845
<b>Error</b>	$9.558848990196967 \times 10^{-11}$
$k$	272
$tejec$	0.374881 s

(Fuente: autor)

La Figura 19 muestra la trayectoria de convergencia hacia la solución reportada en la Tabla 6, y representada por una marca de color magenta.

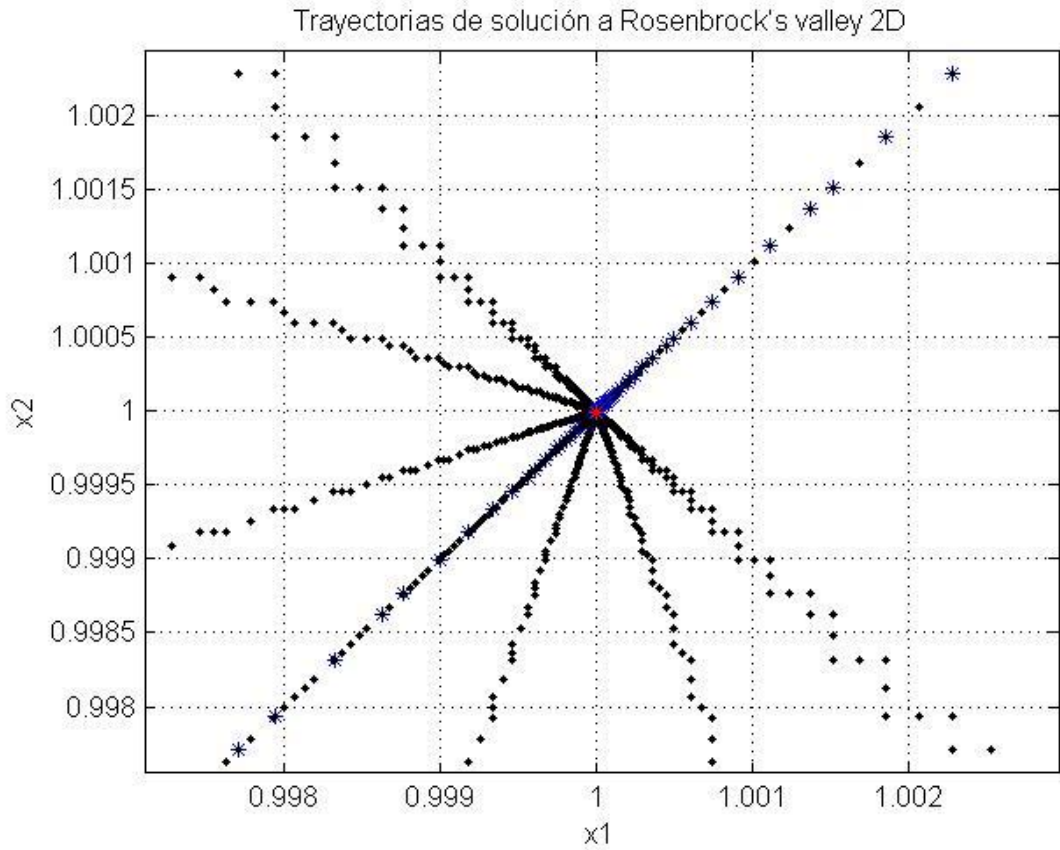
Figura 19. Convergencia para la solución de la ecuación de *Rosenbrock's Valley* en 2D.



(Fuente: autor)

La Figura 20 presenta una ampliación de la gráfica anterior.

**Figura 20. Ampliación de la gráfica mostrada en la Figura 19.**



(Fuente: autor)

### ***Función Rosenbrock's Valley para tres variables***

La ecuación (44) permite obtener la expresión para este caso:

$$f(x_1, x_2, x_3) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 100(x_3 - x_2^2)^2 + (1 - x_2)^2 \quad (56)$$

Cuyo valor mínimo se encuentra en:

$$x_1 = x_2 = x_3 = 1 \quad (57)$$

La Tabla 7 reporta los valores de solución obtenidos con el algoritmo, y se evidencia buena precisión en ellos, con respecto a los presentados en la ecuación anterior.

Tabla 7. Parámetros y resultados de simulación para solución de la función *Rosenbrock's Valley* en 3D.

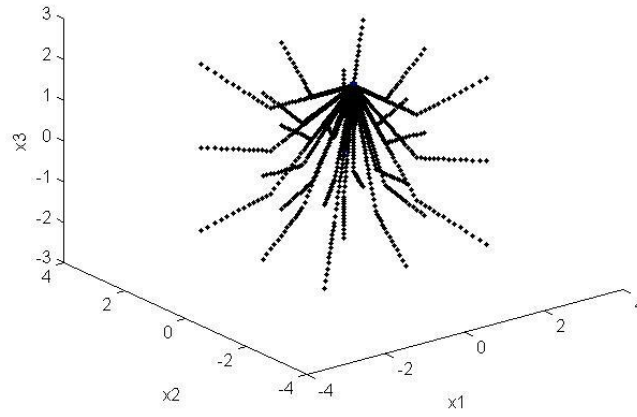
<i>r</i>	0.95
<i>θ</i>	pi/10 <sup>10</sup>
<i>kmax</i>	20000
<i>m</i>	27
<i>ε</i>	10 <sup>-6</sup>
<i>lix1,lsx1,nx1</i>	-2.048, 2.048, 2.048
<i>lix2,lsx2,nx2</i>	-2.048, 2.048, 2.048
<i>lix3,lsx3,nx3</i>	-2.048, 2.048, 2.048
<i>x1</i>	1.000069925245024
<i>x2</i>	1.000069928610191
<i>x3</i>	1.000069931975358
<i>Error</i>	9.877772368209879x10 <sup>-7</sup>
<i>k</i>	188
<i>tejec</i>	0.562458 s

(Fuente: autor)

Las trayectorias de convergencia hacia el valor óptimo final reportado en la Tabla 7, son presentados a continuación, en la Figura 21.

**Figura 21. Convergencia para la solución de la ecuación de *Rosenbrock's Valley* en 3D.**

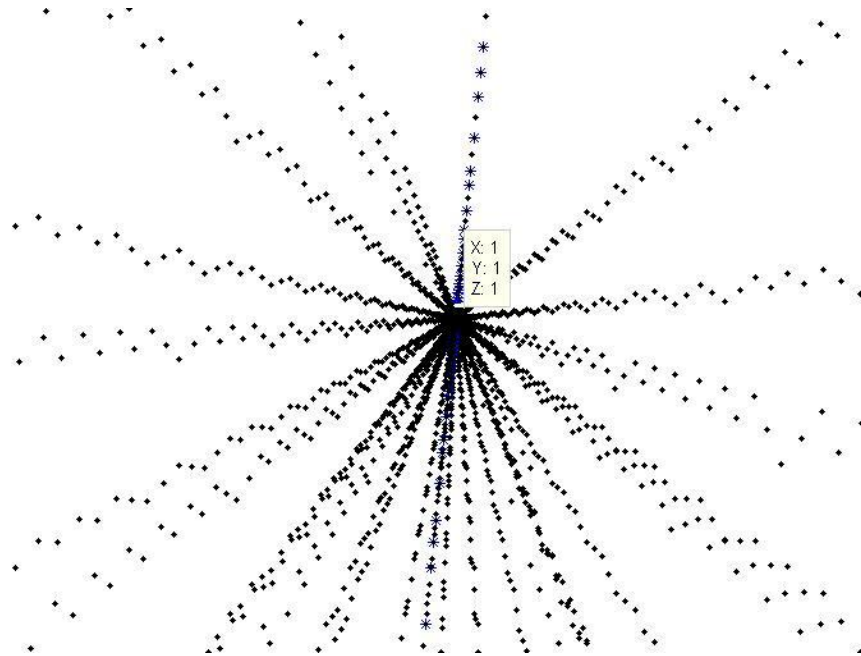
Trayectorias de solución a Rosenbrock's valley 3D



*(Fuente: autor)*

La ampliación de la gráfica anterior, con la etiqueta generada sobre el valor óptimo final, se muestra en la Figura 22.

**Figura 22. Ampliación de la Figura 17.**



*(Fuente: autor)*

### ***Función Rosenbrock's Valley para cuatro variables***

De la ecuación (53) se obtiene la expresión para este caso:

$$\begin{aligned} f(x_1, x_2, x_3, x_4) \\ = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 100(x_3 - x_2^2)^2 + (1 - x_2)^2 \\ + 100(x_4 - x_3^2)^2 + (1 - x_3)^2 \end{aligned} \quad (58)$$

Cuya valor mínimo se encuentra en:

$$x_1 = x_2 = x_3 = x_4 = 1 \quad (59)$$

La Tabla 8 reporta que efectivamente se encuentra una solución bastante precisa con los resultados consignados en la ecuación anterior.

**Tabla 8. Parámetros y resultados de simulación para solución de la función *Rosenbrock's Valley* en 4D.**

<b><i>r</i></b>	0.95
<b><i>θ</i></b>	pi/10 <sup>10</sup>
<b><i>kmax</i></b>	20000
<b><i>m</i></b>	81
<b><i>ε</i></b>	10 <sup>-18</sup>
<b><i>lix1,lsx1,nx1</i></b>	-2.048, 2.048, 2
<b><i>lix2,lsx2,nx2</i></b>	-2.048, 2.048, 2
<b><i>lix3,lsx3,nx3</i></b>	-2.048, 2.048, 2
<b><i>lix4,lsx4,n4</i></b>	-2.048, 2.048, 2
<b><i>x1</i></b>	1.000000001643922
<b><i>x2</i></b>	1.000000003296109
<b><i>x3</i></b>	1.000000006612883
<b><i>x4</i></b>	1.000000013269192

<b>Error</b>	5.753515430748019x10 <sup>-17</sup>
<b>k</b>	20000
<b>tejec</b>	167.667964 s

(Fuente: autor)

### 5.3 Función Rastrigin

Se trata de una función multimodal, obtenida a partir de añadir una componente armónica a la función De Jong's, y su ecuación general es:

$$f(x) = 10n + \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i)] \quad (60)$$

Posee un mínimo en  $x_i = 0$ ,  $i = 1, 2, \dots, n$ ; y su valor es de 0. La región de búsqueda para los algoritmos de optimización es  $-5.12 \leq x_i \leq 5.12$ .

#### **Función Rastrigin para una variable**

De la ecuación (60) se obtiene la expresión para este caso:

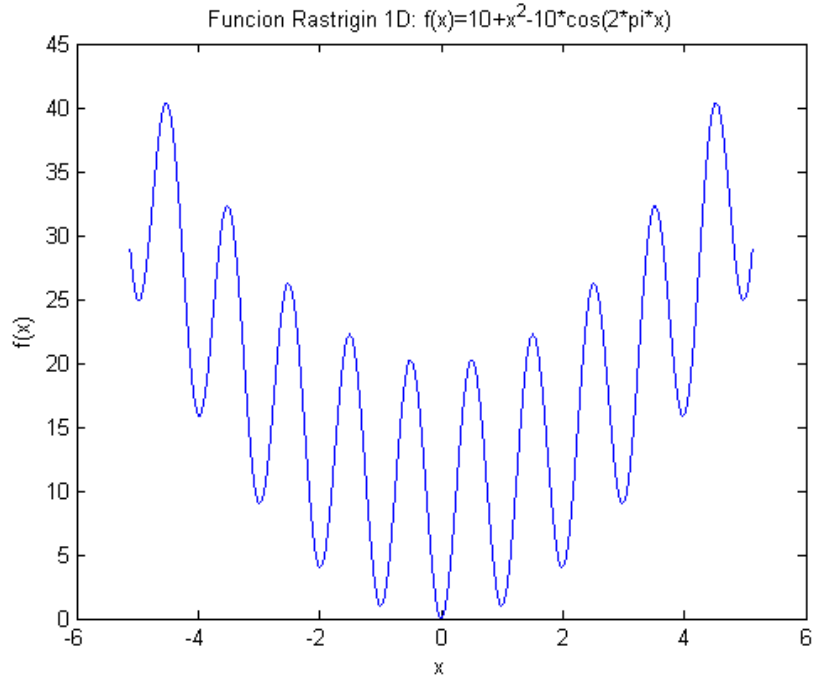
$$f(x_1) = 10 + x_1^2 - 10\cos(2\pi x_1) \quad (61)$$

Cuyo mínimo se encuentra en:

$$x_1 = 0 \quad (62)$$

La gráfica de esta función se presenta en la Figura 23.

Figura 23. Gráfica de la función *Rastrigin* 1D.



(Fuente: autor)

La Tabla 9 a continuación, reporta gran precisión en el resultado del algoritmo para la solución de esta ecuación en particular, de modo que para *Matlab*, el error es cero (menor a  $10^{-324}$  con las características de cómputo utilizadas).

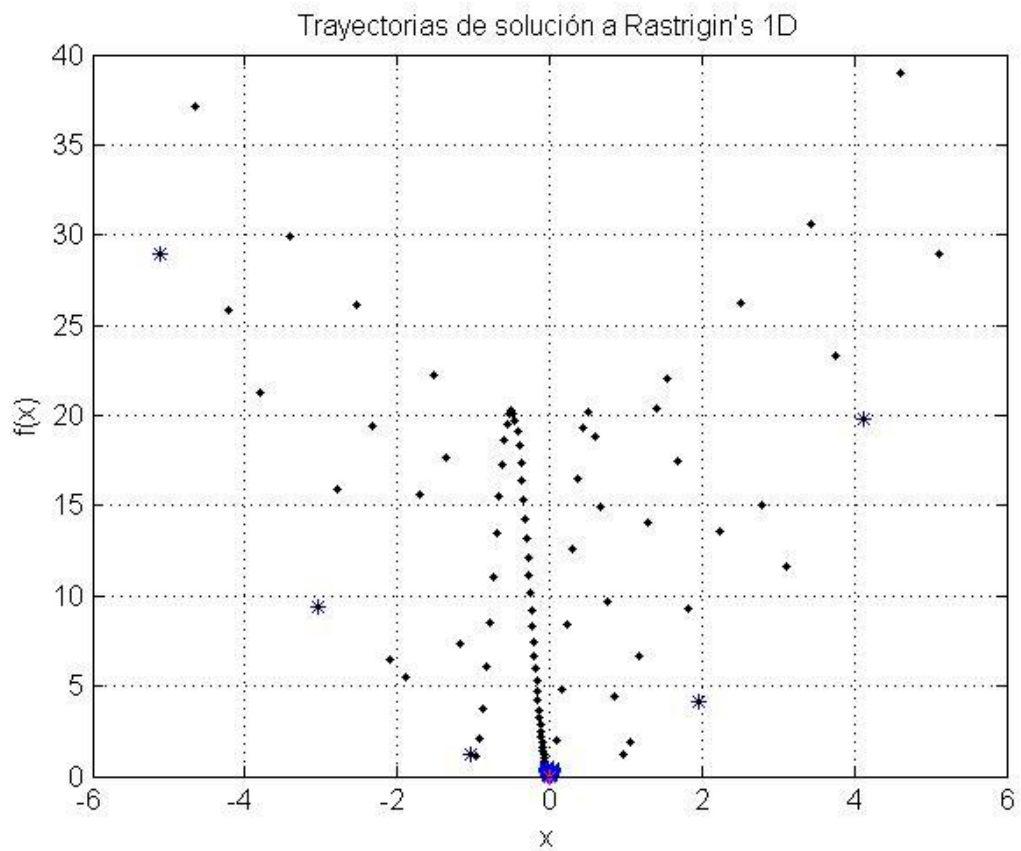
Tabla 9. Parámetros y resultados de simulación para solución de la función *Rastrigin* en 1D.

$r$	0.95
$\theta$	$\pi/10^{10}$
$kmax$	20000
$m$	3
$\varepsilon$	$10^{-15}$
$lix1,lsx1,nx1$	-5.12, 5.12 1
$x_1$	$1.468435978251267 \times 10^{-9}$
<b>Error</b>	0
$k$	420
$tejec$	0.258354 s

(Fuente: autor)

La gráfica presentada en la Figura 24, muestra la trayectoria de convergencia hacia el valor consignado en la tabla anterior, el cual está presentado en color magenta.

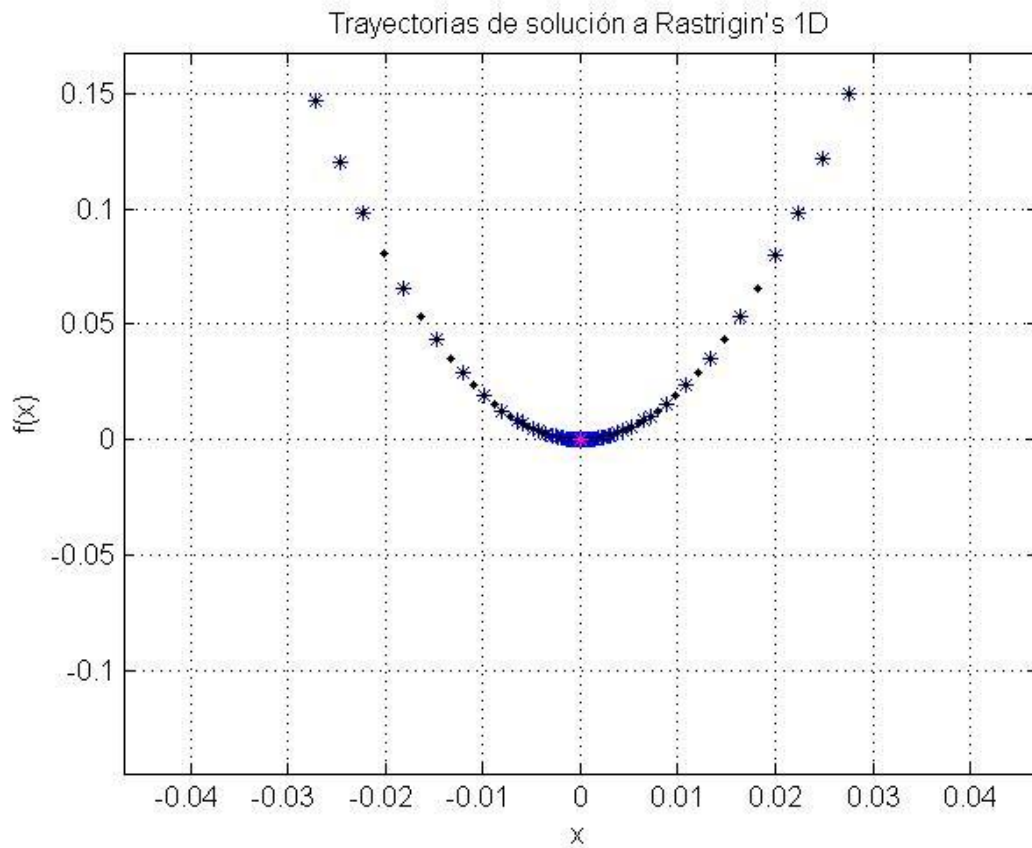
Figura 24. Convergencia hacia la solución de la raíz de la función *Rastrigin* en 1D.



(Fuente: autor)

Realizando una ampliación sobre la figura anterior, se obtiene la Figura 25.

Figura 25. Ampliación de la gráfica de la Figura 20.



(Fuente: autor)

### ***Función Rastrigin para dos variables***

De la ecuación (60), la expresión obtenida para este caso es:

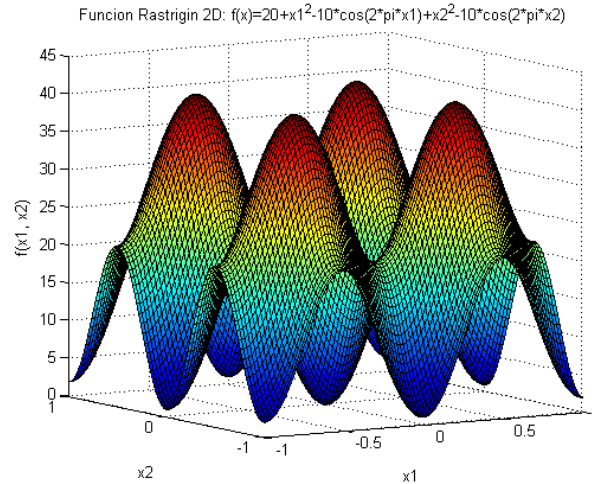
$$f(x_1, x_2) = 20 + x_1^2 - 10\cos(2\pi x_1) + x_2^2 - 10\cos(2\pi x_2) \quad (63)$$

Cuyo valor mínimo teórico se encuentra en:

$$x_1 = x_2 = 0 \quad (64)$$

La gráfica de esta función se presenta en la Figura 26.

Figura 26. Gráfica de la función *Rastrigin* 2D.



(Fuente: autor)

La Tabla 10 presenta los parámetros utilizados y los resultados obtenidos al contar con una región factible conformada con 4 puntos.

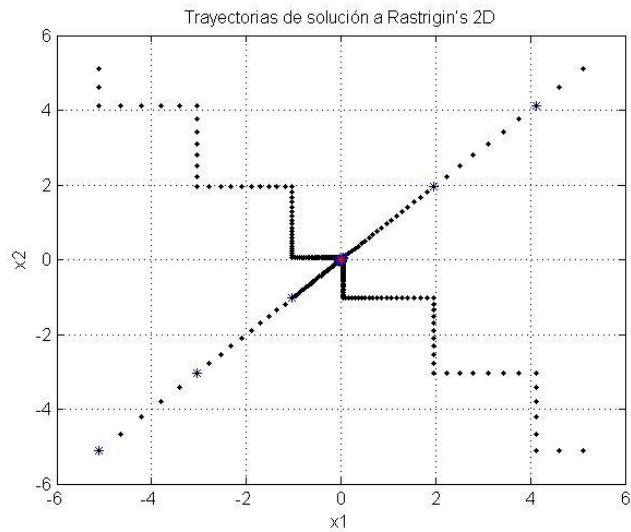
Tabla 10. Parámetros y resultados de simulación para la solución a la raíz de la función *Rastrigin* en 2D.

<i>r</i>	0.95
<i>θ</i>	$\pi/10^{10}$
<i>kmax</i>	20000
<i>m</i>	4
<i>ε</i>	$10^{-15}$
<i>lix1,lsx1,nx1</i>	-5.12, 5.12, 1
<i>lix2,lsx2,nx2</i>	-5.12, 5.12, 1
<i>x1</i>	$1.464684535507087 \times 10^{-9}$
<i>x2</i>	$1.470770761730306 \times 10^{-9}$
<i>Error</i>	0
<i>k</i>	420
<i>tejec</i>	0.361396 s

(Fuente: autor)

La gráfica que se obtiene para las trayectorias de convergencia hacia el valor óptimo final (*marca de color magenta*), se presenta en la Figura 27.

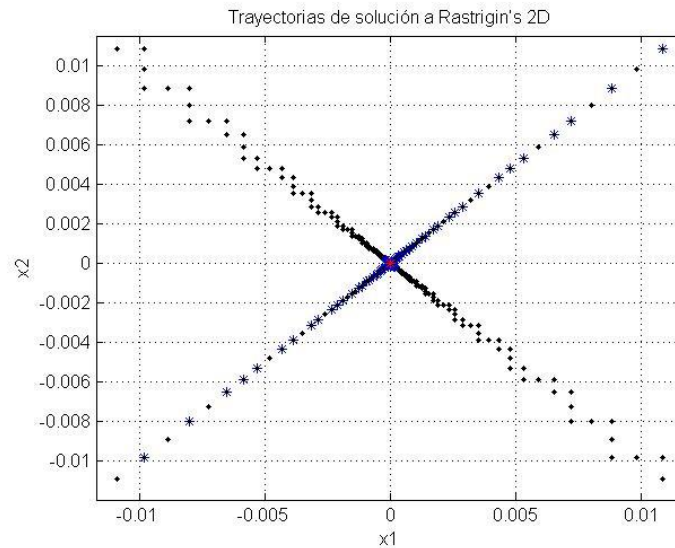
**Figura 27. Convergencia hacia la solución de la raíz de la función *Rastrigin* en 2D.**



(Fuente: autor)

Realizando una ampliación sobre la gráfica anterior, se obtiene la Figura 28.

**Figura 28. Ampliación sobre la gráfica de la Figura 23.**



(Fuente: autor)

### ***Función de Rastrigin para tres variables***

De la ecuación (60), la expresión obtenida para este caso es:

$$f(x_1, x_2, x_3) = 30 + x_1^2 - 10\cos(2\pi x_1) + x_2^2 - 10\cos(2\pi x_2) + x_3^2 - 10\cos(2\pi x_3) \quad (65)$$

Para la cual existe un valor mínimo en:

$$x_1 = x_2 = x_3 = 0 \quad (66)$$

La Tabla 11 reporta la convergencia hacia un resultado bastante preciso, comparado con la solución presentada en la ecuación anterior.

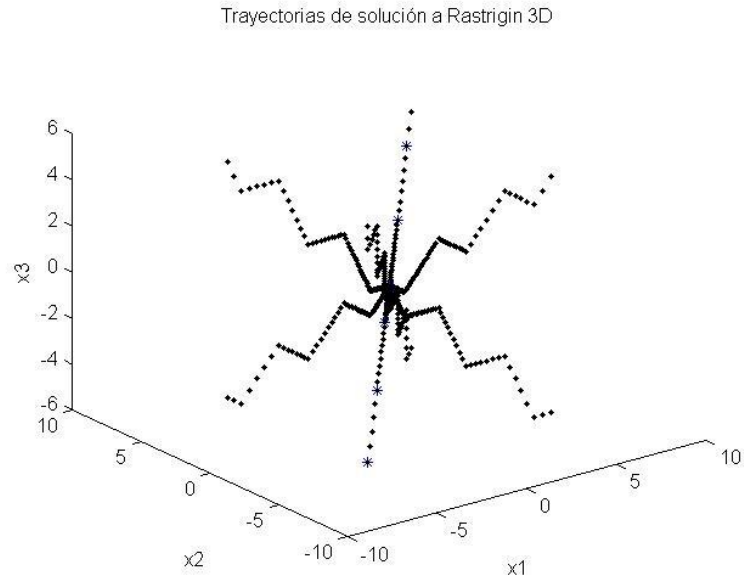
**Tabla 11. Parámetros y resultados de simulación para solución de la función *Rastrigin* en 3D.**

<i>r</i>	0.95
<i>θ</i>	pi/10 <sup>10</sup>
<i>kmax</i>	20000
<i>m</i>	8
<i>ε</i>	10 <sup>-15</sup>
<i>lix1, lsx1, nx1</i>	-5.12, 5.12, 1
<i>lix2, lsx2, nx2</i>	-5.12, 5.12, 1
<i>lix3, lsx3, nx3</i>	-5.12, 5.12, 1
<i>x1</i>	-3.616630722776637x10 <sup>-9</sup>
<i>x2</i>	1.464063462724009x10 <sup>-9</sup>
<i>x3</i>	1.470770689319046 x10 <sup>-9</sup>
<i>Error</i>	0
<i>k</i>	420
<i>tejec</i>	0.468945 s

(Fuente: autor)

La gráfica obtenida para la convergencia hacia el valor óptimo final, es presentada en la Figura 29.

**Figura 29. Convergencia hacia la solución de la raíz de la función *Rastrigin* en 3D.**



(Fuente: autor)

### ***Función de Rastrigin para cuatro variables***

De la ecuación (60), la expresión obtenida para este caso es:

$$\begin{aligned}
 f(x_1, x_2, x_3, x_4) & \\
 &= 40 + x_1^2 - 10\cos(2\pi x_1) + x_2^2 - 10\cos(2\pi x_2) + x_3^2 \\
 &\quad - 10\cos(2\pi x_3) + x_4^2 - 10\cos(2\pi x_4)
 \end{aligned}
 \tag{67}$$

Cuyo valor mínimo se encuentra en:

$$x_1 = x_2 = x_3 = x_4 = 0
 \tag{68}$$

La Tabla 12 muestra que el algoritmo genera una solución bastante precisa con los resultados consignados en la ecuación anterior.

Tabla 12. Parámetros y resultados de simulación para solución de la función *Rastrigin* en 4D.

<i>r</i>	0.95
<i>θ</i>	$\pi/10^{10}$
<i>kmax</i>	20000
<i>m</i>	81
<i>ε</i>	$10^{-15}$
<i>lix1,lsx1,nx1</i>	-5.12, 5.12, 1
<i>lix2,lsx2,nx2</i>	-5.12, 5.12, 1
<i>lix3,lsx3,nx3</i>	-5.12, 5.12, 1
<i>lix4,lsx4,nx4</i>	-5.12, 5.12, 1
<i>x1</i>	$-3.682206729992865 \times 10^{-9}$
<i>x3</i>	$-3.619700099514040 \times 10^{-9}$
<i>x3</i>	$1.470775726892059 \times 10^{-9}$
<i>x4</i>	$1.465887851531423 \times 10^{-9}$
<i>Error</i>	0
<i>k</i>	420
<i>tejec</i>	0.851743 s

(Fuente: autor)

#### 5.4 Función *Schwefel*

Se trata de una función que para resolver por iteración, es engañosa en la obtención de su mínimo global, debido a que posee mínimos locales cercanos a este. Su ecuación general es [14]:

$$f(x) = \sum_{i=1}^n -x_i \text{sen}(\sqrt{|x_i|}) \quad (69)$$

Esta función posee su valor mínimo global en  $x_i = 420.9687464$ ,  $i = 1, 2, \dots, n$ ; y es  $-418.982887n$ . La región estándar de búsqueda para los algoritmos de optimización es  $-500 \leq x_i \leq 500$ .

### ***Función schwefel para una variable***

De la ecuación (69), la expresión obtenida para este caso es:

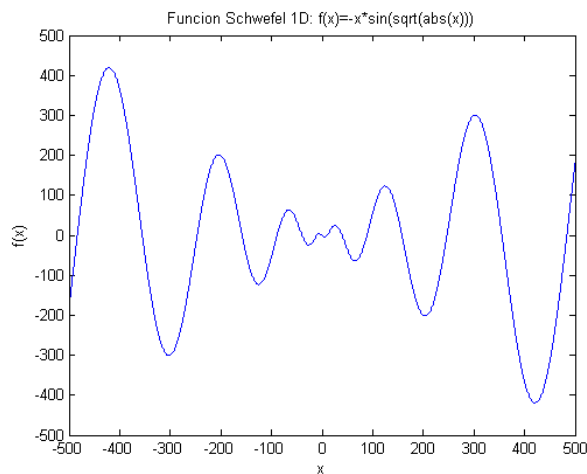
$$f(x_1) = -x_1 \text{sen}(\sqrt{|x_1|}) \quad (70)$$

Cuyo valor mínimo se encuentra en:

$$x_1 = 420.9687464 \quad (71)$$

La gráfica de esta función se presenta en la Figura 30.

**Figura 30. Gráfica de la función Schwefel en 1D.**



(Fuente: autor)

La Tabla 13 reporta la precisión con que el algoritmo se acerca a la respuesta de la ecuación anterior.

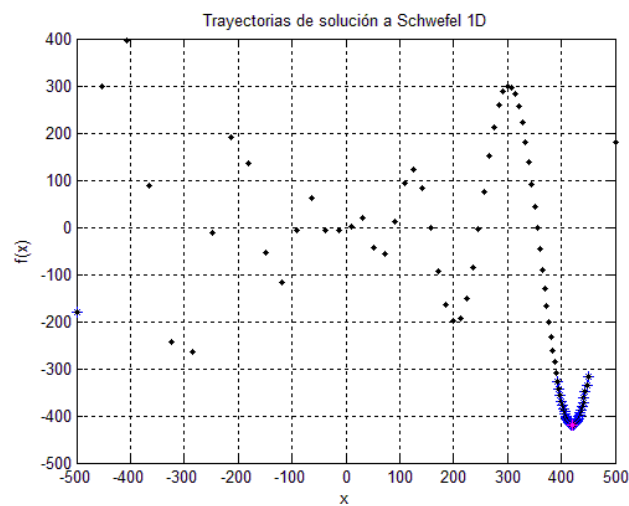
Tabla 13. Parámetros y resultados de simulación para solución de la función *Schwefel* en 1D.

<i>r</i>	0.95
<i>θ</i>	$\pi/10^{10}$
<i>kmax</i>	20000
<i>ε</i>	$10^{-10}$
<i>m</i>	2
<i>lix1,lsx1,nx1</i>	-500, 500, 1
<i>x1</i>	$4.209687204533291 \times 10^2$
<i>Error</i>	$8.401457307627425 \times 10^{-11}$
<i>k</i>	328
<i>tejec</i>	0.269719 s

(Fuente: autor)

La trayectoria de convergencia hacia el valor óptimo final que aparece en color magenta, se muestra en la Figura 31.

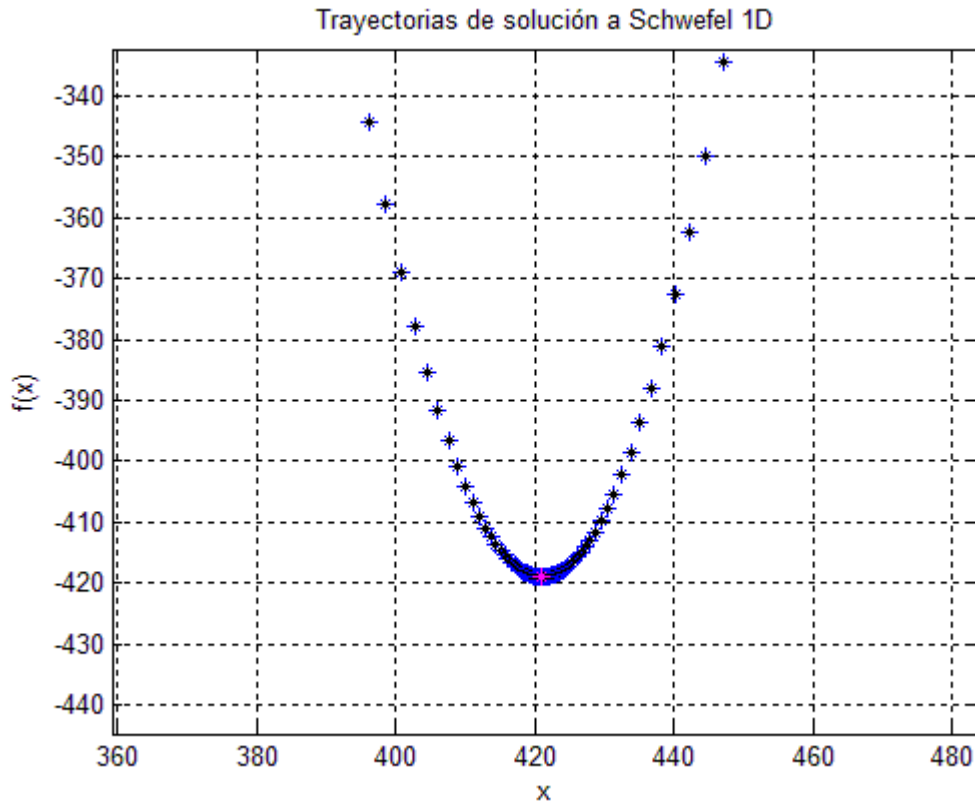
Figura 31. Convergencia a la solución óptima de la función *Schwefel* 1D.



(Fuente: autor)

Una ampliación de la figura anterior es presentada en la Figura 32.

Figura 32. Ampliación de la gráfica de la *Figura 27*.



(Fuente: autor)

### ***Función Schwefel para dos variables***

Se propone utilizar una función basada en *Schwefel*, y obtenida a partir de la ecuación (70):

$$f(x_1, x_2) = (-x_1^2 \text{sen}(\sqrt{x_1}))^2 + (-x_2^2 \text{sen}(\sqrt{x_2}))^2 \quad (72)$$

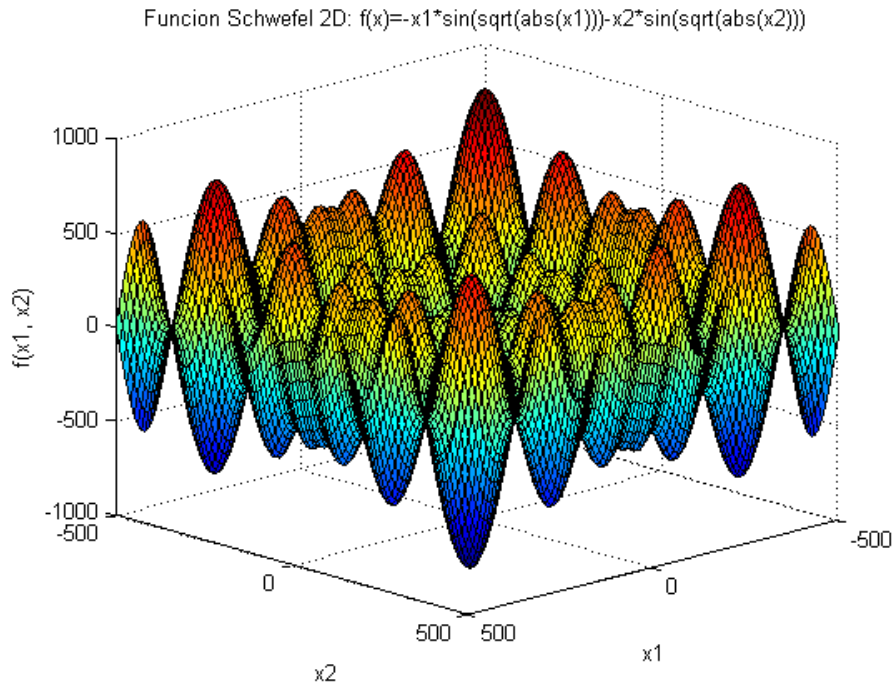
Cuyo valor mínimo se encuentra en:

$$x_1 = x_2 = 420.9687464$$

(73)

La gráfica de esta función se presenta en Figura 33.

Figura 33. Gráfica de la función *Schwefel* en 2D.



(Fuente: autor)

La Tabla 14 presenta los parámetros utilizados y los resultados obtenidos al contar con una región factible conformada con 4 puntos, evidenciándose la precisión con que el algoritmo se acerca al resultado de la ecuación anterior.

Tabla 14. Parámetros y resultados de simulación para función *Schwefel* en 2D.

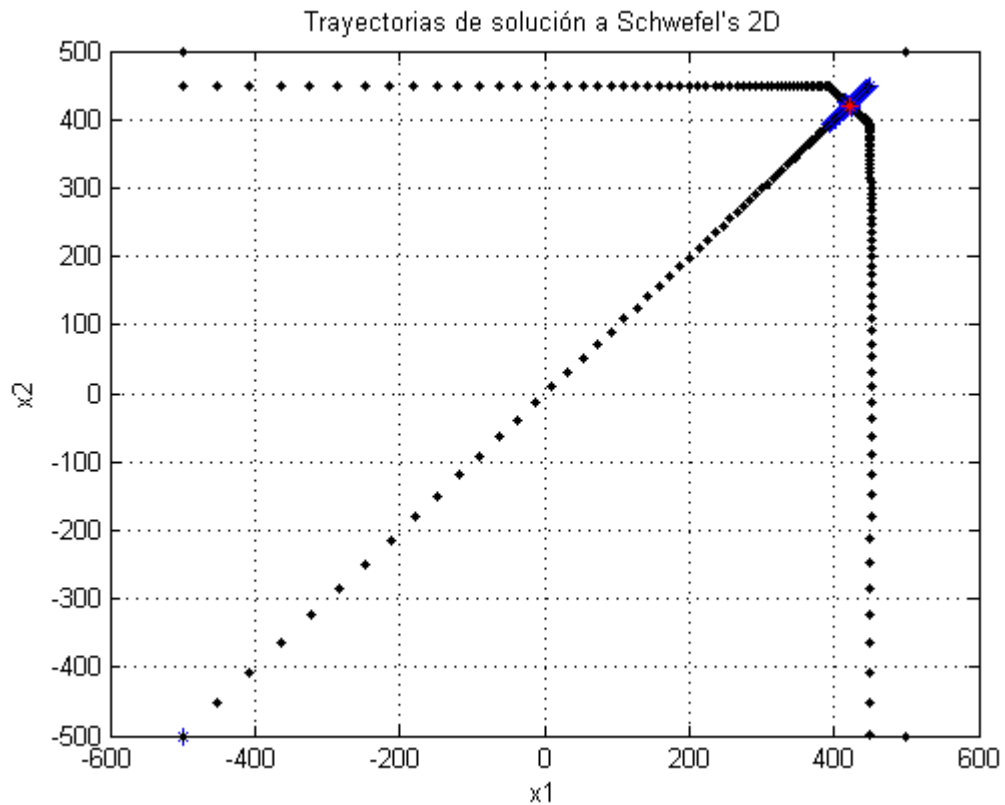
$r$	0.95
$\theta$	$\text{Pi}/10^{10}$
$kmax$	20000
$\varepsilon$	$10^{-10}$
$m$	4
$lix1,lsx1,nx1$	-500, 500, 1

<i>lix2,lsx2,nx2</i>	-500, 500, 1
<i>lix3,lsx3,nx3</i>	-500, 500, 1
<i>x1</i>	$4.209687664181200 \times 10^{-2}$
<i>x2</i>	$4.209687272668258 \times 10^{-2}$
<i>Error</i>	$9.526957001071423 \times 10^{-11}$
<i>k</i>	334
<i>tejec</i>	0.402411 s

(Fuente: autor)

A continuación, en la Figura 34 se presenta la gráfica de la convergencia hacia la solución con el valor óptimo final.

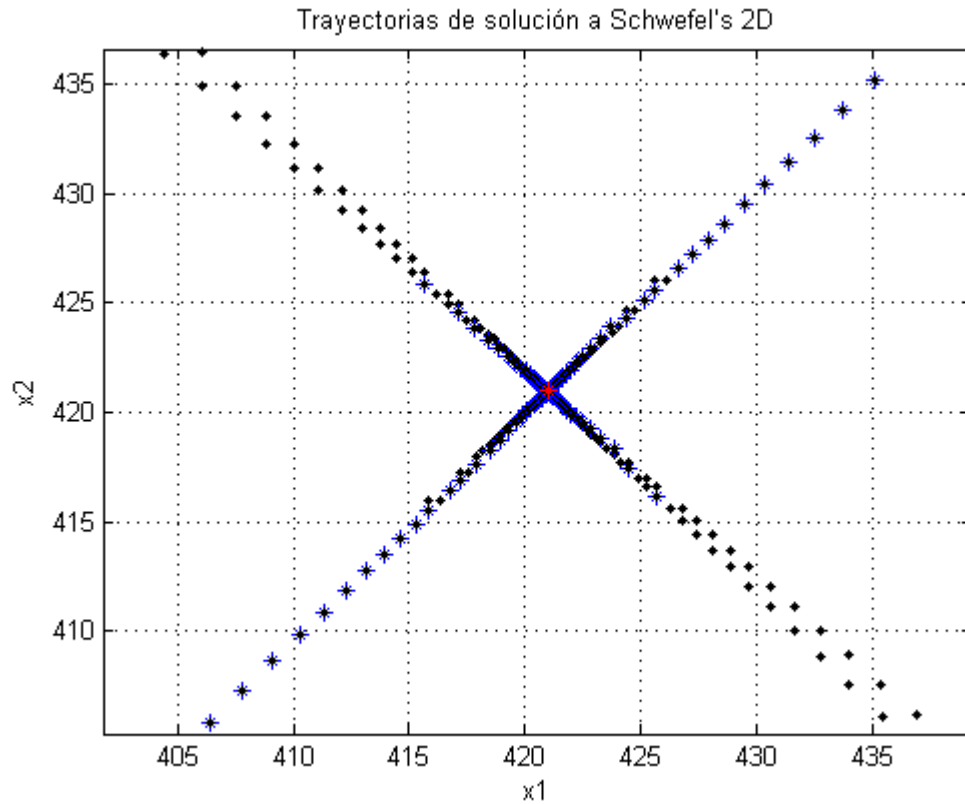
Figura 34. Convergencia a la solución óptima en la función *Schwefel* 2D.



(Fuente: autor)

Realizando una ampliación sobre la Figura 34 se obtiene la siguiente gráfica presentada en la Figura 35.

Figura 35. Ampliación sobre la gráfica de la *Figura 30*.



(Fuente: autor)

### ***Función Schwefel para tres variables***

La función que se propone utilizar, en base a la ecuación (70) es:

$$f(x_1, x_2, x_3) = \left(-x_1 \text{sen}(\sqrt{|x_1|})\right)^2 - \left(x_2 \text{sen}(\sqrt{|x_2|})\right)^2 - \left(x_3 \text{sen}(\sqrt{|x_3|})\right)^2 \quad (74)$$

Cuyo valor mínimo se encuentra en:

$$x_1 = x_2 = x_3 = 420.9687464 \quad (75)$$

La Tabla 15 evidencia la precisión con la que el algoritmo se acerca a los valores de la solución consignada en la ecuación anterior.

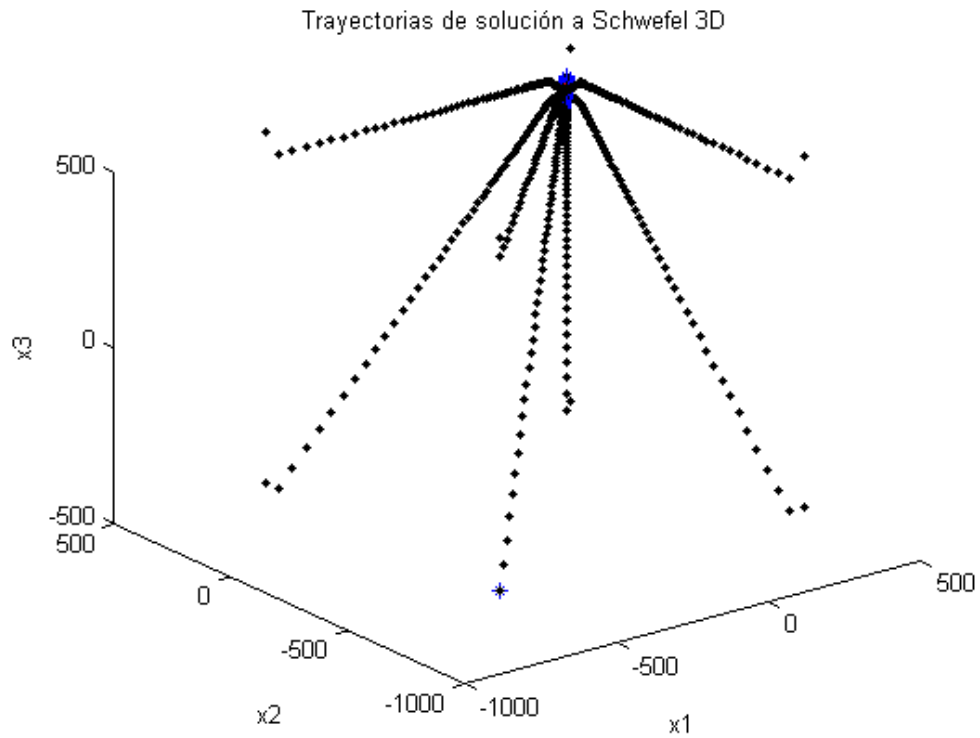
Tabla 15. Parámetros y resultados de simulación para solución de la función *Schwefel* en 3D.

<i>r</i>	0.95
<i>θ</i>	pi/10 <sup>10</sup>
<i>Kmax</i>	20000
<i>ε</i>	10 <sup>-10</sup>
<i>m</i>	8
<i>lix1,lsx1,nx1</i>	-500, 500, 1
<i>lix2,lsx2,nx2</i>	-500, 500, 1
<i>lix3,lsx3,nx3</i>	-500, 500, 1
<i>x1</i>	4.209687307871751x10 <sup>2</sup>
<i>x2</i>	4.209687619152124x10 <sup>2</sup>
<i>x3</i>	4.209687619025881x10 <sup>2</sup>
<i>Error</i>	8.958522812463343x10 <sup>-11</sup>
<i>k</i>	338
<i>tejec</i>	0.404468 s

(Fuente: autor)

La gráfica de convergencia hacia el valor óptimo final, se presenta a continuación, en la Figura 36.

Figura 36. Convergencia al valor óptimo final mínimo de Schwefel en 3D.



(Fuente: autor)

### ***Función Schwefel para cuatro variables***

La función basada en la ecuación (70) es:

$$\begin{aligned}
 f(x_1, x_2, x_3, x_4) &= \left(-x_1 \text{sen}(\sqrt{|x_1|})\right)^2 - \left(x_2 \text{sen}(\sqrt{|x_2|})\right)^2 - \left(x_3 \text{sen}(\sqrt{|x_3|})\right)^2 \\
 &\quad - \left(x_4 \text{sen}(\sqrt{|x_4|})\right)^2
 \end{aligned} \tag{76}$$

Que posee el mínimo global en:

$$x_1 = x_2 = x_3 = x_4 = 420.9687464 \tag{77}$$

La Tabla 16 reporta la precisión con la cual el algoritmo se acerca a los valores consignados en la ecuación anterior.

**Tabla 16. Parámetros y resultados de simulación para solución de la función Schwefel en 4D.**

<i>r</i>	0.95
<i>θ</i>	$\pi/10^{10}$
<i>kmax</i>	20000
$\epsilon$	$10^{-10}$
<i>m</i>	16
<i>lix1, lxx1, nx1</i>	-500, 500, 1
<i>lix2, lxx2, nx2</i>	-500, 500, 1
<i>lix3, lxx3, nx3</i>	-500, 500, 1
<i>lix4, lxx4, nx4</i>	-500, 500, 1
<i>x1</i>	$4.209687603491294 \times 10^2$
<i>x2</i>	$4.209687603526427 \times 10^2$
<i>x3</i>	$4.209687603493705 \times 10^2$
<i>x4</i>	$4.209687603670436 \times 10^2$
<i>Error</i>	$9.640643838793039 \times 10^{-11}$
<i>k</i>	340
<i>tejec</i>	0.902834 s

(Fuente: autor)

### 5.5 Característica de convergencia a una sola solución cuando existe múltiples de ellas.

Es esta una propiedad del algoritmo, en la cual, cuando en una región factible, existen varias soluciones óptimas, el programa convergerá en definitiva, hacia solo una de ellas. Por ejemplo, si se quiere resolver la ecuación:

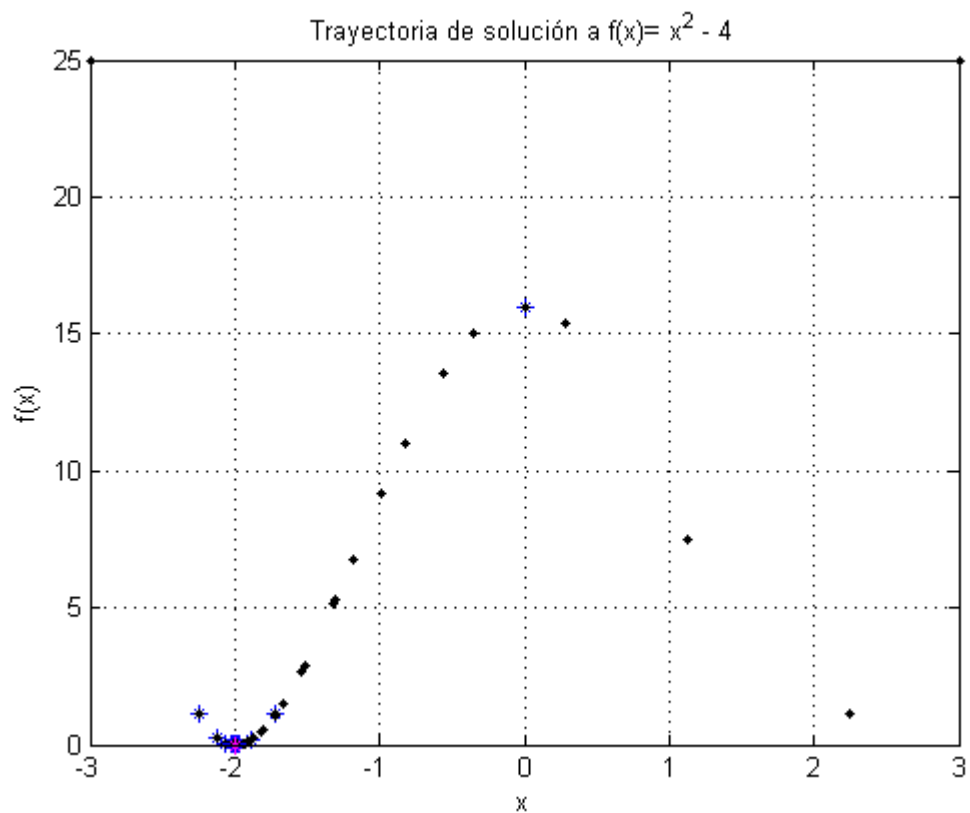
$$x^2 - 4 = 0 \tag{78}$$

La cual analíticamente tiene por solución:

$$x_1 = -2, \quad x_2 = 2 \tag{79}$$

cuya gráfica se presentó en la Figura 3, y su función objetivo está descrita en la ecuación (25). Usando la región factible  $[-3, 3]$ , se presenta en la Figura 37, la gráfica de convergencia hacia el valor óptimo final  $x_1 = -2$  obtenida con el algoritmo, el cual es uno de los dos valores señalados en la ecuación (79).

Figura 37. Convergencia hacia una de las soluciones presentes en una región factible.



(Fuente: autor)

La Tabla 17 evidencia con precisión el resultado mostrada en la Figura 37.

Tabla 17. Resultados y parámetros de la ejecución para resolver ecuación (78).

<i>r</i>	0.75
<i>θ</i>	$\pi/10^{40}$
<i>Kmax</i>	2000
<i>ε</i>	$10^{-29}$
<i>m</i>	3
<i>lix1,lsx1,nx1</i>	-3, 3, 2
<i>x1</i>	-2.0000000000000000
<i>Error</i>	$3.155443620884047 \times 10^{-30}$
<i>k</i>	125
<i>tejec</i>	0.223865 s

(Fuente: autor)

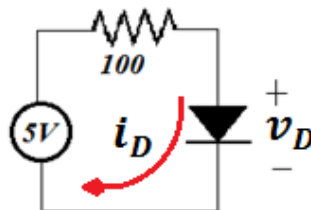
## 6. SOLUCIÓN DE LAS ECUACIONES QUE MODELAN ALGUNOS CIRCUITOS NO- LINEALES DE C.C.

Se muestra el cumplimiento del objetivo general del presente trabajo de investigación (ver página 21), así como también el cumplimiento del tercer objetivo específico; por medio de exponer el funcionamiento de la aplicación realizada en *Matlab*, en la solución de las ecuaciones que modelan algunos circuitos *no-lineales*, con elementos como diodos, transistores *BJT* y *MOSFET*.

### 6.1 Solución a ecuaciones no lineales de un circuito de corriente directa con un diodo (una variable)

Se verifica por medio de la solución de la ecuación que modela al circuito de corriente directa con un diodo de la Figura 38, el funcionamiento del algoritmo implementado para el caso de una variable.

Figura 38. Circuito de un solo diodo.



(Fuente: autor)

Usando *Ley de Voltajes de Kirchhoff* (LVK) para este circuito:

$$-5 + 100i_D + v_D = 0 \quad (80)$$

Para un diodo se cumple de la ecuación (26):

$$i_D = I_s(e^{\frac{v_D}{nV_T}} - 1) \quad (81)$$

Donde se han considerado los siguientes valores para sus parámetros:

$$n = 1$$

$$I_s = 10^{-12} \text{ A}$$

$$V_T = 25.85 \text{ mV}$$

Entonces, la ecuación en términos de una sola variable es:

$$f_1 = -5 + 100I_s(e^{\frac{v_D}{nV_T}} - 1) + v_D = 0 \quad (82)$$

Haciendo uso de la ecuación (23), la ecuación que se implementa en el algoritmo de la espiral de optimización es:

$$f(x_1) = \left( -5 + 100I_s \left( e^{\frac{x_1}{nV_T}} - 1 \right) + v_D \right)^2 \quad (83)$$

donde  $x_1 = v_D$ . Los resultados obtenidos y los parámetros usados en el programa se presentan en la Tabla 18.

**Tabla 18. Parámetros y resultados de simulación para la solución del modelo de un circuito *no-lineal* de c.c. de una malla.**

<i>Ejec</i>	<b>1</b>	<b>2</b>	<b>3*</b>
<i>r</i>	0.95	0.95	0.95
<i>θ</i>	pi/10 <sup>40</sup>	pi/10 <sup>40</sup>	pi/10 <sup>40</sup>
<i>kmax</i>	20000	20000	20000

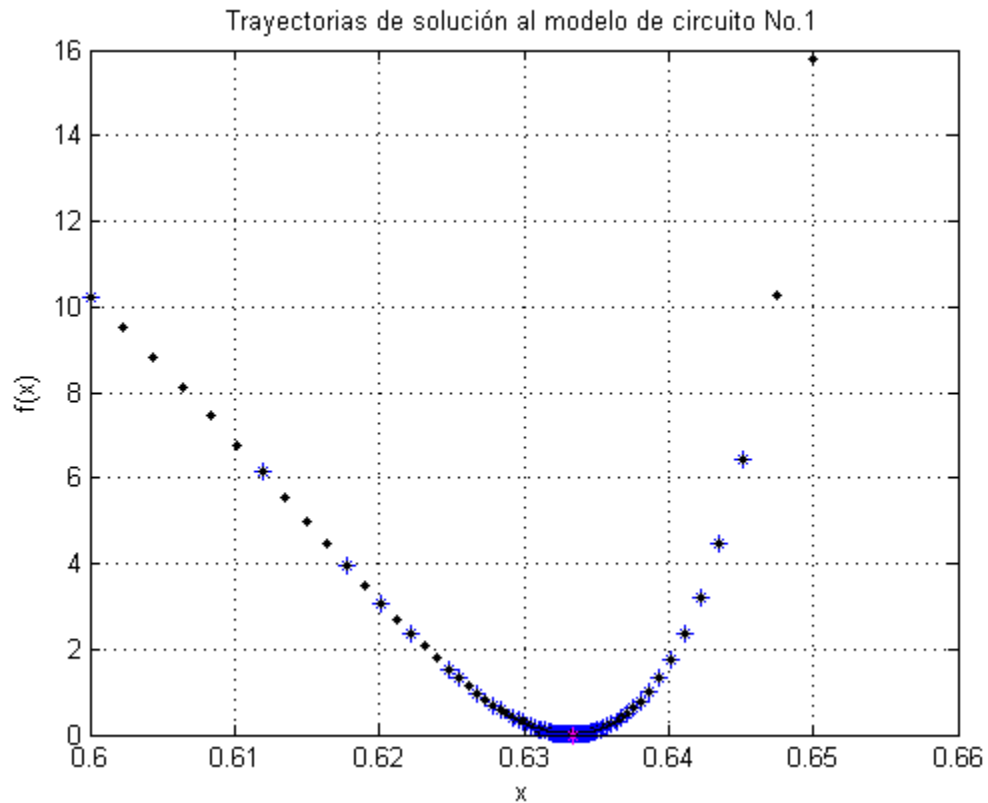
\* Parámetros almacenados previamente para la ejecución por defecto en el programa desarrollado con el presente trabajo de grado.

$\epsilon$	$10^{-20}$	$10^{-25}$	$10^{-25}$
$m$	2	2	2
$lix1,lsx1,nx1$	0.5, 0.8, 1	0.6, 0.7, 1	0.6, 0.65, 1
$x1$	0.633321216757110	0.633321216756519	0.633321216756523
$Error$	$9.9959164510 \times 10^{-21}$	$9.03894574572 \times 10^{-26}$	$9.532842930318 \times 10^{-26}$
$k$	513	605	592
$tejec$	0.321154 s	0.381762 s	0.366416 s

(Fuente: autor)

La gráfica de la Figura 39 muestra la convergencia hacia el valor óptimo final, el cual se encuentra con una marca roja.

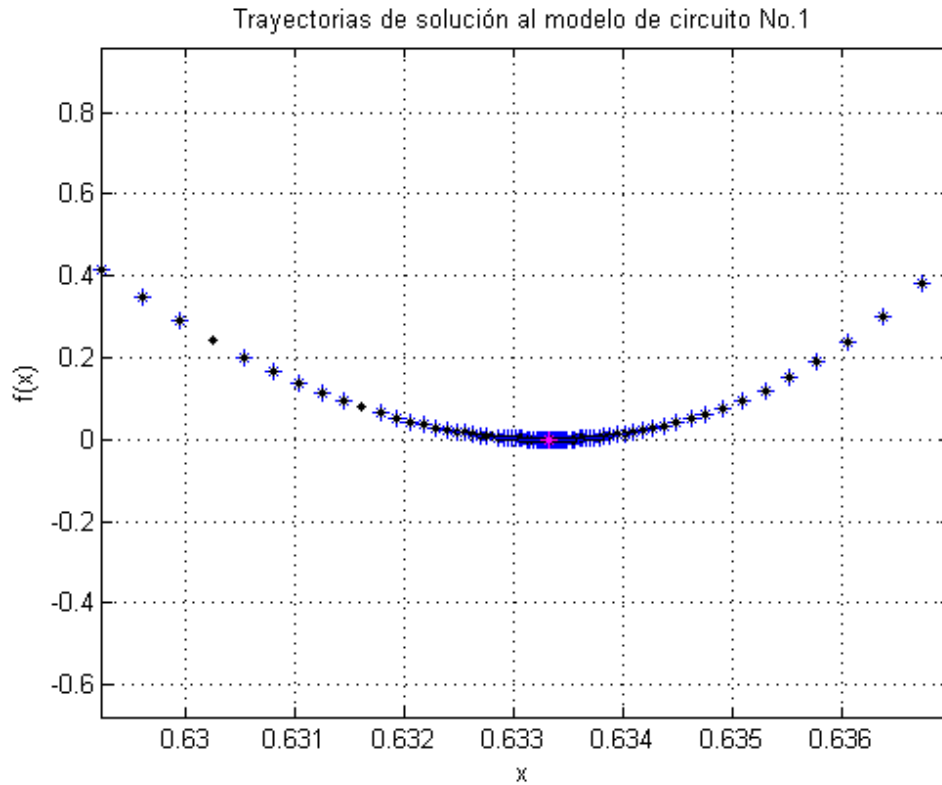
Figura 39. Convergencia para la solución del circuito de la Figura 34.



(Fuente: autor)

A continuación, en la Figura 40, se presenta una ampliación realizada sobre la anterior gráfica.

**Figura 40. Ampliación sobre la gráfica de la Figura 35.**



(Fuente: autor)

La solución de la ecuación al circuito de la Figura 38 realizada con el comando *fsolve* de *MATLAB* para un valor inicial  $x_{01}$  de la variable  $x_1$ , es presentada en la Tabla 19. El valor es muy similar al reportado en la ejecución número 2 de la Tabla 18, lo cual respalda los resultados obtenidos con el método para este caso.

**Tabla 19. Parámetros y resultados de la solución al circuito de un diodo usando el comando *fsolve* de *MATLAB*.**

$x_{01}$	0.5
$x_1$	0.633321216756522

(Fuente: autor)

Las demás variables se hallan por sustitución del valor obtenido con el algoritmo de la espiral:  $v_D = 0.633321216756519V$

$$i_D = I_s \left( e^{\frac{v_D}{nV_T}} - 1 \right) = 43.666787832431 \text{ mA} \quad (84)$$

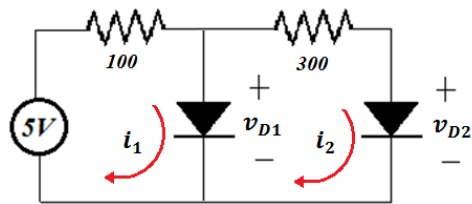
Y con ello, la caída de tensión en la resistencia es

$$v_R = 100i_D = 4.3666787832431 \text{ V} \quad (85)$$

## 6.2 Solución de ecuaciones no lineales de un circuito de corriente directa con dos diodos (dos variables)

Se verifica el funcionamiento del algoritmo implementado para dos variables con el método de la espiral de optimización, por medio de la solución de la ecuación que modela al circuito de corriente directa *no-lineal* de la Figura 41, el cual posee dos mallas.

Figura 41. Circuito de dos mallas con dos diodos.



(Fuente: autor)

Las ecuaciones que modelan a este circuito son, de acuerdo a LVK:

$$-5 + 100i_1 + v_{D1} = 0 \quad (86)$$

$$300i_2 + v_{D2} - v_{D1} = 0 \quad (87)$$

de modo que inicialmente, es éste un modelo con cuatro incógnitas. En un diodo se cumple que [2]:

$$i = I_s(e^{\frac{v_D}{nV_T}} - 1) \quad (88)$$

Donde se han considerado los siguientes valores para sus parámetros:

$$n = 1$$

$$I_s = 10^{-12} A$$

$$V_T = 25.85 mV$$

Por lo cual:

$$i_1 - i_2 = I_s(e^{\frac{v_{D1}}{nV_T}} - 1) \quad (89)$$

Y se llega a un sistema de dos ecuaciones y dos incógnitas:

$$f_1 = -5 + 100 \left[ I_s \left( e^{\frac{v_{D1}}{nV_T}} - 1 \right) + i_2 \right] + V_{D1} = 0 \quad (90)$$

$$f_2 = nV_T \ln \left( \frac{i_2}{I_s} + 1 \right) + 300i_2 - v_{D1} = 0 \quad (91)$$

Asumiendo que:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} v_{D1} \\ i_2 \end{bmatrix} \quad (92)$$

y haciendo uso de la ecuación (23), la función objetivo implementada en el algoritmo es:

$$f(x_1, x_2) = \left( -5 + 100 \left( I_s \left( e^{\left( \frac{x_1}{V_T} \right)} - 1 \right) + x_2 \right) + x_1 \right)^2 + \left( V_T \text{Ln} \left( \left( \frac{x_2}{I_s} \right) + 1 \right) + 300x_2 - x_1 \right)^2 \quad (93)$$

En total se cuenta con 6 puntos para los valores iniciales que corresponden a la región factible y los resultados obtenidos junto con los parámetros usados en el programa se presentan en la Tabla 20.

**Tabla 20. Parámetros y resultados de simulación para un circuito *no-lineal* de c.c. de dos mallas.**

<i>r</i>	0.65
<i>θ</i>	pi/20000
<i>Kmax</i>	40000
<i>m</i>	6
<i>ε</i>	10 <sup>-28</sup>
<i>lix1,lsx1,nx1</i>	0, 1, 2
<i>lix2,lsx2,nx2</i>	0, 0.5, 1
<i>x1</i>	0.633082996765566
<i>x2</i>	4.029438104116737x10 <sup>-4</sup>
<i>Error</i>	5.881944124554169x10 <sup>-29</sup>
<i>k</i>	87
<i>tejec</i>	0.272168 s

(Fuente: autor)

La gráfica de la Figura 42 a continuación, muestra las trayectorias de convergencia hacia el valor óptimo final, el cual se encuentra representado con color rojo.



Con los resultados obtenidos con el algoritmo de la espiral:  $v_{D1} = 633.082996765566 \text{ mV}$ ,  $i_2 = 402.9438104116737 \text{ }\mu\text{A}$ , se calculan los demás valores por sustitución:

De la ecuación (89):

$$i_1 = I_s \left( e^{\frac{v_{D1}}{V_T}} - 1 \right) + i_2 = 43.669170032345 \text{ mA} \quad (94)$$

De la ecuación (88):

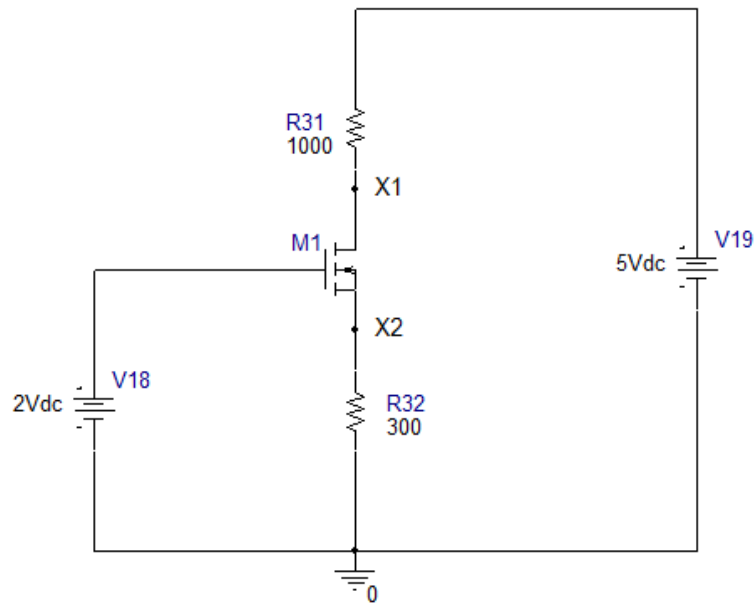
$$v_{D2} = v_{D1} - 300i_2 = 512.199853642064 \text{ mV} \quad (95)$$

### 6.3 Solución de ecuaciones no lineales de un circuito de corriente directa con un *MOSFET* (dos variables)

En la Figura 43, es presentada una topología típica para un amplificador *single* con un *MOSFET*, en la cual se ha considerado modulación de canal, sin el efecto cuerpo ( $V_{SB}=0$ ). El modelo implementado en el algoritmo es el descrito por las ecuaciones (35), (36) y (39), que incluye todas las regiones de operación del transistor, y establece el valor de la *corriente de dren* en cada una de ellas, como función de las tensiones *puerta-fuente* ( $v_{GS}$ ), y *dreno-fuente* ( $v_{DS}$ ), y por ello, se puede expresar de la forma  $i_D(v_{GS}, v_{DS})$ . Los parámetros utilizados para el transistor *NMOS* son [12]:

$$\text{Lambda}=0.1, \text{Phi}=0.9, \text{Gamma}=0.45, \text{Vt0}=0.7, K_n=2 \times 10^{-5}, W/L=20/0.18$$

Figura 43. Circuito con un solo MOSFET.



(Fuente: autor)

En la Figura 43 es posible notar que las variables  $x_1$  y  $x_2$  corresponden con las tensiones de *dren* y de *fuentes* del transistor respectivamente. Utilizando ecuaciones de nodos, se llega a las expresiones:

$$f_1 = i_D(2 - x_2, x_1 - x_2) - \frac{x_1 - x_2}{1000} = 0 \quad (96)$$

$$f_2 = i_D(2 - x_2, x_1 - x_2) - \frac{x_2}{300} = 0 \quad (97)$$

Con lo cual, la función objetivo implementada en el algoritmo, al hacer uso del teorema que conduce a la ecuación (23), es:

$$f(x1, x2) = \left( i_D(2 - x2, x1 - x2) - \frac{x1 - x2}{1000} \right)^2 + \left( i_D(2 - x2, x1 - x2) - \frac{x2}{300} \right)^2 \quad (98)$$

Se realizaron cinco diferentes ejecuciones del algoritmo para resolver este circuito, y los resultados se encuentran registrados en la Tabla 22.

Tabla 22. Parámetros y resultados de ejecución para el circuito de la Figura 43.

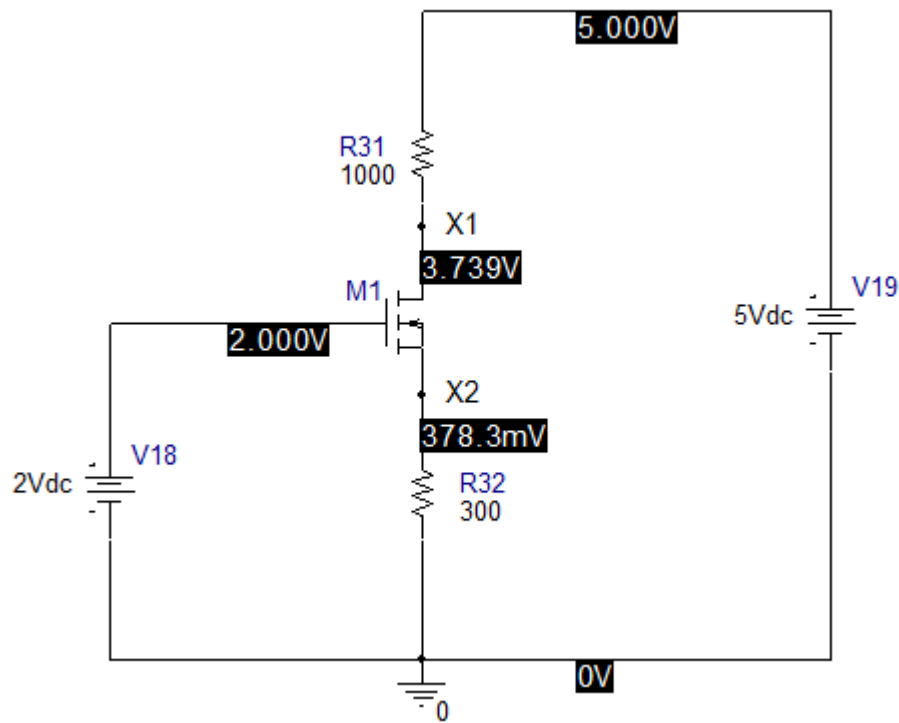
<i>Ejec</i>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4*</b>	<b>5</b>
<i>r</i>	0.95	0.95	0.95	0.95	0.85
<i>θ</i>	Pi/20000	Pi/20000	Pi/20000	Pi/20000	Pi/20000
<i>Kmax</i>	1000000	1000000	1000000	1000000	6764
<i>ε</i>	10 <sup>-20</sup>	10 <sup>-25</sup>	10 <sup>-30</sup>	10 <sup>-35</sup>	10 <sup>-35</sup>
<i>m</i>	100	100	100	100	100
<i>lix1</i>	0	0	0	0	0
<i>lsx1</i>	5	5	5	5	5
<i>nx1</i>	9	9	9	9	9
<i>lix2</i>	0	0	0	0	0
<i>lsx2</i>	5	5	5	5	5
<i>nx2</i>	9	9	9	9	9
<i>x1</i>	3.7389241149	3.7389241203	3.7389241202	3.7389241202	3.7389241202
<i>x2</i>	0.3783227492	0.3783227639	0.3783227639	0.3783227639	0.3783227639
<i>Error</i>	9.28203x10 <sup>-21</sup>	9.87919x10 <sup>-26</sup>	8.29952x10 <sup>-31</sup>	8.51057x10 <sup>-36</sup>	2.35098x10 <sup>-37</sup>
<i>k</i>	328	440	554	659	6764
<i>tejec</i>	7.160202 s	9.668829 s	11.301771 s	2.319220 s	249.900362 s

(Fuente: autor)

\* Parámetros almacenados previamente para la ejecución por defecto en el programa desarrollado con el presente trabajo de grado.

En la Tabla 22 se observa como progresivamente se fue obteniendo un menor error para la respuesta, hasta conseguirse un error que es menor que  $10^{-36}$ . Comparando con *ORCAD CAPTURE*, se tienen resultados muy similares para un nivel 1 (*level=1*), cuando se realiza el análisis de tensiones de *BIAS* al circuito con los mismos parámetros utilizados en el algoritmo tal como lo evidencia la Figura 44.

Figura 44. Circuito de un solo *MOSFET* y tensiones de polarización obtenidas con la simulación en *ORCAD*.

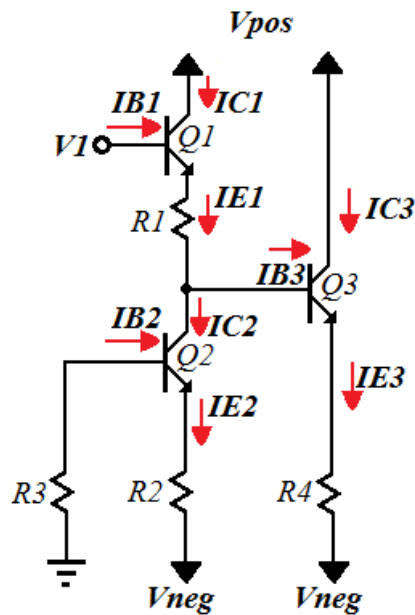


(Fuente: autor)

#### 6.4 Solución de ecuaciones no lineales de un circuito de corriente directa con tres BJT's (tres y cinco variables)

Se presenta a continuación en la Figura 45 a un circuito denominado cambiador de nivel [16]:

Figura 45. Circuito *cambiador de nivel*.



(Basado de [16])

En donde:  $V_1 = 6 V$ ,  $V_{pos} = 12 V$ ,  $V_{neg} = -12 V$ ,  $R_1 = 1 K\Omega$ ,  $R_2 = 2 K\Omega$ ,  $R_3 = 500 \Omega$ ,  $R_4 = 40 K\Omega$

Este circuito se ha modelado con la expresión exponencial básica (ver *pág 34*) y con el modelo *Ebers-Moll* (ver *pág. 34*). Los valores obtenidos para los resultados en cada caso también son distintos, fundamentalmente porque se trata de dos modelos de diferente nivel de complejidad, y que difieren en los parámetros requeridos.

#### 6.4.1 Solución utilizando el modelo exponencial básico

Se plantean leyes básicas de circuitos para la topología presentada en la Figura 45 y se tiene que:

Aplicando LVK:

$$-6 + 100I_{E1} + v_{BE} + 500I_{E3} - 12 = 0 \quad (99)$$

$$R_4 I_{B2} + v_{BE2} + 200I_{E2} - 12 = 0 \quad (100)$$

Aplicando la Ley de Corrientes de Kirchhoff (LCK):

$$I_{E1} - I_{C2} - I_{B3} = 0 \quad (101)$$

Las corrientes del transistor en términos del voltaje *base-emisor* de acuerdo con el modelo señalado en la *página 34* son:

$$I_C = \beta I_B = I_S e^{\frac{v_{BE}}{V_T}} \quad (102)$$

$$I_B = \frac{I_S}{\beta} e^{\frac{v_{BE}}{V_T}} \quad (103)$$

$$I_E = (\beta + 1)I_B = \frac{\beta + 1}{\beta} I_S e^{\frac{v_{BE}}{V_T}} \quad (104)$$

Donde,  $I_S = 10^{-12} A$  y  $V_T = 25.85 mV$ .

Reemplazando las ecuaciones (102), (103) y (104) en (99), (100) y (101); se obtienen las siguientes tres ecuaciones cuyas incógnitas son los voltajes *base-*

emisor de los tres transistores, constituyéndose de esta manera, en un sistema con tres variables.

$$f_1 = -6 + \frac{\beta + 1}{\beta} I_S \left( R_1 e^{\frac{v_{BE1}}{V_T}} + R_3 e^{\frac{v_{BE3}}{V_T}} \right) + v_{BE3} - 12 = 0 \quad (105)$$

$$f_2 = \frac{I_S}{\beta} e^{\frac{v_{BE2}}{V_T}} (R_4 + (\beta + 1)R_2) + v_{BE2} - 12 = 0 \quad (106)$$

$$f_3 = I_S \left( \frac{\beta + 1}{\beta} e^{\frac{v_{BE1}}{V_T}} - e^{\frac{v_{BE2}}{V_T}} - \frac{1}{\beta} e^{\frac{v_{BE3}}{V_T}} \right) = 0 \quad (107)$$

y la función objetivo implementada en el algoritmo es:

$$f(v_{BE1}, v_{BE2}, v_{BE3}) = f_1^2 + f_2^2 + f_3^2 \quad (108)$$

en donde:

$$(x1, x2, x3) = (v_{BE1}, v_{BE2}, v_{BE3}) \quad (109)$$

Tabla 23. Parámetros y resultados de la ejecución del algoritmo para el circuito de la Figura 45.

<i>Ejec</i>	1	2 *	3
<i>r</i>	0.95	0.95	0.95
$\theta$	Pi/10 <sup>40</sup>	Pi/9*10 <sup>24</sup>	Pi/10 <sup>40</sup>
<i>kmax</i>	20000	20000	20000
$\epsilon$	10 <sup>-10</sup>	10 <sup>-6</sup>	10 <sup>-10</sup>
<i>m</i>	8000	729	8000
<i>lix1</i>	0.56	0.5	0.55

\* Parámetros almacenados previamente para la ejecución por defecto en el programa desarrollado con el presente trabajo de grado.

<b>lsx1</b>	0.58	0.7	0.65
<b>nx1</b>	19	8	19
<b>lix2</b>	0.56	0.5	0.55
<b>lsx2</b>	0.58	0.7	0.65
<b>nx2</b>	19	8	19
<b>lix3</b>	0.60	0.5	0.55
<b>lsx3</b>	0.62	0.7	0.65
<b>nx3</b>	19	8	19
<b>x1</b>	0.573523602448310	0.551667471761160	0.575558242761953
<b>x2</b>	0.575814718732461	0.575814722253755	0.575814719071569
<b>x3</b>	0.619694313019193	0.624211660448162	0.618976234470199
<b>Error</b>	$3.7503487700779 \times 10^{-7}$	$9.9301750477419 \times 10^{-6}$	$6.15431143456253 \times 10^{-8}$
<b>k</b>	20000	20000	20000
<b>tejec</b>	7564.331101 s	348.747688 s	6139.853156 s

(Fuente: autor)

Utilizando el comando *fsolve* de *MATLAB*, con los valores iniciales de *x01*, *x02* y *x03* para las variables, se obtienen los resultados consignados en la Tabla 24, los cuales son muy similares a los obtenidos con el algoritmo de la espiral, en la ejecución número 3 de la Tabla 23, la cual presenta el menor error.

Tabla 24. Parámetros y resultados obtenidos con *fsolve* de *MATLAB* en la solución al modelo del circuito con 3 *BJTs*.

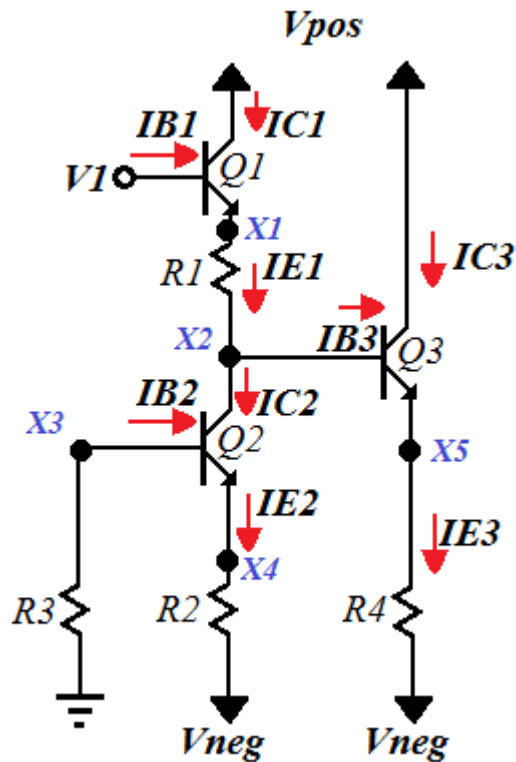
<b>x01</b>	0.4
<b>x02</b>	0.4
<b>x03</b>	0.4
<b>X1</b>	0.576855489322088
<b>X2</b>	0.575814143773232
<b>X3</b>	0.618478451720388

(Fuente: autor)

### 6.4.2 Solución usando el modelo de Ebers-Moll

Se establecen las tensiones de nodo mostradas en la Figura 46:

Figura 46. Designación de nodos para el circuito *cambiador de nivel*.



(Fuente: autor)

En donde:  $V_1 = 6\text{ V}$ ,  $V_{pos} = 12\text{ V}$ ,  $V_{neg} = -12\text{ V}$ ,  $R_1 = 1\text{ K}\Omega$ ,  $R_2 = 2\text{ K}\Omega$ ,  $R_3 = 500\ \Omega$ ,  $R_4 = 40\text{ K}\Omega$

De las ecuaciones (30) a (32), dados unos valores constantes para los parámetros de las ganancias de corriente, el voltaje térmico y la corriente de saturación; el modelo del *BJT* podría interpretarse como una función que posee dos variables independientes que son las tensiones *base-emisor* y *base-colector*, y tres variables dependientes, que son las corrientes de *emisor*, *colector* y de *base*. Así pues, en el algoritmo, se elaboró una función cuyas entradas son las tensiones

mencionadas y los valores fijos correspondientes a los parámetros del dispositivo, y las variables de salida son estas tres corrientes. El encabezado de esta función denominada "*ebersnpn*" se muestra en la siguiente ecuación.

$$[I_E, I_C, I_B] = \text{ebersnpn}(v_{be}, v_{bc}, v_t, is, \beta_{af}, \beta_{ar}) \quad (110)$$

En donde:

$I_E$ : Corriente de emisor.	$v_t$ : Voltaje térmico (25.85 mV).
$I_C$ : Corriente de colector.	$is$ : Corriente de saturación ( $10^{-12}$ A).
$I_B$ : Corriente de base.	$\beta_{af}$ : Ganancia de corriente directa (99).
$v_{be}$ : Tensión <i>base-emisor</i> .	$\beta_{ar}$ : Ganancia de corriente inversa (1/49).
$v_{bc}$ : Tensión <i>base-colector</i> .	

Planteando esta función para cada uno de los transistores del circuito de la Figura 46, se obtienen las siguientes expresiones:

$$[I_{E1}, I_{C1}, I_{B1}] = \text{ebersnpn}(V_1 - x_1, V_1 - V_{pos}, v_t, is, \beta_{af}, \beta_{ar}) \quad (111)$$

$$[I_{E2}, I_{C2}, I_{B2}] = \text{ebersnpn}(x_3 - x_4, x_3 - x_2, v_t, is, \beta_{af}, \beta_{ar}) \quad (112)$$

$$[I_{E3}, I_{C3}, I_{B3}] = \text{ebersnpn}(x_2 - x_5, x_2 - V_{pos}, v_t, is, \beta_{af}, \beta_{ar}) \quad (113)$$

Usando *LCK* sobre el circuito de la Figura 46, se obtienen las ecuaciones (110) a (113), las cuales conforman un sistema de ecuaciones con cinco variables, que son cada una de las tensiones  $x_1, x_2, x_3, x_4$  y  $x_5$ .

$$f_1 = I_{E1} - I_{B3} - I_{C2} = 0 \quad (114)$$

$$f_2 = \frac{x_3}{R_3} + I_{B2} = 0 \quad (115)$$

$$f_3 = \frac{x_4 - V_{neg}}{R_2} - i_{E2} = 0 \quad (116)$$

$$f_4 = \frac{x_5 - V_{neg}}{R_4} - i_{E3} = 0 \quad (117)$$

$$f_5 = \frac{x_1 - x_2}{R_1} - I_{E1} = 0 \quad (118)$$

Y la función objetivo asociada es, de acuerdo al teorema presentado en la página 31:

$$f(x_1, x_2, x_3, x_4, x_5) = f_1^2 + f_2^2 + f_3^2 + f_4^2 + f_5^2 \quad (119)$$

En la Tabla 25 se muestran los resultados obtenidos para una serie de ejecuciones, en donde en todas ellas se usaron trayectorias de espiral de carácter intenso (*ver página 45*), y después de la primera ejecución, se redujo la región factible. La ejecución número 5 presenta el menor error, y se realizó con una región factible conformada por un elevado número de puntos, lo cual permitió diversificar el método, y con ello aumentar en gran manera el tiempo de ejecución, que fue mayor a dos días, a pesar de haberse realizado en un equipo de cómputo de características superiores.

Tabla 25. Parámetros y resultados para la ejecución con el modelo *Ebers-Moll* del circuito *cambiador de nivel* de la Figura 46.

<i>Ejec</i>	1	2	3	4*	5**
<i>r</i>	0.95	0.95	0.95	0.95	0.95
$\theta$	$\pi/10^{40}$	$\pi/10^{40}$	$\pi/10^{40}$	$\pi/10^{40}$	$\pi/10^{40}$
<i>kmax</i>	50	50	150	1000	1000
$\varepsilon$	$10^{-7}$	$10^{-7}$	$10^{-7}$	$10^{-10}$	$10^{-40}$
<i>m</i>	800000	800000	800000	800000	8100000
<i>lix1</i>	4	5	5	5	5
<i>lsx1</i>	6	6	6	6	6
<i>nx1</i>	4	4	4	4	9
<i>lix2</i>	-2	0	0	0	0
<i>lsx2</i>	4	3	3	3	3
<i>nx2</i>	19	19	19	19	29
<i>lix3</i>	-12	-4	-4	-4	-4
<i>lsx3</i>	2	2	2	2	2
<i>nx3</i>	19	19	19	19	29
<i>lix4</i>	-12	-5	-5	-5	-5
<i>lsx4</i>	1	1	1	1	1
<i>nx4</i>	19	19	19	19	29
<i>lix5</i>	-12	-1	-1	-1	-2
<i>lsx5</i>	2	3	3	3	3
<i>nx5</i>	19	19	19	19	29
<i>x1</i>	5.415639856	5.419232320	5.428036783	5.428016614	5.422777450
<i>x2</i>	1.732135680	2.030787850	2.066069366	2.066069366	0.725617499
<i>x3</i>	-1.65932550	-0.10034514	-0.11399620	-0.11399620	0.108893371
<i>x4</i>	-2.24316364	-0.67826738	-0.69049503	-0.69057609	-0.47026300
<i>x5</i>	1.254078729	1.529203945	1.559126644	1.559178268	0.221120149
<i>Error</i>	$2.203688 \times 10^{-5}$	$6.386124 \times 10^{-6}$	$1.715457 \times 10^{-6}$	$1.715064 \times 10^{-6}$	$4.184258 \times 10^{-7}$
<i>k</i>	50	50	150	1000	1000
<i>tejec</i>	34247.8020 s	1312.47042 s	3852.68444 s	84410.2262 s	203740.818 s

(Fuente: autor)

\* Parámetros almacenados previamente para la ejecución por defecto en el programa desarrollado con el presente trabajo de grado.

\*\* Parámetros usados para la ejecución en un equipo del grupo CEMOS con procesador Intel® Core™ i7, 3.4 Ghz, memoria RAM de 8 Gb y funcionando en el modo de alto rendimiento.

Usando los resultados de la ejecución número 5 de la Tabla 25, los valores para las tensiones *base-emisor* de cada uno de los transistores es:

$$v_{BE1} = V_1 - x1 = 0.577222549325253 \text{ V} \quad (120)$$

$$v_{BE2} = x3 - x4 = 0.579156378220489 \text{ V} \quad (121)$$

$$v_{BE3} = x2 - x5 = 0.504497350062152 \text{ V} \quad (122)$$

Resultados que son razonables para transistores *BJT*, pero que en valores de tensiones de nodo, como aparecen en la Tabla 25, difieren de los obtenidos con el comando *fsolve* de *Matlab*, tal como lo muestra la Tabla 26; lo cual indica una dificultad del algoritmo para resolver un modelo del nivel de complejidad que representa *Ebers-Moll*.

Tabla 26. Resultados de la solución al modelo de la Figura 46 usando el comando *fsolve* de *Matlab*.

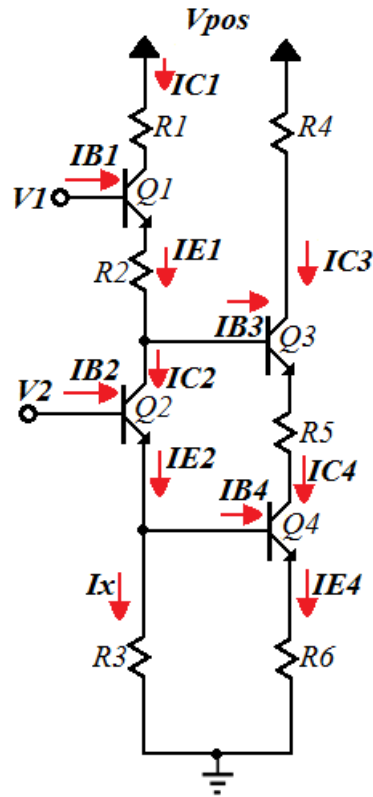
x01	4
x02	-12
x03	-12
x04	-12
x05	-12
X1	5.419839601709665
X2	-0.221516420080363
X3	-0.028434968959210
X4	-0.608842601575470
X5	-0.724221189212759

(Fuente: autor)

## 6.5 Solución a ecuaciones no lineales de un circuito de corriente directa con cuatro *BJT*s (cuatro variables)

Se plantea el siguiente circuito con cuatro transistores *BJT*:

Figura 47. Circuito con cuatro transistores *BJT*.



(Fuente: autor)

En donde:  $V_1 = 6\text{ V}$ ,  $V_2 = 4\text{ V}$ ,  $V_{pos} = 6\text{ V}$ ,  $R_1 = 1\text{ K}\Omega$ ,  $R_2 = 1\text{ K}\Omega$ ,  $R_3 = 500\ \Omega$ ,  
 $R_4 = 1\text{ K}\Omega$ ,  $R_5 = 500\ \Omega$ ,  $R_6 = 200\ \Omega$

Aplicando *LVK*, se tiene que:

$$-V_2 + v_{BE2} + v_{BE4} + I_{E4}R_6 = 0 \quad (123)$$

Aplicando LCK:

$$I_{E2} = I_{B4} + I_x \quad (124)$$

Aplicando LVK en la malla inferior del circuito de la Figura 47:

$$I_x R_3 = v_{BE4} + I_{E4} R_6 \quad (125)$$

Aplicando LCK:

$$I_{E1} = I_{B3} + I_{C2} \quad (126)$$

A su vez se cumple qué:

$$I_{E3} = I_{C4} \quad (127)$$

de acuerdo al *modelo básico exponencial* para un transistor *BJT* (ver página 33), las corrientes en términos de la tensión *base-emisor*, cuando se encuentra operando en la región activa, son:

$$I_C = I_S e^{\frac{v_{BE}}{V_T}} \quad (128)$$

$$I_B = \frac{I_S}{\beta} e^{\frac{v_{BE}}{V_T}} \quad (129)$$

$$I_E = \frac{\beta + 1}{\beta} I_S e^{\frac{v_{BE}}{V_T}} \quad (130)$$

El sistema de cuatro ecuaciones, con las cuatro incógnitas de los voltajes *base-emisor* se describe por las ecuaciones (131) a (134):

$$-V_2 + v_{BE2} + v_{BE4} + \frac{\beta + 1}{\beta} I_S R_6 e^{\frac{v_{BE4}}{V_T}} = 0 \quad (131)$$

$$\frac{\beta + 1}{\beta} I_S e^{\frac{v_{BE2}}{V_T}} - \frac{I_S}{\beta} e^{\frac{v_{BE4}}{V_T}} - \frac{1}{R_3} \left( v_{BE4} + \frac{\beta + 1}{\beta} I_S R_6 e^{\frac{v_{BE4}}{V_T}} \right) = 0 \quad (132)$$

$$\frac{\beta + 1}{\beta} I_S e^{\frac{v_{BE1}}{V_T}} - \frac{I_S}{\beta} e^{\frac{v_{BE3}}{V_T}} - I_S e^{\frac{v_{BE2}}{V_T}} = 0 \quad (133)$$

$$\frac{\beta + 1}{\beta} I_S e^{\frac{v_{BE3}}{V_T}} - I_S e^{\frac{v_{BE4}}{V_T}} = 0 \quad (134)$$

La Tabla 27 presenta los parámetros utilizados para el algoritmo, así como los resultados obtenidos con la ejecución para el valor óptimo final en diversas pruebas. En la ejecución número 1 se cuenta con la región factible más amplia, en la ejecución número 2 se delimita en base a los resultados obtenidos de la primera ejecución, y se obtiene un menor error, en menos número de iteraciones. En base a los resultados de la segunda ejecución, se delimita más la región factible para las otras tres ejecuciones, al tiempo que se varía el parámetro  $r$ , obteniéndose para todas estas, valores muy semejantes en las soluciones. Los valores de los errores obtenidos para las últimas cuatro ejecuciones, son muy similares, y la ejecución con el menor error, se encuentra resaltada en color verde.

Para el caso de la ejecución número 5, se quiso diversificar el método, por medio de aumentar el valor del parámetro  $r$ , al tiempo que se conservó pequeño el valor de  $\theta$ , y sin embargo, su convergencia fue hacia valores muy similares.

Tabla 27. Parámetros y resultados de la ejecución del algoritmo, para el circuito de la Figura 47.

<i>Ejec</i>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<i>r</i>	0.95	0.95	0.95	0.75	0.975
<i>θ</i>	$\pi/10^{40}$	$\pi/10^{40}$	$\pi/10^{40}$	$\pi/10^{40}$	$\pi/10^{40}$
<i>Kmax</i>	20000	1000	20000	20000	20000
<i>ε</i>	$10^{-16}$	$10^{-8}$	$10^{-16}$	$0.11 \cdot 10^{-16}$	$0.11 \cdot 10^{-4}$
<i>m</i>	625	16	625	625	625
<i>lix1</i>	0.4	0.5	0.59	0.59	0.59
<i>lsx1</i>	0.8	0.8	0.61	0.61	0.61
<i>nx1</i>	4	1	4	4	4
<i>lix2</i>	0.4	0.5	0.59	0.59	0.59
<i>lsx2</i>	0.8	0.8	0.61	0.61	0.61
<i>nx2</i>	4	1	4	4	4
<i>lix3</i>	0.4	0.5	0.59	0.59	0.59
<i>lsx3</i>	0.8	0.8	0.61	0.61	0.61
<i>nx3</i>	4	1	4	4	4
<i>lix4</i>	0.4	0.5	0.59	0.59	0.59
<i>lsx4</i>	0.8	0.8	0.61	0.61	0.61
<i>nx4</i>	4	1	4	4	4
<i>x1</i>	0.6404479859	0.5996186944	0.5953718944	0.5953683276	0.5953639252
<i>x2</i>	0.6404479859	0.5995780095	0.5952774218	0.5952738057	0.5952781919
<i>x3</i>	0.6030042229	0.6033943284	0.6034114101	0.6034114431	0.6033941310
<i>x4</i>	0.6032531857	0.6036293059	0.6036686262	0.6036686592	0.6036600107
<i>Error</i>	0.0026265900	$2.516233 \times 10^{-5}$	$1.008081 \times 10^{-5}$	$1.007177 \times 10^{-5}$	$1.096871 \times 10^{-5}$
<i>k</i>	20000	1000	20000	20000	225
<i>tejec</i>	634.672170 s	2.048035 s	1175.87522 s	1279.89178 s	14.157662 s

(Fuente: autor)

A continuación, en la Tabla 28 se muestra los resultados de ejecutar el programa con exactamente los mismos parámetros, y se evidencia la repetitividad de los mismos para cada una de las cinco ejecuciones.

Tabla 28. Resultados de ejecución repetida para los mismos parámetros.

<i>Ejec</i>	<i>1</i> *	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
<i>r</i>	0.95	0.95	0.95	0.95	0.95
$\theta$	$\pi/10^{40}$	$\pi/10^{40}$	$\pi/10^{40}$	$\pi/10^{40}$	$\pi/10^{40}$
<i>Kmax</i>	1000	1000	1000	1000	1000
$\varepsilon$	$10^{-8}$	$10^{-8}$	$10^{-8}$	$10^{-8}$	$10^{-8}$
<i>m</i>	16	16	16	16	16
<i>lix1</i>	0.5	0.5	0.5	0.5	0.5
<i>lsx1</i>	0.8	0.8	0.8	0.8	0.8
<i>nx1</i>	1	1	1	1	1
<i>lix2</i>	0.5	0.5	0.5	0.5	0.5
<i>lsx2</i>	0.8	0.8	0.8	0.8	0.8
<i>nx2</i>	1	1	1	1	1
<i>lix3</i>	0.5	0.5	0.5	0.5	0.5
<i>lsx3</i>	0.8	0.8	0.8	0.8	0.8
<i>nx3</i>	1	1	1	1	1
<i>lix4</i>	0.5	0.5	0.5	0.5	0.5
<i>lsx4</i>	0.8	0.8	0.8	0.8	0.8
<i>nx4</i>	1	1	1	1	1
<i>x1</i>	0.5996186944	0.5996186944	0.5996186944	0.5996186944	0.5996186944
<i>x2</i>	0.5995780095	0.5995780095	0.5995780095	0.5995780095	0.5995780095
<i>x3</i>	0.6033943284	0.6033943284	0.6033943284	0.6033943284	0.6033943284
<i>x4</i>	0.6036293059	0.6036293059	0.6036293059	0.6036293059	0.6036293059
<i>Error</i>	$2.516233 \times 10^{-5}$	$2.516233 \times 10^{-5}$	$2.516233 \times 10^{-5}$	$2.516233 \times 10^{-5}$	$2.516233 \times 10^{-5}$
<i>k</i>	1000	1000	1000	1000	1000
<i>tejec</i>	2.116801	2.161709 s	2.211017 s	2.188958 s	2.191802 s

(Fuente: autor)

---

\* Parámetros almacenados previamente para la ejecución por defecto en el programa desarrollado con el presente trabajo de grado.

Utilizando el comando *fsolve* de *MATLAB* para los valores iniciales  $x01$ ,  $x02$ ,  $x03$  y  $x04$  de las variables, se obtienen resultados similares a los mejores resultados obtenidos con el algoritmo de la espiral (*Ejecución número 4 de la Tabla 27*), tal como lo muestra la Tabla 29.

**Tabla 29.** Parámetros y resultados obtenidos con el comando *fsolve* de *MATLAB* para resolver el circuito de 4 *BJTs*.

$x01$	0.4
$x02$	0.4
$x03$	0.4
$x04$	0.4
$X1$	0.585876040681550
$X2$	0.585622152612941
$X3$	0.603499572726808
$X4$	0.603756708170675

(Fuente: autor)

## 6.6 Solución de ecuaciones no lineales de un circuito de corriente directa con cinco *MOSFET* (4 variables)

En la Figura 48 es presentado un circuito convertidor de corriente a voltaje [17], para el cual en todos los transistores es considerada la modulación de canal, sin el efecto cuerpo ( $V_{SB}=0$ ). El modelo implementado en el algoritmo es el descrito por las ecuaciones (35) a (39), el cual contempla inicialmente posible todas las regiones de operación de los transistores, para establecer el valor de la *corriente de drenó*  $i_D(v_{GS}, v_{DS})$ , que está en función de las tensiones *puerta-fuente* ( $v_{GS}$ ), y *drenó-fuente* ( $v_{DS}$ ). Los parámetros utilizados son [12]:

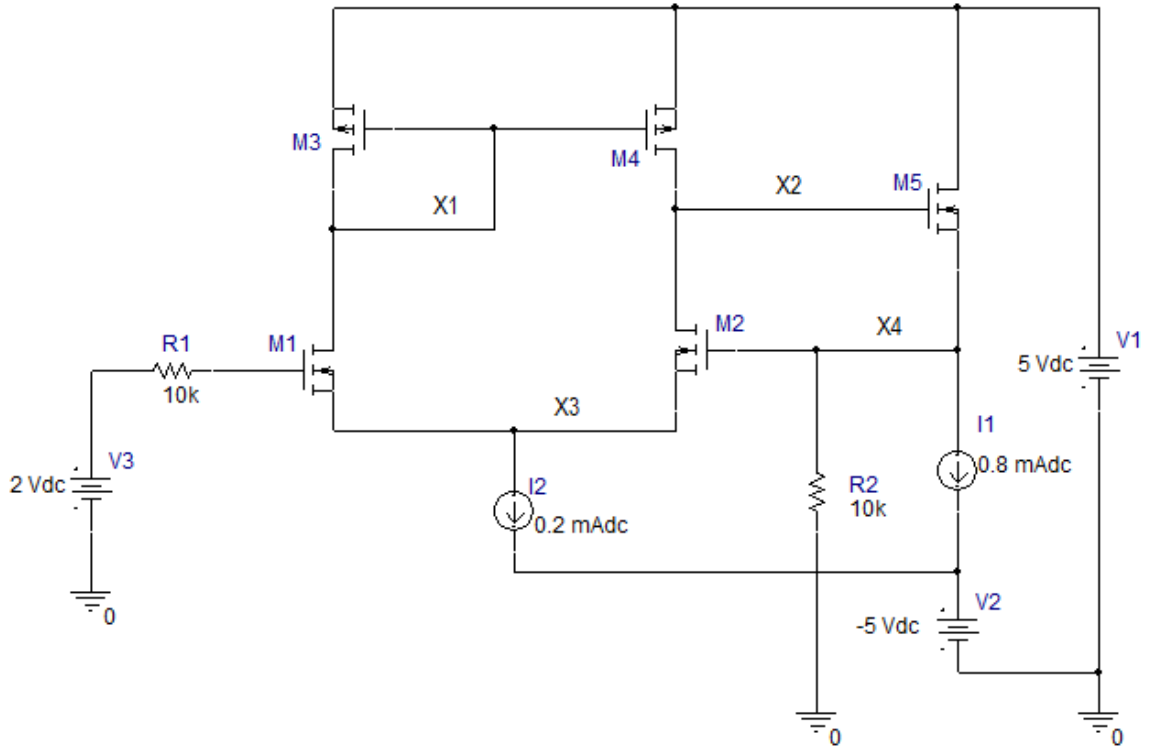
Para todos los transistores *NMOS*:

$\Lambda=0.1$ ,  $\Phi=0.9$ ,  $\Gamma=0.45$ ,  $V_{t0}=0.7$ ,  $K_n=2 \times 10^{-5}$ ,  $W/L=20/0.18$

Para todos los transistores *PMOS*:

$$\lambda=0.2, \Phi=0.8, \Gamma=0.4, V_{t0}=-0.8, K_p=5 \times 10^{-6}, W/L=20/0.18$$

Figura 48. Circuito con 5 *MOSFET*.



(Fuente: autor)

Aplicando *LCK*:

$$f_1 = i_{DN1}(2 - x_3, x_1 - x_3) - i_{DP3}(5 - x_1, 5 - x_1) = 0 \quad (135)$$

$$f_2 = i_{DN2}(x_4 - x_3, x_2 - x_4) - i_{DP4}(5 - x_1, 5 - x_2) = 0 \quad (136)$$

$$f_3 = i_{DN1}(2 - x_3, x_1 - x_3) + i_{DN2}(x_4 - x_3, x_2 - x_4) - 0.2 * 10^{-3} = 0 \quad (137)$$

$$f_4 = i_{DP3}(5 - x_1, 5 - x_1) - \frac{x_4}{10000} - 0.8 * 10^{-3} = 0 \quad (138)$$

En donde,

$i_{DNI}$ : Corriente de drenos para el  $i$ -ésimo transistor NMOS.

$i_{DPI}$ : Corriente de drenos para el  $i$ -ésimo transistor PMOS.

Haciendo uso del teorema representado por la ecuación (23), la función objetivo a implementarse en el algoritmo es entonces:

$$f(x_1, x_2, x_3, x_4) = f_1^2 + f_2^2 + f_3^2 + f_4^2 \quad (139)$$

Los resultados en la ejecución del algoritmo para la ecuación anterior, son presentados en la Tabla 30, en la cual se han realizado 5 ejecuciones en las que se va progresivamente delimitando la región factible, al tiempo que se mejora el error obtenido. Todas ellas se realizaron con trayectorias de espiral de carácter intenso (*ver página 45*), y el mejor resultado obtenido, es el de la ejecución número 5, que se encuentra resaltada en color verde.

Tabla 30. Parámetros y resultados para la solución con el algoritmo del modelo de la Figura 48.

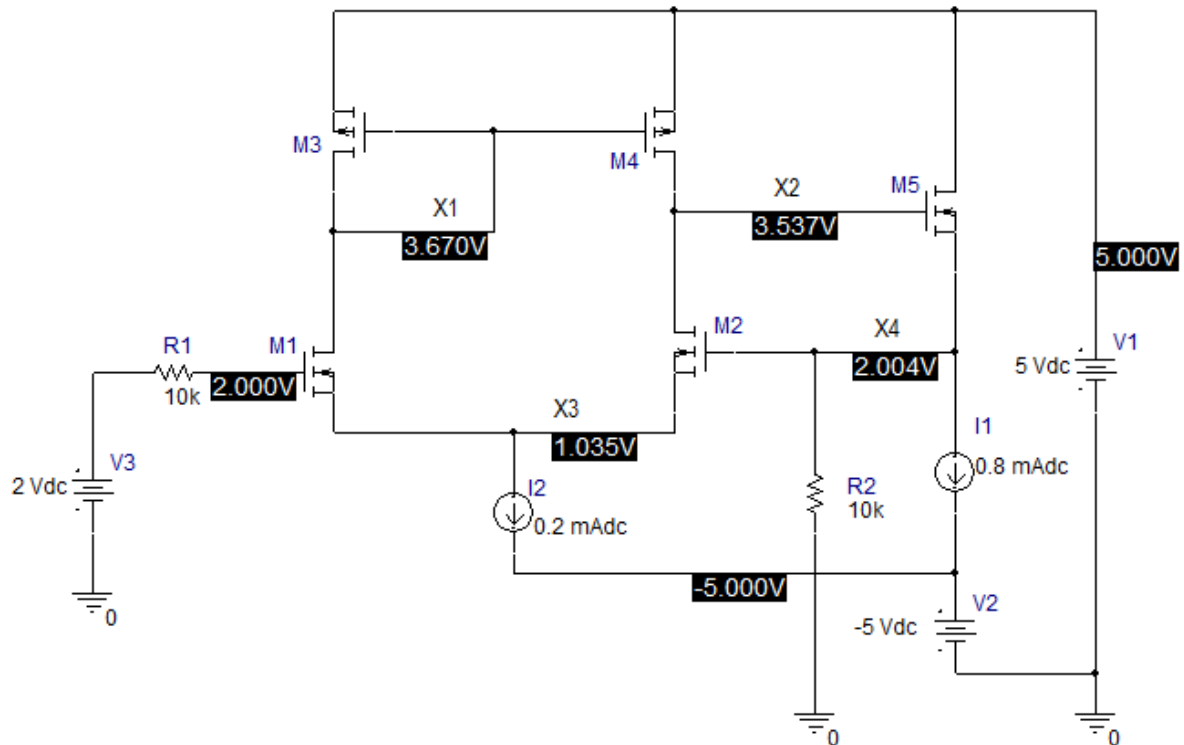
<i>Ejec</i>	1	2	3	4	5*
<i>r</i>	0.95	0.95	0.95	0.95	0.95
$\theta$	$\pi/10^{40}$	$\pi/10^{40}$	$\pi/10^{40}$	$\pi/10^{40}$	$\pi/10^{40}$
<i>Kmax</i>	1000	1000	1000	1000	1000
$\varepsilon$	$10^{-15}$	$10^{-15}$	$10^{-20}$	$10^{-25}$	$10^{-35}$
<i>m</i>	10000	10000	10000	10000	10000
<i>lix1</i>	3	3	3	3	3
<i>lsx1</i>	5	4	4	4	4
<i>nx1</i>	9	9	9	9	9
<i>lix2</i>	3	3	3	3	3
<i>lsx2</i>	5	4	4	4	4
<i>nx2</i>	9	9	9	9	9
<i>lix3</i>	0	0	0	0	0
<i>lsx3</i>	3	2	2	2	2
<i>nx3</i>	9	9	9	9	9
<i>lix4</i>	0	1.5	1.5	1.5	1.5
<i>lsx4</i>	3	2.5	2.5	2.5	2.5
<i>nx4</i>	9	9	9	9	9
<i>x1</i>	3.6700569448	3.6695286488	3.6695434519	3.6695434992	3.6695434993
<i>x2</i>	3.5567536902	3.5365185720	3.5365564793	3.5365565890	3.5365565893
<i>x3</i>	1.0439861885	1.0344737279	1.0345002848	1.0345003648	1.0345003651
<i>x4</i>	2.0222320643	2.0041694788	2.0041972302	2.0041973115	2.0041973118
<i>Error</i>	$1.01129 \times 10^{-10}$	$9.95397 \times 10^{-16}$	$8.55763 \times 10^{-21}$	$8.92852 \times 10^{-26}$	$9.48440 \times 10^{-36}$
<i>k</i>	1000	168	281	393	615
<i>tejec</i>	2925.88738 s	518.816610 s	927.465430 s	1279.88614 s	1790.012846s

(Fuente: autor)

\* Parámetros almacenados previamente para la ejecución por defecto en el programa desarrollado con el presente trabajo de grado.

En la Figura 49 se presenta los resultados del análisis de *BIAS* en tensión, realizado con *ORCAD*, utilizando los mismos valores de los parámetros que se introdujeron en el algoritmo de la espiral, y tomando para los transistores un modelo de nivel 1.

Figura 49. Circuito con 5 *MOSFET* y los valores de tensión obtenidos con la simulación en *ORCAD*.



(Fuente: autor)

Comparativamente, los resultados de la Figura 49, son muy similares a los correspondientes a la ejecución resaltada de la Tabla 30; lo cual indica que el modelo de este circuito ha sido resuelto de manera satisfactoria.

## 7. OBSERVACIONES Y CONCLUSIONES

- Se cumplieron adecuadamente todos los objetivos aprobados en el plan de proyecto (*Ver página 21*).
- Se implementó el algoritmo de la espiral de optimización, para que funcione con 1 a 10 variables, y se documentó su apropiado funcionamiento con *una, dos, tres y cuatro dimensiones* para algunas funciones de prueba, como la función *De Jong's, Rosenbrock's valley, Rastrigin's y Schwefel*. Aunque no se encuentra reportados los resultados en este documento, el algoritmo responde apropiadamente para las mismas pruebas, en los casos de 5 a 10 variables en donde también fue implementado.
- Se propuso una aplicación del método de la espiral, hasta ahora no reportada en la literatura internacional y estrechamente relacionada con un problema existente en la Ingeniería Electrónica: la solución numérica de modelos matemáticos *no-lineales* que describen circuitos electrónicos de c.c.
- Se resolvieron satisfactoriamente, las ecuaciones que modelan circuitos de c.c. con elementos no lineales, tales como diodos, *BJTs*, y *MOSFETs*, desde una sola variable, y hasta un máximo de cuatro variables; y los resultados obtenidos se compararon con otro *software* de simulación, como *ORCAD*, o el mismo *Matlab* y su función *fsolve*, obteniéndose resultados muy similares a los encontrados con el algoritmo de la espiral (*ver páginas 86 a 117*).
- Se encontró que el algoritmo no pudo resolver satisfactoriamente el circuito *cambiador de nivel*, cuando se usa el modelo *Ebers-Moll* para los transistores *BJT* que lo conforman. Esto se debe posiblemente, a que este modelo presenta un muy elevado nivel de convergencia para el algoritmo.

- Respecto al programa basado en el método de la espiral de optimización que también permite la solución de una función objetivo con una sola variable; es de señalar, que esta posibilidad no estaba explícita en el artículo de los autores [1], quienes lo exponen partiendo de dos variables y estableciendo los lineamientos generales para tres variables o más.
- El método de la espiral converge eficientemente dependiendo de los valores escogidos para los puntos de búsqueda iniciales (región factible de búsqueda), y a su vez, los parámetros  $r$  y  $\theta$ . Para todo caso se asume que se tiene un adecuado conocimiento de la región factible.
- Es importante resaltar que se detectó una manera para diversificar, o intensificar la trayectoria de la espiral logarítmica, la cual no se encuentra reportada en las referencias consultadas, y es fruto del análisis y la experiencia alcanzada con el método por parte del autor de esta tesis de grado. Refiriéndose al modelo discreto de la espiral logarítmica de optimización, este resultado se establece de la siguiente manera: *“dados los dominios  $0 < r < 1$ , y  $0 < \theta < 2\pi$ ; para diversificar, es necesario aumentar el valor de  $r$ , manteniendo un valor de  $\theta$  pequeño dentro del dominio permitido para  $\theta$ ; y para intensificar, se requiere disminuir el valor de  $\theta$ , manteniendo un valor superior de  $r$ , y que esté dentro del dominio que rige a  $r$ ” (ver página 39 en adelante).*
- La mejor manera de conseguir una región factible adecuada en las aplicaciones de circuitos con transistores *BJT*, de los cuales se tiene conocimiento que están en la región activa de operación, y por lo cual es posible aplicar el modelo básico exponencial (ver página 33); consiste en introducir un sistema de ecuaciones, cuyas variables sean justamente las tensiones de *base-emisor* de dichos transistores, puesto que para ellos se sabe que la solución está en el entorno aproximado de 0,5 a 0,8 V.

- Se implementó una rutina de *parada* para el método de la espiral basado en el error tolerado para la respuesta, al cual se le denominó épsilon ( $\varepsilon$ ). En cada iteración se compara con este parámetro y si resulta menor o igual a él, se finaliza la búsqueda. Esta difiere a la propuesta por los autores de este método de optimización que finalizan el algoritmo basados solamente en un número máximo de iteraciones, lo cual puede no ser una manera eficiente de ejecutar un programa de optimización.
- Para los valores de los puntos iniciales, se implementó una rutina en *Matlab*, la cual crea una retícula de puntos en la región factible, al establecerse los valores iniciales y finales de esta región, y el número de divisiones que se requiere entre ellos. Si alguno de estos puntos iniciales, es el valor solución, o está muy cerca de ella, el método ha de finalizar o de converger rápidamente a este, según sea el caso. Aumentar el número de divisiones para la región factible, se considera como otra forma de diversificar el método. El tiempo de ejecución disminuye notablemente, cuando se ejecuta el programa sin la opción para graficar (*ver ANEXO*).
- El algoritmo implementado se caracteriza por ser determinista, y con ello también, de resultados repetibles; hecho que se sustenta en la Tabla 28, en donde se reportan los resultados de ejecutar el programa con exactamente los mismos parámetros, obteniéndose los mismos resultados para los valores de las variables.

## 8. RECOMENDACIONES

- Para las aplicaciones del algoritmo en la solución del modelo, es fundamental delimitar en lo más posible la región factible, puesto que cuando existen varias soluciones en una región de búsqueda, el algoritmo converge hacia solo una de ellas (ver página 83).
- Una estrategia de uso del algoritmo puede consistir en ejecutarlo varias veces, y a partir de las soluciones obtenidas cuando el error se hace aceptable, y se percibe una tendencia a converger hacia unos valores dados, ir delimitando la región factible en las ejecuciones posteriores.
- Como resultado del gran número de pruebas que se realizaron con este algoritmo, se propone modificar los parámetros de la siguiente forma: si se ejecuta el programa, y se observa que durante un gran número de iteraciones, el error es todavía considerable, se debe proceder a diversificar el método (no sin antes revisar la región factible). Por otra parte, si se corre el programa, y se observa que el error se hace pequeño, pero no obstante no satisface al parámetro ( $\varepsilon$ ) del error requerido, entonces se procede a intensificar el método.
- Para los circuitos con transistores *BJT*, que se trabajan con el modelo básico exponencial; es conveniente llegar a un sistema de ecuaciones cuyas variables sean las tensiones *base-emisor* de cada uno de ellos, siendo que cuando estos se encuentran en la región activa de operación, la región factible es delimitable al rango [0.5, 0.8] V.
- Cuando se ejecuta el programa, y se obtienen valores grandes para el error, debe verificarse que los parámetros de ejecución sean los adecuados, así como lo debe ser la región factible. Adicionalmente, si se tiene que todos los

parámetros son los apropiados, el error puede ser un indicio de una inconsistencia en el modelo de las ecuaciones, y son entonces las ecuaciones introducidas en el programa las que deben ser revisadas, puesto que posiblemente carecen de solución.

## REFERENCIAS

- [1] K. Tamura, "Spiral optimization-A new multipoint search method," *IEEE*, no. 1, pp. 1759–1764, 2011.
- [2] M. Horenstein, *Microelectrónica: circuitos y dispositivos*, 2nd ed. Prentice Hall, 1997.
- [3] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*, 2nd ed. United Kingdom: Luniver Press, 2010, p. 1.
- [4] E. Maor and J. Pinasco, *e: historia de un número*. Mexico: Consejo Nacional para la Cultura y las Artes, 2006.
- [5] K. Tamura and K. Yasuda, "Primary study of spiral dynamics inspired optimization," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 6, no. S1, pp. S98–S100, Nov. 2011.
- [6] K. Tamura and K. Yasuda, "Spiral Multipoint Search for Global Optimization," *2011 10th International Conference on Machine Learning and Applications and Workshops*, vol. 2, no. 2, pp. 470–475, Dec. 2011.
- [7] I. Amaya, J. Cruz, and R. Correa, "Real Roots of Nonlinear Systems of Equations Through a Metaheuristic Algorithm," *Revista Dyna*, vol. 78, no. 170, pp. 15–23, 2011.
- [8] A. Abbasian, M. Taherzadeh-Sani, B. Amelifard, and A. Afzali-Kusha, "Modeling of MOS Transistors Based on Genetic Algorithm and Simulated Annealing," *2005 IEEE International Symposium on Circuits and Systems*, no. 4, pp. 6218–6221.

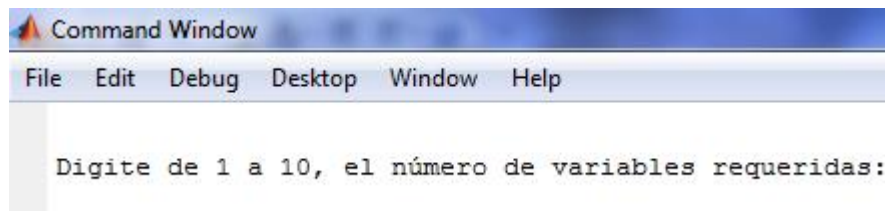
- [9] B. Razavi, *Fundamentals of microelectronics*, Preview. Wiley, 2006, pp. 128–134.
- [10] N. Deshpande, *Electronic Devices & Circuits, Principles and applications*. New Delhi: Tata McGraw-Hill Publishing Company Limited, 2007, pp. 309–312.
- [11] T. Sakurai and A. Newton, “Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas,” *Solid-State Circuits, IEEE Journal of*, p. 584, 1990.
- [12] B. Razavi, *Design of analog CMOS integrated circuits*, 1st ed. Singapore: Mc Graw Hill International Edition, 2001, pp. 9–37.
- [13] P. Gray, *Analysis and design of analog integrated circuits*, 4th ed. John Wiley & Sons, 2001, p. 47.
- [14] M. Molga and C. Smutnicki, “Test functions for optimization needs,” pp. 1–43, 2005.
- [15] B. Ribeiro, R. Albrecht, A. Dobnikar, D. Pearson, and N. Steele, “Adaptive and Natural Computing Algorithms,” *Springer Computer Science*, p. 220, 2005.
- [16] M. Cirovic, *Electrónica fundamental: Dispositivos, circuitos y sistemas*. España: Reverté, 2003, pp. 280–281.
- [17] A. Sedra and K. Smith, *Circuitos microelectrónicos*, 4th ed. Mexico: Oxford University Press, 2002, p. 759.

## ANEXOS

### ANEXO A. Ejecución con los parámetros señalados en el informe

La carpeta del algoritmo posee varios archivos de funciones con la extensión *.m*. No obstante, el *script* principal para la ejecución del algoritmo se llama *Algor\_espiral.m*, el cual se debe correr en primera instancia, y este mismo se encarga de ir llamando las demás funciones según lo vaya requiriendo. Su ejecución hace aparecer en la ventana de comandos el mensaje presentado en la Figura 50.

Figura 50. Mensaje de inicio en la ejecución del *Algoritmo de La Espiral*.



(Fuente: autor)

A lo cual se debe ingresar el número de variables en que se está interesado, bien sea para ejecutar una función de prueba, o para resolver las ecuaciones asociadas a un modelo de circuito del presente informe.

Si por ejemplo se selecciona el número 2, el mensaje de diálogo que aparece en la ventana de comandos, se presenta en la Figura 51.

Figura 51. Primera lista de opciones desplegada por el algoritmo.

```
PROGRAMA QUE UTILIZA EL MÉTODO DE LA ESPIRAL

A continuación el menú de opciones disponibles:

1. Resolver ecuación de De Jongs
2. Resolver ecuación de Rosembrock
3. Resolver ecuación de Rastrigin
4. Resolver ecuación de Schwefel
5. Resolver el modelo del circuito de dos diodos
6. Resolver el modelo del circuito con un solo MOSFET
9. Resolver otro modelo programado previamente por el usuario

Digite la opción deseada:
```

(Fuente: autor)

Suponiendo que se requiere examinar el algoritmo con la ecuación de *Schwefel*, se debe escribir entonces el número *4*, precedido de un *enter*, y el mensaje de diálogo que aparece es presentado a continuación:

*“Las opciones para ejecutar son:*

- 1. Si desea correr el programa con los valores señalados en el informe de la tesis.*
- 2. Si desea introducir usted mismo los valores para estos parámetros.*

*Digite la opción deseada: “*

Si se desea correr esta prueba con los parámetros que se señalan en el informe, se ha de escoger la opción *1*, para la cual aparece una lista de los parámetros con los que se va a ejecutar, así como también el siguiente mensaje de diálogo:

*“La presentación de la gráfica de las trayectorias de solución, aumenta de forma importante el tiempo de ejecución del programa.*

*Digite 1 si desea una gráfica en la solución, o cualquier otra tecla si no la desea: “*

Al seleccionar 1 y luego dar *enter*, el algoritmo iniciará como tal su labor de búsqueda, lo cual indica con la palabra “*Procesando...*” que aparece en la ventana de comandos, y entonces sucederá que dependiendo del valor del retardo preestablecido, la gráfica mostrará secuencialmente a una velocidad dada, el historial de puntos que el algoritmo va encontrando hasta llegar al valor óptimo final, el cual aparecerá como una marca de color magenta. Solo hasta finalizar totalmente la ejecución con los parámetros establecidos, un mensaje en la ventana de comandos aparecerá como se muestra en la Figura 52, y en una salida de audio activa que se tenga en el equipo de cómputo donde se esté corriendo, procederá un sonido de campana (*beep*).

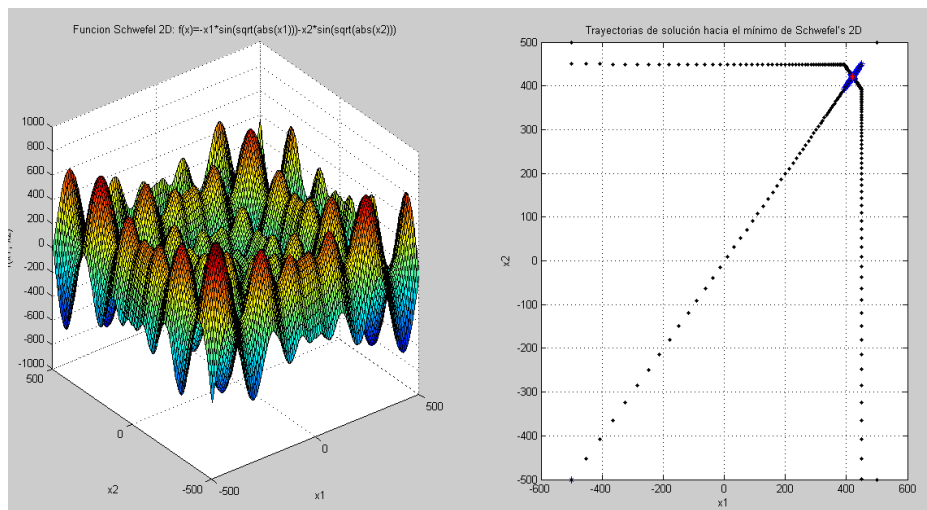
Figura 52. Mensaje de finalización de la ejecución.

```
Ejecución terminada exitosamente!!!
fx >>
```

(Fuente: autor)

La gráfica obtenida finalmente para esta situación, después de maximizar la ventana emergente es presentada en la Figura 53.

Figura 53. Gráficas a modo de ejemplo para la función *Schwefel* en 2D.



(Fuente: autor)

En la parte izquierda de la Figura 53 se encuentra la gráfica de la función de prueba dentro del intervalo estándar de la misma, y en la parte derecha, las trayectorias sobre el dominio de búsqueda para el algoritmo. Las marcas negras corresponden a los puntos que se van obteniendo con la matriz de rotación del algoritmo a medida que transcurren las iteraciones, y entre estas marcas negras estarán las marcas azules que corresponden con valores óptimos previos a la obtención del último valor óptimo, el cual se distingue en color magenta, y está ubicado exactamente en las mismas coordenadas que presenta el mensaje de diálogo, el cual en este caso se muestran en la Figura 54.

Figura 54. Resultados numéricos entregados por el algoritmo.

```
Elapsed time is 219.197498 seconds.  
  
x1=  
    4.209687664181200e+002  
  
x2=  
    4.209687272668258e+002  
  
El número k de iteraciones realizadas es  
    334  
  
El error obtenido es  
    9.526957001071423e-011  
  
Ejecución terminada exitosamente!!!  
fx >>
```

(Fuente: autor)

En la figura anterior, se aprecia que el algoritmo arroja el tiempo empleado para la ejecución (*Elapsed time*), los valores en formato largo para el punto óptimo en  $2D$  que minimiza la función, el número de iteraciones conseguidas, y el error de este cálculo, así como también el mensaje de fin de la ejecución.

Para el caso de tres variables, en algunas circunstancias la marca del óptimo final de color magenta no se encuentra visible en la gráfica de la trayectoria, aún cuando el algoritmo ya ha terminado, lo cual se debe a que ha quedado oculta por las demás que en perspectiva estarían primero para el observador. En tal caso, utilizar repetidamente la herramienta de ampliación de *MATLAB*, podría hacerla visible.

Tal como lo señala uno de los mensajes de diálogo en la ejecución del programa, la generación de las gráficas aumenta de manera significativa el tiempo para finalizar la ejecución, por lo cual no es recomendable esta opción para ejecuciones con un gran número de puntos de la región factible, o proyectadas hacia un gran número de iteraciones.

Internamente, el programa en su *script* principal utiliza un valor asignado a una variable llamada *retardo*, la cual representa el tiempo de permanencia en espera, cuando se localiza un punto de la trayectoria, mientras aparece un nuevo punto. Su utilización es principalmente para comprender la forma de evolución del algoritmo, pero no se hace indispensable para la consecución de los resultados numéricos correspondientes al valor óptimo final, y presentados a modo de ejemplo en la Figura 54.

Para cuatro variables en adelante, no se cuenta con la opción de generación de una gráfica de trayectoria, y para el caso de una variable, la gráfica de la trayectoria de convergencia se realiza sobre el dominio y recorrido de la función objetivo, en tanto que para 2 y de 3 variables las trayectorias se visualizan solo sobre el dominio de la función, de manera similar a como lo sugieren los autores del método.

## ANEXO B. Ejecución con unos parámetros ingresados por el usuario

En caso de no querer ejecutar el programa con los parámetros establecidos previamente, se selecciona la opción 2 ante el mensaje de la Figura 55.

Figura 55. Mensaje de diálogo para seleccionar opción de introducir los parámetros.

```
Las opciones para ejecutar son:  
  
1. Si desea correr el programa con los valores señalados en el informe  
2. Si desea introducir usted mismo los valores para estos parámetros.
```

(Fuente: autor)

De este modo el programa solicitará detalladamente los valores de los parámetros, los cuales a modo de ejemplo, se han ingresado como muestra la Figura 56.

Figura 56. Ejemplo de introducción de parámetros al programa desde el teclado.

```
Digite el error epsilon de tolerancia en la respuesta: 10^-8  
Digite el valor del parámetro theta: pi/10^50  
Digite el valor del parámetro r: 0.95  
Digite el valor del número máximo de iteraciones: 10000  
  
Para la región factible:  
  
Digite el valor del límite inferior de x1: -500  
Digite el valor del límite superior de x1: 500  
Digite el valor del límite inferior de x2: -500  
Digite el valor del límite superior de x2: 500  
Digite el número de divisiones requeridas para x1: 4  
Digite el número de divisiones requeridas para x2: 4
```

(Fuente: autor)

Eligiendo ahora la opción de no graficar, se obtienen los resultados presentados en la Figura 57.

Figura 57. Resultados obtenidos en la ejecución con los parámetros de la Figura 56.

```
Elapsed time is 1.076246 seconds.  
  
x1=  
    4.209685646326940e+002  
  
x2=  
    4.209685646326940e+002  
  
El número k de iteraciones realizadas es  
    263  
  
El error obtenido es  
    8.333017831319012e-009  
  
Ejecución terminada exitosamente!!!  
fx >>
```

(Fuente: autor)

### ANEXO C. Ejecución con ecuaciones y parámetros ingresados por el usuario

En caso de requerir el cálculo para ecuaciones diferentes a las presentadas en este informe, éstas deben ser ingresadas previamente en el script correspondiente al número de variables que involucran, y sus parámetros deben ser registrados de forma semejante a como se encuentran las ya presentes, y debajo de la opción case 9, la cual está reservada para este fin, tal como lo señala la Figura 51.

Por ejemplo, si se quiere resolver el sistema de ecuaciones no-lineales:

$$f_1(x_1, x_2) = x_1^2 + x_2^2 - 2 = 0 \quad (140)$$

$$f_2(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 1)^2 = 0 \quad (141)$$

Cuya función objetivo asociada es:

$$f = (x_1^2 + x_2^2 - 2)^2 + ((x_1 - 1)^2 + (x_2 - 1)^2)^2 \quad (142)$$

En el *script* denominado *funcion2D.m* se escriben estas ecuaciones, así como lo señala la Figura 58, teniendo presente que para esta parte del algoritmo, las variables  $x_1$  y  $x_2$  corresponden a *xx* y *yy* respectivamente.

Figura 58. Ejemplo de escritura de ecuaciones en el programa.

```

case 9

    f1=xx^2+yy^2-2;
    f2=(xx-1)^2+(yy-1)^2;
    val=f1^2+f2^2;

end

```

(Fuente: autor)

Dentro del *script* principal (*Algor\_espiral.m*), se establecen los parámetros de ejecución, y esto se hace dentro del subprograma para dos dimensiones, y la localidad de texto correspondiente a *case 9*, como lo muestra la Figura 59.

Figura 59. Ejemplo de establecimiento de parámetros desde el *script* principal.

```

case 9

    if parametros==1
        titulo=('Trayectorias de solución al modelo del usuario');
        titulox='x1';
        tituloy='x2';
        epsilon=10^-5;
        ang=pi/10^30;
        r=0.95;
        kmax=1000;
        lix=-5;
        lsx=5;
        liy=-5;
        lsy=5;
        nx=4;
        ny=4;
    end

```

(Fuente: autor)

La ejecución de este modelo, con los parámetros señalados, escogiendo la opción correspondiente a *dos* variables y luego la opción 9, además seleccionando no graficar, arroja la solución presentada en la Figura 60.

**Figura 60. Resultados obtenidos para el ejemplo de ingreso de un modelo.**

```
Elapsed time is 0.630404 seconds.  
  
x1=  
    0.965331599563671  
  
x2=  
    1.034468814331350  
  
El número k de iteraciones realizadas es  
    140  
  
El error obtenido es  
    9.675470320852818e-006  
  
Ejecución terminada exitosamente!!!  
fx >>
```

*(Fuente: autor)*