

Una aproximación numérica a los General Linear Methods para la solución de problemas de la cinética química

A numerical approximation to the General Linear Methods for the solution of problems from chemistry kinetics

Mayra Alejandra Vega Niño



Miguel Antonio Caro Candezano



Universidad del Atlántico, Colombia

Jeinny María Peralta Polo



Universidad Nacional Abierta y a Distancia, Colombia

OPEN ACCESS

Recibido: 21/07/2021

Aceptado: 09/09/2021

Publicado: 29/11/2021

Correspondencia de autores:

mayraavega@est.uniatlantico.edu.co



Copyright 2020
by Investigación e
Innovación en Ingenierías

Resumen

Objetivo: Realizar un estudio comparativo entre las soluciones numéricas de sistemas de ecuaciones diferenciales ordinarias tipo Stiff obtenidas con el método General Linear Method (GLM) y dos solvers del software matemático MATLAB como ode45 y ode15s. **Metodología:** Para el estudio comparativo se escogieron dos problemas de la química HIREs y de OREGO, los cuales modelan reacciones de la cinética química. Inicialmente cada problema fue solucionado utilizando un tipo de GLM. De igual forma, fueron calculadas las soluciones numéricas con los solvers ode45 y ode15s del software MATLAB. Finalmente, se establecieron comparaciones teniendo en cuenta el tiempo de ejecución de cada método/solver para resolver el problema, el número de iteraciones totales y el error relativo. **Resultados:** A partir de los resultados numéricos del problema de HIREs se observó que el método DIMSIM tuvo el mejor rendimiento comparando los tiempos de CPU, pero el solver ode45 de MATLAB tuvo la mejor aproximación, seguido por ode15s y los obtenidos por el método DIMSIM, respectivamente. Con el problema OREGO, el mejor tiempo computacional fue alcanzado por el solver ode15s, sin embargo, los resultados obtenidos con el método DIMSIM tuvieron la mejor aproximación. **Conclusiones:** Los resultados numéricos indicaron que el método DIMSIM y el solver ode15s obtuvieron un mejor desempeño en el tiempo de ejecución y el número de iteraciones totales comparados con el solver ode45. Para estudios futuros se pretende comparar los resultados numéricos con aquellos obtenidos con estrategias de paralelización del código computacional.

Palabras claves: General Linear Methods, sistemas de ecuaciones diferenciales, problemas de valor inicial de tipo Stiff.

Abstract

Objective: Comparing the numerical solutions of stiff systems of ordinary differential equations obtained with the General Linear Methods (GLM) with two solvers of the mathematical software MATLAB such as ode45 and ode15s. **Methodology:** For the comparative study, the HIREs and OREGO chemistry problems were chosen, which model kinetic-chemical reactions. Initially, each problem was solved using one type of GLM. Similarly, the numerical solutions were calculated with the ode45 and ode15s solvers of MATLAB software. Finally, comparisons were established taking into account the execution time of each method/solver to solve the problem, the number of total iterations and the relative error. **Results:** From the numerical results of the HIREs problem, it was observed that the DIMSIM method had the best performance comparing CPU times, but the MATLAB ode45 solver had the best approximation, followed by ode15s and those obtained by the DIMSIM method, respectively. With the OREGO problem, the best computational time was achieved by the ode15s solver, however, the results obtained with the DIMSIM method had the best approximation. **Conclusions:** The numerical results showed that the DIMSIM method and the ode15s solver performed better in execution time and number of total iterations compared to the ode45 solver. For future studies, it is intended to compare the numerical results with those obtained with parallelization strategies of the computational code.

Keywords: General Linear Methods, systems of differential equations, stiff initial value problems.

Introducción

Los métodos para resolver sistemas de ecuaciones diferenciales ordinarias se dividen en dos grandes grupos, multipasos y multietapas [1, 2]. El método multietapas más conocido es el método Runge-Kutta, el cual tuvo origen al inicio del siglo XX [3]. Por otra parte, se tienen los métodos multipasos, de los cuales los más conocidos son Adams-Bashforth, Adams-Moulton y la familia de los Backward Differentiation Formula [3]. A pesar de que los métodos de ambos grupos, multipasos y multietapas, poseen algoritmos de fácil implementación y brindan buenas aproximaciones a los sistemas de ecuaciones diferenciales ordinarias, la ejecución suele ser bastante costosa computacionalmente, además, pueden llegar a presentar dificultades si el sistema a solucionar es de tipo *Stiff* [4, 5, 6].

Los sistemas de ecuaciones diferenciales ordinarias tipo *Stiff* fueron estudiados por primera vez en 1952 [7]. En términos generales, un sistema de ecuaciones ordinarias tipo *Stiff* presenta ciertas variables con tasas de cambio bastante rápidas en relación a las demás componentes del sistema, lo que dificulta encontrar su solución numérica utilizando métodos explícitos [5, 7, 8].

Debido a la naturaleza de este tipo de sistemas y su abundancia en diferentes procesos naturales, se requirió el desarrollo de nuevos conceptos de estabilidad que permitieran facilitar la escogencia del método a utilizar para poder encontrar su solución numérica [9]. Estos en general deben ser métodos de tipo implícitos, ya que los métodos explícitos tienden a presentar problemas causados por su región de estabilidad bastante limitada [10, 11, 12].

Por otra parte, los métodos implícitos para solucionar ecuaciones diferenciales ordinarias tienden a tener regiones de estabilidad que contienen a gran parte del plano complejo, lo cual permite su implementación en sistemas diferenciales tipo *Stiff* [10, 13]. A pesar de su naturaleza, su ejecución es bastante costosa computacionalmente, ya que implica la solución en simultaneo de un número elevado de ecuaciones [10,14]. Asimismo, no permiten que el paso h utilizado cambie en cada iteración, requisito propio de la mayoría de sistemas tipo *Stiff* [4, 11].

Con el deseo de superar estas dificultades y obtener un nuevo tipo de método con bajo costo computacional y que realice el cambio del paso h , dependiendo de la necesidad de cada sistema, surgen los *General Linear Methods* (GLM) como respuesta a las dificultades computacionales y numéricas presentadas por los distintos tipos de métodos (multipasos y multietapas), siendo la idea principal de estos, poder generar aproximaciones suficientemente precisas con bajo costo computacional [15]. Este tipo de métodos fueron introducidos en [16], los cuales integran distintas características de los métodos ya mencionados [17].

A continuación, son presentados algunos conceptos relacionados a los sistemas de ecuaciones diferenciales ordinarias tipo *Stiff*. Posteriormente, se expondrán las definiciones y teoremas más importantes relacionados con los *General Linear Methods*. Luego, se presentan los resultados obtenidos al implementar un tipo de GLM a algunos sistemas de ecuaciones diferenciales ordinarias tipo *Stiff*, comparando los resultados con los valores obtenidos al resolver los sistemas mencionados anteriormente mediante dos solvers del software MATLAB. Finalmente, se presentan las conclusiones del trabajo.

Metodología

Ecuaciones diferenciales tipo *Stiff*

Las ecuaciones diferenciales tipo *Stiff* fueron descritas por primera vez en [7], al resolver sistemas de ecuaciones que tenían origen en reacciones químicas y teoría de circuitos eléctricos, fue estudiado que el método explícito que estaban utilizando presentaba problemas para resolver sistemas de este tipo. Al estudiar el comportamiento de cada componente, los cuales exhibieron una tasa de cambio bastante rápida en periodos cortos de tiempo, en comparación con el resto de elementos del problema, que tenían tasas de cambio mucho más lentas.

Consideremos el siguiente problema de valor inicial

$$y' = Ay + \varphi(t), \quad t \geq t_0 \quad (1)$$

$$y(t_0) = y_0 \in \mathbb{R}^m \quad (2)$$

donde $A \in \mathbb{R}^{m \times m}$ es una matriz constante y $\varphi(t)$ es una función que depende de t [18]. Asumamos que A tiene m autovalores $\lambda_1, \dots, \lambda_m$ todos distintos, con autovectores v_1, \dots, v_m . La solución general de (1) viene dada por

$$y(t) = \sum_{i=1}^m c_i v_i e^{\lambda_i(t-t_0)} + \psi(t), \quad t \geq t_0, \quad (3)$$

donde c_i son constantes que dependen de la condición inicial y $\psi(t)$ es la solución particular. Si $|\operatorname{Re}(\lambda_i)| \gg 0$ y λ_i es negativo para algún i , el término $c_i v_i e^{\lambda_i(t-t_0)}$ de (3) tenderá a cero mucho más rápido que los demás componentes de la solución, lo cual clasifica al sistema como *Stiff*.

Bajo las hipótesis anteriores podemos definir el *radio de Stiff* como

$$R = \frac{\max |\operatorname{Re}(\lambda_j)|}{\min |\operatorname{Re}(\lambda_j)|}, \quad j = 1, \dots, m.$$

Si $R \gg 1$ se dice que el problema es *Stiff*. En el caso que el sistema de ecuaciones sea no lineal, se consideran los autovalores de la matriz del Jacobiano J del problema [18].

Observaciones.

- 1) A pesar de que el *radio de Stiff* permite identificar si un problema es *Stiff* o no, puede llegar a presentar dificultades en el caso que algún autovalor sea igual a cero [10].
- 2) Usualmente, para sistemas lineales sólo se consideran los autovalores de la matriz A para comprobar si es *Stiff*. En el caso de los no lineales, es frecuente que se realice un análisis del comportamiento de la estabilidad de los componentes [9].
- 3) La propiedad de ser *Stiff* es propia del sistema y no de la solución del mismo [19].

Debido a que no se cuenta con una única definición o caracterización para sistemas de este tipo, usualmente se dice que un sistema es *Stiff* si cumple la siguiente definición.

Definición. [19] Si un método numérico es forzado a usar, en un intervalo de integración, un paso h excesivamente pequeño relativo al intervalo de tiempo de la solución, para obtener una solución suave de la solución real, se dice que el problema es *Stiff* en ese intervalo.

General Linear Methods (GLM)

Este tipo de método es caracterizado por una matriz de tamaño $(s + r) \times (s + r)$ que contiene cuatro matrices, A , B , U y V , de la siguiente forma:

$$\begin{bmatrix} A_{s \times s} & U_{s \times r} \\ B_{r \times s} & V_{r \times r} \end{bmatrix}.$$

Para un sistema N – dimensional de ecuaciones diferenciales ordinarias, la matriz anterior es equivalente a

$$Y_i^{[n]} = \sum_{j=1}^s a_{ij} hf(Y_j^{[n]}) + \sum_{j=1}^r u_{ij} y_j^{[n-1]}, \quad i = 1, 2, \dots, s. \quad (4)$$

$$y_i^{[n]} = \sum_{j=1}^s b_{ij} hf(Y_j^{[n]}) + \sum_{j=1}^r v_{ij} y_j^{[n-1]}, \quad i = 1, 2, \dots, r. \quad (5)$$

Donde

- $Y_i^{[n]}$: representa las etapas internas como en un método Runge-Kutta.
- $f(Y_i^{[n]})$: representa la derivada de cada etapa.
- $y_i^{[n-1]}$: es la aproximación entrante en el paso n .
- $y_i^{[n]}$: es la aproximación de salida en el paso n .

Los enteros p, q, r, s son tales que

- s : representa el número de etapas del método.
- r : representa la cantidad de valores utilizados en cada paso n del método.
- p : representa el orden del método, lo cual indica que si $y(\bar{x}_n)$ es la solución exacta, se tiene que

$$y(\bar{x}_n) = y(x_n) + O(h^{p+1}).$$

- q : representa el orden de las etapas, lo cual indica que el valor Y_i aproxima la solución en $x_{n-1} + c_i h$ con orden q , esto implica que $Y_i = y(x_{n-1} + c_i h) + O(h^{q+1})$. Aquí, $c = [c_1, \dots, c_s]^T$ es el vector de abscisas.

Teorema [18] Un GLM es convergente si, y sólo si, es estable y consistente.

Método de integración multietapa diagonalmente implícito (DIMSIM)

En este trabajo, como en [10], A es una matriz triangular inferior; los elementos de su diagonal son todos iguales a una constante positiva λ , de modo que se puedan encontrar los valores de cada etapa por separado y de forma secuencial y así reducir el costo de implementación. Al método que satisface las hipótesis anteriores se le conoce como Método de integración multietapa diagonalmente implícito (Diagonally Implicit Multistage Integration Methods (DIMSIM)).

Los métodos DIMSIM se clasifican en los siguientes tipos

Tabla 1. Tipos de DIMSIM según la forma de la matriz A

Tipo	Estructura	Problema
Tipo 1	A estrictamente triangular inferior	No Stiff
Tipo 2	$A = 0$	No Stiff
Tipo 3	$A + \lambda I$ triangular inferior	Stiff
Tipo 4	$A = \lambda I$	Stiff

Fuente: Tomada de [8, 20]

Para la implementación, haremos uso de un DIMSIM del Tipo 3 para el cual se cumple que $p = q = 4$ y $s = r = p + 1 = 5$.

Resultados

En esta sección se mostrarán los resultados obtenidos al resolver distintos sistemas de ecuaciones diferenciales tipo *Stiff* utilizando un DIMSIM de orden $p = q = 4$ además de $s = r = 5$, presentar el resumen de la implementación de estos problemas utilizando los solvers ode45 y ode15s del software MATLAB. El código usado fue adaptado al de Shirley Huang en [10], este fue escrito en el lenguaje de programación FORTRAN y para su ejecución se instalaron el compilador gfortran, junto con los paquetes de Lapack y BLAS, en una máquina virtual de Linux. Además, se utilizó la versión R2020a del software MATLAB.

Observaciones. En todas las soluciones se utilizaron los valores de $tol = h = 10^{-3}$.

Si denotamos como \hat{y} a la aproximación de la solución y en el punto final del intervalo, entonces el error relativo estará dado por

$$\|E\|_{rel} = \max_{\hat{y} \neq 0} \frac{|\hat{y} - y|}{\hat{y}}$$

Problema de OREGO

Este sistema de ecuaciones diferenciales ordinarias describe la reacción de Belusov-Zhabotinskii la cual modela las oscilaciones que se presentan en el ácido malónico en un medio ácido por medio de bromatos [21, 22]. A continuación, el sistema de ecuaciones que describe este problema

$$\begin{cases} y_1' = 77.27(y_2 + y_1(1 - 8.375 \times 10^{-6}y_1 - y_2)), \\ y_2' = \frac{1}{77.27}(y_3 - (1 + y_1)y_2), \\ y_3' = 0.161(y_1 - y_2), \end{cases} \quad (6)$$

con $y_0 = (1,2,3)^T$, $t \in [0,360]$ [18, 21].

Solución Numérica

En la Tabla 2 es presentada la solución numérica en el tiempo $t = 360$ hallada con el método DIMSIM, junto con la solución de referencia del problema obtenida de [21].

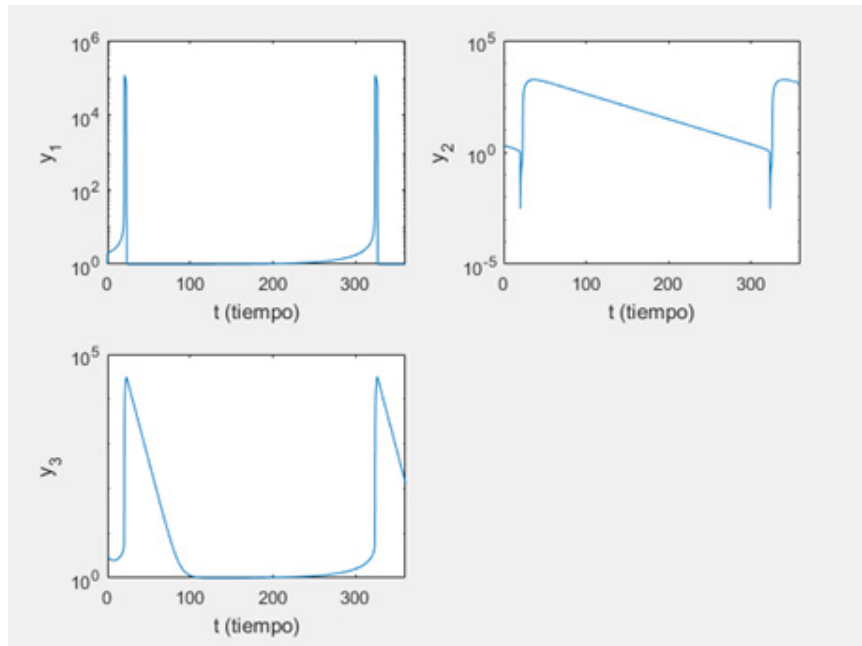
Tabla 2. Solución numérica y de referencia del problema de OREGO en $t = 360$

\vec{y}_i	Solución numérica	Solución de referencia
y_1	1.0008148688420000	1.00081487031852
y_2	1228.1807448950001	1228.17852154988
y_3	132.05683398389999	132.055494284651

Fuente: Elaboración propia

Seguidamente, en la Figura 1, se presenta la gráfica de la solución numérica obtenida en el intervalo de integración $[0, 360]$.

Figura 1. Solución Numérica del problema de OREGO con el método DIMSIM de orden $p = q = 4$, $s = r = 5$



Fuente: Elaboración propia

Comparación

En la Tabla 3 es presentado el resumen de la implementación del código a este problema utilizando el método DIMSIM y los solvers ode45 y ode15s del software MATLAB. Esta tabla incluye el error relativo, el número de iteraciones totales y el tiempo de ejecución del código para hallar la solución.

Tabla 3. Resumen resultados de varios métodos al solucionar el problema de OREGO

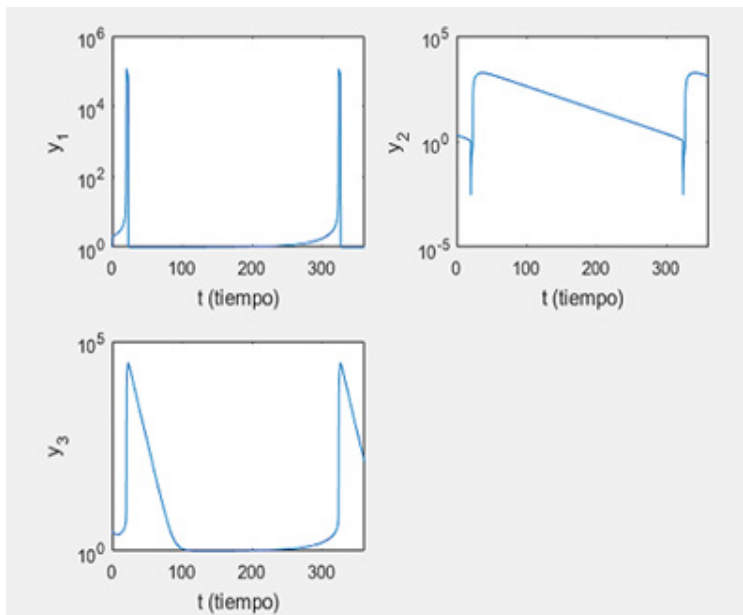
Solver	$\ E\ _{rel}$	Iteraciones totales	Tiempo de ejecución
DIMSIM	$1.193361856000000 \times 10^{-7}$	143026	1.89863896 s
ode45	$7.879608107408603 \times 10^{-5}$	13580769	292.014364 s
ode15s	$1.974911335578400 \times 10^{-2}$	539	0.35330500 s

Fuente: Elaboración propia

Si se hubiera utilizado un método con paso fijo h para resolver el problema de OREGO, al hacer $h = 10^{-3}$, se hubieran obtenido un total de **360000** iteraciones. En la Tabla 3 es observado que el método DIMSIM realizó un total de **143026**, es decir, **216974** iteraciones menos de las esperadas. De igual forma, el DIMSIM tuvo el menor error en el punto final del intervalo, superando notablemente el error del ode15s. El solver que más tardó en encontrar la solución fue ode45, aproximadamente **292.014364 s**, lo cual es bastante alto teniendo en cuenta que el método DIMSIM y el solver ode15s

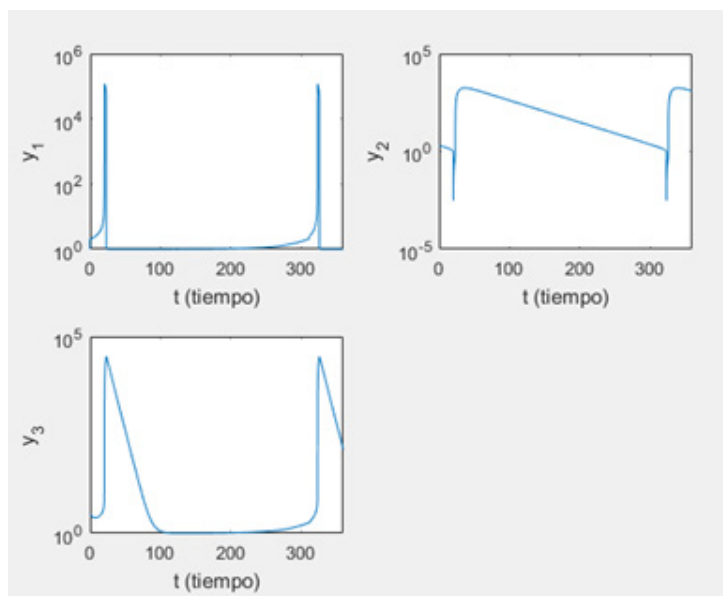
tardaron menos de 2 s cada uno. A continuación, se presentan las gráficas de las soluciones numéricas halladas con ode45 y ode15s.

Figura 2. Solución del problema de OREGO con el solver ode45



Fuente: Elaboración propia

Figura 3. Solución del problema de OREGO con el solver ode15s



Fuente: Elaboración propia

Observando las gráficas de las Figuras 2 y 3 se puede notar que el comportamiento de las soluciones halladas por ode45 y ode15s son bastante parecidas a la encontrada por el DIMSIM (Figura 1). Aunque el método DIMSIM y los dos solvers resolvieron el problema de OREGO, por los resultados del número de iteraciones y el tiempo de ejecución presentados en la Tabla 2 y la Figura 3, se recomienda utilizar el solver ode15s para solucionar este problema.

Problema de HIRES

Este sistema de ecuaciones diferenciales ordinarias está conformado por 8 ecuaciones diferenciales que modelan la morfogénesis proceso por el cual los órganos de las plantas adquieren formas distintivas que les permitirán desarrollar funciones específicas [18, 21, 22]. Este problema es descrito por el siguiente sistema de ecuaciones

$$\begin{cases} y_1' = -1.71y_1 + 0.43y_2 + 8.32y_3 + 0.0007, \\ y_2' = 1.71y_1 - 8.75y_2, \\ y_3' = -10.03y_3 + 0.43y_4 + 0.035y_5, \\ y_4' = 8.32y_2 + 1.71y_3 - 1.12y_4, \\ y_5' = -1.745y_5 + 0.43y_6 + 0.43y_7, \\ y_6' = -280y_6y_8 + 0.69y_4 + 1.71y_5 - 0.43y_6 + 0.69y_7, \\ y_7' = 280y_6y_8 - 1.81y_7, \\ y_8' = -280y_6y_8 + 1.81y_7, \end{cases} \quad (7)$$

con $y_0 = (1,0,0,0,0,0,0,0.0057)^T$, $t \in [0,321.8122]$ [18].

Solución numérica

En la Tabla 4 es presentada la solución numérica en el punto $t = 321.8122$ hallada con DIMSIM, junto con la solución de referencia del problema obtenida de [21].

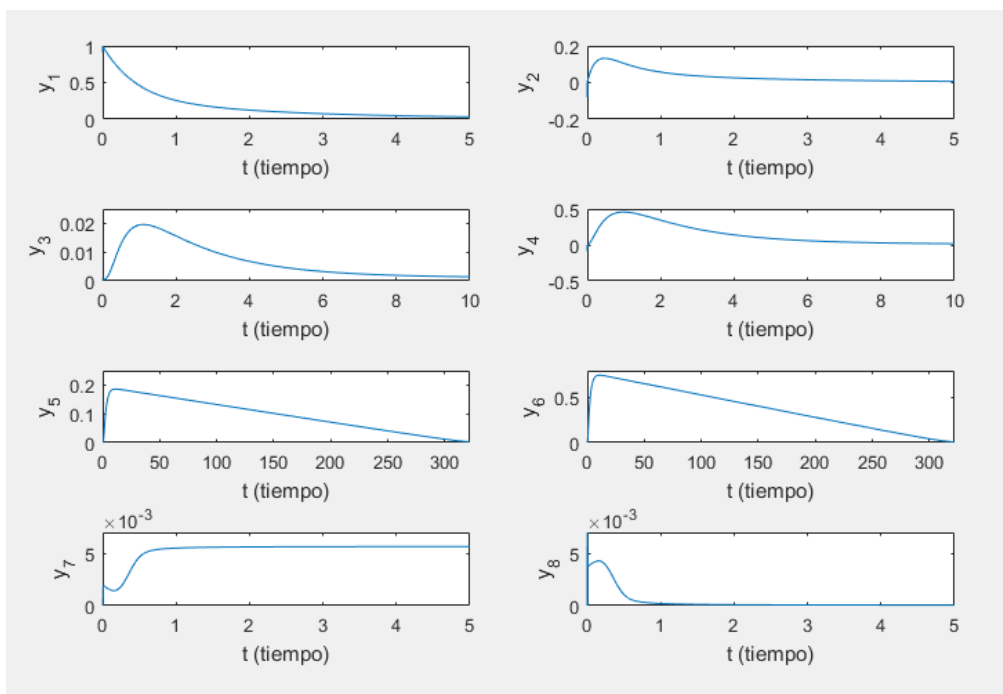
Tabla 4. Solución numérica y de referencia del problema de HIRES en $t = 321.8122$

\bar{y}_i	Solución numérica	Solución de referencia
y_1	$0.73713125733256685 \times 10^{-3}$	$0.7371312573325668 \times 10^{-3}$
y_2	$0.14424857263161851 \times 10^{-3}$	$0.1442485726316185 \times 10^{-3}$
y_3	$0.58887297409675752 \times 10^{-4}$	$0.5888729740967575 \times 10^{-4}$
y_4	$0.11756513432831491 \times 10^{-2}$	$0.1175651343283149 \times 10^{-2}$
y_5	$0.23863561988313308 \times 10^{-2}$	$0.2386356198831331 \times 10^{-2}$
y_6	$0.62389682527427964 \times 10^{-2}$	$0.6238968252742796 \times 10^{-2}$
y_7	$0.28499983951857689 \times 10^{-2}$	$0.2849998395185769 \times 10^{-2}$
y_8	$0.28500016048142308 \times 10^{-2}$	$0.2850001604814231 \times 10^{-2}$

Fuente: Elaboración propia

A continuación, en la Figura 4 es presentada la gráfica de la solución numérica obtenida en el intervalo de integración $[0, 321.8122]$.

Figura 4. Solución Numérica del problema de HIRES con el método DIMSIM de orden $p = q = 4$ y $s = r = 5$



Fuente: Elaboración propia

Observación. No todos los componentes y_i de la solución fueron graficadas en el intervalo $[0, 321.8122]$, debido a que luego de $t = 5$ o $t = 10$ decaen y se mantienen muy cercanas a cero lo cual dificulta percibir gráficamente su comportamiento.

Comparación

En la Tabla 5 se muestra el resumen de la implementación del código a este problema utilizando el DIMSIM antes mencionado y los solvers ode45 y ode15s del software MATLAB. De igual forma, ésta tabla incluye el error relativo, el número de iteraciones totales y el tiempo de ejecución del código para hallar la solución.

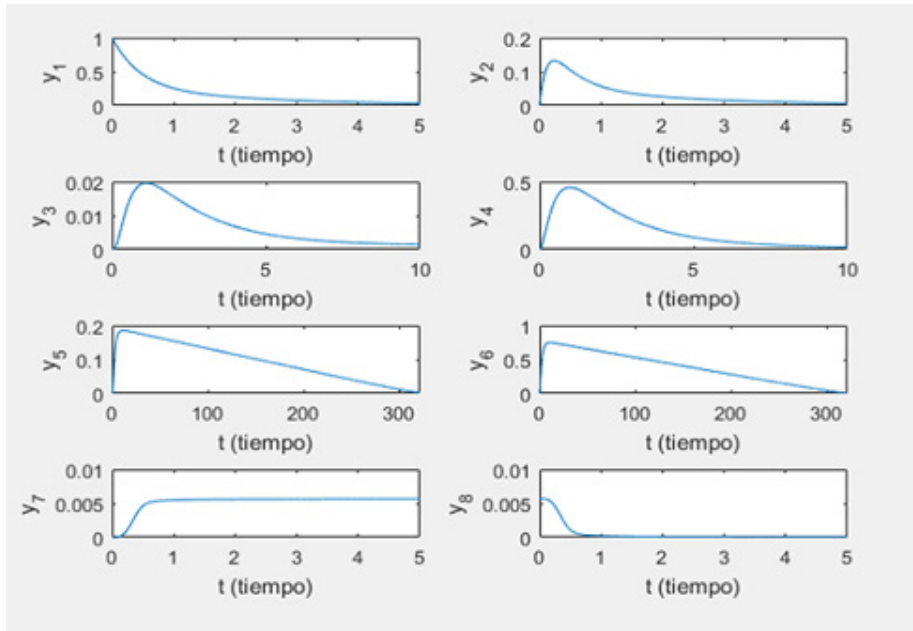
Figura 5. Resumen resultados de varios métodos al solucionar el problema de HIRES

Solver	$\ E\ _{rel}$	Iteraciones totales	Tiempo de ejecución
DIMSIM	0.008179834018000	90	0.002056 s
ode45	0.001318115504546	41521	0.557389 s
ode15s	0.006751333881008	119	0.406032 s

Fuente: Elaboración propia

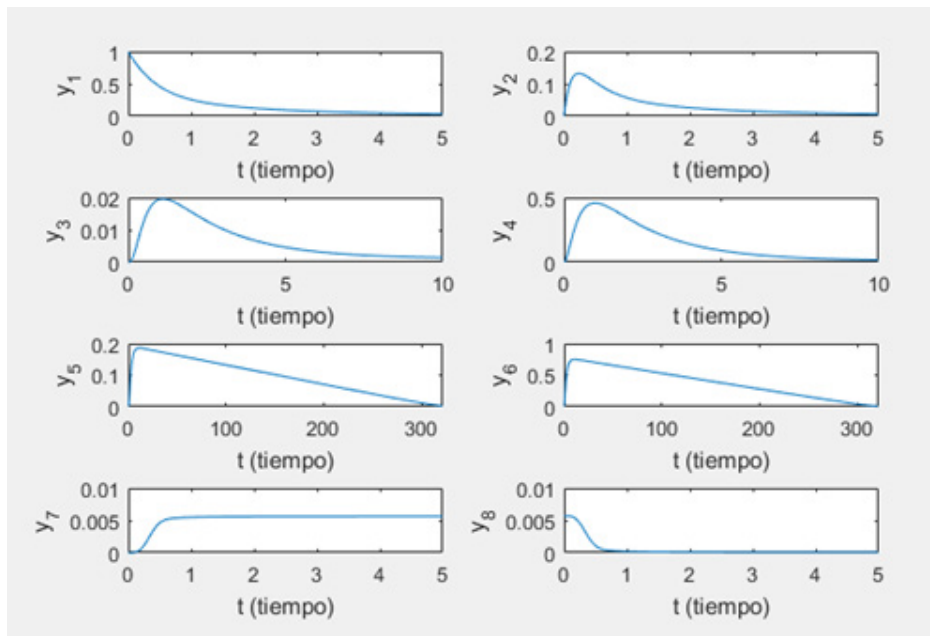
Se puede notar que al implementar el método DIMSIM, este realizó el menor número de iteraciones en menor tiempo. El solver ode45 fue el que presentó el menor error en el punto final del intervalo $t = 321.8122$, siendo este el que realizó el mayor número de iteraciones totales. En general, el método DIMSIM y los dos solvers, tuvieron un buen desempeño en el cálculo de la solución de este problema. A continuación, se presentan las gráficas de las soluciones numéricas halladas con ode45 y ode15s.

Figura 6. Solución del problema de HIRES con el solver ode45



Fuente: Elaboración propia

Figura 7. Solución del problema de HIRES con el solver ode15s



Fuente: Elaboración propia

Observando las gráficas de las Figuras 5 y 6 se puede notar que el comportamiento de las soluciones halladas por ode45 y ode15s son bastante parecidas a la encontrada por el DIMSIM (Figura 4). A pesar de que el método DIMSIM y los dos solvers resolvieron el problema de HIREs, por los resultados presentados en la Tabla 5 y la Figura 4, se recomienda utilizar DIMSIM para solucionar este problema debido a que este encontró la solución en menor tiempo y con un bajo error relativo.

Conclusiones

El propósito principal de este trabajo fue realizar una implementación numérica del método DIMSIM, basado en el GLM, para resolver dos problemas de la cinética química, reacciones de OREGO y HIREs, para posteriormente comparar estos resultados con las soluciones halladas en la implementación de dos solvers del software MATLAB ode45 y ode15s. Con base en los resultados presentados, se concluye que el solver ode15s tuvo mejores resultados que el método DIMSIM en el número total de iteraciones y tiempo de ejecución en el problema de OREGO, esto se puede explicar debido a que este utiliza métodos de orden variable (orden 1 a orden 5) lo cual favorece su rendimiento en algunos problemas [23, 24]. Sin embargo, el método DIMSIM obtuvo la aproximación más cercana en este problema.

Por otra parte, en el problema de HIREs, el método DIMSIM obtuvo mejores resultados que los solvers ode45 y ode15s en el número total de iteraciones y tiempo de ejecución, siendo el solver ode45 el que tuvo la mejor aproximación. Teniendo en cuenta que el solver ode45 no fue diseñado para resolver sistemas de ecuaciones diferenciales ordinarias tipo Stiff, pudimos observar que, aunque puede ser más demorado (puede tardar hasta **291.661059 s** más que el método DIMSIM y el solver ode15s), es igualmente capaz de encontrar la solución de algunos sistemas de este tipo. Finalmente, una diferencia notable entre el DIMSIM utilizado y los solvers ode45 y ode15s, es que estos últimos no necesitan que el usuario proporcione el Jacobiano del sistema, ellos lo aproximan internamente, lo cual puede ser una ventaja en sistemas con un gran número de ecuaciones debido a que calcular su Jacobiano analíticamente puede ser un proceso costoso computacionalmente.

Referencias bibliográficas

1. C. Klingenberg y M. Westdickenberg, *Theory, Numerics and Applications of Hyperbolic Problems II*. Aachen, Alemania: Springer, 2018.
2. J. V. Lambers y A. C. Sumner, *Explorations in Numerical Analysis*. World Scientific, 2018.
3. C. Brezinski y L. Wuytack, *Historical Developments in the 20th Century*. Elsevier Science, 2001.
4. J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*. Nueva York, Estados Unidos: Wiley & Sons, Ltd, 2016.
5. H. Ricardo, *A Modern Introduction to Differential Equations*. Academic Press, 2020.
6. R. Glandon, M. Narayanamurthi y A. Sandu. (2020). "Linearly implicit multistep methods for time integration". *arXiv preprint arXiv:2011.10685*.
7. C. F. Curtiss y J. O. Hirschfelder. (1952). "Integration of Stiff Equations". *Proceedings of the National Academy of Sciences of the United States of America [En línea]*. 38(3), 235. Disponible en: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1063538/pdf/pnas01576-0089.pdf>

8. A. D. Polyanin y V. F. Zaitsev, *Handbook of Ordinary Differential Equations: Exact Solutions, Methods, and Problems*. CRC Press, 2017.
9. D. Garfinkel y C. Marbach. (1977). "Stiff Differential Equations". *Annual Review of Biophysics [En línea]*. 6(1), 525-542. Disponible en: <https://www.annualreviews.org/doi/abs/10.1146/annurev.bb.06.060177.002521?journalCode=biophys.1>
10. S. Huang, "Implementation of General Linear Methods for Stiff Ordinary Differential Equations", Tesis Ph.D., Dept. Matemáticas, Univ. de Auckland, Nueva Zelanda, 2005.
11. D. Xue, *Differential Equation Solutions With MATLAB: Fundamentals and Numerical Implementations*. Walter de Gruyter GmbH & Co KG, 2020.
12. G. Izzo y Z. Jackiewicz. (2018). "Strong stability preserving transformed DIMSIMs". *Journal of Computational and Applied Mathematics [En línea]*. 343, 174-188. DOI: <https://doi.org/10.1016/j.cam.2018.03.018>
13. G. Califano, G. Izzo y Z. Jackiewicz. (2018). "Strong Stability Preserving General Linear Methods with Runge-Kutta Stability". *Journal of Scientific Computing [En línea]*, 76(2), 943-968. DOI: <https://doi.org/10.1007/s10915-018-0646-5>
14. A. Moradi, M. Sharifi y A. Abdi. (2020). "Transformed implicit-explicit second derivative diagonally implicit multistage integration methods with strong stability preserving explicit part". *Applied Numerical Mathematics*. DOI: <https://doi.org/10.1016/j.apnum.2020.04.007>
15. A. J. Kadhim y A. Gorgey. (2020). "Extrapolation of type-1 diagonally implicit multistage integration methods for solving ordinary differential equations". *Advances in Mathematics: Scientific Journal [En línea]*. 9(12), 10949-10955. DOI: <https://doi.org/10.37418/amsj.9.12.78>
16. J. C. Butcher. (1966). "On the convergence of numerical solutions to ordinary differential equations". *Mathematics of Computation [En línea]*. 20(93), 1-10. Disponible en: <https://www.ams.org/journals/mcom/1966-20-093/S0025-5718-1966-0189251-X/S0025-5718-1966-0189251-X.pdf>
17. A. Abdi y B. Behzad. (2018). "Efficient Nordsieck second derivative general linear methods: construction and implementation". *Calcolo [En línea]*. 55(3). DOI: <https://doi.org/10.1007/s10092-018-0270-7>
18. Z. Jackiewicz, *General linear methods for ordinary differential equations*. Hoboken, Estados Unidos: John Wiley & Sons, Ltd, 2009.
19. J. D. Lambert, *Numerical Methods for Ordinary Differential Systems*. Nueva York, Estados Unidos: John Wiley & Sons, Ltd, 1991.
20. M. Braš y Z. Jackiewicz. (2020). "A new class of efficient general linear methods for ordinary differential equations". *Applied Numerical Mathematics [En línea]*. 151, 282-300. DOI: <https://doi.org/10.1016/j.apnum.2019.12.022>
21. F. Mazzia y F. Iavernaro, "Test Set for Initial Value Problem Solvers", Universidad de Bari, Febr. 2008. Disponible en: <http://archimede.dm.uniba.it/~testset/report/prologue.pdf>
22. B. Soleimani, B., & Weiner, R. (2017). "A class of implicit peer methods for stiff systems". *Journal of Computational and Applied Mathematics [En línea]*. 316, 358-368. DOI: <https://doi.org/10.1016/j.cam.2016.06.014>
23. S. Eshkabilov, *Practical Matlab Modeling With Simulink: Programming and Simulating Ordinary and Partial Differential Equations*. Apress, 2020.
24. K. Postawa, J. Szczygieł y M. Kułazyński. (2020). "A comprehensive comparison of ODE solvers for biochemical problems". *Renewable Energy [En línea]*. 156, 624-633. DOI: <https://doi.org/10.1016/j.renene.2020.04.089>