

ALGORITMO PREDICTOR-CORRECTOR PARA LA ESTIMACIÓN DE PARÁMETROS EN ECUACIONES DIFERENCIALES PARCIALES

Maria de la Luz López-González¹, Norma Leticia Flores-Martínez²
Hugo Jiménez Islas³

ARTÍCULO DE DIVULGACIÓN

Recibido: 17/01/2023 Aceptado: 20/06/2023 Publicado: 30/06/2023

Resumen.- Las técnicas de optimización por búsqueda directa son usadas extensamente para resolver problemas de optimización de interés de ingeniería bioquímica y química. En este trabajo se presenta el algoritmo predictor-corrector (Luus-Jaakola y Levenberg-Marquardt), con la validación de dos casos de estudio con estimación de parámetros: 1) El primer caso de estudio fue manufacturado a partir de la solución analítica con 4 parámetros conocidos, de los cuales se diseñó un sistema de 3 ecuaciones diferenciales parciales (EDPs). 2) se presenta un caso tomado de la literatura donde se estudia el fenómeno de difusión en un gel de agarosa considerando la comparación de la solución numérica, la solución analítica y una solución que considera una mejora en las condiciones de frontera del modelo propuesto en la literatura. Las pruebas realizadas del algoritmo predictor-corrector muestran la viabilidad del método para ser adaptado en la estimación de parámetros con modelos en ecuaciones diferenciales parciales cuando existe disparidad en la magnitud de estos parámetros. La factibilidad del algoritmo Luus-Jaakola de ser acoplado con el método determinístico Levenberg-Marquardt (LM) es de utilidad para complementar la convergencia en la solución de problemas de estimación de parámetros cuando la selección de una aproximación inicial adecuada es indispensable.

Palabras Clave: Parámetros, estimación, predictor, corrector, ecuaciones.

PREDICTOR-CORRECTOR ALGORITHM FOR PARAMETER ESTIMATION IN PARTIAL DIFFERENTIAL EQUATIONS

Abstract.- Direct search optimization techniques are widely used to solve optimization problems of interest in biochemical and chemical engineering. In this work, the predictor-corrector algorithm (Luus-Jaakola and Levenberg-Marquardt) is presented, with the validation of two case studies with parameter estimation: 1) The first case study was manufactured from the analytical solution with 4 known parameters, from which a system of 3 partial differential equations (PDEs) was designed. 2) A case taken from the literature is presented where the phenomenon of diffusion in an agarose gel is studied considering the comparison of the numerical solution, the analytical solution and a solution that considers an improvement in the boundary conditions of the model proposed in the literature. The tests performed on the predictor-corrector algorithm show the feasibility of the method to be adapted for parameter estimation with models in partial differential equations when there is disparity in the magnitude of these parameters. The feasibility of the Luus-Jaakola algorithm to be coupled with the deterministic Levenberg-Marquardt (LM) method is useful to complement the convergence in the solution of parameter estimation problems when the selection of a suitable initial approximation is indispensable.

Keywords: parameters, estimation, predictor, corrector, equations.

Introducción

Las técnicas de optimización por búsqueda directa, como las redes neuronales, los algoritmos genéticos y la optimización por Luus-Jaakola (LJ), son ampliamente utilizadas en ingeniería bioquímica y química para resolver problemas de optimización. El algoritmo Luus-Jaakola ha demostrado su eficacia al alcanzar el óptimo global en sistemas de ecuaciones algebraicas y diferenciales ordinarias. Sin embargo, en problemas altamente no lineales, encontrar el óptimo global puede ser complicado y no se garantiza con certeza (Luus, 2001). En este contexto, se han realizado comparaciones entre el algoritmo LJ y algoritmos genéticos para analizar su velocidad de convergencia y confiabilidad en la búsqueda del óptimo global (Liao y Luus, 2007).

¹ Profesor de tiempo parcial. Universidad de Guanajuato. Departamento de enfermería y obstetricia.
<https://orcid.org/0000-0002-9612-6219> mdll.lopezgonzalez@ugto.mx

² Profesor de tiempo completo. Universidad Politécnica de Guanajuato. Departamento de Ingeniería Agroindustrial.
<https://orcid.org/0000-0002-4415-7224> nlflores@upgto.edu.mx

³ Profesor de tiempo completo. Tecnológico Nacional de México en Celaya. Departamento de Ingeniería Bioquímica.
<https://orcid.org/0000-0002-1084-5520> hugo.jimenez@itcelaya.edu.mx (**Autor corresponsal**).

Desde su presentación original en 1973, Rein Luus ha continuado mejorando el algoritmo LJ, enfocándose en su implementación y convergencia para resolver ecuaciones diferenciales ordinarias no lineales en problemas de ingeniería química, como reactores de tanque agitado continuo, procesos de fermentación y "cracking catalítico" (Luus, 2001; Liao y Luus, 2007). Algunos autores han abordado limitaciones del algoritmo y propuesto mejoras, como el Improved New Luus-Jaakola (INLJ) que utiliza la búsqueda aleatoria adaptativa con el método de identificación y diversificación (RasID) (Li D. *et al.*, 2016). Esta mejora es especialmente útil cuando se consideran restricciones en el sistema de estudio y la generación de valores aleatorios es esencial para la solución del problema.

La eficiencia y precisión del algoritmo Luus-Jaakola ha sido motivo de debate, ya que se considera un método heurístico, generando comparaciones con métodos determinísticos como el método Newton-Raphson (Wang y Luus, 1978; Spaans y Luus, 1998; Luus, 2001; Gümus y Floudas, 2001). Sin embargo, su principal ventaja radica en la capacidad para encontrar una solución óptima en el sistema de estudio independientemente de la aproximación inicial, a diferencia de los métodos determinísticos que requieren una aproximación cercana para lograr la convergencia. Además, su facilidad de implementación, con solo 125 líneas de código y 4 condiciones de operación, lo hace atractivo en términos de esfuerzo computacional (Liao y Luus, 2007).

Aunque el algoritmo Luus-Jaakola es eficaz en la solución de sistemas de ecuaciones, enfrenta limitaciones al abordar la complejidad de los problemas de optimización del mundo real para encontrar el mínimo global. Por esta razón, se ha explorado el uso de algoritmos combinados/híbridos que combinan métodos heurísticos y determinísticos en diversas áreas de la ingeniería. La viabilidad de utilizar el algoritmo Luus-Jaakola como estrategia de búsqueda directa previa al uso de un método determinístico radica en garantizar la convergencia mediante la predicción cercana con el algoritmo LJ y luego corregir con el método determinístico para estimar parámetros (Ramírez-Martínez, 2008; Mendes-Platt, 2014; Li N. *et al.*, 2019; de Jesús Rubio, 2020; Oliveros-Muñoz *et al.*, 2021; Pal y Kaushik, 2023; Meena *et al.*, 2023).

En este trabajo se introduce una estrategia de estimación de parámetros basada en el algoritmo predictor-corrector para abordar problemas gobernados por ecuaciones diferenciales parciales no lineales. En primer lugar, el algoritmo predice una aproximación inicial cercana a la solución del sistema utilizando el algoritmo Luus-Jaakola. Luego, esta aproximación se refina mediante el método determinístico Levenberg-Marquardt (Marquardt, 1963; Ji *et al.*, 2021; Luo *et al.*, 2021; Wang *et al.*, 2022).

Esta estrategia puede aplicarse de manera efectiva en diversos problemas de ingeniería que involucren la búsqueda de mínimos globales o relativos durante el proceso de optimización, ya sea en sistemas no lineales regidos por ecuaciones algebraicas, diferenciales ordinarias o ecuaciones diferenciales parciales no lineales.

Metodología

Implementación del algoritmo Luus-Jaakola

La implementación de algoritmo Luus-Jaakola (Luus y Jaakola, 1973), contempla los siguientes puntos. 1) El algoritmo busca maximizar o minimizar la función $P = f(x_1, x_2, \dots, x_n)$, según sea el caso, sujeta a las restricciones: $g_i(x_1, x_2, \dots, x_n) \leq 0$; $i = 1, 2, \dots, m$, $h_j(x_1, x_2, \dots, x_n) \geq 0$; $j = 1, 2, \dots, r$ y $q_k(x_1, x_2, \dots, x_n) = 0$; $k = 1, 2, \dots, s$. 2) El algoritmo genera números aleatorios para cada conjunto $\{x_1, x_2, \dots, x_n\}$ y se evalúa cada conjunto con respecto a las restricciones $g_i \leq 0$ y $h_j \geq 0$. De esto se podrán tener dos resultados: un conjunto de valores $\{x_1, x_2, \dots, x_n\}$ que satisfagan las restricciones y otro grupo de valores que no cumpla las restricciones. 3) Con el conjunto de valores que cumplen con las restricciones se evalúa la función $P = f(x_1, x_2, \dots, x_n)$ y se almacena el valor que logre maximizar o minimizar, según sea el caso.

El algoritmo Luus-Jaakola tiene la finalidad de minimizar la suma de residuos cuadrática (S); considerando el sistema de ecuaciones, $\phi_{mod(t,x;xp_n)}$, y un conjunto de datos experimentales, ϕ_{exp} , de tal manera que la función objetivo (F) se define en la Ecuación 1.

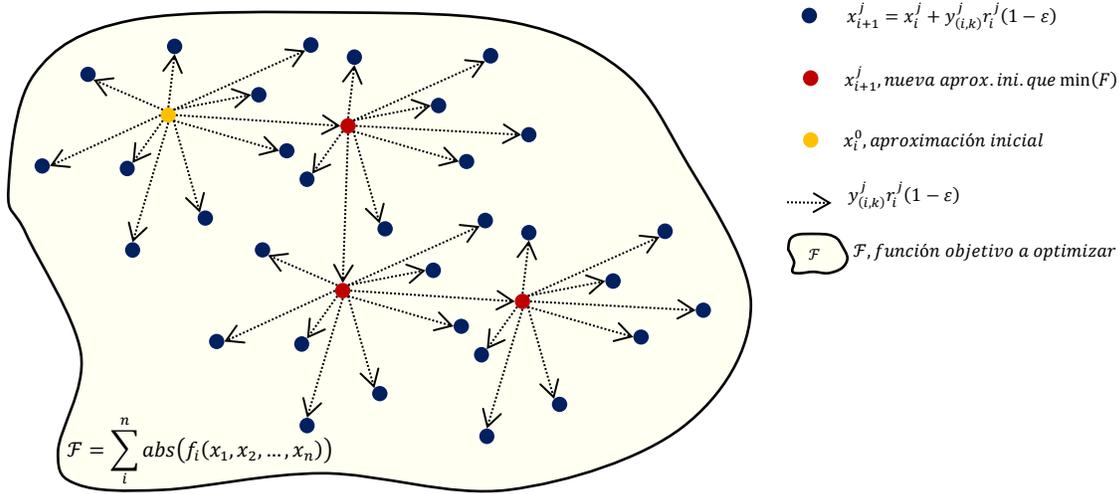
$$\mathcal{F} = \sum_j^m \sum_i^l (\phi_{exp} - \phi_{mod(t,x;xp_n)})^2 \quad (1)$$

Donde $\phi_{mod(t,x;xp_n)}$ es la ecuación j, x_{p_n} son los parámetros por estimar, m el número de ecuaciones, l el número de datos experimentales y n variables independientes. Una vez identificada la función objetivo, se toman los valores iniciales para cada parámetro por estimar ($x_1^0, x_2^0, \dots, x_n^0$) y un valor inicial para el rango de búsqueda de cada parámetro ($r_1^0, r_2^0, \dots, r_n^0$). Posteriormente, se genera un conjunto de valores aleatorios ($Y_{i,k}$) con el fin de generar un nuevo valor (Ecuación 2);

$$x_i^j = x_i^{j-1} + y_{ki} r_i^{j-1} \tag{2}$$

Donde $j = 1, 2, \dots$, iteraciones máximas, $i = 1, 2, \dots, n$ y $k = 1, 2, \dots$, número máximo de datos aleatorios. Del conjunto \hat{x} de parámetros estimados, los valores $[x_1, x_2, \dots, x_n]$ que minimicen la función objetivo (\mathcal{F}) en cada iteración j , serán asignados como conjunto inicial $[x_1^0, x_2^0, \dots, x_n^0]$ en la iteración $j + 1$. Durante cada iteración j , el rango es reducido con respecto al factor de contracción ($r_i^j = (1 - \varepsilon)r_i^{j-1}, \varepsilon > 0$), hasta llegar al máximo número de iteraciones (Salcedo, 1992; Spaans y Luus, 1998). La Figura 1 muestra una representación esquemática del funcionamiento del algoritmo Luus-Jaakola.

Figura 1. Principio del algoritmo Luus-Jaakola.



Nota. Descripción del funcionamiento del algoritmo Luus-Jaakola a partir de una aproximación inicial. Fuente: Elaboración propia.

Implementación del algoritmo Levenberg-Marquardt

Con la aproximación del vector inicial dada por el algoritmo Luus-Jaakola, se determinará la aproximación numérica del vector de parámetros con el método determinístico de Levenberg-Marquardt (Marquardt, 1963; Ji *et al.*, 2021; Luo *et al.*, 2021; Wang *et al.* 2022). El algoritmo de Levenberg-Marquardt está definido por:

$$Xp_n^{i+1} = Xp_n^i - [H_m + \lambda \text{diag}(H_m)]^{-1} \nabla S \tag{3}$$

Donde ∇S es el gradiente de la función $S = \sum_j^m \sum_i^l (\phi_{exp} - \phi_{mod}(t, x; Xp_n))^2$ con respecto a cada parámetro por estimar;

$$\nabla S = \left[\frac{\partial S}{\partial Xp_1} \quad \frac{\partial S}{\partial Xp_2} \quad \dots \quad \frac{\partial S}{\partial Xp_n} \right]^T \tag{4}$$

H_m es la matriz Hessiana $n \times n$, donde n corresponde al número de parámetros a estimar, se calcula:

$$\text{Hessiano} = \begin{bmatrix} \frac{\partial^2 S}{\partial Xp_1^2} & \frac{\partial^2 S}{\partial Xp_1 \partial Xp_2} & \frac{\partial^2 S}{\partial Xp_1 \partial Xp_n} \\ \frac{\partial^2 S}{\partial Xp_2 \partial Xp_1} & \frac{\partial^2 S}{\partial Xp_2^2} & \frac{\partial^2 S}{\partial Xp_2 \partial Xp_n} \\ \frac{\partial^2 S}{\partial Xp_n \partial Xp_1} & \frac{\partial^2 S}{\partial Xp_n \partial Xp_2} & \frac{\partial^2 S}{\partial Xp_n^2} \end{bmatrix} \tag{5}$$

$\lambda \cdot \text{diag}(H_m)$ Corresponde al parámetro de Levenberg-Marquardt LAMBDA (λ), que multiplica a la diagonal principal de la matriz Hessiana y está definida por;

$$\lambda \cdot \text{diag}(\text{Hessiano}) = \begin{bmatrix} \lambda \frac{\partial^2 S}{\partial Xp_1^2} & 0 & 0 \\ 0 & \lambda \frac{\partial^2 S}{\partial Xp_2^2} & 0 \\ 0 & 0 & \lambda \frac{\partial^2 S}{\partial Xp_n^2} \end{bmatrix} \tag{6}$$

Para evitar el cálculo de la matriz inversa $[H_m + \lambda \text{diag}(H_m)]^{-1}$, se define que $\frac{\nabla S}{[H_m + \lambda \text{diag}(H_m)]} = b$. De tal forma que sí $\nabla S [H_m + \lambda \text{diag}(H_m)]^{-1} = b$, puede multiplicarse por $[H_m + \lambda \text{diag}(H_m)]$, y obtener $\nabla S [H_m +$

$\lambda \text{diag}(H_m)]^{-1}[H_m + \lambda \text{diag}(H_m)] = b[H_m + \lambda \text{diag}(H_m)]$. Considerando que $[H_m + \lambda \text{diag}(H_m)]^{-1}[H_m + \lambda \text{diag}(H_m)] = [I]$, se reescribe el sistema de ecuaciones como $\{VS\} = \{b\}[H_m + \lambda \text{diag}(H_m)]$. Este sistema de ecuaciones se resuelve empleando el método de factorización LU, para obtener el vector $\{b\}$. Por lo tanto, la aproximación con el método Levenberg-Marquardt es definida en la Ecuación 7.

$$\{Xp_n^{i+1}\} = \{Xp_n^i\} - \{b\} \quad (7)$$

Durante la implementación del algoritmo, el parámetro de Levenberg-Marquardt LAMBDA (λ) tiene un valor inicial de 1.0 y es modificado con respecto al valor de la suma de residuos cuadrática (S). Si $S^{i+1} < S^i$, disminuye el valor de LAMBDA (λ), en caso contrario LAMBDA (λ) aumenta con respecto a un factor de 1.1. Para saber si se ha alcanzado la convergencia, se compara el valor absoluto del gradiente con la tolerancia. Si el valor absoluto del gradiente de S, ($abs(\nabla S)$), es menor a 1.0E-05, de tal manera que si $(abs(\frac{\partial S}{\partial Xp_1}) \leq TOL, abs(\frac{\partial S}{\partial Xp_2}) \leq TOL, \dots, abs(\frac{\partial S}{\partial Xp_n}) \leq TOL)$; se considera que el método cumple con la tolerancia y se detienen las iteraciones.

Discretización de la coordenada espacial

Después de identificar el modelo y los parámetros a estimar, es necesario adimensionalizar el modelo antes de proceder a su discretización en su coordenada espacial utilizando el método de diferencias finitas. La discretización tiene como objetivo convertir la ecuación diferencial parcial en un sistema de ecuaciones diferenciales ordinarias. El número de ecuaciones diferenciales ordinarias resultantes dependerá del número de nodos definidos durante el proceso de discretización.

$$\frac{\partial \phi}{\partial t} = A_0 \left[\frac{\partial^2 \phi}{\partial x^2} + \frac{A_1}{x} \frac{\partial \phi}{\partial x} \right] + \frac{A_2 \phi}{A_3 + A_4 \phi} \rightarrow \text{discretización} \rightarrow \frac{d\phi}{dt} \quad (8)$$

Para discretizar la ecuación diferencial parcial se utilizan las diferencias finitas centrales para los nodos interiores.

$$\frac{d\phi_i}{dt} = A_0 \left[\frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{dx^2} + \frac{A_1}{x_i} \left(\frac{\phi_{i+1} - \phi_{i-1}}{2dx} \right) \right] + \frac{A_2 \phi_i}{A_3 + A_4 \phi_i} \quad (9)$$

$i = 2, 3, \dots, (\text{nodos}-1)$

La condición de frontera 1 representa al nodo 1 y es discretizada con diferencias finitas hacia adelante.

$$X_1 = 0, C_1 \left(\frac{\phi_2 - \phi_1}{dx} \right) + C_2 \phi_1 = C_3 \quad (10)$$

La condición de frontera 2 representa al último nodo y es discretizada con diferencias finitas hacia atrás.

$$X_{\text{nodos}} = 1, C_4 \left(\frac{\phi_{\text{nodos}} - \phi_{(\text{nodos}-1)}}{dx} \right) + C_5 \phi_{\text{nodos}} = C_6 \quad (11)$$

El valor de dx, es definido por $dx = \frac{1}{\text{nodos}-1}$.

Resultados

Estimación de 4 parámetros en un caso de estudio manufacturado

El caso de estudio 1 (caso manufacturado) está compuesto por tres ecuaciones diferenciales parciales y se plantea a partir de la solución analítica (Ecuación 12, Ecuación 13 y Ecuación 14)

$$T_1 = X_{p1} \sin(x) + \frac{2x}{t+X_{p2}} \quad (12)$$

$$T_2 = X_{p3} x e^{-t} \quad (13)$$

$$T_3 = x^3 + \frac{2x}{t+X_{p4}} \quad (14)$$

De la solución analítica, se plantean los parámetros x_{p1} , x_{p2} , x_{p3} y x_{p4} con vector solución $x_{pn}^T = [3.0, 1.0, 2.0, 0.5]^T$. Se diseñó un sistema de tres ecuaciones diferenciales, cada una con sus correspondientes condiciones de frontera y condición inicial con una estructura distinta por ecuación diferencial;

$$\frac{\partial T_1}{\partial t} = 2 \left(\frac{\partial^2 T_1}{\partial x^2} \right) + \left(\frac{\partial T_2}{\partial x} \right)^2 - \frac{2x}{(t+X_{p2})^2} + 2X_{p1} \sin(x) - X_{p3} e^{-t} \quad (15)$$

Donde la condición inicial es en $t = 0$, $T_1 = X_{p1} \sin(x) + \frac{2x}{X_{p2}}$. La condición de frontera 1 en $x = 0$, $T_1 = 0$ y la condición de frontera 2 en $x = 1$, $T_1 = 3 \sin(1) + \frac{2x}{t+X_{p2}}$.

$$\frac{\partial T_2}{\partial t} = 3 \left(\frac{\partial^2 T_2}{\partial x^2} \right) - X_{p3} x e^{-t} \quad (16)$$

Donde la condición inicial en $t = 0, T_2 = X_{p3}x$. La condición de frontera 1 en $x = 0, T_2 = 0$ y la condición de frontera 2 en $x = 1, T_2 = 2e^{-t}$.

$$\frac{\partial T_3}{\partial t} = \frac{\partial T_1}{\partial x} + \frac{\partial^2 T_3}{\partial x^2} - \frac{2x}{(t+X_{p4})^2} - X_{p1} \cos(x) - \frac{2}{t+X_{p2}} - 6x \quad (17)$$

Donde la condición inicial es en $t = 0, T_3 = x^3 + \frac{2x}{x_{p2}}$. La condición de frontera 1 en $x = 0, T_3 = 0$ y la condición de frontera 2 en $x = 1, T_3 = 1 + \frac{2}{t+X_{p4}}$.

Los datos experimentales manufacturados para la estimación de parámetros se obtienen de la solución analítica, donde se establece $x = 0.5, x_{pn}^T = [3.0, 1.0, 2.0, 0.5]^T$, abarcando un rango de tiempo de 0 a 1. Para resolver el sistema de ecuaciones en derivadas parciales (EDPs), se realizó una discretización de la coordenada espacial de cada EDP, junto con las condiciones de frontera correspondientes, generando tres sistemas de ecuaciones en derivadas ordinarias (EDOs). Estos sistemas están conformados por mallas de 15, 21 y 35 nodos, respectivamente. Se utilizó el método de Runge-Kutta de 4to orden para resolver dichos sistemas. Además, se incorpora el algoritmo Luus-Jaakola para ajustar el factor de contracción al modificar el número de nodos durante la estimación de parámetros.

En este caso, se puede apreciar que el número de iteraciones impacta la precisión del algoritmo en la aproximación obtenida mediante el uso del método Luus-Jaakola. Es fundamental considerar que, al aumentar el número de nodos en la solución numérica de las ecuaciones diferenciales parciales con diferencias finitas, también se incrementa la cantidad de ecuaciones a resolver. Del mismo modo, si se aumenta el número de iteraciones durante la ejecución del algoritmo Luus-Jaakola, el tiempo de cómputo aumenta, tal como se muestra en la Tabla 1.

Tabla 1. Comparación de diferentes factores de contracción(ε) del rango de búsqueda del algoritmo Luus-Jaakola.

Nodos	Vector de parámetros encontrado	Suma de residuos cuadráticos (S)	Tiempo de cómputo (s)
15	[2.25796, 0.61267, 1.16962, 0.51642] ^{a1}	3.04299	54.91
	[2.85905, 0.89233, 1.98194, 0.50124] ^{a2}	0.02591	54.96
	[2.99448, 0.99971, 1.98668, 0.50146] ^{a3}	0.00027	54.87
	[2.25795, 0.61267, 1.16960, 0.51643] ^{b1}	3.04312	27.65
	[2.84861, 0.88345, 1.98602, 0.49980] ^{b2}	0.03068	27.59
	[2.96929, 0.98880, 1.97909, 0.50283] ^{b3}	0.00223	27.51
21	[2.25801, 0.61280, 1.16964, 0.51659] ^{a1}	2.98875	2054.00
	[2.93414, 0.95307, 1.97235, 0.50324] ^{a2}	0.00555	2387.00
	[3.00096, 1.00956, 1.97129, 0.50273] ^{a3}	0.00130	1381.37
	[2.25800, 0.61281, 1.16962, 0.51660] ^{b1}	2.98888	673.00
	[2.92489, 0.94595, 1.97954, 0.50200] ^{b2}	0.00737	691.18
	[2.98567, 0.98985, 1.97134, 0.50100] ^{b3}	0.00200	691.12
35	[2.25247, 0.61618, 1.17278, 0.52120] ^{a1}	2.78752	1936.68
	[2.96340, 0.99266, 1.91425, 0.50918] ^{a2}	0.01140	1924.10
	[2.98273, 1.00899, 1.91535, 0.50900] ^{a3}	0.01134	1964.54
	[2.25246, 0.61619, 1.17277, 0.52121] ^{b1}	2.78764	945.29
	[2.95631, 0.98824, 1.91714, 0.50943] ^{b2}	0.01164	1226.00
	[2.93932, 0.97236, 1.92994, 0.50622] ^{b3}	0.01417	925.93

Nota: Se presenta el Caso de estudio 1 utilizando diferente número de malla para la discretización de la coordenada espacial. Condiciones de operación del algoritmo Luus-Jaakola para obtener cada vector de parámetros: ^{a1}100 iteraciones, 20 valores aleatorios por iteración, h=1E-04 y ε=0.80. ^{a2}100 iteraciones, 20 valores aleatorios por iteración, h=1E-04 y ε=0.90. ^{a3}100 iteraciones, 20 valores aleatorios por iteración, h=1E-04 y ε=0.95. ^{b1}50 iteraciones, 20 valores aleatorios por iteración, h=1E-04 y ε=0.80. ^{b2}50 iteraciones, 20 valores aleatorios por iteración, h=1E-04 y ε=0.80. ^{b3}50 iteraciones, 20 valores aleatorios por iteración, h=1E-04 y ε=0.80.

A continuación, se presentan los resultados de la estimación de parámetros mediante el algoritmo predictor-corrector, utilizando diferentes mallas para la discretización de la coordenada espacial y sus correspondientes condiciones de operación (ver Tabla 2). En este caso, se observa que la estimación de parámetros con una malla de 35 nodos obtuvo la menor suma de residuos cuadráticos (3.24000 E-08). Sin embargo, también requirió un mayor tiempo de cómputo debido a que se deben resolver simultáneamente 99 ecuaciones diferenciales ordinarias y 6 ecuaciones algebraicas en cada paso de integración utilizando el método de Runge-Kutta de 4to orden.

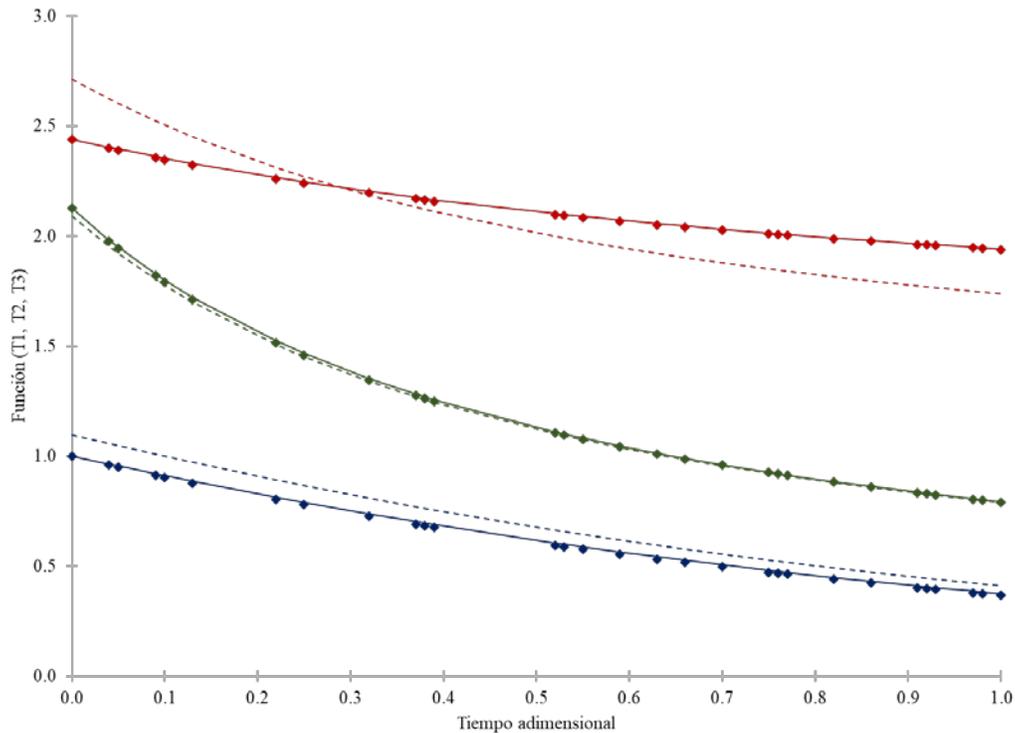
Tabla 2. Resultados de la estimación de parámetros utilizando el algoritmo predictor-corrector para el Caso de estudio 2

Nodos	Estimación de parámetros con el algoritmo predictor-corrector (LJ-LM)	Suma de residuos cuadráticos (S) con el algoritmo predictor-corrector (LJ-LM))	Tiempo de cómputo(s)
7	[2.96524, 0.97154, 1.98258, 0.50055] ^a	2.18579E-03	20.46
15	[2.99937, 0.99998, 1.99861, 0.50005] ^b	2.29720E-06	75.90
21	[2.99969, 0.99996, 1.99943, 0.50001] ^c	4.20800E-07	416.56
35	[2.99989, 0.99994, 1.99995, 0.49998] ^d	3.24000E-08	2129.95

Nota: Condiciones de operación del algoritmo predictor-corrector: ^aLuus-Jaakola (50 iteraciones, 20 valores aleatorios por iteración, $h=1E-04$ y $\epsilon=0.95$) y Levenberg-Marquardt ($\lambda_{ini}=1.0$, No. De iteraciones=22 y $h=1E-04$), ^bLuus-Jaakola (50 iteraciones, 20 valores aleatorios por iteración, $h=1E-04$ y $\epsilon=0.95$) y Levenberg-Marquardt ($\lambda_{ini}=1.0$, No. De iteraciones=22 y $h=1E-05$), ^cLuus-Jaakola (50 iteraciones, 20 valores aleatorios por iteración, $h=1E-04$ y $\epsilon=0.95$) y Levenberg-Marquardt ($\lambda_{ini}=1.0$, No. De iteraciones=22 y $h=2E-06$) y ^dLuus-Jaakola (50 iteraciones, 20 valores aleatorios por iteración, $h=1E-04$ y $\epsilon=0.95$) y Levenberg-Marquardt ($\lambda_{ini}=1.0$, No. De iteraciones=22 y $h=1E-06$).

A continuación, en la Figura 2 se observa la comparación de la estimación de los parámetros para una malla de 35 nodos, se observa el modelo con la primera estimación de parámetros (LJ) y su segunda estimación (LM).

Figura 2. Estimación de parámetros con el algoritmo predictor-corrector para una malla de 35 nodos



Nota: Resultados de la estimación de parámetros del Caso 1, se presenta la estimación con Luus-Jaakola y la refinación de la solución con Levenberg-Marquardt al comparar con los datos experimentales manufacturados. En color rojo: línea sólida (LM), línea punteada (LJ), rombos(datos experimentales manufacturados para T_1); en color azul: línea sólida (LM), línea punteada (LJ), rombos(datos experimentales manufacturados para T_2) y en color verde: línea sólida (LM), línea punteada (LJ), rombos(datos experimentales manufacturados para T_3) Fuente: Elaboración propia.

En los casos de estudio manufacturados, es crucial utilizar un algoritmo predictor-corrector, como el algoritmo Luus-Jaakola, para asegurar la convergencia del método de Levenberg-Marquardt. Esto se debe a que una aproximación inicial precisa y cercana a los valores reales de los parámetros es fundamental para que el método converja exitosamente. Si la aproximación inicial no es adecuada, no hay garantía de que el método converja. Al estimar los cuatro parámetros conocidos, el algoritmo Luus-Jaakola proporciona una aproximación inicial adecuada, lo que garantiza la convergencia deseada. Por otro lado, si se utiliza únicamente el método de Levenberg-Marquardt con una aproximación inicial de 0 o 1 para cada parámetro, el método diverge. En resumen, la implementación del algoritmo predictor-corrector Luus-Jaakola es esencial en los casos de estudio manufacturados, ya que proporciona una

aproximación inicial precisa que asegura el éxito de la convergencia del método de Levenberg-Marquardt al estimar los parámetros conocidos, evitando problemas de convergencia causados por aproximaciones inadecuadas.

Difusión controlada de un péptido antimicrobiano para películas de biopolímeros

Se propone como caso de estudio 2 para validar el algoritmo predictor-corrector el modelo propuesto por Sebti *et al.* (2003) para determinar el coeficiente de difusión aparente de una solución de nisina en contacto con un gel de agarosa empleando la segunda ley de Fick. El procedimiento experimental del fenómeno de difusión estudiado se realizó con una concentración inicial de nisina en la solución de 376 µg/ mL, con una temperatura de 22.3°C y el gel de agarosa fue cortado en rebanadas delgadas de 1 mm para medir la concentración de nisina a los 5.9 días del experimento.

Sebti *et al.* (2003), proponen que modelo para la difusividad en el gel de agarosa obedece la segunda Ley de Fick dentro de un cilindro semi-infinito, donde el coeficiente de difusividad (D_{eff}) es constante (Ecuaciones 18, 19 y 20).

$$\frac{\partial C_A}{\partial t} = D_{eff} \frac{\partial^2 C_A}{\partial z^2} \quad (18)$$

$$CF1: z = 0, \frac{\partial C_A}{\partial z} = 0 \quad (19)$$

$$CF2: z \rightarrow \infty, C_A = C_A^* = 0 \quad (20)$$

Este problema fue resuelto de forma analítica con la transformada de Laplace (Sebti *et al.*, 2013), obteniendo la siguiente solución:

$$W_A = \frac{C_0^* - C_A}{C_0^* - C_{init}} = erf\left(\frac{z}{2\sqrt{D_{eff}t}}\right) \quad (21)$$

La validación de este caso de estudio se realizó a partir de los datos experimentales reportados por Sebti *et al.* (2003) del experimento E, los cuales se obtuvieron con el software Engauge Digitizer. La solución analítica fue replicada para realizar la estimación de parámetros con el algoritmo predictor-corrector, considerando la suma de residuos:

$$\min S = \min \sum_{i=1}^{datos} \left(W_{A(experimental)} - erf\left(\frac{z}{2\sqrt{D_{eff}t}}\right) \right)^2 \quad (22)$$

De forma complementaria, se propone un modelo adicional (Flores-Martínez *et al.*, 2018). Se considera que durante el experimento de difusión la solución de nisina fue agitada lentamente, por lo tanto, se presume la existencia de resistencia interfacial y se propone una modificación al modelo de propuesto originalmente por Sebti *et al.* (2003). La mejora al modelo consiste en modificar la condición de frontera donde el gel de agarosa está en contacto con la solución de nisina (CF2). De tal forma que el modelo propuesto para este caso de estudio es;

$$\frac{\partial C_A}{\partial t} = D_{eff} \frac{\partial^2 C_A}{\partial z^2} \quad (23)$$

$$CF1: z = 0, \frac{\partial C_A}{\partial z} = 0 \quad (24)$$

$$CF2: z = H \text{ es } -D_{eff} \frac{\partial C_A}{\partial z} = k_C(C_A - C_{SOL}) . \quad (25)$$

Este modelo se resuelve empleando el método de separación de variables, resultando la siguiente solución analítica:

$$C_A = \left(2 \sum_{n=1}^{\infty} \frac{e^{-\lambda_n^2 D_{eff} t}}{\lambda_n + \sin \lambda_n \cos \lambda_n} \sin \lambda_n \cos\left(\lambda_n * \left(1 - \frac{z}{H}\right)\right) \right) (C_A^0 - \bar{C}_{Asol}) + \bar{C}_{Asol} \quad (26)$$

Con la función trascendente:

$$\lambda_n \text{sen} \lambda_n - \text{Bimcos} \lambda_n = 0 \quad n = 1,2,3,4,5, \dots \quad (27)$$

Este modelo fue empleado para validar la solución numérica obtenida al emplear el algoritmo predictor-corrector a partir de la Ecuación 22 con sus respectivas condiciones de frontera y condición inicial. El modelo con resistencia interfacial fue resuelto al discretizar la coordenada espacial (z), para posteriormente resolver el sistema de ecuaciones diferenciales ordinarias con el método de Runge-Kutta de 4to. Orden. Se discretizó el modelo con una malla de 51 nodos, para este caso de estudio en particular al conocer aproximadamente el orden de magnitud de los parámetros a estimar se agrega un factor de escala adicional al algoritmo Luus-Jaakola, de tal forma que la estimación con $Xp_i^j = (Xp_i^{j-1} + y_{ki}r_i^{j-1}) * \text{factor de escala}$. Tanto el modelo propuesto por Sebti *et al.* (2003) y el modelo sugerido por

Flores-Martínez *et al.* (2018) fueron utilizados para evaluar el algoritmo predictor-corrector al minimizar la suma de residuos cuadráticos (Tabla 3).

Tabla 3. Resultados de la estimación de parámetros con el algoritmo predictor-corrector

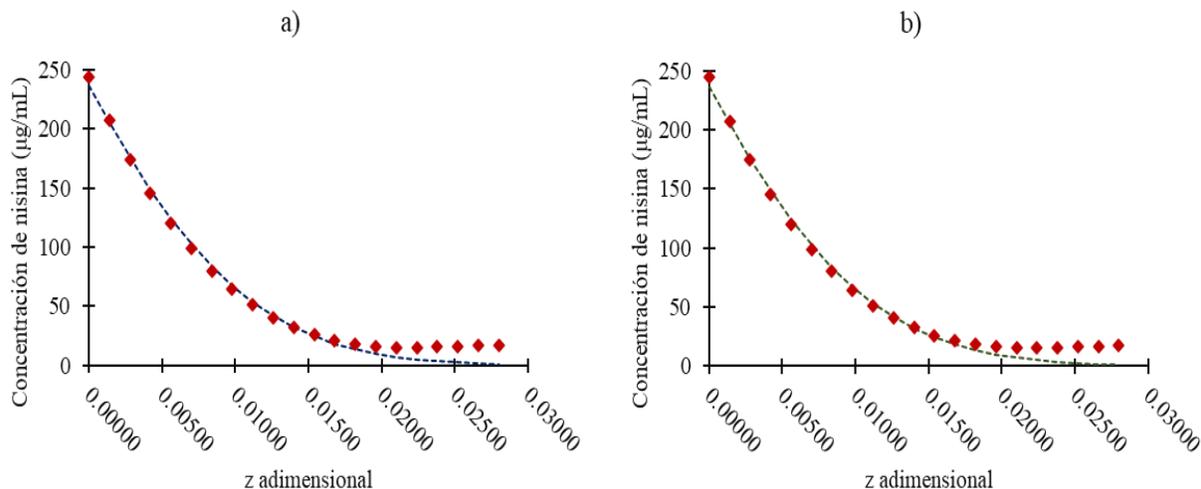
	Solución analítica con el modelo de Sebti <i>et al.</i> (2003) ^a	Solución analítica con el modelo de Flores-Martínez <i>et al.</i> (2018) ^b	Solución numérica con el modelo de Flores-Martínez <i>et al.</i> (2018) ^c
$D_{eff} \left(\frac{m^2}{s}\right)$	8.34183×10^{-11}	1.08731×10^{-10}	1.08731×10^{-10}
$K_c \left(\frac{m}{s}\right)$	-----	1.80632×10^{-8}	1.80632×10^{-8}
$C_A^0 \left(\frac{\mu g}{mL}\right)$	230.8	236.65	236.62
Error promedio (%)	33.28	30.89	27.87
Suma de residuos cuadráticos (S)	2005.89577	1923.42294	1225.95416
Coefficiente de correlación (R^2)	0.98968	0.98943	0.996398
Tiempo de cómputo (s)	0.31	-----	1533.64

Nota: Condiciones de operación del algoritmo predictor-corrector: ^aLuus-Jaakola (200 iteraciones, 100 valores aleatorios por iteración, y $\epsilon=0.90$) y Levenberg-Marquardt ($\lambda_{ini}=1.0$, No. De iteraciones=1000), ^bLuus-Jaakola (100 iteraciones, 200 valores aleatorios por iteración y $\epsilon=0.90$) y Levenberg-Marquardt ($\lambda_{ini}=1.0$, No. De iteraciones=1000), ^cLuus-Jaakola (50 iteraciones, 50 valores aleatorios por iteración, factor1=1E-08, factor=1E-09, h=1E-05 y $\epsilon=0.90$) y Levenberg-Marquardt ($\lambda_{ini}=1.0$, No. De iteraciones=100 y h=1E-06).

Los resultados del algoritmo predictor-corrector muestran una mejora en la estimación de parámetros, reduciendo el error promedio de un 33.28% utilizando la solución analítica basada en la transformada de Laplace (Sebti *et al.*, 2003) a un 30.89% utilizando la solución analítica por separación de variables (Flores-Martínez *et al.*, 2018). Además, se logra una estimación numérica con un error promedio del 27.55% utilizando las Ecuaciones 23, 24 y 25.

Es importante destacar que el modelo propuesto por Flores-Martínez *et al.* (2018) presenta una menor suma de residuos cuadráticos, lo que indica un mejor ajuste a los datos experimentales (Figura 3) en comparación con el modelo original propuesto por Sebti *et al.* (2003). Esto se debe a que el modelo de Flores-Martínez *et al.* (2018) incorpora la presencia de resistencia interfacial en la frontera donde la solución de nisina y el gel de agarosa están en contacto, lo cual es consecuencia de la agitación lenta de la solución de nisina.

Figura 3. Ajuste del modelo con los datos experimentales. a) Solución analítica propuesta por Sebti *et al.* (2003). b) Solución numérica obtenida de la estimación de parámetros con el algoritmo predictor-corrector del modelo propuesto por Flores-Martínez *et al.* (2018)



Nota: Datos experimentales adaptados de Sebti *et al.* (2003). Fuente: Elaboración propia.

Flores-Martínez *et al.* (2018) emplean el método de Levenberg-Marquardt para la estimación de parámetros obteniendo un coeficiente de correlación del modelo con los datos experimentales de 0.98, la estimación de parámetros partir de la solución numérica del modelo con el algoritmo predictor-corrector tiene un coeficiente de correlación de 0.99. Lo

anterior, permite validar que la estimación de parámetros puede realizarse sin conocer la solución analítica de un sistema gobernado por ecuaciones diferenciales parciales del tipo parabólico, esto sucede cuando existe la dificultad de encontrar la solución analítica de dicho sistema.

Conclusiones

El propósito de implementar un método predictor-corrector es obtener una aproximación inicial cercana a la solución al utilizar el método de búsqueda local de Levenberg-Marquardt. En este sentido, el algoritmo Luus-Jaakola, al ser un método de búsqueda directa, puede generar un vector de parámetros adecuado que luego será refinado por el método de Levenberg-Marquardt.

Las pruebas realizadas con el algoritmo Luus-Jaakola demuestran su viabilidad para adaptarse a la estimación de parámetros en modelos que involucran ecuaciones diferenciales parciales, especialmente en casos donde exista una disparidad en la magnitud de dichos parámetros. Esto es relevante, por ejemplo, al estimar coeficientes de difusividad y la constante de transferencia de masa en sistemas gobernados por la segunda Ley de Fick. El algoritmo Luus-Jaakola es capaz de proporcionar un vector inicial de aproximación apropiado. Además, se observa que el algoritmo tiene la capacidad de encontrar múltiples mínimos locales al modificar la aproximación inicial, siempre que la topología de la función lo permita.

De esta manera, el método predictor-corrector propuesto garantiza la identificación de al menos un mínimo local al estimar parámetros en sistemas de ecuaciones diferenciales parciales parabólicas, considerando sus respectivas condiciones de contorno e inicial. Esta misma estrategia puede extrapolarse a la estimación de parámetros en sistemas de ecuaciones algebraicas, sistemas de ecuaciones diferenciales ordinarias con condiciones iniciales, y sistemas de ecuaciones diferenciales parciales elípticas o hiperbólicas, junto con sus condiciones de contorno e inicial correspondientes. Por último, se agradece el apoyo del Tecnológico Nacional de México (TecNM) con los proyectos 13754.22-P y 17152.23-P.

Referencias bibliográficas

- De Jesús Rubio, J. (2020). Stability analysis of the modified Levenberg–Marquardt algorithm for the artificial neural network training. *IEEE transactions on neural networks and learning systems*, 32(8), 3510-3524. DOI: 10.1109/TNNLS.2020.3015200.
- Flores-Martínez, N. L., Pérez-Pérez, M. C. I., Oliveros-Muñoz, J. M., López-González, M. L., y Jiménez-Islas, H. (2018). Estimation of diffusion coefficients of essential oil of *Pimenta dioica* in edible films formulated with Aloe vera and gelatin, using Levenberg-Marquardt method. *Revista Mexicana de Ingeniería Química*, 17(2), 485-506. DOI: <https://doi.org/10.24275/uam/izt/dcbi/revmexingquim/2018v17n2/flores>
- Gümüş, Z. H., y Floudas, C. A. (2001). Global optimization of nonlinear bilevel programming problems. *Journal of Global Optimization*, 20(1), 1-31. DOI: <https://doi.org/10.1023/A:1011268113791>
- Ji, Y., Kang, Z., y Liu, X. (2021). The data filtering based multiple-stage Levenberg–Marquardt algorithm for Hammerstein nonlinear systems. *International Journal of Robust and Nonlinear Control*, 31(15), 7007-7025. DOI: <https://doi.org/10.1002/rnc.5675>
- Li, D., Lu, T., Liu, L. y Jin, Q. (2016). Fractional modeling of heating process using improved NLJ algorithm. *Proceedings of the 35th Chinese Control Conference*, pp. 10447-10452. DOI: <https://doi.org/10.1109/chicc.2016.7555011>
- Li, N., Yang, H., y Mu, A. (2019). Improved Grey Particle Swarm Optimization and New Luus-Jaakola Hybrid Algorithm Optimized IMC-PID Controller for Diverse Wing Vibration Systems. *Complexity*, 2019, 1-21. DOI: <https://doi.org/10.1155/2019/8283178>
- Liao, B. y Luus, R. (2007). Comparison of the Luus–Jaakola optimization procedure and the genetic algorithm. *Engineering Optimization*, 37(4), pp. 381-396. DOI: <https://doi.org/10.1080/03052150512331328312>
- Luo, G., Zou, L., Wang, Z., Lv, C., Ou, J., y Huang, Y. (2021). A novel kinematic parameters calibration method for industrial robot based on Levenberg-Marquardt and Differential Evolution hybrid algorithm. *Robotics and Computer-Integrated Manufacturing*, 71, 102165. DOI: <https://doi.org/10.1016/j.rcim.2021.102165>
- Luus, R. (1998). Comments on "Dynamic Optimization of Batch Reactors Using Adaptive Stochastic Algorithms". *Ind. Eng. Chem. Res.*, p. 305. DOI: <https://doi.org/10.1021/ie9708283>
- Luus, R. (2001). Comments on "Global Optimization for the Parameter Estimation of Differential-Algebraic Systems". *Ind. Eng. Chem. Res.*, 40, pp. 488-489. DOI: <https://doi.org/10.1021/ie000602r>
- Marquardt, D. W. (1963). An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2), pp. 431-441. DOI: <https://doi.org/10.1137/0111030>

- Meena, V. P., Yadav, U. K., Mathur, A., Singh, V. P., Guerrero, J. M., y Khan, B. (2023). Location and size selection using hybrid Jaya-Luus-Jaakola algorithm for decentralized generations in distribution system considering demand-side management. *IET Renewable Power Generation*, 17(6), 1535-1544. DOI: <https://doi.org/10.1049/rpg2.12692>
- Mendes Platt, G., Pinheiro Domingos, R., y Oliveira de Andrade, M., (2014) Application of the Firefly and Luus-Jaakola algorithms in the calculation of a double reactive azeotrope. *Computational Science & Discovery*, 7, p. 015002. DOI: <https://doi.org/10.1088/1749-4699/7/1/015002>
- Oliveros-Muñoz, J. M., Martínez-Villalba, J. A., Jiménez-Islas, H., Luna-Porres, M. Y., Escamilla-Alvarado, C., y Ríos-Fránquez, F. J. (2021). Luus-Jaakola method and ADM1 based optimization of hydrogen sulfide in anaerobic digestion of cow manure. *Biochemical Engineering Journal*, 171, 108012. DOI: <https://doi.org/10.1016/j.bej.2021.108012>
- Pal, T., y Kaushik, M. (2023). Aircraft parameter estimation using a novel hybrid Luus-Jaakola/Hooke-Jeeves neural-network based optimization technique. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 237(9), 2196-2208. DOI: <https://doi.org/10.1177/09544100221140980>
- Ramírez Rodríguez, J. F. (2008) Estimación de parámetros en ecuaciones diferenciales parciales parabólicas y análisis de convergencia. Tesis de maestría en ciencias, Instituto Tecnológico de Celaya.
- Salcedo, R. L. (1992) Solving nonconvex nonlinear programming and mixed-integer nonlinear programming problems with adaptive random search. *Ind. Eng. Chem. Res.* 31 262-273. DOI: <https://doi.org/10.1021/ie00001a037>
- Sebti, I., Carnet, A. R., Blanc, D., Saurel, R., y Coma, V. (2003). Controlled diffusion of an antimicrobial peptide from a biopolymer film. *Chemical Engineering Research and Design*, 81(9), 1099-1104. DOI: <https://doi.org/10.1205/026387603770866245>
- Spaans, R., y Luus, R. (1992) Importance of search-domain reduction in random optimization. *J. Optim. Theory Appl.*, 75, 635-638. DOI: <https://doi.org/10.1007/bf00940497>
- Wang, B. C., y Luus, R. (1978) Reliability of optimization procedures for obtaining global optimum. *AIChE J.* 24, 619-626. DOI: <https://doi.org/10.1002/aic.690240409>
- Wang, K., Gong, W., Liao, Z., y Wang, L. (2022). Hybrid niching-based differential evolution with two archives for nonlinear equation system. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(12), 7469-7481. DOI: 10.1109/TSMC.2022.3157816