

UNIVERSIDAD DE GUANAJUATO
División de Ciencias e Ingenierías
Campus León



Proyecto 2

Solución de Ecuaciones Diferenciales Ordinarias Cinética de
Reacción de Primer Orden

Métodos Numéricos - NELI06028

Fecha de Entrega: 28 de Noviembre del 2025

Profesor: Dra. Alma Xochitl González Morales

Estudiante:

Pablo Ignacio Zorrilla Romo – 427762

Introducción

En este proyecto se resolverá una ecuación diferencial ordinaria como lo es la Cinética de Reacción de Primer Orden. La selección de esta ecuación no es algo trivial, puesto que se conoce bien su solución analítica, tiene una estructura matemática lineal y separable que nos va a permitir analizar bien las variables de los métodos como a estabilidad, convergencia y error de truncamiento.

Se explorará la solución de esta ecuación para la cinética de reacción de primer orden empleando tres métodos. El método de Euler, el método de Rung-Kutta de Segundo Orden y el método de Runge-Kutta de cuarto orden.

Se revisará si se es capaz de converger esta teoría cinética y los métodos numéricos. Con la implementación de tres códigos en C.

Objetivos

- Establecer el modelo matemático de una Reacción Cinética de Primer Orden
- Deducir la Solución Analítica
- Desarrollar códigos en C para los métodos de Euler, Rung-Kutta de Segundo Orden y Rung-Kutta de cuarto orden para resolver la ecuación diferencial ordinaria.

Planteamiento del Problema

La cinética química estudia la velocidad de las reacciones y los mecanismos mediante los cuales ocurren. Mientras que la termodinámica se encarga de determinar si una reacción es viable y en qué dirección ocurre con base en los estados iniciales y finales, la cinética se ocupa del tiempo, respondiendo la pregunta de la rapidez con la que se alcanza el equilibrio. (*INSTITUTO NACIONAL DEPARTAMENTO de QUÍMICA COORDINACIÓN 4º MEDIO COORDINADOR: MARIO TORO, n.d.*)

En el caso de este proyecto, nos centraremos en la reacción reversible de primer orden de la forma:



La “ley de velocidad” que gobierna a esta reacción indica que la velocidad de desaparición del reactivo A es directamente proporcional a su concentración instantánea. Matemáticamente, esto se expresa mediante la ecuación diferencial ordinaria. (“2.3: First-Order Reactions,” 2013)

$$\frac{dC}{dt} = -kC$$

Donde:

- $C(t)$ es la concentración molar del reactivo (mol/L) en un tiempo t .
- k es la constante específica de velocidad de reacción, con unidades de tiempo inverso (s^{-1})
- El signo negativo en esta reacción en específico indica que la concentración del reactivo disminuye en el tiempo

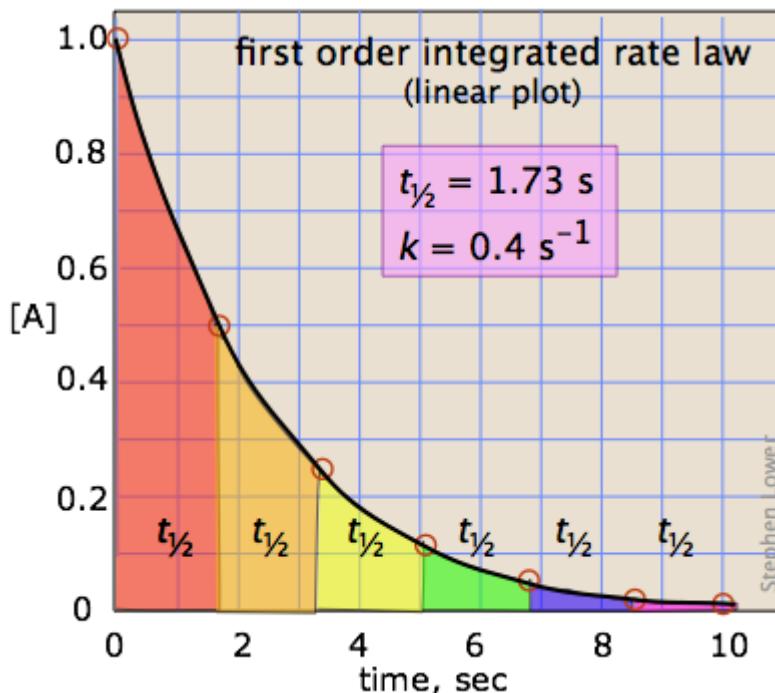


Ilustración 1 Gráfica de reacción de primer orden. ("2.3: First-Order Reactions," 2013)

Naturaleza de la constante K y ecuación de Arrhenius

Para los fines de este proyecto se tiene que comprender que k es “constante” en condiciones isotérmicas únicamente. La dependencia de la velocidad de reacción con la temperatura es descrita por la ecuación de Arrhenius. (*Chemical Kinetics*, 2025)

$$k = Ae^{-\frac{Ea}{RT}}$$

Esta relación exponencial indica que pequeños incrementos en la temperatura resultan en aumentos drásticos en la velocidad de reacción. En una simulación, si la reacción fuera exotérmica y el sistema no fuera isotérmico, la ecuación de la cinética, tendría que acoplarse a una ecuación de balance de energía (dT/dt) convirtiendo el problema en un sistema de EDOs. (Batiot et al., 2021)

Solución Analítica para Posterior Comparación

Gracias a la estructura de la ley de velocidad para una reacción de primer orden, se puede obtener una solución exacta, lo cual puede ayudar a validar los métodos numéricos

aplicados. Ya que se podrían comparar los resultados aproximados de los algoritmos (Euler, RK2, RK4) contra esta solución exacta. Por lo tanto, se podían cuantificar el error global y verificar la convergencia.

Derivación

$$\frac{dC}{C} = -k \, dt$$

Es una EDO por lo tanto se puede separar variables e integrar desde $t = 0$ y $C = C_0$ hasta algún otro estado.

$$\int_{C_0}^C \frac{1}{C'} dC' = -k \int_0^t dt'$$

$$\ln(C) - \ln(C_0) = -kt$$

$$\ln\left(\frac{C}{C_0}\right) = -kt$$

Despejando $C(t)$:

$$C(t) = C_0 e^{-kt}$$

Esta función es la de un decaimiento exponencial. Una propiedad de las reacciones de primer orden es que su vida media ($t_{1/2}$), es constante y no depende de la concentración inicial.

Códigos y Resultados

Para todos los códigos se emplearán los siguientes valores de prueba (también para la versión analítica)

$$\begin{aligned} k &= 0.5 \\ C_0 &= 100 \text{ M} \\ \text{Tiempo Inicial } t_0 &= 0 \\ \text{Tiempo Final } t_f &= 4 \\ \text{Paso del tiempo } h &= 1 \end{aligned}$$

Empleando el siguiente método para resolver analíticamente la EDO y además con la implementación del método de Euler, se tiene lo siguiente.

Codigo:

```
#include <stdio.h>
#include <math.h>

#define MAXP 10000

double k = 0.3;

double f(double t, double C){ return -k*C; }
double C_exacta(double t, double t0, double C0){ return C0*exp(-k*(t - t0)); }

int euler(double t0, double C0, double tf, double h, double t[], double C[], int maxp){
    int n = 0;
    t[0] = t0; C[0] = C0;
    while (t[n] < tf && n+1 < maxp) {
        double paso = h;
        if (t[n] + paso > tf) paso = tf - t[n];
        double Cn1 = C[n] + paso * f(t[n], C[n]);
        double tn1 = t[n] + paso;
        n++;
        t[n] = tn1; C[n] = Cn1;
        if (paso == 0.0) break;
    }
    return n+1;
}

int main(void){
    double t0 = 0.0, C0 = 1.0, tf = 5.0, h = 0.5;
    printf("EDO: dC/dt = -k C\n");
    printf("Ingresa k, C0, t0, tf, h: ");
    if (scanf("%lf %lf %lf %lf %lf", &k, &C0, &t0, &tf, &h) != 5) {
        printf("Entrada invalida. Se usan valores por defecto.\n");
    }
    if (h <= 0.0) h = 0.1;
    if (tf <= t0) tf = t0 + 1.0;
    if (k < 0.0) k = -k;

    double t[MAXP], C[MAXP];
    int m = euler(t0, C0, tf, h, t, C, MAXP);

    printf("\nMetodo de Euler para dC/dt = -k C\n");
    printf("k=% .8f, C0=% .8f, t0=% .8f, tf=% .8f\n", k, C0, t0, tf);
    printf("%10s %18s %18s %18s\n", "k", "t_k", "C_k (Euler)", "C_exacta");
    for (int i=0; i<m; ++i){
        double Ce = C_exacta(t[i], t0, C0);
        printf("%10d %18.10f %18.10f %18.10f\n", i, t[i], C[i], Ce);
    }

    double Cfin = C[m-1];
    double Cref = C_exacta(t[m-1], t0, C0);
    double err = fabs(Cfin - Cref);
```

```

    printf("\nAproximacion final: C(tf)=%.12f en t=%.6f\n", Cfin, t[m-1]);
    printf("Referencia exacta: C(tf)=%.12f    Error=%e\n", Cref, err);
    return 0;
}

```

Resultado:

$k=0.500000000$, $C_0=100.000000000$, $t_0=0.000000000$, $tf=4.000000000$, $h=1.000000000$

k	t_k	C_k (Euler)	C_{exacta}
0	0.0000000000	100.000000000000	100.000000000000
1	1.0000000000	50.000000000000	60.6530659713
2	2.0000000000	25.000000000000	36.7879441171
3	3.0000000000	12.500000000000	22.3130160148
4	4.0000000000	6.250000000000	13.5335283237

Aproximacion final: $C(tf)=6.2500000000000$ en $t=4.000000$

Método de Rung-Kutta

Código:

```

#include <stdio.h>
#include <math.h>

#define MAXP 10000
/*
 * Este metodo promedia dos pendientes pero para esta EDO se puede
simplificar
 * por lo tanto en la función rk2 el metodo se vuelve:
 * C_n+1=C_n[1-hk+1/2*(hk)^2]
 */
double k = 0.3;

double f(double t, double C){ return -k*C; }
double C_exacta(double t, double t0, double C0){ return C0*exp(-k*(t - t0)); }

int rk2(double t0, double C0, double tf, double h, double t[], double C[], int maxp) {
    int n = 0;
    t[0] = t0; C[0] = C0;
    while (t[n] < tf && n+1 < maxp) {
        double paso = h;
        if (t[n] + paso > tf) paso = tf - t[n];
        double k1 = f(t[n], C[n]);
        double k2 = f(t[n] + paso, C[n] + paso*k1);
        double Cn1 = C[n] + 0.5*paso*(k1 + k2);
        double tn1 = t[n] + paso;
        n++;
        t[n] = tn1; C[n] = Cn1;
        if (paso == 0.0) break;
    }
    return n+1;
}

```

```

}

int main(void){
    double t0 = 0.0, C0 = 1.0, tf = 5.0, h = 0.5;
    printf("EDO: dC/dt = -k C (Runge-Kutta de 2o orden - Heun)\n");
    printf("Ingresa k, C0, t0, tf, h: ");
    if (scanf("%lf %lf %lf %lf", &k, &C0, &t0, &tf, &h) != 5){
        printf("Entrada invalida. Se usan valores por defecto.\n");
    }
    if (h <= 0.0) h = 0.1;
    if (tf <= t0) tf = t0 + 1.0;
    if (k < 0.0) k = -k;

    double t[MAXP], C[MAXP];
    int m = rk2(t0, C0, tf, h, t, C, MAXP);

    printf("\nk=% .8f, C0=% .8f, t0=% .8f, tf=% .8f, h=% .8f\n\n", k, C0, t0,
    tf, h);
    printf("%10s %18s %18s %18s\n", "k", "t_k", "C_k (RK2)", "C_exacta");
    for (int i=0; i<m; ++i){
        double Ce = C_exacta(t[i], t0, C0);
        printf("%10d %18.10f %18.10f %18.10f\n", i, t[i], C[i], Ce);
    }

    double Cfin = C[m-1];
    double Cref = C_exacta(t[m-1], t0, C0);
    double err = fabs(Cfin - Cref);
    printf("\nAproximacion final: C(tf)=%.12f en t=%.6f\n", Cfin, t[m-
    1]);
    printf("Referencia exacta: C(tf)=%.12f Error=%.3e\n", Cref, err);
    return 0;
}

```

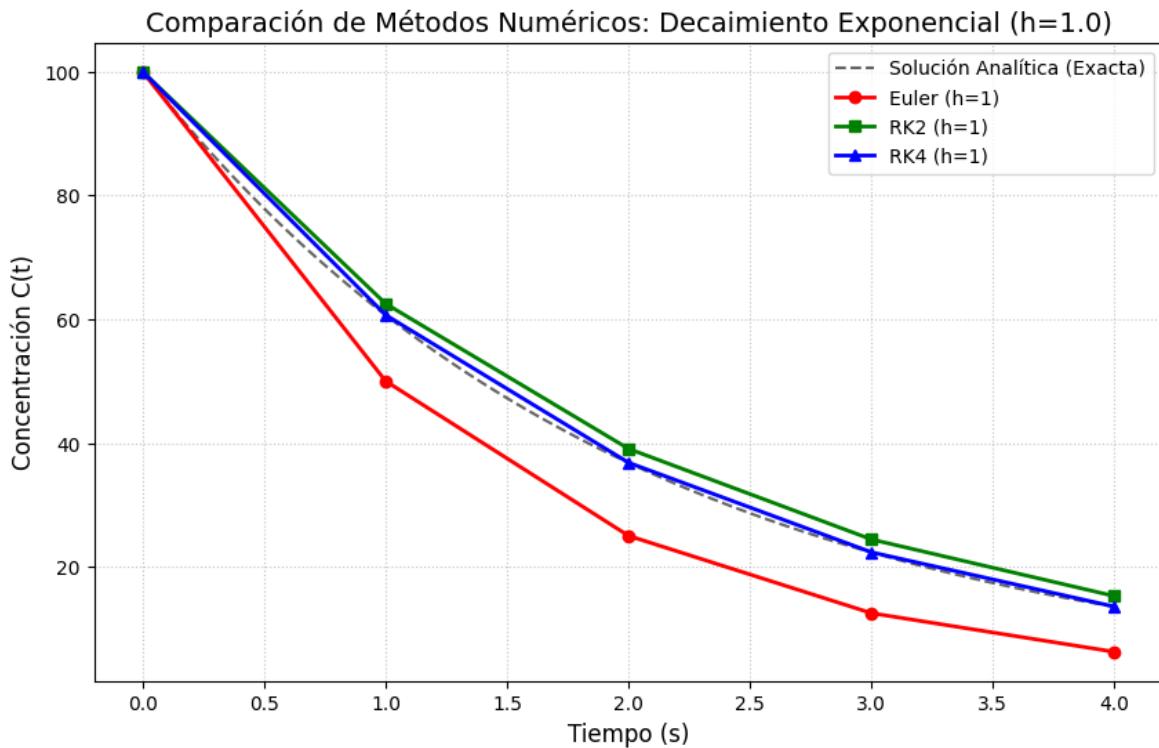
Resultados:

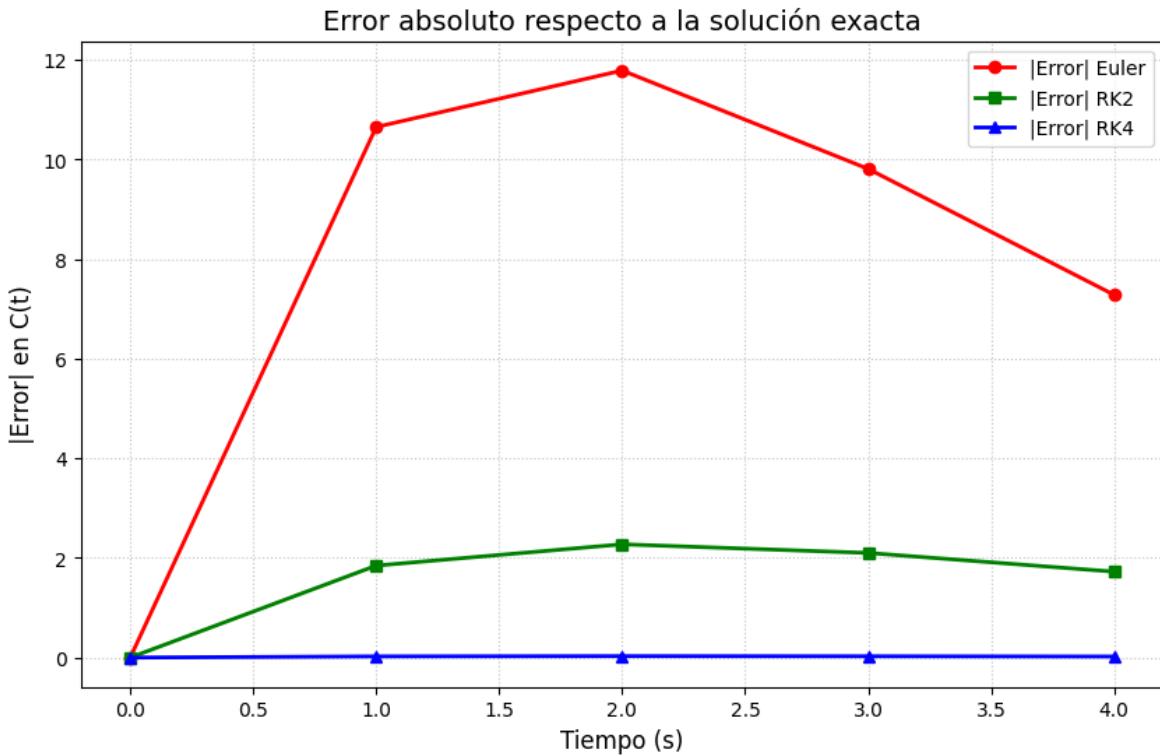
k	t_k	C_k (RK2)	C_exacta
0	0.0000000000	100.0000000000	100.0000000000
1	1.0000000000	62.5000000000	60.6530659713
2	2.0000000000	39.0625000000	36.7879441171
3	3.0000000000	24.4140625000	22.3130160148
4	4.0000000000	15.2587890625	13.5335283237

Tabla de Resultados:

Tiempo (t)	Exacta (Cex)	Euler (Ceu)	% Error Euler	RK2 (Crk2)	% Error RK2	RK4 (Crk4)	% Error RK4
0	100	100	0.00%	100	0.00%	100	0.00%
1	60,653	50,000	17.56%	62,500	3.05%	60,677	0.04%
2	36,788	25,000	32.04%	39,063	6.18%	36,817	0.08%
3	22,313	12,500	43.98%	24,414	9.42%	22,340	0.12%
4	13,534	6,250	53.82%	15,259	12.75%	13,555	0.16%

Por el comportamiento de estos métodos, también se gráfico el error absoluto en cada tiempo (cada segundo) para cada uno de los métodos y la gráfica de los datos obtenidos:





Discusión y Conclusión

Al realizar un análisis al comportamiento y a como funcionan cada uno de los tres métodos empleados se puede observar lo siguiente.

Se observa una caída fuerte en la precisión de Euler, esto es porque se uso un paso de 1, Euler asume que la pendiente al inicio del intervalo es constante durante todo el segundo. Pero como la curva real es una exponencial decreciente la pendiente realmente se suaviza, mientras que Euler baja demasiado rápido y por eso al final nos queda un error de alrededor de 54%

Para el caso de RK2, en este método se “mejora” un poco y se usa el promedio de las pendientes. En este caso se tiende a sobreestimar la curva, quedando por encima de la solución exacta.

Y, por último, tenemos a un caso de éxito y a nuestro ganador, el RK4, a pesar de un paso de tiempo enorme RK4 logra un error menor al 0.2%. Esto demuestra el por que según el libro (C. Chapra) es un estándar en ingeniería.

Para concluir, a lo largo de este proyecto se demostró la eficacia de los métodos numéricos (especialmente el RK4) para simular la cinética de una reacción cinética de primer orden, casi igualando en algunos casos a la solución analítica.

Con la comprensión de las debilidades y fortalezas de cada uno de los métodos, en el futuro se podrá implementar con éxito estos métodos en problemas reales, en la industria o en el estudio de la Ingeniería Química.

Referencias

2.3: *First-Order Reactions*. (2013, October 2). Retrieved November 28, 2025, from Chemistry LibreTexts website:

[https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Supplemental_Modules_\(Physical_and_Theoretical_Chemistry\)/Kinetics/02%3A_Reaction_Rates/2.03%3A_First-Order_Reactions](https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Supplemental_Modules_(Physical_and_Theoretical_Chemistry)/Kinetics/02%3A_Reaction_Rates/2.03%3A_First-Order_Reactions)

Batiot, B., Rogaume, T., Richard, F., Luche, J., Collin, A., Guillaume, E., & José Luis Torero. (2021). Origin and Justification of the Use of the Arrhenius Relation to Represent the Reaction Rate of the Thermal Decomposition of a Solid. *Applied Sciences*, 11(9), 4075–4075. <https://doi.org/10.3390/app11094075>

Chemical Kinetics. (2025). Retrieved November 28, 2025, from Google Books website: https://books.google.com.mx/books?id=nHux3YED1HsC&redir_esc=y

INSTITUTO NACIONAL DEPARTAMENTO DE QUÍMICA COORDINACIÓN 4° MEDIO COORDINADOR: MARIO TORO. (n.d.). Retrieved from https://institutonacional.cl/wp-content/uploads/2020/04/4-Quimica-U_Cinetica.pdf