Resolución Numérica de Ecuaciones Diferenciales

Métodos Numéricos

Docente: Dra. Alma Xóchitl González Morales

Universidad de Guanajuato – Campus León
División de Ciencias e Ingenierías
León, Guanajuato, México

Noviembre 2025

Resolución Numérica de Ecuaciones Diferenciales

Oliva-Villar David Isaac
División de Ciencias e Ingenierías Campus León, Universidad de Guanajuato
Loma del Bosque 103, 37150 León Guanajuato, México
Licenciatura en Ingeniería Química Sustentable
(Fecha: 27 de Noviembre de 2025)

Resumen

El presente documento aborda la implementación de códigos realizados en lenguaje C de métodos numéricos para la resolución de Ecuaciones Diferenciales Ordinarias (EDO). Se comparan los métodos de Euler, Runge-Kutta de 2do orden (Heun) y Runge-Kutta de 4to orden (RK4) analizando su precisión en un problema de valor inicial estándar. Adicionalmente, se extiende el análisis a sistemas de ecuaciones diferenciales mediante la simulación de una cinética de reacciones consecutivas $A \to B \to C$ en un reactor batch.

1. Planteamiento del problema

La modelación de procesos en ingeniería química, como reactores, transferencia de calor transitoria o dinámica de fluidos, frecuentemente resulta en Ecuaciones Diferenciales Ordinarias (EDOs) de la forma:

$$\frac{dy}{dx} = f(x, y), \quad y(x_0) = y_0 \tag{1}$$

Rara vez estas ecuaciones tienen solución analítica exacta en casos reales, por lo que se recurre a métodos numéricos que discretizan el dominio. El objetivo de este trabajo es implementar y comparar tres algoritmos clásicos y aplicar el más robusto (RK4) a un problema de cinética química.

1.1. Métodos Numéricos Empleados

1. Método de Euler (Primer Orden): Es el método más simple, basado en la truncación de la serie de Taylor a primer orden.

$$y_{i+1} = y_i + hf(x_i, y_i) \tag{2}$$

2. Método de Runge-Kutta 2do Orden (Heun): Mejora la estimación promediando la pendiente al inicio y al final del intervalo estimado.

$$k_1 = f(x_i, y_i) \tag{3}$$
$$k_2 = f(x_i + h, y_i + hk_1) \tag{4}$$
$$y_{i+1} = y_i + \frac{h}{2}(k_1 + k_2) \tag{5}$$

3. Método de Runge-Kutta 4to Orden (RK4): Es el estándar en ingeniería debido a que su error local es de orden $O(h^5)$. Utiliza un promedio ponderado de cuatro pendientes.

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{6}$$

2. Caso de Estudio: Cinética en Serie

Para la aplicación en sistemas de ecuaciones, se analiza una reacción consecutiva irreversible en un reactor intermitente (Batch) isotérmico. El esquema de reacción es:

$$\mathrm{A} \xrightarrow{k_1} \mathrm{B} \xrightarrow{k_2} \mathrm{C}$$

Donde A es el reactivo, B es el producto deseado (intermediario) y C es un subproducto de degradación. El balance de materia genera el siguiente sistema de EDOs acopladas:

$$\frac{d[A]}{dt} = -k_1[A] \tag{7}$$

$$\frac{d[B]}{dt} = k_1[A] - k_2[B] \tag{8}$$

$$\frac{d[C]}{dt} = k_2[B] \tag{9}$$

Parámetros de simulación:

- $k_1 = 0{,}5\,\text{min}^{-1}$
- $k_2 = 0{,}2\,\text{min}^{-1}$
- Condiciones iniciales: $[A]_0 = 1{,}0$ M, $[B]_0 = 0$, $[C]_0 = 0$.
- Tiempo de simulación: 0 a 20 min.

El objetivo es encontrar numéricamente el tiempo t_{max} donde la concentración de B es máxima.

3. Resultados y Discusión

3.1. Comparativa de Métodos (EDO simple)

Se resolvió $\frac{dy}{dx} = x + y$ con $y(0) = 1$. A continuación se presenta la tabulación detallada de los resultados obtenidos experimentalmente con los tres programas desarrollados. Se puede observar claramente cómo la precisión varía entre los métodos a medida que avanza el dominio x.

x	Euler	RK2 (Heun)	RK4
0.0	1.000000	1.000000	1.000000
0.1	1.100000	1.110000	1.110342
0.2	1.220000	1.242050	1.242805
0.3	1.362000	1.398465	1.399717
0.4	1.528200	1.581804	1.583648
0.5	1.721020	1.794894	1.797441
0.6	1.943122	2.040857	2.044236
0.7	2.197434	2.323147	2.327503
0.8	2.487178	2.645578	2.651079
0.9	2.815895	3.012364	3.019203
1.0	3.187485	3.428162	3.436559

Tabla 1: Evolución paso a paso de y_{num} para los tres métodos.

Al comparar el valor final en $x = 1{,}0$ con la solución analítica exacta $y(x) = 2e^x - x - 1$, se obtienen los siguientes errores relativos:

Método	Valor Aprox ($x = 1$)	Valor Exacto	Error Relativo (%)
Euler	3.187485	3.436564	7.25 %
RK2	3.428162	3.436564	0.24 %
RK4	**3.436559**	**3.436564**	**0.0001 %**

Tabla 2: Comparación de precisión final.

Se observa claramente que el método de Euler presenta un error significativo para pasos de $h = 0{,}1$, mientras que RK4 ofrece una precisión excelente, prácticamente idéntica a la solución analítica.

3.2. Resultados del Sistema Cinético (RK4)

Se procesaron los datos generados por el algoritmo RK4 para el sistema de reacciones en serie. A continuación se presenta un resumen de los puntos clave obtenidos de la simulación, seleccionando intervalos representativos para observar la evolución de las especies.

Tiempo (min)	[A] (mol/L)	[B] (mol/L)	[C] (mol/L)
0.0	1.0000	0.0000	0.0000
1.0	0.6065	0.3537	0.0398
3.0	**0.2231**	**0.5428**	**0.2341**
6.0	0.0498	0.4190	0.5312
9.0	0.0111	0.2570	0.7319
12.0	0.0025	0.1471	0.8505
15.0	0.0006	0.0821	0.9174
20.0	0.0000	0.0305	0.9695

Tabla 3: Resumen de la evolución de especies en el reactor batch.

Los resultados confirman el comportamiento cinético esperado de un par de reacciones en serie, el cual se ilustra detalladamente en forma de gráfico en la Figura 1.

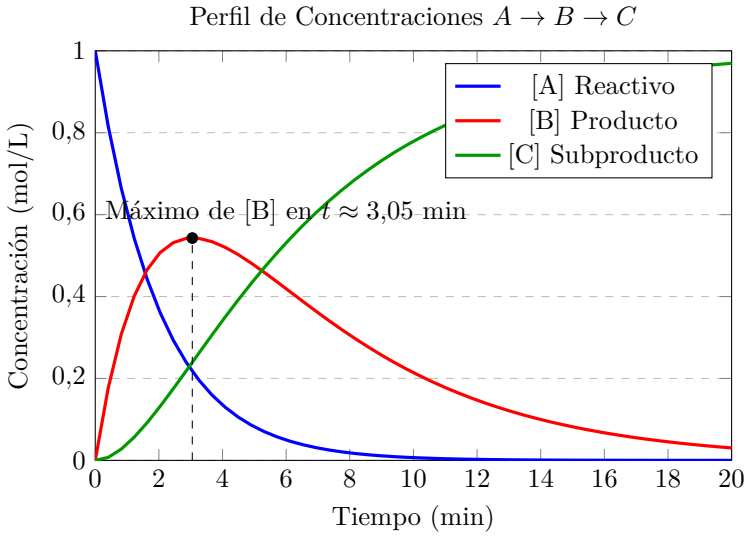

Figura 1: Evolución de las especies químicas simulada con RK4.

Análisis del Gráfico:

El gráfico revela la dinámica de competencia cinética típica de reacciones en serie:

1. **Decaimiento de A (Curva Azul):** La concentración del reactivo A disminuye exponencialmente desde el inicio. Esto es consistente con una cinética de primer orden ($r_A = -k_1[A]$), donde la velocidad de consumo es proporcional a la concentración remanente.

2. **Formación y Consumo de B (Curva Roja):** Esta es la curva crítica para el ingeniero de procesos. Al inicio, $[B]$ aumenta rápidamente porque la concentración de A es alta, favoreciendo la reacción $A \to B$. Sin embargo, a medida que se acumula B y se agota A, la velocidad de degradación $B \to C$ ($r_{deg} = k_2[B]$) empieza a competir con la formación.

3. **El Punto Óptimo:** El gráfico muestra un máximo claro. Según los datos tabulados, este pico ocurre entre $t = 3{,}0$ y $t = 3{,}1$ minutos, alcanzando una concentración máxima de $[B]_{max} \approx 0{,}5428$ M. Este instante representa el **tiempo de residencia óptimo** (τ_{opt}) para el reactor. Operar por menos tiempo resultaría en baja conversión de A, y operar por más tiempo causaría la pérdida del producto valioso hacia el subproducto no deseado C.

4. **Acumulación de C (Curva Verde):** Su formación es lenta al principio (periodo de inducción) porque depende de la existencia previa de B, y se acelera justo después del pico de B.

4. Conclusiones

La implementación de estos métodos numéricos en lenguaje C permitió resolver exitosamente ecuaciones diferenciales aplicadas a problemas de ingeniería química, siendo el presente, una aplicación mediante sistemmas de cinética química (característicos por ser sistemas de ecuaciones), esto para posteriormente poder modelar el empleo y diseño de un reactor. El análisis comparativo evidenció que el método de Runge-Kutta de 4to Orden (RK4) es la herramienta más óptima, ofreciendo un equilibrio entre facilidad de implementación y alta cercanía a la realidad en sus resultados. Esta exactitud es vital en el diseño y simulación de reactores, donde desviaciones cinéticas menores pueden derivar en la degradación de productos importantes de la reacción, generando desde accidentes hasta problemas económicos.

Anexo: Códigos en Lenguaje C

A continuación se presentan los códigos fuente desarrollados para este reporte.

A. Método de Euler

```c
//Autor: David Isaac Oliva Villar
//Método: Euler (Primer Orden)


#include <stdio.h>

// Definimos la función f(x, y) = dy/dx
double f(double x, double y) {
    return x + y; // Ejemplo: dy/dx = x + y
}

int main() {
    double x0 = 0.0, y0 = 1.0; // Condiciones iniciales
    double x_final = 1.0;      // Hasta dónde queremos calcular
    double h = 0.1;            // Tamaño de paso
    double x = x0, y = y0;

    printf("--- METODO DE EULER ---\n");
    printf("x\t\ty\n");
    printf("%.6f\t%.6f\n", x, y);

    while (x < x_final) {
        y = y + h * f(x, y); // Fórmula de Euler
        x = x + h;

        printf("%.6f\t%.6f\n", x, y);
    }

    return 0;
}
```

B. Método Runge-Kutta 2do Orden (Heun)

```c
//Autor: David Isaac Oliva Villar
//Método: Runge-Kutta 2do Orden (Método de Heun)

#include <stdio.h>

double f(double x, double y) {
    return x + y;
}

int main() {
    double x0 = 0.0, y0 = 1.0;
    double x_final = 1.0;
    double h = 0.1;
    double x = x0, y = y0;
    double k1, k2;

    printf("--- METODO DE RK2 (HEUN) ---\n");
    printf("x\t\ty\n");
    printf("%.6f\t%.6f\n", x, y);

    while (x < x_final) {
```

```
        k1 = f(x, y);
        k2 = f(x + h, y + h * k1);

        // Promedio de pendientes
        y = y + (h / 2.0) * (k1 + k2);
        x = x + h;

        printf("%.6f\t%.6f\n", x, y);
    }

    return 0;
}
```

C. Método RK4 (Univariable)

```
//Autor: David Isaac Oliva Villar
 //Método: Runge-Kutta 4to Orden (RK4)

#include <stdio.h>

double f(double x, double y) {
    return x + y;
}

int main() {
    double x0 = 0.0, y0 = 1.0;
    double x_final = 1.0;
    double h = 0.1;
    double x = x0, y = y0;
    double k1, k2, k3, k4;

    printf("--- METODO DE RK4 ---\n");
    printf("x\t\ty\n");
    printf("%.6f\t%.6f\n", x, y);

    while (x < x_final) {
        k1 = h * f(x, y);
        k2 = h * f(x + 0.5 * h, y + 0.5 * k1);
        k3 = h * f(x + 0.5 * h, y + 0.5 * k2);
        k4 = h * f(x + h, y + k3);

        // Promedio ponderado (1:2:2:1)
        y = y + (1.0 / 6.0) * (k1 + 2 * k2 + 2 * k3 + k4);
        x = x + h;

        printf("%.6f\t%.6f\n", x, y);
    }

    return 0;
}
```

D. Sistema de EDOs (Cinética Química)

```
// Autor: David Isaac Oliva Villar
//Tema: Cinética Química - Reacciones en Serie (A -> B -> C)
//Método: Runge-Kutta 4to Orden para Sistemas

#include <stdio.h>

```

```c
#define N 3 // Tenemos 3 especies: A, B, C

// Constantes de velocidad de reacción (unidades 1/min)
// k1: Velocidad de formación de B
// k2: Velocidad de degradación de B a C
const double K1 = 0.5;
const double K2 = 0.2;

// Función que define el sistema de ecuaciones diferenciales
// y[0] = Concentración de A
// y[1] = Concentración de B
// y[2] = Concentración de C
void cinetica_reaccion(double t, double y[], double f[]) {
    // d[A]/dt = -k1*[A]
    f[0] = -K1 * y[0];

    // d[B]/dt = k1*[A] - k2*[B]
    f[1] = (K1 * y[0]) - (K2 * y[1]);

    // d[C]/dt = k2*[B]
    f[2] = K2 * y[1];
}

int main() {
    // Configuración del tiempo (minutos)
    double t = 0.0;
    double t_final = 20.0; // Simulamos 20 minutos de reacción
    double h = 0.1;        // Paso de integración

    // Condiciones iniciales (Moles/Litro)
    // Empezamos solo con A puro.
    double y[N] = {1.0, 0.0, 0.0};

    // Variables auxiliares RK4
    double k1[N], k2[N], k3[N], k4[N];
    double y_temp[N];
    double f[N];

    printf("--- SIMULACION REACTOR BATCH (A->B->C) ---\n");
    printf("Tiempo(min)\t[A]\t\t[B]\t\t[C]\n");
    printf("%.2f\t\t%.4f\t\t%.4f\t\t%.4f\n", t, y[0], y[1], y[2]);

    while (t < t_final) {

        // --- Paso 1 ---
        cinetica_reaccion(t, y, f);
        for(int i=0; i<N; i++) k1[i] = h * f[i];

        // --- Paso 2 ---
        for(int i=0; i<N; i++) y_temp[i] = y[i] + 0.5 * k1[i];
        cinetica_reaccion(t + 0.5*h, y_temp, f);
        for(int i=0; i<N; i++) k2[i] = h * f[i];

        // --- Paso 3 ---
        for(int i=0; i<N; i++) y_temp[i] = y[i] + 0.5 * k2[i];
        cinetica_reaccion(t + 0.5*h, y_temp, f);
        for(int i=0; i<N; i++) k3[i] = h * f[i];

        // --- Paso 4 ---
        for(int i=0; i<N; i++) y_temp[i] = y[i] + k3[i];
        cinetica_reaccion(t + h, y_temp, f);
        for(int i=0; i<N; i++) k4[i] = h * f[i];
```

```
        // --- Actualización de concentraciones ---
        for(int i=0; i<N; i++) {
            y[i] = y[i] + (1.0/6.0) * (k1[i] + 2*k2[i] + 2*k3[i] + k4[i]);
        }

        t = t + h;

        // Imprimimos los resultados
        printf("%.2f\t\t%.4f\t\t%.4f\t\t%.4f\n", t, y[0], y[1], y[2]);
    }

    return 0;
}
```

Referencias

[1] S. C. Chapra y R. P. Canale, *Métodos Numéricos para Ingenieros*, 7ma ed., McGraw-Hill, 2015.

[2] H. S. Fogler, *Elements of Chemical Reaction Engineering*, 5ta ed., Pearson, 2016.