



**Regresión de modelos cinéticos no lineales
mediante homotopías.**

Juliana Atehortúa Muñoz

Trabajo de grado presentado como requisito parcial para optar al título de:
Ingeniera Química.

Asesores

Luis Alberto Díaz Montes, Ingeniero Químico PhD.

Rolando de Jesús Barrera Zapata, Ingeniero Químico PhD.

Universidad de Antioquia

Facultad de ingeniería, Departamento de Ingeniería Química

Medellín, Colombia

2023.

Cita	(Atehortúa, 2023)
Referencia	Atehortúa Muñoz, J. (2023). <i>Regresión de modelos cinéticos no lineales mediante homotopías, 2016 - 2023</i> [trabajo de grado]. Universidad de Antioquia, Medellín.
Estilo APA 7 (2020)	



Grupo de Investigación CERES.



Centro de Documentación Ingeniería (CENDOI)

Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

Rector: Jhon Jairo Arboleda Céspedes.

Decano/Director: Jesús Francisco Vargas Bonilla.

Jefe departamento: Lina María González Rodríguez.

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

Agradecimientos

A mis estimados asesores de trabajo de grado, PhD. Luis Alberto Díaz Montes y PhD. Rolando de Jesús Barrera Zapata, por su invaluable orientación, apoyo inquebrantable y contribuciones constantes a lo largo de este recorrido de investigación. Su experiencia y dedicación han sido fundamentales para moldear la trayectoria de mi trabajo y enriquecer mi comprensión en este inmenso universo donde los números es lo menos abstracto.

También estoy profundamente agradecida con todos los docente dedicados que han compartido su conocimiento y sabiduría durante mi educación de pregrado. Sus enseñanzas han sentado las bases sobre las cuales he construido mis logros académicos.

Por último, mi gratitud se extiende a mi familia por su constante aliento, paciencia y amor. Su creencia constante en mí ha sido la fuerza impulsora detrás de mis logros.

Tabla de contenido

Resumen	8
Abstract	9
Introducción	10
1. Objetivos	11
1.1 Objetivo general	11
1.2 Objetivos específicos.....	11
2. Marco teórico.....	12
2.1 Modelos cinéticos.....	12
2.2 Regresión por mínimos cuadrados	13
2.3 Métodos para solución de ecuaciones no lineales (SENL).	15
Métodos locales	17
Métodos globales	19
Métodos homotópicos.....	20
Métodos homotópicos frente a los métodos numéricos convencionales	29
Estado del arte sobre la aplicación de métodos homotópicos en el campo de la ingeniería química.....	31
3. Metodología.....	33
3.1 Búsqueda bibliográfica de los modelos cinéticos de ingeniería química con múltiples soluciones.....	33
• Problema 1.....	33
• Problema 2.....	35
3.2 Desarrollo de los planteamientos matemáticos a partir de los modelos seleccionados implementado mínimos cuadrados o diferenciación numérica.	36
3.3 Desarrollo de los códigos de las funciones de los modelos y de la homotopía esférica	38
Búsqueda de valores solución a partir de la variación de valores iniciales	39

3.4 Creación y ejecución de los programas.....	44
4. Resultados.....	46
5. Análisis.....	51
6. Conclusiones.....	53
Referencias.....	55
Anexos.....	57

Lista de tablas

Tabla 3.1 Datos y condiciones experimentales del proceso.....	34
Tabla 3.2 Resultados para los parámetros de cada modelo.....	35
Tabla 3.3 Datos experimentales del sistema.	36
Tabla 3.4 Conjuntos de valores iniciales para los modelos del 3.10 al 3.17.....	40
Tabla 3.5 Rango de valores iniciales para el modelo 3.18.....	41
Tabla 3.6 Algoritmo método de homotopía Newton con rastreo mediante hiperesferas y método de Newton-Raphson (Montes, 2019)	43
Tabla 4.1 Número de soluciones encontradas para cada modelo.....	46
Tabla 4.2 Valores encontrados para los modelos con única solución.....	46
Tabla 4.3 Múltiples soluciones para los modelos 3.16 y 3.18	47
Tabla 4.4 Múltiples soluciones para modelos 3.16 y 3.18	48
Tabla 4.5 Indicadores en cada solución encontrada.....	48
Tabla 0.1 Código en MATLAB para minimizar la ecuación 2.15 del modelo 3.18.....	57
Tabla 0.2 Código de la homotopía con hiperesferas en Visual Basic for Application de Microsoft Excel. Tomado de (Montes, 2019)	58

Lista de figuras

Figura 2.1 Diferentes ajustes para el mismo sistema de datos	14
Figura 2.2 Seguimiento homotópico en el método clásico	24
Figura 2.3 Errores o fallas frecuentes del seguimiento por el método clásico.....	24
Figura 2.4 Método de seguimiento diferencial.....	25
Figura 2.5 Errores o fallas en el método de seguimiento diferencial	26
Figura 2.6 Estimativo en la ruta homotópica con predictor y sin predictor (Montes, 2019)	28
Figura 2.7 Seguimiento de la ruta homotópica	29
Figura 2.8 Diferencia en el dominio de la solución del problema mediante tres modelos.	30
Figura 3.1 Mejores ajustes para los datos experimentales.	35
Figura 4.1 Parámetro t en función del avance de la ruta homotópica (p)	49
Figura 4.2 Comportamiento de a_0 en función de t	49
Figura 4.3 Comportamiento de a_1 en función de t	50
Figura 4.4 Comportamiento de a_3 en función de t	50

Resumen

En el ámbito de la ingeniería química, el ajuste de modelos cinéticos desafía a los investigadores a encontrar múltiples soluciones para sistemas no lineales. En este trabajo, se abordó este desafío mediante la implementación de homotopías con hiperesferas, una técnica innovadora para resolver ecuaciones no lineales. El objetivo principal fue evaluar los ajustes a modelos cinéticos y aplicar homotopía tipo Newton mediante hiperesferas para encontrar múltiples soluciones a los parámetros cinéticos. En la presente investigación se estudiaron nueve modelos, inicialmente mediante herramientas como Solver y ReCongSolver de Microsoft Excel y la función `fsolve` de MATLAB fue posible filtrar los modelos más complejos. De los nueve modelos sólo dos registraron múltiples soluciones; a estos dos modelos se les aplicó el método homotópico desarrollado en ambos softwares (Microsoft Excel y MATLAB) con el fin de obtener las diferentes raíces de manera simultánea. El modelo 3.16 no convergió a múltiples soluciones, mientras que el modelo 3.18 satisfactoriamente reportó dos raíces válidas. Durante la convergencia se encontró que el método es altamente sensible a pequeñas variaciones en los valores iniciales, lo que podría afectar la estabilidad numérica. En pocas palabras, la implementación exitosa del método y sus resultados en la búsqueda de soluciones para sistemas complejos subrayan su potencial como herramienta valiosa en la ingeniería química y la investigación de procesos químicos.

Palabras clave: ajuste, modelos cinéticos, múltiples soluciones, homotopías, hiperesferas, parámetros, Microsoft Excel, MATLAB.

Abstract

In the field of chemical engineering, adjusting kinetic models challenges researchers to find multiple solutions for nonlinear systems. In this work, this challenge was addressed by implementing hypersphere homotopies, an innovative technique for solving nonlinear equations. The main objective was to evaluate the adjustments to kinetic models and apply Newton-type homotopy using hyperspheres to find multiple solutions to kinetic parameters. In this research, nine models were studied; initially, tools like Microsoft Excel's Solver and ReCongSolver, as well as MATLAB's fsolve function, were used to filter the more complex models. Out of the nine models, only two exhibited multiple solutions; the homotopic method developed in both software (Microsoft Excel and MATLAB) was applied to these two models to simultaneously obtain different roots. Model 3.16 did not converge to multiple solutions, while model 3.18 successfully yielded two valid roots. During convergence, it was found that the method is highly sensitive to small variations in initial values, which could impact numerical stability. In essence, the successful implementation of the method and its results in searching for solutions in complex systems underscore its potential as a valuable tool in chemical engineering and chemical process research.

Keywords: adjustment, kinetic models, multiple solutions, homotopies, hyperspheres, parameters, Microsoft Excel, MATLAB.

Introducción

En la industria química, el desarrollo de modelos cinéticos precisos y confiables es fundamental para el diseño, optimización y control de procesos químicos. Existen varios métodos para ajustar los parámetros cinéticos de los modelos cinéticos, tales como el método de Newton, el método de Levenberg-Marquardt y otros métodos basados en gradiente que requieren el cálculo de la derivada. Estos métodos pueden no encontrar una solución o hallar solo una cuando se tienen múltiples opciones.

En este contexto, los métodos homotópicos con hiperesferas se presentan como una alternativa prometedora para ajustar los parámetros cinéticos de los modelos cinéticos. Estos métodos se basan en la deformación continua del modelo y de los datos experimentales a lo largo de una trayectoria homotópica esférica. A medida que la trayectoria homotópica se acerca al modelo óptimo, los parámetros cinéticos se ajustan hasta que se alcanza la solución deseada. Además, estos métodos pueden manejar sistemas no lineales y tienen la capacidad de encontrar múltiples soluciones.

Por lo tanto, el objetivo de este trabajo de grado es aplicar homotopías hiperesféricas para ajustar nuevos parámetros cinéticos. Se espera que los resultados de este estudio proporcionen una herramienta más eficiente y precisa para ajustar los parámetros cinéticos de los modelos, lo que permitirá una mejor comprensión y control de las reacciones químicas.

1. Objetivos

1.1 Objetivo general

Evaluar ajustes a modelos cinéticos, aplicando el concepto de homotopías como método de solución de ecuaciones no lineales.

1.2 Objetivos específicos

- Definir los modelos cinéticos que se emplean actualmente en la ingeniería química.
- Establecer la homotopía a evaluar para el ajuste de los modelos cinéticos.
- Implementar la homotopía seleccionada sobre los modelos.
- Verificar el comportamiento (o los resultados) de las regresiones desarrolladas.

2. Marco teórico

2.1 Modelos cinéticos

La velocidad de una reacción química r_A nos indica cuán rápido se consumen ciertas moles de una especie química para formar otra diferente. La ecuación de velocidad es una expresión algebraica que es sólo una función de las propiedades químicas de las especies que reaccionan y de las condiciones de reacción como: concentración de las especies, temperatura, presión y de la presencia y tipo de catalizador (si lo hay) (Fogler, 2010).

$$-r_A = [k_A(T)][fn(C_A, C_B, \dots)] \quad (2.1)$$

Los modelos cinéticos son herramientas fundamentales en la industria de la ingeniería química ya que permiten describir y predecir la cinética de las reacciones químicas. Lo que conlleva a una optimización en los procesos de producción, reducción de los costos y a una mejora en la calidad de los productos.

Algunos de los modelos cinéticos que se utilizan comúnmente en la industria de la ingeniería química son, por ejemplo: modelos que obedecen a la ley de potencia o modelos de cinética enzimática. Los modelos que obedecen a ley de potencia establecen que, la velocidad de reacción corresponde al producto de las concentraciones de las especies individuales reaccionantes, cada una de ellas elevada a una potencia.

$$r = k[C_A]^\alpha [C_B]^\beta \quad (2.2)$$

Los exponentes α y β de las concentraciones conducen al concepto de orden de reacción. El orden de una reacción se refiere a las potencias a las cuales se elevan las concentraciones en la ley de velocidad cinética; y k corresponde a la constante de velocidad de reacción. Dentro de los modelos que obedecen a esta ley y que más se emplean para diversos sistemas son los de orden cero, uno y dos.

Por otra parte, los modelos de cinética enzimática se utilizan para describir la velocidad de reacción de las enzimas. Las enzimas son proteínas que actúan como catalizadores en las reacciones

bioquímicas, acelerando la velocidad de reacción sin ser consumidas en el proceso. Estos modelos se basan en la teoría de Michaelis-Menten que describe cómo la velocidad de reacción depende de la concentración del sustrato y de la enzima. El modelo se compone de dos pasos: la formación del complejo enzima-sustrato (ES) y la liberación del producto (P). La ecuación de velocidad de reacción de Michaelis-Menten se expresa como:

$$r_S = \frac{V_{\text{máx}} * [S]}{K_M + [S]} \quad (2.3)$$

Donde r es la velocidad de reacción, $[S]$ es la concentración del sustrato, $V_{\text{máx}}$ es la velocidad máxima de reacción (cuando la enzima está totalmente saturada con sustrato), y K_M es la constante de Michaelis-Menten, que es una medida de la afinidad de la enzima por el sustrato.

Por otra parte, el artículo "A Review of Models for Complex Chemical Kinetics with Multiple Steady States" de Johnson y Song (2005) discute varios modelos cinéticos que presentan múltiples estados estacionarios en reacciones químicas complejas.

2.2 Regresión por mínimos cuadrados

La regresión por mínimos cuadrados es un método utilizado para encontrar la línea o curva que mejor se ajusta a un conjunto de datos. Consiste en encontrar los parámetros de una función matemática que minimicen la suma de los cuadrados de las diferencias entre los valores observados y los valores predichos por la función.

Se parte de un conjunto de puntos (datos experimentales) y se busca encontrar los parámetros $(a_0, a_1, a_2, \dots, a_p)$ de un modelo matemático que represente de manera óptima el comportamiento de los datos.

$$y_{\text{modelo}} = f(a_0, a_1, a_2, \dots, a_p, x_1, x_2, \dots, x_m) \quad (2.13)$$

Para un mismo conjunto de datos se pueden ajustar múltiples modelos. (Díaz, 2022)

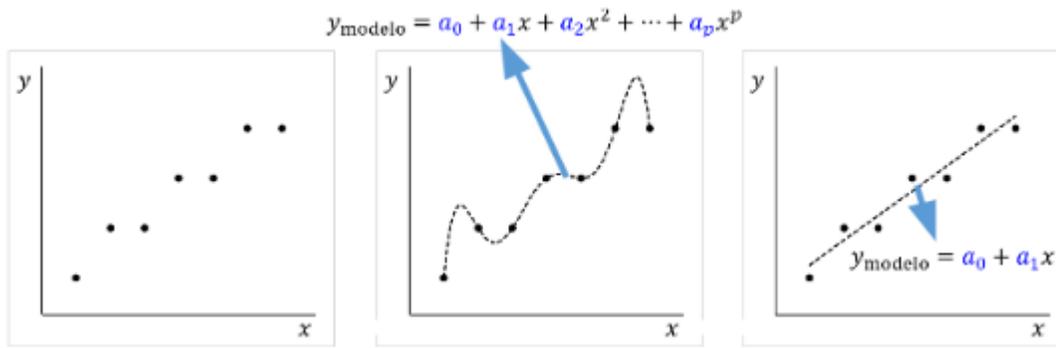


Figura 2.1 Diferentes ajustes para el mismo sistema de datos

Para todos los diferentes ajustes, los parámetros desconocidos se pueden calcular minimizando las desviaciones o el error (ε) entre los datos experimentales y la curva que representa el modelo.

El criterio mayormente empleado para cuantificar la discrepancia general es el de la suma de los cuadrados de las desviaciones:

$$SS_{residual} = \sum_{i=1}^n (\varepsilon_i)^2 = \sum_{i=1}^n (y_i - y_{i,modelo})^2 \quad (2.14)$$

Es decir:

$$SS_{residual} = \sum_{i=1}^n \left(y_i - f(a_0, a_1, a_2, \dots, a_p, x_{1,i}, x_{2,i}, \dots, x_{m,i}) \right)^2 \quad (2.15)$$

Dado que los valores $\{x_{1,i}, x_{2,i}, \dots, x_{m,i}, y_i\}$ son datos conocidos, $SS_{residual}$ es una función de $p+1$ variables: los parámetros desconocidos del modelo $a_0, a_1, a_2, \dots, a_p$.

Para encontrar el mejor ajuste es necesario resolver el problema de optimización no restringido:

$$\min SS_{residual} = \sum_{i=1}^n \left(y_i - f(a_0, a_1, a_2, \dots, a_p, x_{1,i}, x_{2,i}, \dots, x_{m,i}) \right)^2$$

Aplicando condición necesaria de primer orden (puntos estacionarios) para un punto solución de este problema de optimización:

$$\nabla SS_{\text{residual}}(a_0, a_1, \dots, a_p) = \begin{bmatrix} \frac{\partial}{\partial a_0} SS_{\text{residual}}(a_0, a_1, a_2, \dots, a_p) \\ \frac{\partial}{\partial a_1} SS_{\text{residual}}(a_0, a_1, a_2, \dots, a_p) \\ \vdots \\ \frac{\partial}{\partial a_p} SS_{\text{residual}}(a_0, a_1, a_2, \dots, a_p) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Es decir:

$$\nabla SS_{\text{residual}} = \begin{bmatrix} -2 \sum_{i=1}^n \left((y_i - f(a_0, a_1, \dots, a_p, x_{1,i}, x_{2,i}, \dots, x_{m,i})) \cdot \frac{\partial}{\partial a_0} f(a_0, a_1, \dots, a_p, x_{1,i}, \dots, x_{m,i}) \right) \\ -2 \sum_{i=1}^n \left((y_i - f(a_0, a_1, \dots, a_p, x_{1,i}, x_{2,i}, \dots, x_{m,i})) \cdot \frac{\partial}{\partial a_1} f(a_0, a_1, \dots, a_p, x_{1,i}, \dots, x_{m,i}) \right) \\ \vdots \\ -2 \sum_{i=1}^n \left((y_i - f(a_0, a_1, \dots, a_p, x_{1,i}, x_{2,i}, \dots, x_{m,i})) \cdot \frac{\partial}{\partial a_p} f(a_0, a_1, \dots, a_p, x_{1,i}, \dots, x_{m,i}) \right) \end{bmatrix} \\ = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Para varios modelos comunes (lineal simple, lineal múltiple, polinomial, logarítmico, ...) la solución de este sistema de ecuaciones es única y representa la solución del problema de optimización no restringida (Díaz, 2022)

2.3 Métodos para solución de ecuaciones no lineales (SENL).

El problema de solucionar un sistema de ecuaciones no lineales puede definirse como encontrar el vector \mathbf{x}_f que satisfaga la expresión

$$f(\mathbf{x}_f) = 0 \quad (2.15)$$

Con

$$f(\mathbf{x}) = \begin{bmatrix} f_1(x_1, x_2, \dots, x_n) \\ f_2(x_1, x_2, \dots, x_n) \\ \vdots \\ f_n(x_1, x_2, \dots, x_n) \end{bmatrix} \quad (2.16)$$

donde, $f(\mathbf{x})$ es una función vectorial multivariable que representa el sistema de ecuaciones igualadas a cero a resolver, y $f_i(x_1, x_2, \dots, x_n)$ es la función escalar multivariable de la ecuación i (Montes, 2019). En la búsqueda de soluciones aproximadas a problemas numéricos, se utilizan diversas estrategias que se basan en el uso de fórmulas recurrentes. Estas fórmulas permiten obtener una aproximación aceptable de la solución exacta del problema. Para aplicar estas fórmulas, es necesario contar con un valor inicial que se conoce como estimación inicial o valor semilla $\mathbf{x}^{[0]}$, y que es esencial para la convergencia del método numérico.

Para determinar si se ha alcanzado una solución numérica satisfactoria, se establece un criterio de convergencia que implica comparar la norma del vector 2.16 en el punto evaluado con una cantidad fija y subjetiva ε_f , que suele ser cercana a cero, como por ejemplo 1×10^{-5} . Si la norma del vector 2.16 es menor que ε_f , se considera que el punto analizado está lo suficientemente cerca de la solución exacta para declarar la convergencia.

$$\|f(\mathbf{x}^{[k]})\| \leq \varepsilon_f \quad (2.17)$$

El método numérico se ejecuta hasta que se cumpla el criterio de convergencia establecido, momento en el cual se declara que se ha alcanzado una solución satisfactoria. Además, es común fijar un límite máximo de iteraciones para lograr la solución numérica. Si el criterio de convergencia no se satisface dentro del límite de iteraciones establecido, el método se detiene y se informa que no se ha alcanzado la convergencia.

Los métodos para encontrar los ceros de una función no lineal se pueden dividir en métodos locales, que se basan en información sobre la función en puntos particulares de su dominio, y métodos globales, que dependen de alguna propiedad de la función que se cumple en todo el dominio (Wayburns & Seader, 1987).

Métodos locales

Los métodos locales para la solución de ecuaciones no lineales son algoritmos que buscan encontrar una única solución dentro de un rango determinado de valores. Algunos de los métodos locales más comunes y empleados son el método de Newton-Raphson o el método de Powell.

Método de Newton-Raphson

Este método es un método iterativo que se basa en la idea de que una función se puede aproximar por una función lineal en un punto dado, y que la solución al mínimo de la función se puede encontrar al encontrar la raíz de la derivada de la función. El método de Newton utiliza esta aproximación lineal para encontrar la raíz de la derivada, y luego utiliza esta raíz para actualizar la posición del punto de búsqueda en cada iteración.

En particular, el método de Newton-Raphson para sistemas de ecuaciones no lineales utiliza la fórmula:

$$x_{i+1} = x_i - J_f^{-1}(x_i) \cdot f(x_i) \quad (2.18)$$

donde x_i es la aproximación inicial del vector de soluciones y $J(F(x_i))^{-1}$ es la inversa de la matriz jacobiana evaluada en x_i . $F(x_i)$ es el vector de funciones evaluadas en la aproximación x_i .

Este proceso se repite hasta que se alcance una precisión deseada. Una limitación importante del método de Newton-Raphson es que solo encuentra una solución local cerca de la aproximación inicial. En otras palabras, si hay varias raíces de la ecuación en el intervalo de búsqueda, el método puede converger a una de ellas, pero no a las demás.

Método de Powell

El método de Powell surge como una alternativa al método de Newton cuando se enfrenta a la limitación de matrices jacobianas singulares o cuasi-singulares. En lugar de utilizar la dirección de

descenso determinada por la inversa de la matriz jacobiana, el método de Powell utiliza una combinación lineal de las direcciones de búsqueda anteriores. Una dirección de búsqueda comúnmente utilizada es $\mathbf{s}^{[k]} = \mathbf{s}_G^{[k]}$, la dirección en la que la función $f(\mathbf{x}) = \frac{1}{2} \cdot \|\mathbf{f}(\mathbf{x})\|^2 = \frac{1}{2} \cdot [\mathbf{f}(\mathbf{x})]^T \cdot \mathbf{f}(\mathbf{x})$ desciende más rápidamente; es decir, la dirección dada por el negativo del vector gradiente de $f(\mathbf{x})$, (Montes, 2019)

$$\mathbf{s}_G^{[k]} = -\nabla f(\mathbf{x}^{[k]}) = -\nabla \left(\frac{1}{2} \cdot [\mathbf{f}(\mathbf{x}^{[k]})]^T \cdot \mathbf{f}(\mathbf{x}^{[k]}) \right) \quad (2.21)$$

donde

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(x_1, x_2, \dots, x_n)}{\partial x_1} \\ \frac{\partial f(x_1, x_2, \dots, x_n)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x_1, x_2, \dots, x_n)}{\partial x_n} \end{bmatrix} \quad (2.22)$$

por lo que el vector de búsqueda sería

$$\mathbf{s}_G^{[k]} = -[\mathbf{J}_f(\mathbf{x}^{[k]})]^T \cdot \mathbf{f}(\mathbf{x}^{[k]}) \quad (2.23)$$

El método de Powell se inicia con el cálculo del tamaño de paso $\beta[k]$ con el que se minimiza una aproximación cuadrática de $f(\mathbf{X})$ en la dirección de $\mathbf{S}_G[k]$. Es decir:

$$\beta^{[k]} = \frac{\|\mathbf{s}_G^{[k]}\|^2}{\|\mathbf{J}_f(\mathbf{x}^{[k]}) \cdot \mathbf{s}_G^{[k]}\|^2} \quad (2.24)$$

De este modo, para un tamaño de paso máximo permitido en cada iteración (γ), se construye el vector de búsqueda de acuerdo con

$$\mathbf{s}^{[k]} = \begin{cases} \gamma \frac{\mathbf{s}_G^{[k]}}{\|\mathbf{s}_G^{[k]}\|} & \text{cuando } \gamma \leq \beta^{[k]} \cdot \|\mathbf{s}_G^{[k]}\| \\ \eta^{[k]} \cdot \mathbf{s}_N^{[k]} + (1 - \eta^{[k]}) \cdot \beta^{[k]} \cdot \mathbf{s}_G^{[k]} & \text{cuando } \beta^{[k]} \cdot \|\mathbf{s}_G^{[k]}\| < \gamma < \|\mathbf{s}_N^{[k]}\| \\ \mathbf{s}_N^{[k]} & \text{cuando } \gamma \geq \|\mathbf{s}_N^{[k]}\| \end{cases} \quad (2.25)$$

donde el factor de ponderación $\eta^{[k]}$ se evalúa con

$$\eta^{[k]} = (\gamma - \beta^{[k]} \cdot \|\mathbf{s}_G^{[k]}\|) / (\|\mathbf{s}_N^{[k]}\| - \beta^{[k]} \cdot \|\mathbf{s}_G^{[k]}\|) \quad (2.26)$$

Este método aprovecha las ventajas de dos direcciones de búsqueda y evita los problemas asociados a la implementación de cada una de ellas.

Métodos globales

Los métodos globales, también conocidos como métodos directos, son técnicas numéricas para encontrar soluciones de sistemas de ecuaciones no lineales. Estos métodos pueden encontrar todas las soluciones posibles en un intervalo dado y son particularmente útiles para problemas de diseño y optimización en ingeniería. Algunos de los métodos globales más comunes son:

Métodos de región de confianza

Los métodos de búsqueda de confianza buscan la solución óptima de un problema de optimización no lineal, donde la función objetivo y las restricciones pueden no ser lineales. El objetivo es minimizar o maximizar una función objetivo $f(x)$ sujeta a restricciones $g(x) \leq 0$ y $h(x) = 0$. Puede describirse matemáticamente de la siguiente manera:

Se comienza con una solución inicial x_0 y una región de confianza inicial alrededor de la solución, la cual se define por un radio $\delta > 0$.

Se evalúa la función objetivo en la solución inicial x_0 : $f_0 = f(x_0)$. Posteriormente, se construye un modelo aproximado de la función objetivo en torno a la solución inicial x_0 . Este modelo puede ser una aproximación de primer o segundo orden, y puede basarse en la expansión de Taylor de la función objetivo:

$$m(x) = f(x_0) + \nabla f(x_0)^T(x - x_0) + \frac{1}{2}(x - x_0)^T H(x - x_0) \quad (2.27)$$

donde $\nabla f(x_0)$ es el gradiente de f en x_0 , H es la matriz Hessiana de f en x_0 , y T denota la transposición.

Se busca una nueva solución dentro de la región de confianza que mejore el valor del modelo de la función objetivo. Este proceso se puede realizar utilizando un método de optimización local, como el método de Newton o el método del gradiente conjugado:

$$x_n = \operatorname{argmin}(m(x)) \text{ sujeto a: } \|x - x_0\| \leq \delta$$

Si el modelo mejora la función objetivo, se acepta la nueva solución y se expande la región de confianza para incluir la nueva solución: $x_0 = x_n$; $f_0 = f_n$; $\delta = \delta * \gamma$. Donde $\gamma > 1$ es un factor de expansión de la región de confianza.

Si el modelo no mejora la función objetivo, se rechaza la nueva solución y se reduce la región de confianza: $\delta = \delta / \gamma$; donde $0 < \gamma < 1$ es un factor de reducción de la región de confianza.

Se repite el proceso hasta que se cumpla un criterio de convergencia, como la convergencia de la función objetivo o la convergencia de la norma del gradiente: $\|\nabla f(x)\| \leq \varepsilon$; donde $\varepsilon > 0$ es una tolerancia predefinida.

Métodos homotópicos

Es un método iterativo que busca una solución a través de una familia continua de ecuaciones, llamadas ecuaciones homotópicas, que van desde una ecuación inicial para la cual se conoce la solución hasta la ecuación deseada para la cual se busca la solución.

El método de homotopía es un enfoque general para resolver ecuaciones no lineales, y se puede utilizar para resolver una amplia variedad de problemas, incluyendo sistemas de ecuaciones, problemas de optimización y problemas de valor inicial y de frontera.

La idea básica detrás del método de homotopía es transformar la ecuación original no lineal en una ecuación lineal que sea más fácil de resolver. Esto se logra mediante la introducción de un parámetro adicional en la ecuación original, que se utiliza para crear una familia continua de ecuaciones homotópicas.

Por esta razón, el objetivo es encontrar la solución x_F del problema 2.9, en este caso, haciendo uso de un sistema de ecuaciones auxiliar cuya solución, X_e , es conocida.

$$e(x_e) = 0 \quad (2.28)$$

A partir de la función $e(x)$ y la función $f(x)$, se define la función homotópica $h(t, x)$, buscando que se presente la siguiente

$$\begin{cases} t = 0 \rightarrow h(x) = e(x) \\ t = 1 \rightarrow h(x) = f(x). \end{cases} \quad (2.29)$$

La forma de la función homotópica más habitual es

$$h(t, x) = t \cdot f(x) + (1 - t) \cdot e(x) \quad (2.30)$$

La estrategia básica del uso de esta construcción en la solución del problema original $f(x_f) = 0$, proviene del siguiente razonamiento: para $t = t_{[0]} = 0 \rightarrow h_{[0]}(x) = e(x)$, se conoce la solución $x_{[0]} = x_e$. Si se hace un pequeño incremento sobre $t_{[0]}$, $\Delta t_{[0]}$, tal que:

$$t_{[1]} = t_{[0]} + \Delta t_{[0]} \cong t_{[0]},$$

se espera que $h_{[1]}(x) \cong h_{[0]}(x)$, y por tanto que el vector solución $x_{[0]}$ que satisface $h_{[0]}(x_{[0]}) = 0$ sea cercano al vector solución $x_{[1]}$ que satisface $h_{[1]}(x_{[1]}) = 0$. Así, la raíz de la ecuación:

$$h_{[1]}(X) = t_{[1]} \cdot f(x) + (1 - t_{[1]}) \cdot e(x)$$

puede encontrarse utilizando un método de convergencia local como el de Newton–Raphson, y tomando como valor inicial $x_{[0]}$, el cual debe ser cercano al vector a encontrar $x_{[1]}$. Una vez que se encuentra el vector $x_{[1]}$, se realiza un nuevo incremento sobre el parámetro t , $\Delta t_{[1]}$, tal que:

$$t_{[2]} = t_{[1]} + \Delta t_{[1]} \cong t_{[1]}$$

y se espera que $h_{[2]}(x) \cong h_{[1]}(x)$, y por tanto, que el vector solución $x_{[1]}$ que satisface $h_{[1]}(x_{[1]}) = 0$ sea cercano al vector solución $X_{[2]}$ que satisface $h_{[2]}(x_{[2]}) = 0$.

Así se crea un algoritmo recurrente, donde utilizando como valor inicial la raíz conocida $x_{[h-1]}$ de la función homotópica en un punto $[h - 1]$, puede utilizarse un método de convergencia local para encontrar el valor $x_{[h]}$ de la solución al problema

$$h_{[h]}(x_{[h]}) = t_{[h]} \cdot f(x_{[h]}) + (1 - t_{[h]}) \cdot e(x_{[h]}) = 0 \quad (2.31)$$

Dentro de este proceso recurrente, cuando $t = 1$, entonces $h(1, x) = f(x)$ y nuevamente se aplicará el método adecuado de convergencia local, disponiendo de un valor inicial lo suficientemente cercano a la respuesta (proveniente de la solución del punto discreto anterior en la ruta homotópica) para garantizar la obtención de la solución al problema original $f(x_f) = 0$.

El método de homotopía se puede implementar de varias maneras, incluyendo el método de seguimiento de trayectoria, el método de continuación de Newton y el método de continuidad de grado. Cada una de estas técnicas tiene sus propias ventajas y desventajas, y la elección de un método específico dependerá del problema particular que se esté tratando de resolver.

Cabe resaltar que el carácter global o local de un método homotópico depende de la trayectoria que se utiliza para conectar las soluciones. En general, los métodos homotópicos pueden ser globales o locales, dependiendo de la trayectoria elegida.

Un método homotópico se considera global si la trayectoria utilizada conecta todas las soluciones del sistema de ecuaciones no lineales. Mientras que, se considera local si la trayectoria utilizada sólo conecta algunas soluciones cercanas en el espacio de soluciones.

En la práctica, muchos métodos homotópicos son locales y sólo son capaces de encontrar soluciones cercanas a la solución inicial. Sin embargo, existen métodos homotópicos globales que pueden encontrar todas las soluciones posibles del sistema de ecuaciones no lineales. Estos métodos suelen ser más complejos y requieren más recursos computacionales que los métodos homotópicos locales.

Tipos de homotopías

La función $e(x)$ a incluir en el algoritmo de la solución homotópica se debe elegir de tal manera que $e(x_e) = 0$, es decir, $x_{[0]} = x_e$. Dicha elección da lugar a diferentes tipos de homotopías:

- Homotopía punto fijo, con $e(x) = x - x_{[0]}$
- Homotopía afín, con: $e(x) = A \cdot (x - x_{[0]})$, donde A es una matriz de escalamiento, usualmente la matriz jacobiana de $f(x)$ evaluada en el punto $x_{[0]}$.
- Homotopía Newton, con: $e(x) = f(x) - f(x_{[0]})$.

Dado que el camino homotópico no se conoce de antemano, es necesario encontrar una manera de determinar su ubicación. En la literatura, se han descrito diferentes enfoques para llevar a cabo este seguimiento, como el Método Clásico y el Método de Seguimiento Diferencial (Wayburns & Seader, 1987). Estos métodos han demostrado ser efectivos para localizar el camino homotópico en muchos casos, pero pueden presentar dificultades si la trayectoria homotópica es muy compleja o si hay múltiples caminos homotópicos en la región de análisis (Jiménez-Islas, 1996).

- Método clásico: es un enfoque que se basa en proponer predictores horizontales con una longitud predefinida (también conocida como Δt) para calcular un valor X^P . Este valor X^P se utiliza como base para generar un vector normal, el cual se emplea como una corrección de Newton. El objetivo es encontrar el punto X^C mediante esta corrección y una vez que se obtiene, se utiliza como base para generar un nuevo predictor.

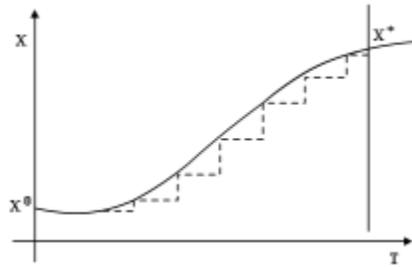


Figura 2.2 Seguimiento homotópico en el método clásico

Tomada de (Jiménez-Islas, 1996)

Este método tiene algunas desventajas. Una de ellas es que, el valor de Δt debe ser relativamente pequeño para evitar el riesgo de utilizar muchas iteraciones de Newton durante el proceso correctivo. Adicionalmente, existe el riesgo de perder el camino homotópico en situaciones como la que se ilustra en la Figura 2.3, las cuales son bastante comunes.

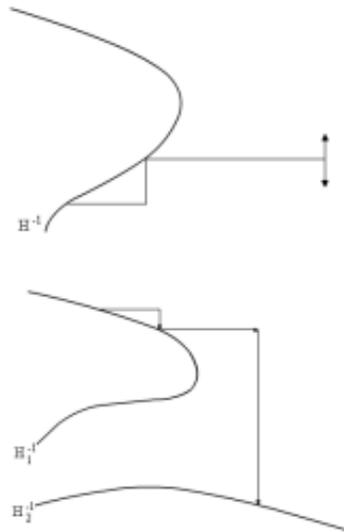


Figura 2.3 Errores o fallas frecuentes del seguimiento por el método clásico

Tomada de (Jiménez-Islas, 1996)

- Método de seguimiento diferencial: El método propuesto implica tomar la ecuación (2.31) y diferenciarla con respecto a un parámetro llamado "p". Este parámetro se utiliza como una variable para formular un problema de valores iniciales. Luego, se resuelve este problema utilizando un predictor explícito, como el método de Euler o el método de Runge-Kutta-Fehlberg. Después de eso, se aplica una corrección de Newton siguiendo el proceso

mostrado en la Figura 2.4. La longitud del predictor se ajusta utilizando el algoritmo de Georg (1981).

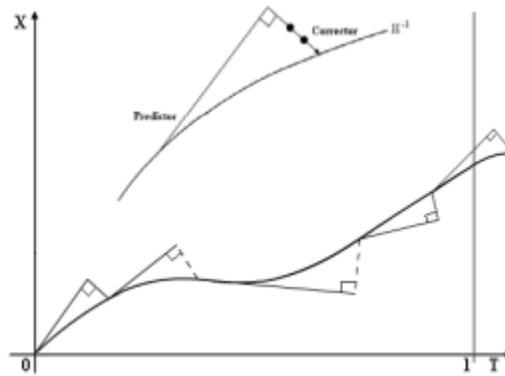


Figura 2.4 Método de seguimiento diferencial

Tomada de (Jiménez-Islas, 1996)

Este algoritmo se considera más robusto que el enfoque anterior, ya que permite seguir el camino homotópico de manera más eficiente. Esto se debe a que se basa en la pendiente de la trayectoria homotópica. Sin embargo, no tiene en cuenta casos como los ilustrados en la Figura 2.5. Además, para encontrar varias soluciones vectoriales, se sugiere utilizar varios valores iniciales " X^0 ". La desventaja es que no se puede determinar de antemano si diferentes valores iniciales " X^0 " conducirán a soluciones diferentes " X ", ni se garantiza que se encontrarán todas las soluciones vectoriales del problema con un número determinado de valores iniciales " X^0 ".

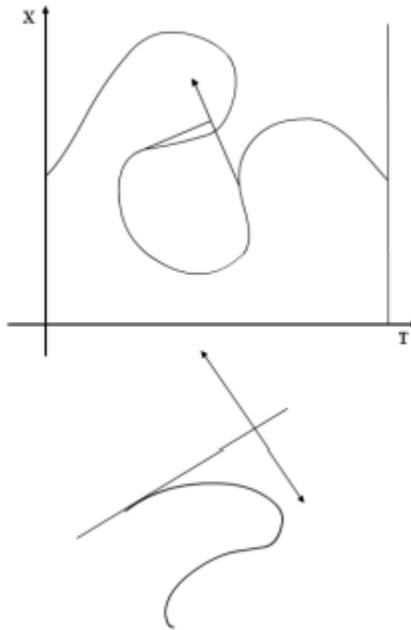


Figura 2.5 Errores o fallas en el método de seguimiento diferencial

Tomada de (Jiménez-Islas, 1996)

La necesidad de plantear un seguimiento diferente surge cuando se enfrenta a situaciones en las que el método clásico o el método de seguimiento diferencial pueden ser insuficientes. En ciertos casos, la trayectoria homotópica puede ser altamente compleja, presentando curvas cerradas, bucles o múltiples caminos homotópicos en la región de análisis.

Para abordar estas situaciones desafiantes, se propone el uso de hiperesferas en el proceso de seguimiento.

- Seguimiento con hiperesferas: Las hiperesferas son objetos geométricos multidimensionales que permiten una mayor flexibilidad y adaptabilidad al rastrear el camino homotópico. La idea principal es ubicar hiperesferas alrededor de puntos clave en la trayectoria homotópica y utilizar su propiedad de expansión o contracción para seguir el camino deseado. Estas hiperesferas pueden ajustarse y moverse de manera iterativa a medida que se recopila información sobre la dirección y el comportamiento del camino homotópico.

La metodología empleada para realizar el seguimiento homotópico con hiperesferas consiste en (Quemada et al., 2022):

1. Comenzamos con un punto inicial hiperdimensional $x^0 = [x_1^0, x_2^0, x_3^0, \dots, x_n^0]^t$, el cual se convertirá en el centro de la primera hiperesfera. Esta hiperesfera tiene un radio previamente establecido r .¹
2. A partir de este punto central, generamos un vector que es tangente al camino homotópico y que interseca la superficie de la hiperesfera en un nuevo punto hiperdimensional $x^P = [x_1^P, x_2^P, x_3^P, \dots, x_n^P]^t$
3. Tomando este punto como una aproximación, buscamos la intersección entre la hiperesfera y el camino homotópico resolviendo un sistema de ecuaciones que representa ambos elementos mediante el método de Newton-Raphson. De esta manera, obtenemos un nuevo punto hiperdimensional $x^c = [x_1^c, x_2^c, x_3^c, \dots, x_n^c]^t$ (corregido).
4. El punto hiperdimensional x^c se convierte en el centro para generar la siguiente hiperesfera, y también se utiliza como punto inicial para el siguiente vector tangente. Esto nos permite encontrar un nuevo punto x^P que nos llevará a un nuevo x^c , y así sucesivamente, continuando este proceso iterativo.

Generalmente, para mejorar el estimativo inicial en el siguiente punto y por ende, un favorable avance de la ruta homotópica, se emplea el esquema predictor - corrector de Euler; que está dado por:

$$\mathbf{x}_{[h]}^{\text{Inicial}} = \mathbf{x}_{[h-1]} + (t_{[h]} - t_{[h-1]}) \cdot [\mathbf{J}_h(\mathbf{x}_{[h-1]})]^{-1} \cdot [\mathbf{e}(\mathbf{x}_{[h-1]}) - \mathbf{f}(\mathbf{x}_{[h-1]})] \quad (2.32)$$

¹ Generalmente, por heurística el radio r inicial de la hiperesfera se toma como 0.1

$$\text{Con} \quad \mathbf{J}_h(\mathbf{x}_{[h-1]}) = t_{[h-1]} \cdot \mathbf{J}_f(\mathbf{x}_{[h-1]}) + (1 - t_{[h-1]}) \cdot \mathbf{J}_e(X_{[h-1]}) \quad (2.33)$$

En este esquema, primero se implementa la Ec. 2.32 sobre el punto $(t_{[h-1]}, \mathbf{x}_{[h-1]})$ para encontrar un mejor valor inicial para el siguiente punto de la ruta (Montes, 2019).

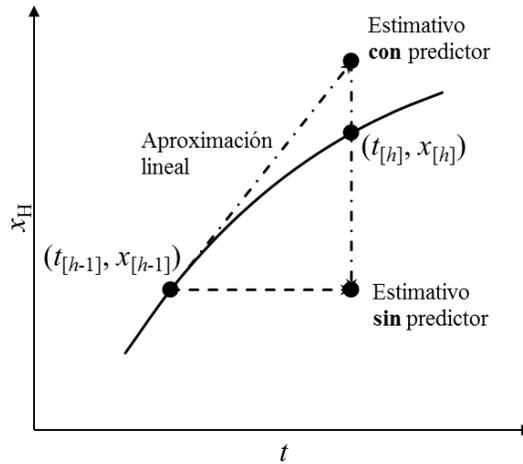


Figura 2.6 Estimativo en la ruta homotópica con predictor y sin predictor (Montes, 2019)

En consecuencia, la trayectoria de la ruta homotópica está definida por una sucesión de hipersferas (figura 2.7), donde V^P corresponde a los vectores de predicción y H^{-1P} es el camino homotópico que conforman; V^c corresponde a los vectores corregidos por medio de la corrección hipersférica y H^{-1C} es el camino homotópico formado por estos; C^N corresponde a los puntos de corrección hipersférica realizada por el método de Newton-Raphson y H^{-1} sería el camino homotópico cuya trayectoria en principio, es desconocida (Quemada et al., 2022).

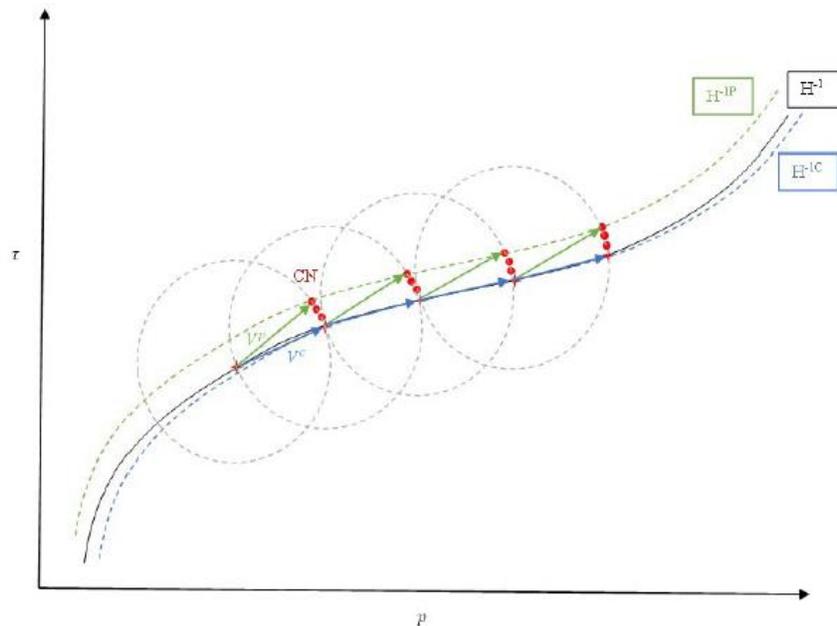


Figura 2.7 Seguimiento de la ruta homotópica
Tomada de (Quemada et al., 2022)

Métodos homotópicos frente a los métodos numéricos convencionales

La diferencia entre los métodos homotópicos y métodos como el de Newton y Powell se ve ilustrada en la figura 2.2 donde se exponen como ejemplo dos ecuaciones no lineales. En este caso, se buscan las raíces positivas reales, (1,4) y (4.07, 0.65). Con el método de Newton, la suposición inicial para x_1 y x_2 debe estar dentro de cualquiera de las regiones rayadas para lograr la convergencia a cualquiera de las raíces. Con el método de Powell, la región de convergencia se expande hacia las líneas discontinuas.

Sin embargo, ambos métodos fallan a partir del valor inicial de (15, 15), que está marcada con un círculo. La continuación de la homotopía diferencial con $g(x) = f(x) - f(x')$ es convergente desde el punto (15, 15) así como desde cualquier otro punto del cuadrante positivo.

Dos ramas de la trayectoria de la homotopía que comienzan en el punto (15, 15) y conducen a las dos soluciones se muestran como curvas -- -. Al continuar más allá de cualquiera de las soluciones, se obtiene un camino de homotopía cerrado que contiene ambas soluciones. (Wayburns & Seader, 1987)

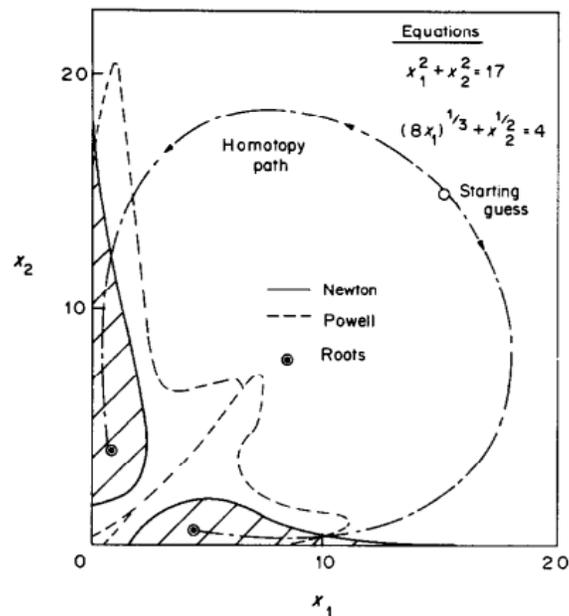


Figura 2.8 Diferencia en el dominio de la solución del problema mediante tres modelos.

Por esta razón, la utilización de homotopías en la solución de ecuaciones no lineales tiene diversas ventajas en el campo de la ingeniería química. A continuación, se presentan algunos argumentos para destacar su importancia:

- **Resolución de problemas no lineales:** Los métodos homotópicos permiten resolver ecuaciones no lineales que son difíciles de abordar con otros métodos numéricos, especialmente cuando se trata de sistemas complejos en ingeniería química (Allgower & Kurt, 2012).
- **Reducción del esfuerzo computacional:** Los métodos homotópicos reducen el esfuerzo computacional necesario para encontrar soluciones numéricas aproximadas, lo que significa que se pueden resolver problemas más grandes y complejos en un tiempo razonable (Allgower & Kurt, 2012).
- **Flexibilidad en la elección de parámetros:** La elección de parámetros en los métodos homotópicos es muy flexible, lo que permite adaptarlos a las características del problema

en cuestión (Asadi & Jalali Farahani, 2018). Esto es particularmente importante en la ingeniería química, donde los problemas son muy diversos y requieren soluciones personalizadas.

- **Mejora de la precisión:** Los métodos homotópicos pueden mejorar la precisión de las soluciones numéricas, especialmente en problemas que presentan múltiples soluciones o singularidades (Asadi & Jalali Farahani, 2018).

Estado del arte sobre la aplicación de métodos homotópicos en el campo de la ingeniería química

Chang y Seader en “Simulation Of Continuous Reactive Distillation By A Homotopy-Continuation Method” estudiaron teóricamente la destilación reactiva para la esterificación del ácido acético con etanol para producir acetato de etilo y agua, utilizando cinética reversible de segundo orden. Debido al alto grado de interacción entre las variables de iteración, la convergencia resultó difícil o incluso imposible de lograr con el método de Newton. Por lo tanto, se aplicó un método de continuación homotópica robusto para resolver las ecuaciones no lineales simultáneas utilizadas para modelar el sistema de destilación reactiva.

Las ecuaciones de modelado, llamadas $f(x) = 0$, se incorporaron en una homotopía de Newton:

$$H(x, t) = f(x) - (1 - t)f(x^0) = 0$$

donde t es el parámetro escalar de homotopía, que se varía de 0 a 1 mientras se utiliza la continuación para rastrear el camino desde la suposición inicial x_0 (en $t = 0$) hasta la solución en $t = 1$. El rastreo se realizó mediante la continuación de la longitud diferencial del arco utilizando un predictor de Euler y un corrector de Newton, como se describe en Waybum y Seader. Inicialmente, se intentó un tamaño máximo de paso equivalente a moverse desde $t = 0$ hasta $t = 1$, utilizando el método de Newton, el cual se iteró con y sin búsqueda de línea (Chang & Seader, 1988).

En esta investigación se analizaron 6 casos diferentes producto de variaciones en alguna de las propiedades o variables de la torre de destilación, se realizaron los cálculos mediante el método de Newton, método de Newton with a Golden-section line search y el de continuación homotópica; los dos primeros fallaron y el último convergió para todos los casos.

De las conclusiones extraídas de los estudios citados anteriormente, que demuestran una estrecha correspondencia entre el método de continuación paramétrica y las ventajas del seguimiento con hiperesferas estudiadas al final de la sección 2.3, se ha decidido aplicar esta estrategia en la búsqueda de múltiples soluciones para los parámetros.

3. Metodología

3.1 Búsqueda bibliográfica de los modelos cinéticos de ingeniería química con múltiples soluciones.

Para la selección de algunos modelos cinéticos con múltiples soluciones se realizó una búsqueda rigurosa de investigaciones donde se hayan realizado regresiones a partir de datos experimentales. Los problemas seleccionados se presentan a continuación:

- **Problema 1:** En el estudio “A new kinetic model to predict substrate inhibition and better efficiency in an airlift reactor on deammonification process”, se probaron diferentes modelos empíricos matemáticos basados en Monod para predecir la afinidad e inhibición de las bacterias involucradas en el proceso de deamonificación en aguas residuales con altas concentraciones de amonio y nitrito. El único reactor se expuso a diferentes concentraciones de sustrato del digestato del biodigestor de estiércol de cerdo (de 60 a 1200 mg NH₃-N L⁻¹) (Bonassa et al., 2021).

Para este estudio, se seleccionaron los modelos cinéticos empíricos de Tessier (Ec. 3.10), Monod (Ec. (3.11)), Moser (Ec. (3.12)), Haldane (Ec. (3.13)), Aiba et al. (Ec. (3.14)), Edwards I (Ec. (3.15)), Luong (considerando la inhibición del sustrato, Ec. (3.16)) y Han y Levenspiel (1988) (Ec. (3.17))

$$\mu = \mu_{max} * \left(1 - e^{-\frac{S}{K_S}}\right) \quad (3.10)$$

$$\mu = \mu_{max} \times \frac{S}{K_S + S} \quad (3.11)$$

$$\mu = \mu_{max} \times \frac{S^n}{K_S + S^n}, n > 0 \quad (3.12)$$

$$\mu = \mu_{max} \times \frac{S}{S + K_S + \frac{S^2}{K_I}} \quad (3.13)$$

$$\mu = \mu_{max} \times \frac{S}{S + K_S} \times e^{-\frac{S}{K_I}} \quad (3.14)$$

$$\mu = \mu_{max} \times \left(e^{-\frac{S}{K_I}} - e^{-\frac{S}{K_S}} \right) \quad (3.15)$$

$$\mu = \mu_{max} \times \frac{S}{S+K_S} \times \left(1 - \frac{S}{S_m} \right)^m \quad (3.16)$$

$$\mu = \mu_{max} \times \left(1 - \frac{S}{S_m} \right)^n \times \frac{S}{S+K_S \times \left(1 - \frac{S}{S_m} \right)^m} \quad (3.17)$$

Estos modelos son evoluciones constructivas del modelo Monod, considerando e incorporando más parámetros de proceso, como K_I (constante de inhibición), S_m (constante máxima de inhibición de sustrato por encima de la cual las células dejan de crecer), constantes m y n (relacionadas con la tasa de conversión de sustrato (μ) y la concentración de sustrato). Además, algunos otros modelos ofrecen nuevas interpretaciones y un mayor número de parámetros, como la constante de inhibición exponencial y la concentración crítica de sustrato.

Los datos experimentales para este estudio se muestran en la tabla 3.1

Tabla 3.1 Datos y condiciones experimentales del proceso.

$x_i = S$	$y_i = \mu$
61.4	3.3
72.9	6.1
148.2	9.2
195.9	11.6
255.3	17.5
290.3	19.8
407.1	25.9
512.3	34.1
653.7	37.8
792.8	39.2
902.1	39.3
980.4	34.4
1110.2	24.6
1192.6	19.6

Tomado de (Bonassa et al., 2021)

La tabla 3.2 muestra los parámetros arrojados por cada modelo y los valores estadísticos a modo comparativo.

Tabla 3.2 Resultados para los parámetros de cada modelo.

Modelo	R ²	μ _{máx}	K _s (mg NH ₃ -N gVSS-1 h-1)	K _i (mgNH ₃ -N L-1)	S _m (mgNH ₃ -N L-1)	m	n
Aiba	0.8397	195.9	1744.3	1245.7			
Edwards I	0.8598	568.2	736	860.4			
Haldane	0.8369	156	1575.6	435.9			
Han and Levenspiel	0.9926	12.9	122.7		1317.3	7.9975	-7.195
Luong	0.9744	487.1	5897.5		1220.6	0.408	
Monod	0.7099	43.5	323.8				
Moser	0.7626	35.7	10503				1.7031
Teissier	0.7596	33.73	294.01				

Tomado de (Bonassa et al., 2021)

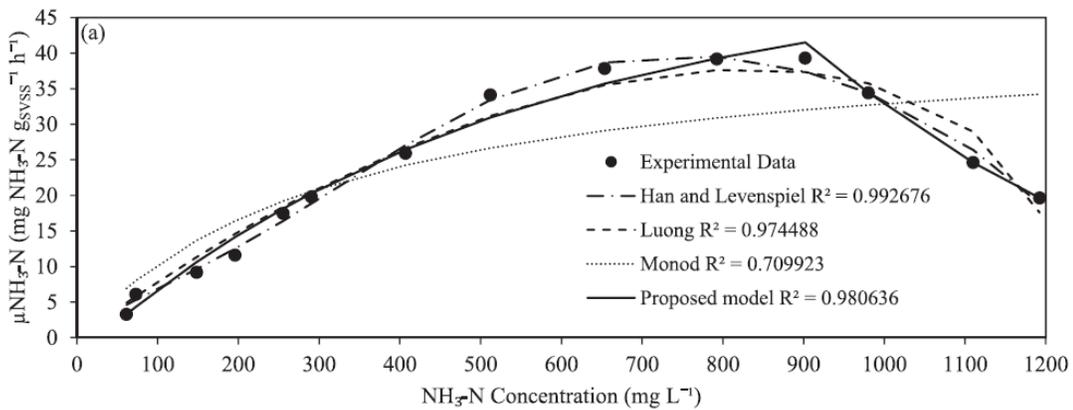


Figura 3.1 Mejores ajustes para los datos experimentales.

Tomado de (Bonassa et al., 2021)

- **Problema 2:** Se cree que, en un cierto sistema químico reactivo a condiciones de temperatura y presión dadas, la relación entre la velocidad de consumo de un reactivo “A” (r_A) y su concentración (C_A) en el sistema está dada por el modelo:

$$r_A = a_0 \cdot \frac{C_A^{a_1}}{(1+a_2 \cdot C_A)^{a_3}} \quad (3.18)$$

Tabla 3.3 Datos experimentales del sistema.

Datos	
$x_i = C_A$	$y_i = r_A$
0	0
0.2	0.088
0.7	0.529
2	1.441
4.5	2.476
7.1	2.517
8.7	2.876

3.2 Desarrollo de los planteamientos matemáticos a partir de los modelos seleccionados implementado mínimos cuadrados o diferenciación numérica.

Para los modelos del 3.10 al 3.17, se realizó el cálculo del vector gradiente mediante diferencias finitas para evitar expresiones de larga extensión y complejidad mayor. Para estos modelos se busca minimizar el vector gradiente (∇E_r) y para ello se empleó diferencias finitas centradas que tiene en cuenta la diferencia entre los valores de la función en puntos adyacentes tanto hacia adelante como hacia atrás para aproximar la derivada.

$$\nabla E_r = \frac{E_r(x_i+h) - E_r(x_i-h)}{2h} \quad (3.19)$$

$$E_r = (y_{dato} - y_{modelo})^2 \quad (3.20)$$

Donde x_i hace referencia a cada uno de los parámetros a encontrar y h representa el tamaño del incremento (también se conoce como paso de discretización), en este caso se estableció un valor de $1e-08$.

Por otro lado, para el modelo (3.18) se inició la búsqueda de múltiples soluciones para los parámetros a_0 , a_1 , a_2 y a_3 del modelo propuesto. El objetivo fue explorar diversas combinaciones de valores para estos parámetros, y determinar cómo afectaban la relación entre la velocidad de consumo del reactivo A (r_A) y su concentración (C_A).

Para esto, se pretendió minimizar la ecuación (2.15) y luego, aplicando la condición necesaria de primer orden (puntos estacionarios) para un punto solución de este problema de optimización:

$$\nabla SS_{\text{residual}}(a_0, a_1, a_2, a_3) = \begin{bmatrix} \frac{\partial}{\partial a_0} SS_{\text{residual}}(a_0, a_1, a_2, \dots, a_p) \\ \frac{\partial}{\partial a_1} SS_{\text{residual}}(a_0, a_1, a_2, \dots, a_p) \\ \frac{\partial}{\partial a_2} SS_{\text{residual}}(a_0, a_1, a_2, \dots, a_p) \\ \frac{\partial}{\partial a_3} SS_{\text{residual}}(a_0, a_1, a_2, \dots, a_p) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

De modo que:

$$\nabla SS_{\text{residual}}(a_0, a_1, a_2, a_3) = \begin{bmatrix} -2 \sum_{i=1}^n \left(\left(y_i - a_0 \cdot \frac{C_A^{a_1}}{(1 + a_2 \cdot C_A)^{a_3}} \right) \cdot \frac{C_A^{a_1}}{(1 + a_2 \cdot C_A)^{a_3}} \right) \\ -2 \sum_{i=1}^n \left(\left(y_i - a_0 \cdot \frac{C_A^{a_1}}{(1 + a_2 \cdot C_A)^{a_3}} \right) \cdot \frac{a_0 C_A^{a_1} \log(C_A)}{(1 + a_2 \cdot C_A)^{a_3}} \right) \\ -2 \sum_{i=1}^n \left(\left(y_i - a_0 \cdot \frac{C_A^{a_1}}{(1 + a_2 \cdot C_A)^{a_3}} \right) \cdot -\frac{a_0 a_3 C_A^{a_1+1}}{(1 + a_2 \cdot C_A)^{a_3+1}} \right) \\ -2 \sum_{i=1}^n \left(\left(y_i - a_0 \cdot \frac{C_A^{a_1}}{(1 + a_2 \cdot C_A)^{a_3}} \right) \cdot -\frac{a_0 C_A^{a_1} \log(1 + a_2 C_A)}{(1 + a_2 C_A)^{a_3}} \right) \end{bmatrix}$$

3.3 Desarrollo de los códigos de las funciones de los modelos y de la homotopía esférica

Para dar solución al problema de optimización descrito en las ecuaciones 2.15 y 3.19, se utilizaron dos softwares, Microsoft Excel® y MATLAB® con el fin de verificar la veracidad de los diferentes valores solución. En Microsoft Excel se usaron los complementos *Solver* y *ReCongSolver*, y en MATLAB (versión R2021b) se utilizó la función *fsolve*. Se exploraron diferentes valores iniciales para los parámetros con el fin de obtener múltiples soluciones que cumplan con el criterio de convergencia predeterminado por las herramienta de ambos softwares.

Durante el proceso de búsqueda, se tuvieron en cuenta las características predeterminadas de *fsolve* en MATLAB R2021b. Estas características incluyen:

- Tolerancia de convergencia: La función *fsolve* utiliza una tolerancia predeterminada de convergencia de $1e-6$. Esto implica que *fsolve* considerará que ha convergido cuando el cambio relativo en las variables de salida sea menor que $1e-6$.
- Método numérico: *fsolve* implementa el método de Newton-Raphson para resolver sistemas de ecuaciones no lineales. Este método está descrito en la sección 2.3 (métodos locales).
- Estimación inicial: Es un conjunto de estrategias implementadas en la función para proporcionar una estimación inicial de las variables de salida del sistema de ecuaciones no lineales. Esta heurística combina diferentes enfoques, como la consideración de valores iniciales proporcionados por el usuario, el análisis de la estructura del sistema, el comportamiento cercano a cero de la función y los valores típicos conocidos o esperados.

El complemento *Solver* de Microsoft Excel funciona en un nivel más abstracto dado que consta con una amigable interfaz donde el usuario define el problema de optimización o las ecuaciones no lineales a resolver, establece las restricciones y los criterios de optimización. Esta

herramienta utiliza una combinación de métodos iterativos y algoritmos de optimización en función de la naturaleza y la complejidad del problema específico que esté tratando de resolver. Los detalles exactos de los algoritmos específicos utilizados están integrados en la implementación del software y no están expuestos directamente al usuario. Las características o criterios predeterminados de esta herramienta son: precisión de restricciones de 0.000001, convergencia de 0.0001 y valor de inicialización aleatorio de 0.

Por otra parte, el complemento ReCongSolver utiliza un método de región de confianza para buscar la solución de un sistema de ecuaciones no lineales, utilizando como plataforma una hoja de cálculo de Microsoft Excel® para el planteamiento y solución del problema (Díaz Montes & Barrera, s. f.). El algoritmo que se sigue se fundamenta en el método de región de confianza junto con el método de Dogleg.

Por consiguiente, para los modelos 3.10 al 3.17 se utilizaron los complementos Solver y ReCongSolver ya que ofrecen la oportunidad de obtener de manera rápida e infalible soluciones relativamente cercanas al conjunto de valores iniciales que se estableció para el sistema de funciones.

Para el modelo 3.18 también se usaron los complementos anteriores, junto con la función `fsolve` de MATLAB. El código para la ecuación 3.18, que busca minimizar la ecuación 2.15 se muestra en sección de anexos (Tabla 0.1).

Búsqueda de valores solución a partir de la variación de valores iniciales

El objetivo de esta etapa era verificar la existencia de múltiples valores que minimizaran la función objetivo de los modelos cinéticos y evaluar cómo estos valores variaban con los ajustes de los parámetros iniciales.

Para llevar a cabo esta exploración, se definieron los parámetros del sistema cinético y la función objetivo correspondiente. A través de un proceso iterativo, se modificaron los valores

iniciales de los parámetros del sistema y se ejecutaron las herramientas anteriormente mencionadas, para encontrar los valores que minimizaban la función objetivo. Cabe destacar que este enfoque de búsqueda no pretendía ser exhaustivo ni determinar soluciones óptimas, sino más bien explorar la diversidad de soluciones que podrían existir en el espacio de parámetros.

Se generaron múltiples conjuntos de valores iniciales para los parámetros del sistema, cada uno de los cuales representaba una configuración diferente. En la tabla 3.5 algunos de los diferentes valores iniciales para cada modelo del 3.10 al 3.17.

Tabla 3.4 Conjuntos de valores iniciales para los modelos del 3.10 al 3.17

Modelo	$\mu_{\text{máx}}$	K_s	K_i	S_m	m	n	Número de combinaciones de valores iniciales exploradas
Tessier (Ec. 3.10)	0,5, 10, 15, 20, 25, 30, 35, 40, 45, 50.	5, 10, 50, 100, 150, 200, 250, 300, 350, 400.	N.A	N. A	N.A	N.A	100
Monod (Ec. 3.11)	0,5, 10, 15, 20, 25, 30, 35, 40, 45, 50.	5, 10, 50, 100, 150, 200, 250, 300, 350, 400.	N.A	N. A	N.A	N.A	100
Moser (Ec. 3.12)	20, 30, 40, 45, 50.	5, 7, 10, 20, 40, 50, 75, 100.	N.A	N. A	N.A	0,5, 1, 1.5, 2.	160
Haldane (Ec. 3.13)	100, 125, 150, 175, 200, 225, 250.	750, 1000, 1250, 1500, 2000	150, 300, 450, 650.	N. A	N.A	N. A	140
Aiba (Ec. 3.14)	50, 80, 100, 150, 175, 200, 250, 300.	1000, 1500, 1750, 2000.	1000, 1250, 1500, 1750, 2000.	N. A	N.A	N. A	160
Edwards (Ec. 3.15)	150, 300, 450, 650.	500, 750, 1000, 1250, 1500	500, 750, 1000, 1250, 1500	N. A	N.A	N. A	100
Luong (Ec. 3.16)	150, 300, 450, 650.	4000, 5000, 6000, 7000.	N. A	1000, 1500, 1750, 2000.	0,2, 0,6, 0,8, 1, 1,5, 2.	N. A	384
			N. A	1000, 1500,			9216

Han	y	0,5, 10, 15, 5, 10, 50,				
Levenspiel		20, 25, 30, 100, 150,	1750, 2000.	-8, -6, -4, -2.	-8, -6, -4, -2.	
(Ec. 3.17)		200		2, 4, 6, 8	2, 4, 6, 8	

N.A: No aplica

La selección de los valores iniciales no se realizó de manera aleatoria, sino que, por el contrario, se tuvo en cuenta la tendencia, comportamiento y características químicas de cada modelo cinético. En “A new kinetic model to predict substrate inhibition and better efficiency in an airlift reactor on deammonification process” se reportaron resultados para todos los parámetros estudiados, por lo que a partir de ellos se estableció un amplio rango para la búsqueda de otras posibles soluciones.

En cuanto al modelo descrito por la ecuación 3.18, también se trabajó con un gran conjunto de valores iniciales ya que al ser un modelo con cuatro parámetros implica mayor cantidad de combinaciones. Los valores iniciales para este modelo se encuentran en la tabla 3.6.

Tabla 3.5 Rango de valores iniciales para el modelo 3.18

Modelo	a₀	a₁	a₂	a₃
Ec. 3.18	[0.01-20]	[0.1-4]	[0.01-1]	[1-3]

Para la selección de los diferentes valores iniciales se tuvo en cuenta el comportamiento de la velocidad de reacción frente a la variación de cada parámetro. Para a_0 , que representa la velocidad de reacción cuando la concentración del reactivo es muy baja, se establecieron valores iniciales desde 0.01 a 20 con saltos de una unidad (1), ya que típicamente a_0 sería un número positivo pequeño, pero no necesariamente entero. Para a_1 , se analizaron valores de 0.1 a 4 ya que al a_1 ser positivo indica una relación directamente proporcional entre la concentración del reactivo A y su velocidad de reacción. Por otro lado, un valor positivo de a_2 en la ecuación cinética indica que la concentración del reactivo tiene un efecto activador sobre la velocidad de reacción, lo que significa que a medida que C_A aumenta, la velocidad de reacción también aumenta. La magnitud

de a_2 depende de la fuerza de esta influencia activadora. En modelos cinéticos similares, el rango común para el parámetro a_2 suele estar entre 0.01 y 0.5, por lo que este parámetro se analizó en el intervalo de 0.01 a 1.

Por último, el parámetro a_3 se establecieron valores enteros de 1 a 3, ya que en modelos similares este parámetro cinético adquiere valores enteros en este intervalo, pues debido a la naturaleza de los sistemas químicos puede existir saturación gradual de los sitios de reacción ($a_3 \approx 1$) o situaciones más complejas de interacción entre los reactantes ($a_3 \approx 2$) o ($a_3 \approx 3$).

A medida que se ejecutaba Solver de Microsoft Excel, se registraban los valores de los parámetros resultantes y los correspondientes valores de la función objetivo. Estos conjuntos de resultados proporcionaron una visión inicial de cómo los valores iniciales influían en las soluciones obtenidas y permitieron identificar patrones de comportamiento.

Durante la ejecución repetida del código variando los valores iniciales, se lograron obtener diferentes resultados para los parámetros a_0 , a_1 , a_2 y a_3 . Estas soluciones fueron registradas para luego, intentar obtenerlas por el método homotópico implementado. Es importante destacar que se realizaron los análisis y las interpretaciones correspondientes a los resultados obtenidos, considerando las limitaciones y las implicaciones de cada solución encontrada.

Esta fase exploratoria proporcionó una base valiosa para el posterior enfoque utilizando la homotopía con hiperesferas. Al tener una comprensión inicial de cómo los valores iniciales afectan las soluciones, se pudo diseñar un enfoque más estratégico para explorar una gama más amplia de soluciones utilizando métodos avanzados.

Algoritmo de la homotopía tipo Newton con el esquema predictor corrector.

En la tabla 3.7 se plantea el algoritmo utilizado para la solución del sistema de ecuaciones no lineales de la homotopía Newton, con el uso del predictor de Newton, y utilizando como corrector el método de Newton-Raphson (Montes, 2019).

Tabla 3.6 Algoritmo método de homotopía Newton con rastreo mediante hiperesferas y método de Newton-Raphson (Montes, 2019)

Entrada: \mathbf{x}_0 , n , $\mathbf{f}(\mathbf{x})$, ε , R_{init} , Cp , TT , I_{max} , t_l (opcional) y t_u (opcional)

Pasos

1. Inicialización: $t_0 = 0$, $p_0 = 0$, $k = 0$

1.1 Cálculo de $\|\mathbf{f}(\mathbf{x}_0)\|$. Si $\|\mathbf{f}(\mathbf{x}_0)\| < \varepsilon$ Entonces SALIR

2. Cálculo:

$\mathbf{J}_f(\mathbf{x}_0)$: calculada con diferencias finitas de segundo orden de error. Para la Homotopía de Newton: $\mathbf{J}_h = \mathbf{J}_f(\mathbf{x}_0)$.

$$\frac{dt}{dp} = (-1)^{n+1} \det(\mathbf{J}_h)$$

2.1 Control de terminación

Dominio automático de t :

Si $(|t| > 100n \text{ and } \left|\frac{dt}{dp}\right| \geq 10^n)$ or $(\left|\frac{dt}{dp}\right| > 10^{10})$ Entonces, el seguimiento de esta rama de la ruta Homotópica termina. Si solo una rama ha sido recorrida, Entonces regresar al paso 1 para seguir la otra rama. Si ambas ramas ya fueron recorridas (valores positivos y negativos de p), Entonces reportar resultados y SALIR

Dominio fijo de t :

Si t se encuentra por fuera del intervalo $t_l < t < t_u$, Entonces el seguimiento de esta rama de la ruta Homotópica termina. Si solo una rama ha sido recorrida, Entonces regresar al paso 1 para seguir la otra rama. Si ambas ramas ya fueron recorridas (valores positivos y negativos de p), Entonces reportar resultados y SALIR.

2.2 Control de retorno: Cuando $\frac{dt}{dp}$ cambia de signo, se alcanza un punto de retorno. Al ocurrir esto, el radio es reducido a la mitad y el algoritmo regresa a la hiperesfera anterior hasta que $\left|\frac{dt}{dp}\right| < Cp$ en ambos lados del punto de retorno.

3. Con los elementos de los anteriormente encontrados, se utiliza una subrutina del método de eliminación Gaussiana para encontrar $\frac{dx}{dp}$ del problema lineal $\mathbf{J}_h \frac{\partial \mathbf{x}}{\partial p} = -\frac{\partial t}{\partial p} \cdot \frac{\partial \mathbf{h}}{\partial t}$, con $\frac{\partial \mathbf{h}}{\partial t} = \mathbf{f}(\mathbf{x}_0)$ para la Homotopía de Newton aquí utilizada.

4. El radio de la hiperesfera es ajustado con el siguiente criterio heurístico

$$R_k = \begin{cases} R_{init} & \text{cuando } p = 0 \\ R_{k-1}[P_1 - (P_1 - P_2)] \exp\left(-P_3 \left|\frac{dt}{dp_{k-1}}\right| \left|\frac{dt}{dp}\right|\right) & \text{cuando } p \neq 0 \end{cases}$$

El subíndice $k - 1$ indica elementos de la iteración anterior a la actual.

5. Los valores iniciales t^0 y x^0 para el método de convergencia local de Newton son calculados utilizando el paso predictor de Euler

$$\Delta p = \pm \sqrt{\frac{R^2}{1 + \left\|\frac{d\mathbf{x}}{dp}\right\|^2 + \left(\frac{dt}{dp}\right)^2}}$$

$$t^0 = t + \Delta p \frac{dt}{dp}, \quad \mathbf{x}^0 = \mathbf{x} + \frac{d\mathbf{x}}{dp} \Delta p$$

(Continúa siguiente página)

(Continuación)

6. Paso corrector: Los valores de t^0 y \mathbf{x}^0 alimentan el método de Netwon–Raphson para resolver el sistema de ecuaciones:

$$\mathbf{h}(t, \mathbf{x}) = \mathbf{f}(\mathbf{x}) - (1 - t)\mathbf{f}(\mathbf{x}_0) = \mathbf{0}$$

$$c = \|\mathbf{x} - \mathbf{x}_{k-1}\|^2 + (t - t_{k-1})^2 - R^2 = 0,$$

e.g.,

$$\mathbf{h}^s = \begin{bmatrix} \mathbf{h} \\ c \end{bmatrix} = \begin{bmatrix} \mathbf{f}(\mathbf{x}) - (1 - t)\mathbf{f}(\mathbf{x}_0) \\ \|\mathbf{x} - \mathbf{x}_{k-1}\|^2 + (t - t_{k-1})^2 - R^2 \end{bmatrix} = \mathbf{0}.$$

La matriz jacobiana necesaria para la implementación del método es

$$\mathbf{J}_{\mathbf{h}^s} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \dots & \frac{\partial h_1}{\partial x_n} & \frac{\partial h_1}{\partial t} \\ \vdots & \ddots & \vdots & \vdots \\ \frac{\partial h_n}{\partial x_1} & \dots & \frac{\partial h_n}{\partial x_n} & \frac{\partial h_n}{\partial t} \\ \frac{\partial c}{\partial x_1} & \dots & \frac{\partial c}{\partial x_n} & \frac{\partial c}{\partial t} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{\mathbf{f}} & \mathbf{f}(\mathbf{x}_0) \\ 2(x_1 - x_1^c) & \dots & 2(x_n - x_n^c) & 2(t - t_c) \end{bmatrix}$$

$$\mathbf{J}_{\mathbf{h}^s} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} & f_1(x_1^0, \dots, x_n^0) \\ \vdots & \ddots & \vdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} & f_n(x_1^0, \dots, x_n^0) \\ 2(x_1 - x_1^c) & \dots & 2(x_n - x_n^c) & 2(t - t_c) \end{bmatrix}$$

y se calcula mediante diferencias finitas de segundo orden de error. Si después de I_{max} iteraciones el sistema no satisface el criterio de convergencia $\|\mathbf{h}^s\| < \varepsilon$, Entonces el radio de la hiperesfera es reducido a la mitad y el algoritmo continúa en el paso 5.

- 6.1 Si la ruta Homotópica cruza el hiperplano $t = 1$, los valores de t ambos lados del hiperplano deben satisfacer $|1 - t| < TT$. Si esta condición no se satisface, Entonces el radio es reducido a la mitad y el algoritmo continúa en el paso 5. Si la condición si se satisface, Entonces se almacena la raíz encontrada.

El valor encontrado en el paso 6 es el nuevo punto in la ruta homotópica que se está discretizando y el centro de la siguiente hiperesfera. El algoritmo continúa en el paso 2 utilizando este nuevo punto.

3.4 Creación y ejecución de los programas

Los diferentes códigos implementados de cada uno de los métodos o estrategias numéricas utilizadas fueron ejecutados en una unidad con sistema operativo Windows 11, un procesador Intel (R) Core (TM) i5-1135G7 CPU @2.40GHz 2.42 GHz y una memoria RAM 8.00 GB. Se utilizará

el software Matlab [versión: 2021b (9.11.0.1809720), 64-bit (win64), November 02, 2021, License Number 40903832].

El código para aplicar la homotopía esférica se diseñó y ejecutó en ambos softwares, sin embargo, se decidió trabajar con los resultados en Microsoft Excel, ya que permite navegar en una interfaz más amigable y de mayor entendimiento para el usuario. Adicionalmente, el código diseñado en Visual Basic for Applications de Microsoft Excel permite realizar un seguimiento a la ruta homotópica pues se observa la variación del parámetro t , la norma de la función, el número de esferas y el número de raíces encontradas. El código desarrollado para la homotopía esférica se muestra en la tabla 0.2 de la sección Anexos.

4. Resultados

En la tabla 4.1 se presenta el número de soluciones encontradas para cada uno de los modelos.

Tabla 4.1 Número de soluciones encontradas para cada modelo

Modelo	Número de soluciones encontradas		
	1	2	3
Tessier (3.10)	✓		
Monod (3.11)	✓		
Moser (3.12)	✓		
Haldane (3.13)	✓		
Aiba (3.14)	✓		
Edwards (3.15)	✓		
Luong (3.16)	✓	✓	
Han y Levenspiel (3.17)	✓		
Problema 2 (3.18)	✓	✓	✓

El número de raíces fueron corroboradas mediante las tres herramientas (Solver, ReCongSolver y fsolve de MATLAB), esto con el fin de validar la veracidad de los valores encontrados.

En la tabla 4.2 se observan las raíces encontradas para los modelos de única solución.

Tabla 4.2 Valores encontrados para los modelos con única solución

Modelo	Parámetro					
	μ_{max}	K_s	K_i	S_m	m	n
Tessier (3.10)	33.71	293.50				
Monod (3.11)	43.48	322.81				
Moser (3.12)	35.28	12028.86				1.73
Haldane (3.13)	156.09	1576.41	435.49			
Aiba (3.14)	470.62	4179.28	948.64			
Edwards (3.15)	443.11	719.71	878.99			
Han y Levenspiel (3.17)	12.66	121.66		1314.92	-8.14	7.34

En la tabla 4.3 se observan las diferentes soluciones que fueron obtenidas mediante los complementos Solver y ReCongSolver en Microsoft Excel y mediante la función fsolve de MATLAB.

Tabla 4.3 Múltiples soluciones para los modelos 3.16 y 3.18

Modelo	Parámetro	Solución 1	Solución 2	Solución 3
Luong (3.16)	μ_{max}	363.57	795.27	
	K_s	3676.76	9476.89	
	S_m	1110.91	1235.79	
	m	0.43	0.47	
Problema 2 (3.18)	a_0	1.42	7.74	12.82
	a_1	2.08	2.70	2.99
	a_2	0.30	1.30	1.53
	a_3	3.00	3.00	3.00

Con base en los resultados obtenidos en la tabla 4.3 se procedió a ejecutar el código de la homotopía de Newton mediante hiperesferas descrito en la tabla 3.8, esto con el fin de obtener los múltiples valores solución previamente explorados con las herramientas de Microsoft Excel y de MATLAB. En esta etapa del proceso se partió de los mismos valores iniciales reportados en las tablas 3.5 y 3.6 de la sección 3.4 de la metodología, sin embargo, para ambos modelos el programa no convergió.

Tras navegar reiterativamente en el amplio rango establecido y después de modificar ligeramente condiciones previamente pautadas como la tolerancia y el dominio para el parámetro t , se decidió para el modelo 3.18 fijar el parámetro a_3 en el valor de tres (3). Esta decisión se tomó teniendo en cuenta la naturaleza y la tendencia de a_3 , ya que en cada iteración o prueba se comportó como un parámetro cinético típico de un sistema donde los reactantes pueden tener una interacción compleja ($a_3 \approx 3$).

Los resultados obtenidos se muestran en la tabla 4.4.

Tabla 4.4 Múltiples soluciones para modelos 3.16 y 3.18

Modelo	Parámetro	Solución 1	Solución 2	Solución 3
	μ_{max}			
Luong (3.16)	K_s	No convergencia		
	S_m			
	m			
	a_0			
Problema 2 (3.18)	a_1	2.08	1.24	-0.94
	a_2	0.30	0.08	-0.06
	a_3	3.00	3.00	3.00

Por lo anterior, el modelo 3.18 demostró convergencia en tres raíces únicamente cuando el parámetro a_3 fue restringido. Las especificaciones de la ruta homotópica como el tiempo de ejecución, número de esferas y norma de la función se observan en la tabla 4.5.

Tabla 4.5 Indicadores en cada solución encontrada.

Modelo	Parámetro	Solución 1	Solución 2	Solución 3
	Norma	9.26E-03	9.29E-03	7.63E-06
Problema 2 (3.18)	t	0.9995	0.9996	1.0000
	r^2	0.9895	0.9928	-1.56
	Número de esferas	220	228	1096

En la figura 4.1 se observa el comportamiento del parámetro t (eje y) a lo largo del avance de la ruta homotópica p (eje x).

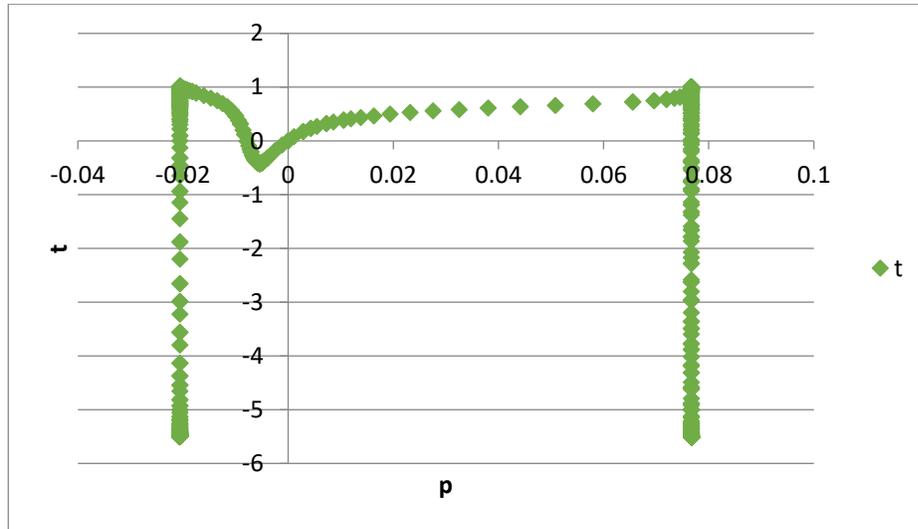


Figura 4.1 Parámetro t en función del avance de la ruta homotópica (p)

En la figura 4.2 se observa el comportamiento del parámetro a_0 (eje y) en función de t (eje x).

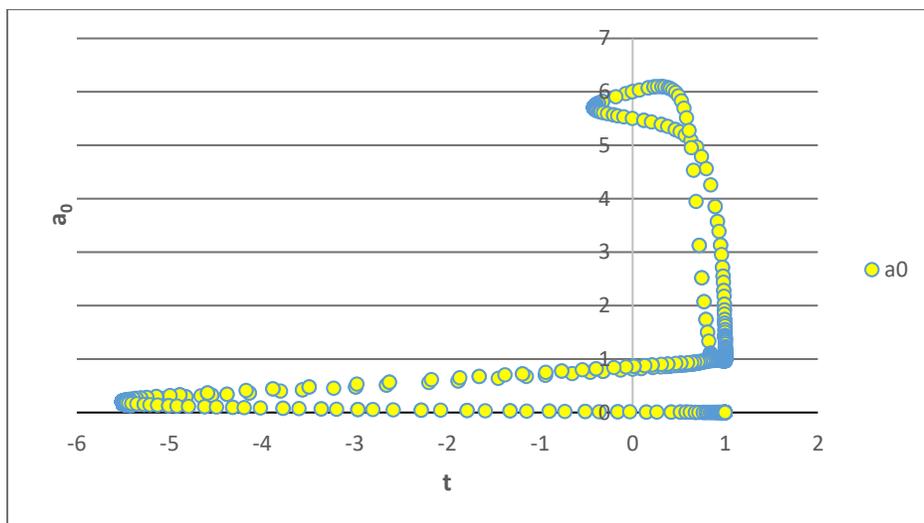


Figura 4.2 Comportamiento de a_0 en función de t .

En la figura 4.3 se observa el comportamiento del parámetro a_1 (eje y) en función de t (eje x).

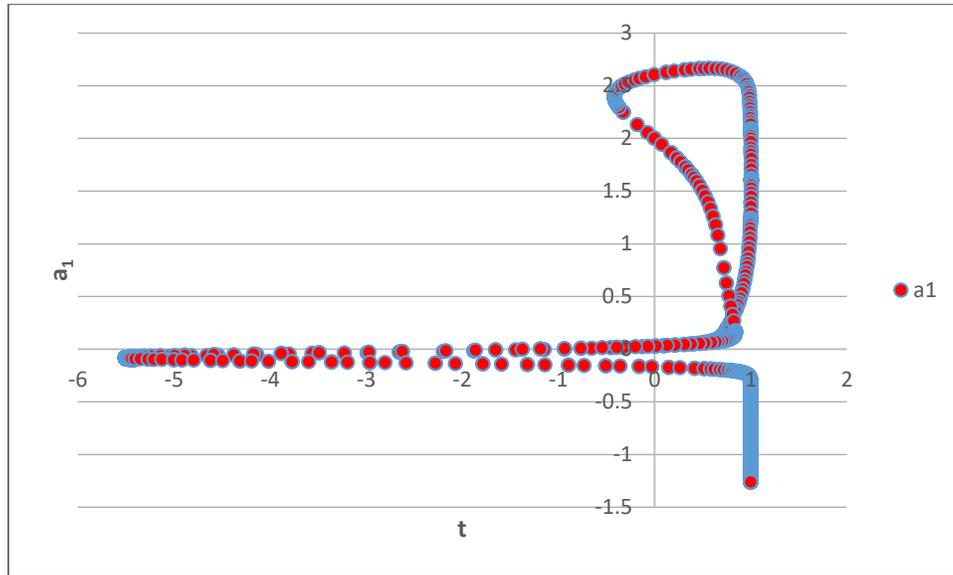


Figura 4.3 Comportamiento de a_1 en función de t .

En la figura 4.4 se observa el comportamiento del parámetro a_2 (eje y) en función de t (eje x).

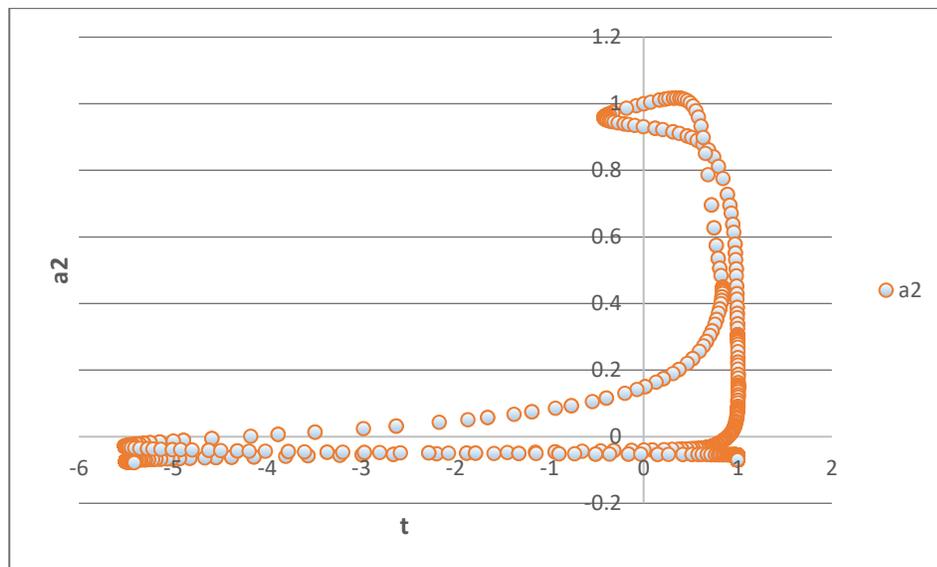


Figura 4.4 Comportamiento de a_3 en función de t .

5. Análisis

El tratamiento previo que se realizó a los modelos 3.10-3.18 mediante los complementos Solver y ReCongSolver de Microsoft Excel y la función fsolve de MATLAB permitió realizar una decantación de intervalos y de valores iniciales donde los modelos no convergían a ninguna solución satisfactoria.

Pese a que no se obtuvieron resultados favorables para más de una solución en los modelos 3.10, 3.11, 3.12, 3.13, 3.14, 3.15, 3.17, se decidió aplicar el método homotópico en los modelos 3.16 y 3.18 ya que éstos sí habían arrojado múltiples valores solución para los parámetros.

Durante la ejecución del programa descrito en la tabla 3.8, el modelo de Luong (3.16) presentó dificultades ya que el parámetro t tenía una tasa de avance despreciable, así como la norma de la función no mostraba tendencia decreciente. Cabe aclarar que, esta problemática con el modelo se presentó tomando todos los valores iniciales descritos en la tabla 3.5.

A partir de los resultados obtenidos, se puede evidenciar la sensibilidad del método homotópico empleado frente a variaciones en los valores iniciales o en características establecidas para la ejecución del programa como la tolerancia o el dominio para t .

En cuanto al modelo 3.18, la búsqueda del conjunto de valores iniciales también supuso un gran reto, esto porque inicialmente la exploración simultánea de los cuatro parámetros (a_0 , a_1 , a_2 , a_3) en el amplio conjunto de los números reales resultó compleja e imposible de asentar con las herramientas informáticas al alcance.

Es por esto que, al establecer como necesaria la restricción para el parámetro a_3 permite al programa navegar en un conjunto más acotado, disminuyendo el tiempo de convergencia y el número de esferas empleadas para encontrar las soluciones descritas en la tabla 4.4.

La figura 4.1 permite observar el comportamiento del parámetro t en función del avance de la ruta homotópica p , por lo que resulta importante analizar el número de veces donde t es igual a 1, ya que concuerda con el número de raíces encontradas.

En cuanto al dominio tomado para t fue de -5.5083 a 1.0054, cabe aclarar que el programa siguió esta ruta ya que estaba activada la opción de autodomínio, lo que le permitió acotar el intervalo de manera automática gracias al predictor-corrector de Euler.

En cuanto al comportamiento de los parámetros a_0 , a_1 y a_2 en las figuras 4.2, 4.3 y 4.4 se logra observar un patrón en la ruta homotópica para cada parámetro; esto puede sugerir que a medida que t va variando, los valores de los parámetros a_0 , a_1 y a_2 también cambian de manera gradual y coherente. Este patrón refleja cómo las soluciones del modelo cinético (3.18) evolucionan de manera sistemática a lo largo de la ruta homotópica.

Los puntos exactos donde a_0 , a_1 y a_2 tocan a $t=1$ simultáneamente corresponden a los valores solución. Cabe señalar que la solución 3 para el modelo cinético fue descartada debido a la magnitud de los valores arrojados, especialmente el de a_0 , ya que su valor muy cercano a cero corresponde a un mínimo local o a un punto estacionario que no es un mínimo global verdadero, ya que no es una solución válida o significativa en el contexto del problema original.

Se intentó obtener las soluciones (1) y (2) a partir de diversos valores iniciales, sin embargo, en ninguna de las combinaciones probadas (tabla 3.6) el programa convergió.

Esta sensibilidad del método a pequeñas variaciones en los valores iniciales puede indicar que el método es altamente dependiente de las condiciones iniciales. Esto puede dificultar la replicación de resultados y la obtención de soluciones consistentes. Además, la falta de convergencia ante variaciones sugiere que el método puede ser inestable en ciertas regiones del espacio de parámetros.

6. Conclusiones

En primer lugar, se logró una comprensión profunda de los modelos cinéticos ampliamente utilizados en la ingeniería química. Esta búsqueda bibliográfica proporcionó los fundamentos teóricos necesarios para abordar los objetivos propuestos, pues al examinar los modelos cinéticos existentes, se estableció un contexto para evaluar su aplicabilidad y las limitaciones en el trabajo de trabajo de grado propuesto. En pocas palabras, la selección adecuada de modelos cinéticos es fundamental para garantizar la precisión en la representación de reacciones químicas complejas y poder establecer las bases para su ajuste y optimización.

La búsqueda bibliográfica permitió identificar posibles brechas en el conocimiento actual. Esto condujo a la identificación de áreas poco exploradas en el campo de la ingeniería química y a la detección de enfoques innovadores que podrían aplicarse en el desarrollo de la investigación de sistemas químicos que presentan múltiples soluciones.

La implementación exitosa de la técnica de homotopías con hipersferas para ajustar modelos cinéticos fue un logro significativo. La homotopía seleccionada proporcionó una ruta sistemática para transformar ecuaciones no lineales en sistemas más manejables y lineales, facilitando así su solución. Este enfoque demostró ser una herramienta poderosa para abordar la no linealidad intrínseca en los modelos cinéticos, además de que la obtención de múltiples resultados precisos y consistentes es esencial para tomar decisiones fundamentales en el diseño y optimización de procesos químicos y bioquímicos.

La exploración reveló múltiples combinaciones de valores que minimizaban la función objetivo del sistema cinético. La sensibilidad a pequeñas variaciones iniciales fue un aspecto destacado, subrayando la presencia de óptimos locales en el espacio de solución. La evaluación exhaustiva del comportamiento y los resultados de las regresiones respaldó la robustez de la técnica de homotopías para lograr un ajuste confiable de los modelos cinéticos. A pesar de la singularidad de la matriz jacobiana, el enfoque demostró ser efectivo en la obtención de soluciones válidas y consistentes.

No obstante, la falta de robustez frente a variaciones en los valores iniciales limita la capacidad para un análisis exhaustivo. La posibilidad de obtener las mismas raíces con diferentes condiciones iniciales es crucial para validar y comprender la solución en una amplia gama de situaciones. Para mejorar este aspecto, se sugiere abordar la inestabilidad numérica derivada de la singularidad de la matriz jacobiana, mediante técnicas de precondicionamiento, optimización en la formulación de las ecuaciones y la selección de algoritmos numéricos más estables.

En pocas palabras, la implementación exitosa de los métodos homotópicos para ajustar modelos cinéticos y resolver sistemas no lineales destaca como un logro importante en el campo de la ingeniería. Estos enfoques abren nuevas puertas para abordar desafíos complejos y aprovechar al máximo el potencial de los métodos homotópicos para encontrar múltiples soluciones que se apliquen en el diseño y en la optimización de procesos ingenieriles.

Referencias

- Allgower, E., & Kurt, G. (2012). *Introduction to Numerical Continuation Methods*.
<http://avaxhome.ws/blogs/ChrisRedfield>
- Asadi, J., & Jalali Farahani, F. (2018). Optimization of dimethyl ether production process based on sustainability criteria using a homotopy continuation method. *Computers and Chemical Engineering*, *115*, 161-178.
<https://doi.org/10.1016/j.compchemeng.2018.03.014>
- Bonassa, G., Chiapetti Bolsan, A., Venturin, B., Celant De Prá, M., Goldschmidt Antes, F., Ester Hollas, C., Johann, G., Coldebella, A., & Kunz, A. (2021). A new kinetic model to predict substrate inhibition and better efficiency in an airlift reactor on deammonification process. *Bioresource Technology*, *319*. <https://doi.org/10.1016/j.biortech.2020.124158>
- Chang, Y. A., & Seader, J. D. (1988). SIMULATION OF CONTINUOUS REACTIVE DISTILLATION BY A HOMOTOPY-CONTINUATION METHOD. En *Comput. them. Engng* (Vol. 12, Número 12).
- Díaz, Luis. (2022). *M. Especiales Unidad 5. Aproximación funcional e interpolación*.
- Díaz Montes, L., & Barrera, R. (s. f.). Manual de usuario RegConSolver. 2019.
- Fogler, H. S. (2010). Fogler, H. Scott - Essentials of Chemical Reaction Engineering-Prentice Hall (2010). *Prentice Hall*.
- Jiménez-Islas, H. (1996). *SEHPE: Programa para la solución de sistemas de ecuaciones no lineales mediante método homotópico con seguimiento hiperesférico* *Rheological modelling View project PHYSICAL, BARRIER, MECHANICAL AND MICROSTRUCTURAL PROPERTIES OF Aloe vera-GELATIN-GLYCEROL EDIBLE FILMS INCORPORATED WITH Pimenta dioica L. MERRILL ESSENTIAL OIL View project*. <https://www.researchgate.net/publication/273441761>
- Montes, L. A. (2019). *Simulaciones en Ingeniería Química potenciadas por Métodos Homotópicos*.
- Quemada, M., López, M. de la L., Oliveros, J. M., & Jiménez, H. (2022). *Enseñanza del método de continuación homotópica con seguimiento hiperesférico para estudiantes de ingeniería*.

Wayburns, T. L., & Seader, J. D. (1987). HOMOTOPY CONTINUATION METHODS FOR COMPUTER-AIDED PROCESS DESIGN. En *comput. &em. &gag* (Vol. 11, Número 1).

Anexos

Tabla 0.1 Código en MATLAB para minimizar la ecuación 2.15 del modelo 3.18

```

function gradssres = fcnapplication5(vari)
% Summary of this function goes here
% Detailed explanation goes here
C_A = [0.2 0.7 2 4.5 7.1 8.7]; % Vector x
r_A = [0.088 0.529 1.441 2.476 2.517 2.876]; % Vector y

a_0 = vari(1);
a_1 = vari(2);
a_2 = vari(3);
a_3 = vari(4);

fun1 = (r_A - (a_0 .* C_A .^ a_1) ./ (1 + a_2 .* C_A).^a_3) .* ((C_A .^ a_1) ./ (1 +
a_2 .* C_A).^a_3);
% Calcula la sumatoria del numerador
suma_fun1 = sum(fun1);

gradssres(1) = -2 * suma_fun1;

fun2 = (r_A - (a_0 .* C_A .^ a_1) ./ (1 + a_2 .* C_A).^a_3) .* ((a_0 .* C_A .^ a_1 .* log
(C_A)) ./ (1 + a_2 .* C_A).^a_3);

suma_fun2 = sum(fun2);

% Calcula el resultado de la función
gradssres(2) = -2 * suma_fun2;

fun3 = (r_A - (a_0 .* C_A .^ a_1) ./ (1 + a_2 .* C_A).^a_3) .* (-a_0 * a_3 .* C_A .^ (a_1 +
1) .* (1 + a_2 .* C_A).^((-a_3)-1));

suma_fun3 = sum(fun3);

% Calcula el resultado de la función
gradssres(3) = -2 * suma_fun3;

fun4 = (r_A - (a_0 .* C_A .^ a_1) ./ (1 + a_2 .* C_A).^a_3) .* (-a_0 .* C_A .^ a_1 .* log(1 +
a_2 .* C_A) .* (1 + a_2 .* C_A).^(-a_3));

```

```
suma_fun4 = sum(fun4);  
  
% Calcula el resultado de la función  
gradssres(4) = -2 * suma_fun4;  
  
end
```

Tabla 0.2 Código de la homotopía con hiperesferas en Visual Basic for Application de Microsoft Excel. Tomado de (Montes, 2019)

Option Explicit

Public Functions As Range

Public Variables As Range

Public N As Integer

Public RInitial As Double

Public Tol As Double

Public MaxIter As Integer

'Public InicializarX0ConValor As Boolean

Public ConstanteX0 As Double

Public Salir As Boolean

Public ShowHPath As Boolean

Public ReturnTol As Double

Public T1Tol As Double

Public AutTDomain As Boolean

Public Low_t As Double

Public Up_t As Double

Public TimeRec As Boolean

Dim ExecTime As Double

Public RootsX() As Double

Public NRootsX As Integer

Public ErrNumber As Long

Dim HomotopicPathp() As Double

Dim HomotopicPatht() As Double

Dim HomotopicPathX() As Double

Dim OldStatusBar As String

Dim NSpheres As Long

Public X0() As Double

Const ErrorMaquina As Double = 0.000000000000001

Dim EsSingular As Boolean

Sub PrincipalHomotoSolver()

Dim Iter As Integer

Dim i As Integer

Dim j As Integer

Dim p As Double

Dim p_1 As Double

Dim tc As Double

Dim tc_1 As Double

Dim Xc() As Double

Dim Xc_1() As Double

Dim X() As Double 'Used for Newton-Raphson

Dim t As Double 'Used for Newton-Raphson

Dim JFx() As Double 'Used for Newton-Raphson

Dim FuncX0() As Double

```
Dim FuncX() As Double
Dim NormFuncX As Double
Dim B() As Double
Dim JFxc() As Double
Dim JFxc_1() As Double
Dim dXdpc() As Double
Dim dXdpc_1() As Double
Dim FHY() As Double
Dim NormFHY As Double
Dim JHY() As Double
Dim SNewt() As Double
Dim NormFuncX0 As Double 'FuncX0 norm
Dim R As Double
Dim dtdpc As Double
Dim dtdpc_1 As Double
Dim SecondBranch As Boolean
Dim RepeatSphere As Boolean
Dim Deltap As Double
Dim PathBranchTermination As Boolean

Salir = True
'Input routine
EntradaDeDatos

If Salir = True Then Exit Sub

ReDim X0(1 To N) As Double
ReDim Xc(1 To N) As Double
ReDim Xc_1(1 To N) As Double
```

```
ReDim X(1 To N) As Double
ReDim JFx(1 To N, 1 To N) As Double
ReDim FuncX0(1 To N) As Double
ReDim FuncX(1 To N) As Double
ReDim B(1 To N) As Double
ReDim JFxc(1 To N, 1 To N) As Double
ReDim JFxc_1(1 To N, 1 To N) As Double
ReDim dXdpc(1 To N) As Double
ReDim dXdpc_1(1 To N) As Double
ReDim FHY(1 To N + 1) As Double
ReDim JHY(1 To N + 1, 1 To N + 1) As Double
ReDim SNewt(1 To N + 1) As Double
```

```
Application.ScreenUpdating = False
    Application.EnableCancelKey = xlErrorHandler
On Error GoTo ErrorHandler
If TimeRec = True Then
    ExecTime = Timer
End If
```

```
ErrNumber = 65534 'Random high value
```

```
For i = 1 To N
    X0(i) = Variables.Cells(i).Value
    Xc(i) = Variables.Cells(i).Value
Next i
```

```
Call FX(X0, FuncX0)
```

```
NormFuncX0 = 0
```

```
For i = 1 To N
    NormFuncX0 = NormFuncX0 + (FuncX0(i)) ^ 2
Next i
NormFuncX0 = Sqr(NormFuncX0)

If NormFuncX0 <= Tol Then
    MsgBox "El valor inicial es una raíz."
    Exit Sub
End If

Initialization
SecondBranch = False
NSpheres = 0
NRootsX = 0
OldStatusBar = Application.DisplayStatusBar
Application.DisplayStatusBar = True

p = 0
tc = 0
RepeatSphere = False
Do
    If ErrNumber <> 65534 Then Exit Do
    Do
        If ErrNumber <> 65534 Then Exit Do
        If RepeatSphere = False Then
            'If tc > 0.9 Then
                'Elements to display the advance of the method
                If p = 0 Then NormFuncX = NormFuncX0
                If p <> 0 Then
```

```

NormFuncX = 0

For i = 1 To N

    NormFuncX = NormFuncX + (FuncX(i)) ^ 2

Next i

NormFuncX = Sqr(NormFuncX)

End If

If Abs(tc) < 0.001 Or Abs(tc) > 999 Then 'The display format of tc is different depending of its
magnitude

    If SecondBranch = False Then Application.StatusBar = "Presione Esc lentamente una o dos veces
para cambiar la dirección en la ruta homotópica. " & "Norma=" & (Format(NormFuncX, "0.00E+0")) & "
t=" & (Format(tc, "0.000E+0")) & " N. Esferas=" & (NSpheres) & " N. Raíces=" & (NRootsX)

    If SecondBranch = True Then Application.StatusBar = "Presione Esc lentamente una o dos veces
para detener el programa. " & "Norma=" & (Format(NormFuncX, "0.00E+0")) & " t=" & (Format(tc,
"0.000E+0")) & " N. Esferas=" & (NSpheres) & " N. Raíces=" & (NRootsX)

Else

    If SecondBranch = False Then Application.StatusBar = "Presione Esc lentamente una o dos veces
para cambiar la dirección en la ruta homotópica. " & "Norma=" & (Format(NormFuncX, "0.00E+0")) & "
t=" & (Format(tc, "0.0000")) & " N. Esferas=" & (NSpheres) & " N. Raíces=" & (NRootsX)

    If SecondBranch = True Then Application.StatusBar = "Presione Esc lentamente una o dos veces
para detener el programa. " & "Norma=" & (Format(NormFuncX, "0.00E+0")) & " t=" & (Format(tc,
"0.0000")) & " N. Esferas=" & (NSpheres) & " N. Raíces=" & (NRootsX)

End If

'Continuing with the method

Call dFdX(Xc(), JFxc())

If ErrNumber <> 65534 Then Exit Do 'Termination treatment

With Application.WorksheetFunction

    dtdpc = .MDeterm(JFxc)

End With

dtdpc = (-1) ^ (N + 1) * dtdpc

If p = 0 Then

    R = RInitial

    For i = 1 To N

        Xc_1(i) = Xc(i)

```

```

Next i
tc_1 = tc
p_1 = p
'Find dXdpc with the system JHx*dXdpc=-dtdp*F(X0)
For i = 1 To N ' B represents -dtdp*F(X0)
    B(i) = -dtdpc * FuncX0(i)
Next i
Call Gauss(JFxc, B, dXdpc)
'If ErrNumber <> 65534 Then Err.Raise (ErrNumber) 'Error treatment
If ErrNumber <> 65534 Then Exit Do 'Termination treatment

Else 'p <> 0
    'Termination criteria
    PathBranchTermination = False
    If AutTDomain Then
        If (Abs(tc) > 100 * N And Abs(dtdpc) >= 10 ^ N) Or (Abs(dtdpc) >= 10 ^ 10) Then
            PathBranchTermination = True
        Else
            If tc < Low_t Or Up_t < tc Then PathBranchTermination = True
        End If
    'Termination control
    If PathBranchTermination Then
        Exit Do
    End If
    If p <> 0 Then 'Here p could be 0 again if the branches are switching
        'Return control
        If (Sgn(dtdpc_1) <> Sgn(dtdpc)) And (Abs(dtdpc_1) >= ReturnTol Or Abs(dtdpc) >= ReturnTol)
Then
        'If (Sgn(dtdpc_1) <> Sgn(dtdpc)) Then

```

```

R = R / 2
For i = 1 To N
    dXdpc(i) = dXdpc_1(i)
    Xc(i) = Xc_1(i)
Next i
tc = tc_1
p = p_1
dtdpc = dtdpc_1
Else 'Return control
R = R * (1 - (1 - 2) * Exp(-1 * Abs(dtdpc_1) / Abs(dtdpc)))
For i = 1 To N
    Xc_1(i) = Xc(i)
Next i
tc_1 = tc
p_1 = p
'Find dXdpc with the system JHx*dXdpc=-dtdp*F(X0)
For i = 1 To N ' B represents -dtdp*F(X0)
    B(i) = -dtdpc * FuncX0(i)
Next i
Call Gauss(JFxc, B, dXdpc)
If ErrNumber <> 65534 Then Exit Do 'Termination treatment
End If 'Return control
End If
End If 'p
End If 'RepeatSphere = False

'Approximation of the change in the curvature length
Deltap = 0
For i = 1 To N

```

```

    Deltap = Deltap + (dXdpc(i)) ^ 2
Next i
If SecondBranch = False Then
    Deltap = R / Sqr(1 + Deltap + (dtdpc) ^ 2)
Else
    Deltap = -R / Sqr(1 + Deltap + (dtdpc) ^ 2)
End If

```

If NSpheres = 0 Then 'Este condicional se adicionó para que no surjan errores al ejecutar el complemento en diferentes hojas.

```

    ReDim HomotopicPathp(0 To NSpheres) As Double
    ReDim HomotopicPatht(0 To NSpheres) As Double
    ReDim HomotopicPathX(1 To N, 0 To NSpheres) As Double
Else
    ReDim Preserve HomotopicPathp(0 To NSpheres) As Double
    ReDim Preserve HomotopicPatht(0 To NSpheres) As Double
    ReDim Preserve HomotopicPathX(1 To N, 0 To NSpheres) As Double
End If

```

```

HomotopicPathp(NSpheres) = p
HomotopicPatht(NSpheres) = tc
For i = 1 To N
    HomotopicPathX(i, NSpheres) = Xc(i)
Next i
NSpheres = NSpheres + 1

```

```

'Predictor Step (Euler)
For i = 1 To N
    X(i) = Xc(i) + Deltap * dXdpc(i)
Next i

```

```

t = tc + Deltap * dtdpc

'Corrector (Newton-Raphson)
For Iter = 0 To MaxIter

  'Function vector construction
  Call FX(X, FuncX)

  'If ErrNumber <> 65534 Then Err.Raise (ErrNumber) 'Error treatment
  If ErrNumber <> 65534 Then Exit Do 'Termination treatment

  For i = 1 To N
    FHY(i) = FuncX(i) - (1 - t) * FuncX0(i)
  Next i

  'Sphere equation
  FHY(N + 1) = 0

  For i = 1 To N
    FHY(N + 1) = FHY(N + 1) + (X(i) - Xc(i)) ^ 2
  Next i

  FHY(N + 1) = FHY(N + 1) + (t - tc) ^ 2 - (R ^ 2)

  'Checking convergence
  NormFHY = 0

  For i = 1 To N + 1
    NormFHY = NormFHY + (FHY(i)) ^ 2
  Next i

  NormFHY = Sqr(NormFHY)

  If NormFHY <= Tol Then

    If Sgn(1 - tc_1) <> Sgn(1 - t) Then 'Verify this part : tc_1 or tc
      If Abs(1 - tc_1) > T1Tol Or Abs(1 - t) > T1Tol Then 'Verify this part : tc_1 or tc
        Iter = MaxIter + 1
      Else
        NRrootsX = NRrootsX + 1

        If NRrootsX = 1 Then 'Este condicional se adicionó para que no surjan errores al ejecutar el
complemento en diferentes hojas.

```

```
ReDim RootsX(1 To N, 1 To NRootsX) As Double
Else
    ReDim Preserve RootsX(1 To N, 1 To NRootsX) As Double
End If
For i = 1 To N
    RootsX(i, NRootsX) = X(i)
Next i
Exit For
End If
End If 'Sgn(1 - tc_1) <> Sgn(1 - t)
Exit For
End If 'NormFHY <= Tol
If Iter = MaxIter Then 'This avoids unnecessary steps
    Iter = MaxIter + 1
    Exit For
End If

'Jacobian construction
Call dFdX(X(), JF_x())
If ErrNumber <> 65534 Then Err.Raise (ErrNumber) 'Error treatment
If ErrNumber <> 65534 Then Exit Do 'Termination treatment
For i = 1 To N
    For j = 1 To N
        JHY(i, j) = JF_x(i, j)
    Next j
Next i
If p = 0 And SecondBranch = False Then 'It is always the same value
    For i = 1 To N
        JHY(i, N + 1) = FuncX0(i)
```

```
    Next i
End If
For j = 1 To N
    JHY(N + 1, j) = 2 * (X(j) - Xc(j))
Next j
JHY(N + 1, N + 1) = 2 * (t - tc_1)

Call GaussForNR(JHY, FHY, SNewt)

If ErrNumber <> 65534 Then Err.Raise (ErrNumber) 'Error treatment
If ErrNumber <> 65534 Then Exit Do 'Termination treatment

For i = 1 To N
    X(i) = X(i) - SNewt(i)
Next i

t = t - SNewt(N + 1)

Next Iter

If Iter <= MaxIter Then
    RepeatSphere = False

    tc = t

    For i = 1 To N
        Xc(i) = X(i)
        dXdpc_1(i) = dXdpc(i)

        For j = 1 To N
            JFxc_1(i, j) = JFxc(i, j)
        Next j
    Next i

    dtdpc_1 = dtdpc

    p_1 = p

    p = p_1 + Deltap

Else
```

```
RepeatSphere = True
  R = R / 2
End If
Loop

TerminationTreatment:
  If SecondBranch = False And ErrNumber <> 65534 Then
    SecondBranch = True
    p = 0
    tc = 0
    For i = 1 To N
      Xc(i) = X0(i)
    Next i
    ErrNumber = 65534
    RepeatSphere = False
  ElseIf SecondBranch = False Then
    SecondBranch = True
    p = 0
    tc = 0
    For i = 1 To N
      Xc(i) = X0(i)
    Next i
    RepeatSphere = False
  ElseIf ErrNumber <> 65534 Then
    GoTo ErrorHandler
  Else
    Exit Do
  End If
Loop
```

'Insert the initial given values on the cells

For i = 1 To N

 Variables.Cells(i).Value = X0(i)

Next i

'Turn screen updating on again.

Application.ScreenUpdating = True

Call ResultDownload

Exit Sub

ErrorHandling:

If Err.Number <> 0 Then ErrNumber = Err.Number

If ErrNumber <> 65534 And SecondBranch = False Then GoTo TerminationTreatment:

 'Insert the initial given values on the cells

 For i = 1 To N

 Variables.Cells(i).Value = X0(i)

 Next i

 'Turn screen updating on again.

 Application.ScreenUpdating = True

Select Case Err.Number

Case 18

 MsgBox "Ejecución interrumpida por el usuario"

Case 11

 MsgBox ("División por cero" & Chr\$(10) & "¡Revise sus ecuaciones o cambie sus valores iniciales!")

Case 6

```
MsgBox ("Desbordamiento" & Chr$(10) & "¡Revise sus ecuaciones o cambie sus valores iniciales!")
```

```
Case 13
```

```
MsgBox ("No coinciden los tipos" & Chr$(10) & "¡Revise sus ecuaciones o cambie sus valores  
iniciales!")
```

```
Case Else
```

```
MsgBox "Error inesperado"
```

```
End Select
```

```
Call ResultDownload
```

```
End Sub
```

```
Sub EntradaDeDatos()
```

```
Salir = True
```

```
Form.Show
```

```
End Sub
```

```
Sub ResultDownload()
```

```
Dim i As Long
```

```
Dim j As Integer
```

```
Application.EnableCancelKey = xlErrorHandler
```

```
On Error GoTo ErrorHandler1
```

```
If TimeRec = True Then
```

```
ExecTime = Round(Timer - ExecTime, 2)
```

```
MsgBox ("Tiempo de ejecución: " & ExecTime & " segundos.")
```

```
End If
```

If NRrootsX = 0 Then

MsgBox "No se encontraron raíces. Compruebe que las ecuaciones estén correctamente ingresadas o intente desde otros valores iniciales"

'Return Microsoft Excel control of the status bar

Application.StatusBar = False

Application.DisplayStatusBar = OldStatusBar

Else 'NRrootsX <> 0

MsgBox "HomotoSolver ha encontrado " & NRrootsX & " solución(ones)"

'Return Microsoft Excel control of the status bar

Application.StatusBar = False

Application.DisplayStatusBar = OldStatusBar

FormRoots.Show

End If 'NRrootsX = 0

If ShowHPath = True Then

ActiveWorkbook.Sheets.Add After:=Worksheets(Worksheets.Count)

'Labels

ActiveSheet.Cells(3, 3).Value = "Datos de la Ruta Homotópica"

ActiveSheet.Cells(4, 3).Value = "p"

ActiveSheet.Cells(4, 4).Value = "t"

For j = 1 To N

 ActiveSheet.Cells(4, 4 + j).Value = "x" & j

Next j

```
'Format
ActiveSheet.Cells.HorizontalAlignment = xlCenter
ActiveSheet.Cells.VerticalAlignment = xlCenter
ActiveSheet.Cells(3, 3).HorizontalAlignment = xlGeneral
For j = 1 To N
    ActiveSheet.Cells(4, 4 + j).Characters(Start:=2, Length:=6).Font.Subscript = True
Next j
'Format for p
With ActiveSheet.Cells(4, 3)
    .Interior.ThemeColor = xlThemeColorDark1
    .Borders(xlEdgeLeft).LineStyle = xlContinuous
    .Borders(xlEdgeTop).LineStyle = xlContinuous
    .Borders(xlEdgeBottom).LineStyle = xlContinuous
    .Borders(xlEdgeRight).LineStyle = xlContinuous
End With
With ActiveSheet.Range(Cells(5, 3), Cells(4 + NSpheres, 3))
    .Interior.ThemeColor = xlThemeColorDark1
    .Borders(xlEdgeLeft).LineStyle = xlContinuous
    .Borders(xlEdgeTop).LineStyle = xlContinuous
    .Borders(xlEdgeBottom).LineStyle = xlContinuous
    .Borders(xlEdgeRight).LineStyle = xlContinuous
End With
'Format for t
With ActiveSheet.Cells(4, 4)
    .Interior.ThemeColor = xlThemeColorDark1
    .Borders(xlEdgeLeft).LineStyle = xlContinuous
    .Borders(xlEdgeTop).LineStyle = xlContinuous
    .Borders(xlEdgeBottom).LineStyle = xlContinuous
    .Borders(xlEdgeRight).LineStyle = xlContinuous
```

End With

With ActiveSheet.Range(Cells(5, 4), Cells(4 + NSpheres, 4))

.Interior.ThemeColor = xlThemeColorDark1

.Borders(xlEdgeLeft).LineStyle = xlContinuous

.Borders(xlEdgeTop).LineStyle = xlContinuous

.Borders(xlEdgeBottom).LineStyle = xlContinuous

.Borders(xlEdgeRight).LineStyle = xlContinuous

End With

'Format for x

With ActiveSheet.Range(Cells(4, 5), Cells(4, 4 + N))

.Interior.ThemeColor = xlThemeColorDark1

.Borders(xlEdgeLeft).LineStyle = xlContinuous

.Borders(xlEdgeTop).LineStyle = xlContinuous

.Borders(xlEdgeBottom).LineStyle = xlContinuous

.Borders(xlEdgeRight).LineStyle = xlContinuous

End With

With ActiveSheet.Range(Cells(5, 5), Cells(4 + NSpheres, 4 + N))

.Interior.ThemeColor = xlThemeColorDark1

.Borders(xlEdgeLeft).LineStyle = xlContinuous

.Borders(xlEdgeTop).LineStyle = xlContinuous

.Borders(xlEdgeBottom).LineStyle = xlContinuous

.Borders(xlEdgeRight).LineStyle = xlContinuous

End With

'Data download

For i = 0 To NSpheres - 1

ActiveSheet.Cells(5 + i, 3).Value = HomotopicPathp(i)

ActiveSheet.Cells(5 + i, 4).Value = HomotopicPatht(i)

For j = 1 To N

```
        ActiveSheet.Cells(5 + i, 4 + j).Value = HomotopicPathX(j, i)
    Next j
Next i

'Chart 'Pending
ActiveSheet.Shapes.AddChart2(240, xlXYScatter).Select
'ActiveChart.SetSourceData Source:=Range(Cells(4, 3), Cells(4 + NSpheres, 4))
ActiveSheet.Shapes.AddChart.Select
ActiveChart.ChartType = xlXYScatter
ActiveChart.SetSourceData Source:=Range(Cells(4, 3), Cells(4 + NSpheres, 4))
ActiveChart.ChartTitle.Delete
ActiveChart.Axes(xlCategory).HasTitle = True
ActiveChart.Axes(xlCategory).AxisTitle.Text = "p"
ActiveChart.Axes(xlValue).HasTitle = True
ActiveChart.Axes(xlValue).AxisTitle.Text = "t"

Cells(1, 1).Select
End If 'ShowHPath
```

Exit Sub

ErrorHandling1:

```
'Insert the initial given values on the cells
For i = 1 To N
    Variables.Cells(i).Value = X0(i)
Next i
```

```
'Turn screen updating on again.
```

```
Application.ScreenUpdating = True

Select Case Err.Number

Case 18
    MsgBox "Ejecución interrumpida por el usuario"

Case 11
    MsgBox ("División por cero" & Chr$(10) & "¡Revise sus ecuaciones o cambie sus valores iniciales!")

Case 6
    MsgBox ("Desbordamiento" & Chr$(10) & "¡Revise sus ecuaciones o cambie sus valores iniciales!")

Case 13
    MsgBox ("No coinciden los tipos" & Chr$(10) & "¡Revise sus ecuaciones o cambie sus valores
iniciales!")

Case Else

    MsgBox "Error inesperado"

End Select

End Sub

Sub FX(X() As Double, FX() As Double)

'ENTRADA: - La dimensión del problema en la variable "N" Integer
'    - Un vector de dimensión N en la variable "X" Double con subíndice inicial en 1
'    - Las funciones deben estar contenidas en el rango "Funciones"

'OBJETIVO: Calcular La función "F" en el punto "X"

'SALIDA: La variable "FX" Double con subíndice inicial en 1

    Dim i As Integer

On Error GoTo ErrorHandler2

'Permite la interrupción del código

Application.EnableCancelKey = xlErrorHandler
```

```
' Introduce los valores X donde se evalua FX y la calcula
For i = 1 To N
    'If SonPositivos = True Then Variables.Cells(i).Value = Abs(X(i))
    'If SonPositivos = False Then Variables.Cells(i).Value = X(i)
    Variables.Cells(i).Value = X(i)
Next i
'Almacena los valores de F evaluado en X
For i = 1 To N
    FX(i) = Functions.Cells(i).Value
Next i
Exit Sub

ErrorHandling2:
    ErrNumber = Err.Number
End Sub

Sub dFdX(X() As Double, dFdX() As Double)

'ENTRADA: - La dimensión del problema en la variable "N" Integer
'    - Un vector de dimensión N en la variable "X" Double con subíndice inicial en 1
'    - En cualquier hoja las funciones deben estar contenidas en el rango "Funciones"
'    - El delta para el cálculo de la jacobiana en el rango "h"
'    - Si la búsqueda se limitará a los positivos en la variable "SonPositivos" Boolean
'OBJETIVO: Calcular la derivada de "F" en "X"
'SALIDA: La variable "dFdX" Double con subíndice inicial en 1

Dim i As Integer
Dim j As Integer
Dim k As Integer
Dim s As Integer
```

```
Dim h() As Double
```

```
ReDim h(1 To N) As Double
```

```
Dim Temporal As Double
```

```
Dim Fun() As Double
```

```
ReDim Fun(1 To N, 1 To N, 1 To 4) As Double
```

```
On Error GoTo ErrorHandler3
```

```
'Permite la interrupción del código
```

```
Application.EnableCancelKey = xlErrorHandler
```

```
'Introduce los valores X donde se evalúa la dFdX y calcula el tamaño de paso h para evaluar las derivadas numéricamente
```

```
For i = 1 To N
```

```
    'Ubica las funciones en el punto de interés X en donde se evaluará la dFdX
```

```
    'If SonPositivos = True Then Variables.Cells(i).Value = Abs(X(i))
```

```
    'If SonPositivos = False Then Variables.Cells(i).Value = X(i)
```

```
    Variables.Cells(i).Value = X(i)
```

```
'Calcula los tamaños de paso hi con los que se minimiza el error de redondeo en el cálculo de la derivada
```

```
If X(i) <> 0 Then h(i) = Abs(X(i)) * ErrorMaquina ^ (1 / 5)
```

```
If X(i) = 0 Then h(i) = 0.00000001
```

```
'Reduce aún más el error de redondeo en la asignación del valor hi
```

```
    Temporal = X(i) + h(i)
```

```
    h(i) = Temporal - X(i)
```

```
Next i
```

```
'Calcula la función en los 2 puntos necesarios para evaluar una diferencia centrada de orden 2
```

```
For k = 1 To 2
```

```
    If k = 1 Then s = -1
```

```
    If k = 2 Then s = 1
```

```
    For j = 1 To N
```

```
        'If SonPositivos = True Then Variables.Cells(j).Value = Abs(X(j) + s * h(j))
```

```

'If SonPositivos = False Then Variables.Cells(j).Value = X(j) + s * h(j)

Variables.Cells(j).Value = X(j) + s * h(j)

For i = 1 To N
    Fun(i, j, k) = Functions.Cells(i).Value
Next i

'If SonPositivos = True Then Variables.Cells(j).Value = Abs(X(j))

'If SonPositivos = False Then Variables.Cells(j).Value = X(j)

Variables.Cells(j).Value = X(j)

Next j

Next k

'Calcula dF/dX mediante una diferencia de orden 2

For j = 1 To N
    For i = 1 To N
        dFdX(i, j) = (1 / (2 * h(j))) * (Fun(i, j, 2) - Fun(i, j, 1))
    Next i
Next j

Exit Sub

ErrorHandling3:
    ErrNumber = Err.Number

End Sub

Sub Gauss(A() As Double, B() As Double, X() As Double)

'ENTRADA: - Una matriz de dimensiones NxN en la variable "A" Double con valores iniciales de 1 en los
subíndices
'
' - Un vector de dimensiones Nx1 en la variable "B" Double con valor inicial de 1 en el subíndice
'
' - Si se desea emplear escalamiento en la variable "UsarEscalado" Boolean
'
' - La dimensión del problema en la variable "N" Integer

'OBJETIVO: Resolver el sistema lineal de ecuaciones AX=B por eliminación Gaussiana con pivoteo

'SALIDA: - El vector solución X en la variable "X" Double

```

' - Si el problema es singular en la variable "EsSingular" Boolean

Dim i As Integer

Dim j As Integer

Dim k As Integer

Dim l As Integer

Dim Epsilon As Double

Dim C As Double

Dim AMax As Double

Dim AiMax() As Double

Dim AkMax As Double

Dim Temporal As Double

Dim Z As Double

Dim MatrizAumentada() As Double

On Error GoTo ErrorHandler4

'Permite la interrupción del código

Application.EnableCancelKey = xlErrorHandler

'Dimensiona la matriz aumentada con subíndices iniciales en 0, y referencias en la fila 0 y la columna 0

ReDim MatrizAumentada(0 To N, 0 To N + 1) As Double

'Dimensiona el vector que contendrá los coeficientes mas grandes de cada fila de la matriz aumentada

ReDim AiMax(1 To N) As Double

'Construye la matriz aumentada con subíndices iniciales en 1

For i = 1 To N

For j = 1 To N

MatrizAumentada(i, j) = A(i, j)

Next j

MatrizAumentada(i, N + 1) = B(i)

```
Next i
'Inicializa la variable AMax
EsSingular = False
k = 1
AMax = 0
'Calcula el máximo de los valores absolutos entre los coeficientes de la matriz que sean diferentes de cero
For i = 1 To N
  For j = 1 To N
    If (MatrizAumentada(i, j) <> 0 And Abs(MatrizAumentada(i, j)) > AMax) Then AMax =
Abs(MatrizAumentada(i, j))
  Next j
Next i
'Verifica singularidad
If AMax = 0 Then
  EsSingular = True
  Exit Sub
End If
'Calcula el épsilon que luego será empleado en el criterio de singularidad de la matriz
Epsilon = AMax * ErrorMaquina ^ 3
'Eliminación Gaussiana con pivoteo
Do While (EsSingular = False And k < N)
  'Pivoteo
  'Si es necesario, utiliza el escalado
  For i = 1 To N
    'Encuentra los coeficiente con mayor valor absoluto en cada una de las filas de la matriz aumentada
    AiMax(i) = 0
    For j = 1 To N + 1
      If Abs(MatrizAumentada(i, j)) > AiMax(i) Then AiMax(i) = Abs(MatrizAumentada(i, j))
    Next j
  Next i
Next i
```

'Cambio de filas

$AkMax = Abs(MatrizAumentada(k, k)) / AiMax(k)$

$l = k$

For $i = k + 1$ To N

If $Abs(MatrizAumentada(i, k)) / AiMax(i) > AkMax$ Then

$AkMax = Abs(MatrizAumentada(i, k)) / AiMax(i)$

$l = i$

End If

Next i

If $l \neq k$ Then

For $j = k$ To $N + 1$

$Temporal = MatrizAumentada(k, j)$

$MatrizAumentada(k, j) = MatrizAumentada(l, j)$

$MatrizAumentada(l, j) = Temporal$

Next j

End If

'Verificación de singularidad

If $Abs(MatrizAumentada(k, k)) < Epsilon$ Then

$EsSingular = True$

Else

'Eliminación

For $i = k + 1$ To N

If $MatrizAumentada(i, k) \neq 0$ Then

$C = MatrizAumentada(i, k) / MatrizAumentada(k, k)$

$MatrizAumentada(i, k) = 0$

For $j = k + 1$ To $N + 1$

$MatrizAumentada(i, j) = MatrizAumentada(i, j) - C * MatrizAumentada(k, j)$

Next j

End If

```
    Next i
  End If
  k = k + 1
Loop
'Última verificación de singularidad
If EsSingular = False Then
  If Abs(MatrizAumentada(k, k)) < Epsilon Then
    EsSingular = True
  'Retrosustitución
Else
  X(N) = MatrizAumentada(N, N + 1) / MatrizAumentada(N, N)
  For k = N - 1 To 1 Step -1
    Z = MatrizAumentada(k, N + 1)
    For j = k + 1 To N
      Z = Z - X(j) * MatrizAumentada(k, j)
    Next j
    X(k) = Z / MatrizAumentada(k, k)
  Next k
End If
End If
Exit Sub

ErrorHandling4:
  ErrNumber = Err.Number
End Sub

Sub GaussForNR(A() As Double, B() As Double, X() As Double)
```

'ENTRADA: - Una matriz de dimensiones NxN en la variable "A" Double con valores iniciales de 1 en los subíndices

- ' - Un vector de dimensiones Nx1 en la variable "B" Double con valor inicial de 1 en el subíndice
- ' - Si se desea emplear escalamiento en la variable "UsarEscalado" Boolean
- ' - La dimensión del problema en la variable "N" Integer

'OBJETIVO: Resolver el sistema lineal de ecuaciones $AX=B$ por eliminación Gaussiana con pivoteo

'SALIDA: - El vector solución X en la variable "X" Double

- ' - Si el problema es singular en la variable "EsSingular" Boolean

Dim i As Integer

Dim j As Integer

Dim k As Integer

Dim l As Integer

Dim NNR As Double

Dim Epsilon As Double

Dim C As Double

Dim AMax As Double

Dim AiMax() As Double

Dim AkMax As Double

Dim Temporal As Double

Dim Z As Double

Dim MatrizAumentada() As Double

On Error GoTo ErrorHandler4

'Permite la interrupción del código

Application.EnableCancelKey = xlErrorHandler

NNR = N + 1

'Dimensiona la matriz aumentada con subíndices iniciales en 0, y referencias en la fila 0 y la columna 0

ReDim MatrizAumentada(0 To NNR, 0 To NNR + 1) As Double

'Dimensiona el vector que contendrá los coeficientes mas grandes de cada fila de la matriz aumentada

```
ReDim AiMax(1 To NNR) As Double
'Construye la matriz aumentada con subíndices iniciales en 1
For i = 1 To NNR
    For j = 1 To NNR
        MatrizAumentada(i, j) = A(i, j)
    Next j
    MatrizAumentada(i, NNR + 1) = B(i)
Next i
'Inicializa la variable AMax
EsSingular = False
k = 1
AMax = 0
'Calcula el máximo de los valores absolutos entre los coeficientes de la matriz que sean diferentes de cero
For i = 1 To NNR
    For j = 1 To NNR
        If (MatrizAumentada(i, j) <> 0 And Abs(MatrizAumentada(i, j)) > AMax) Then AMax =
Abs(MatrizAumentada(i, j))
    Next j
Next i
'Verifica singularidad
If AMax = 0 Then
    EsSingular = True
    Exit Sub
End If
'Calcula el épsilon que luego será empleado en el criterio de singularidad de la matriz
Epsilon = AMax * ErrorMaquina ^ 3
'Eliminación Gaussiana con pivoteo
Do While (EsSingular = False And k < NNR)
    'Pivoteo
    'Si es necesario, utiliza el escalado
```

```
For i = 1 To NNR
    ' Encuentra los coeficiente con mayor valor absoluto en cada una de las filas de la matriz aumentada
    AiMax(i) = 0
    For j = 1 To NNR + 1
        If Abs(MatrizAumentada(i, j)) > AiMax(i) Then AiMax(i) = Abs(MatrizAumentada(i, j))
    Next j
Next i
'Cambio de filas
AkMax = Abs(MatrizAumentada(k, k)) / AiMax(k)
l = k
For i = k + 1 To NNR
    If Abs(MatrizAumentada(i, k)) / AiMax(i) > AkMax Then
        AkMax = Abs(MatrizAumentada(i, k)) / AiMax(i)
        l = i
    End If
Next i
If l <> k Then
    For j = k To NNR + 1
        Temporal = MatrizAumentada(k, j)
        MatrizAumentada(k, j) = MatrizAumentada(l, j)
        MatrizAumentada(l, j) = Temporal
    Next j
End If
'Verificación de singularidad
If Abs(MatrizAumentada(k, k)) < Epsilon Then
    EsSingular = True
Else
    'Eliminación
    For i = k + 1 To NNR
```

```
If MatrizAumentada(i, k) <> 0 Then
    C = MatrizAumentada(i, k) / MatrizAumentada(k, k)
    MatrizAumentada(i, k) = 0
    For j = k + 1 To NNR + 1
        MatrizAumentada(i, j) = MatrizAumentada(i, j) - C * MatrizAumentada(k, j)
    Next j
End If
Next i
End If
k = k + 1
Loop
'Última verificación de singularidad
If EsSingular = False Then
    If Abs(MatrizAumentada(k, k)) < Epsilon Then
        EsSingular = True
    'Retrosustitución
    Else
        X(NNR) = MatrizAumentada(NNR, NNR + 1) / MatrizAumentada(NNR, NNR)
        For k = NNR - 1 To 1 Step -1
            Z = MatrizAumentada(k, NNR + 1)
            For j = k + 1 To NNR
                Z = Z - X(j) * MatrizAumentada(k, j)
            Next j
            X(k) = Z / MatrizAumentada(k, k)
        Next k
    End If
End If
Exit Sub
ErrorHandling4:
```

ErrNumber = Err.Number

End Sub
