ESC113 Group 8

Numerical Solution of Reaction Kinetics Using MATLAB

Team Members: Rounak Mandal,Rishi Gupta, Rudra Dwivedi ,Krishna Yadav ,Krish Agrawal ,Roushan Upadhyay,Kunal Chandra

Introduction :

Chemical reaction kinetics is fundamental to understanding reactor design and optimizing industrial processes. This project aims to develop a MATLAB-based computational tool to simulate reaction kinetics in a batch reactor. The tool provides a generic framework for solving ordinary differential equations (ODEs) governing reactant and product concentrations over time, accommodating various reaction orders and numerical methods. By implementing Explicit Euler, Implicit Euler, and Runge-Kutta 4 (RK4) algorithms, the tool enables users to analyze the trade-offs between computational efficiency and accuracy.

Project Objectives

1 MATLAB App Development

Create a MATLAB app with a graphical user interface (GUI) for simulating reaction kinetics.

2 Numerical Methods Implementation

Implement numerical methods (Explicit Euler, Implicit Euler, RK4) to solve ODEs for reactions of arbitrary order.

3 User Parameter Input

Enable users to input parameters such as rate constants, initial concentrations, time steps, and reaction orders.

4 Visualization

Visualize concentration profiles of reactants and products over time.

Mathematical Formulation

The core ODE for a reaction of order n is:

$$\frac{dC_A}{dt} = -k \cdot C_A^n$$

Where:

- C_A : Concentration of reactant A (mol/L)
- k: Rate constant
- n: Reaction order (0 to 10)

For a first-order reaction (n = 1), the analytical solution is:

$$C_A(t) = C_{A0} \cdot e^{-kt}$$

For higher orders, numerical methods are required.

Numerical Methods

Explicit Euler

$$C_A^{i+1} = C_A^i + h \cdot \left(-k \cdot (C_A^i)^n\right)$$

Advantages

• Simple Implementation: Easy to code and understand.
• Low Computational Cost: Only one function evaluation per step.
• Explicit Form: No need for iterative solver

Limitations

• Conditionally Stable: Requires small time steps (h) for stiff ODEs.
• Error Accumulation: Global error ∝h(first-order accuracy).
• Overshoots/Instability: May produce non-physical results (e.g., negative concentrations) if h *is* too large.

Implicit Euler

$$C_A^{i+1} = C_A^i - h \cdot k \cdot (C_A^{i+1})^n$$

Solved iteratively using the Newton-Raphson method.

Advantages

• Unconditionally Stable: Works for large h (no stability restrictions).
• Better for Stiff Equations: Suitable for reactions with fast/slow timescales.
• No Oscillations: Avoids artificial instability.

Limitations

• Higher Computational Cost: Requires solving nonlinear equations at each step.
• First-Order Accuracy: Global error still ∝*h*.
• Complex Implementation: Needs iterative solvers (e.g., Newton-Raphson).

Runge-Kutta 4th Order (RK4)

$$k_1 = f(t^i, C_A^i)$$
$$k_2 = f\left(t^i + \frac{h}{2}, C_A^i + \frac{hk_1}{2}\right)$$
$$k_3 = f\left(t^i + \frac{h}{2}, C_A^i + \frac{hk_2}{2}\right)$$
$$k_4 = f(t^i + h, C_A^i + hk_3)$$
$$C_A^{i+1} = C_A^i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

Advantages

- **4th-Order Accuracy:** Global error $\propto h^4$
- **Balanced Stability:** Works well for moderately stiff systems.
- **No Iterations Required:** Explicit but more accurate than Euler.

Limitations & Computational Cost

- **Higher Cost per Step:** 4 function evaluations per step.
- **Not Fully Stiff-Stable:** May still fail for extremely stiff problems.
- **Step Size Sensitivity:** Requires careful h selection.

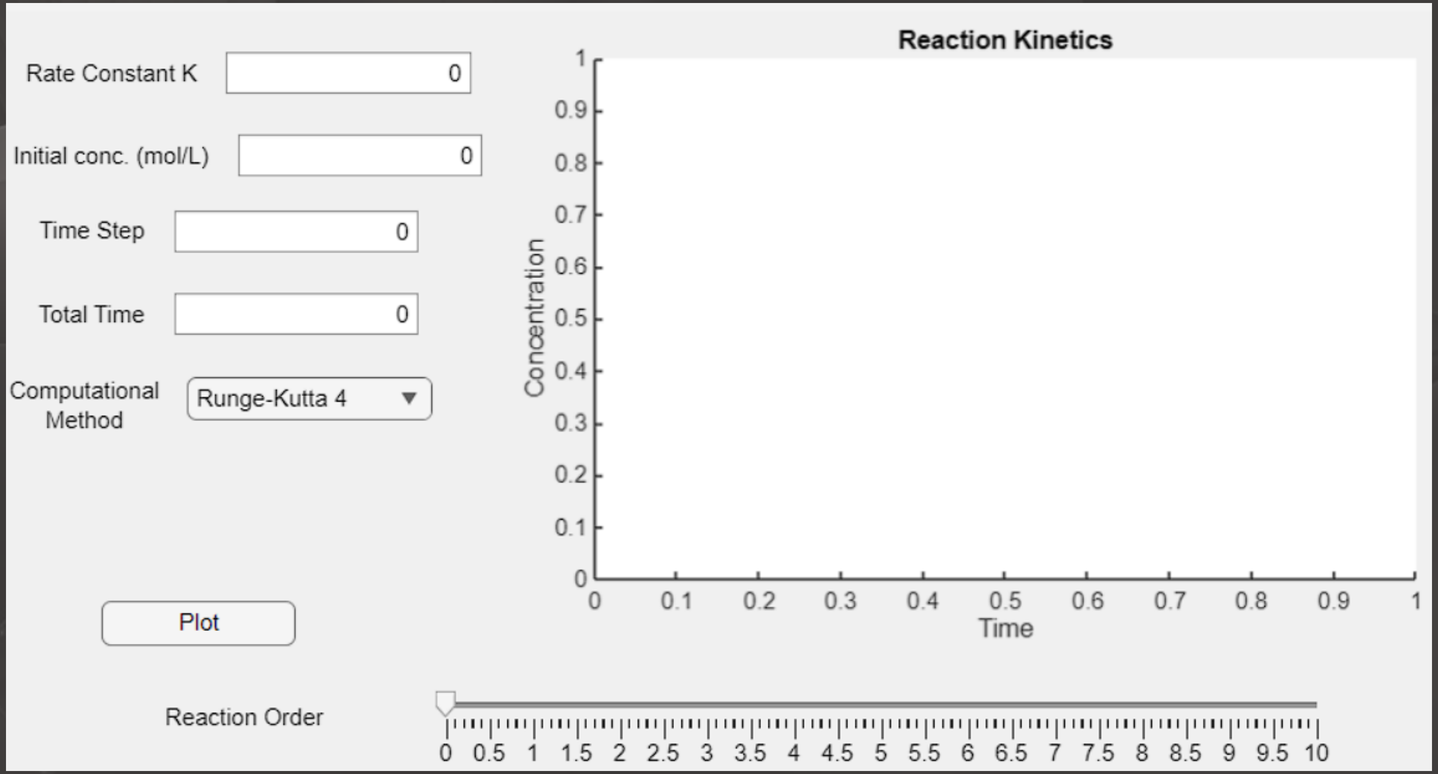
Implementation

MATLAB App Structure

- Input fields for k, initial concentration, time step, and total time.
- A slider for reaction order (n = 0 to 10).
- A drop down menu to select the numerical method.
- A plot panel to visualize results.

Code Workflow

- Input Validation: Ensures positive values for k, concentration, and time parameters.
- ODE Solver: Computes $C_A(t)$ and $C_B(t) = C_{A0} - C_A(t)$ (Product concentration) using the selected method.
- Plotting: Generates concentration-time curves for reactants and products.



Rate Constant K: 0

Initial conc. (mol/L): 0

Time Step: 0

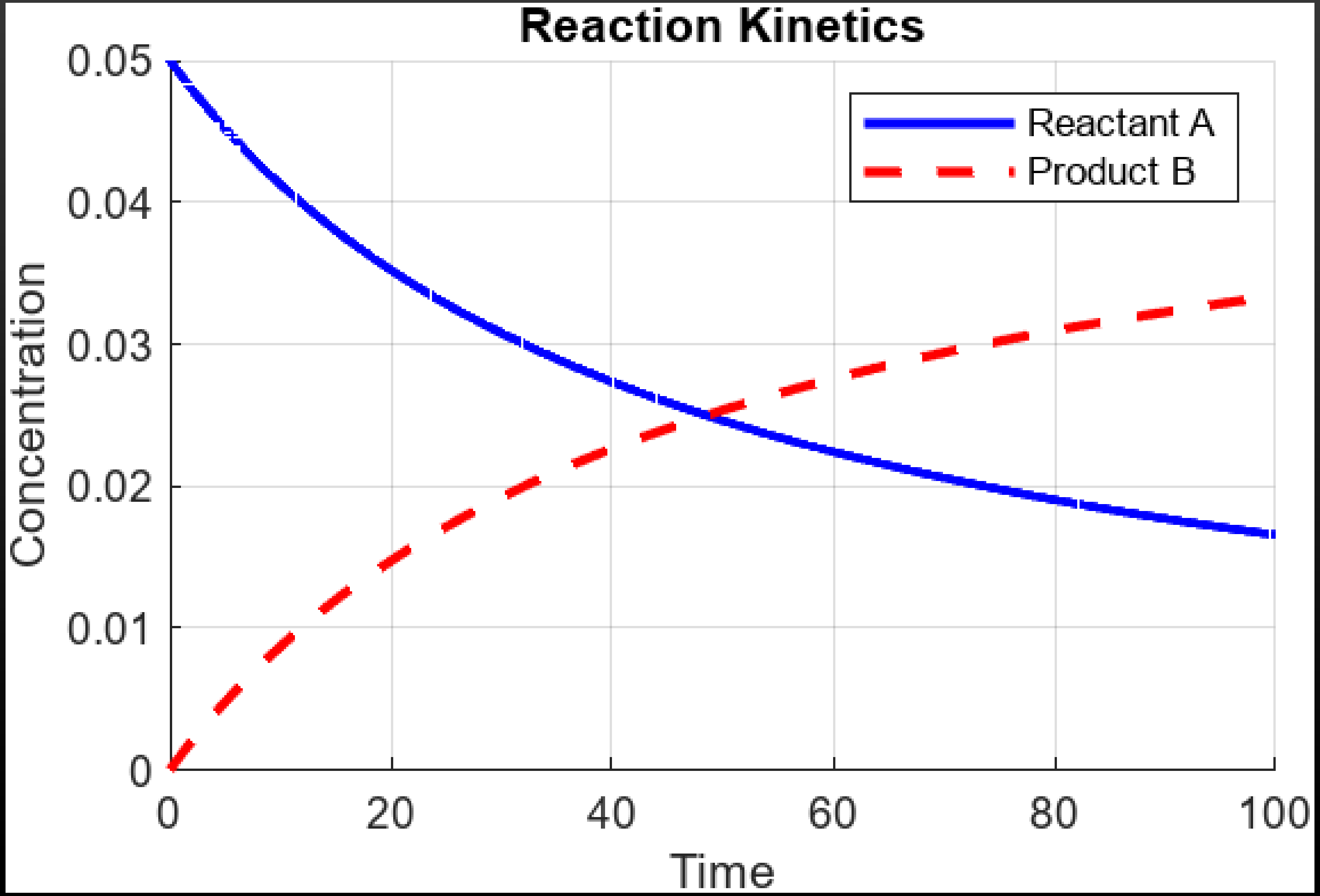
Total Time: 0

Computational Method: Runge-Kutta 4

Problem statement

- **Reaction:** 2 $NO_2 \rightarrow 2NO + O_2$
- **Rate constant:** k=0.54 $M^{-1}s^{-1}$
- **Initial concentration:** $[NO_{2_0}]$=0.050 M
- **Time span:** 100 seconds

Order of the Reaction=2



Method Comparison

Method	Accuracy	Stability	Computational Cost
Explicit Euler	Low	Low	Low
Implicit Euler	Moderate	High	High
RK4	High	Moderate	Moderate

Conclusion

This project successfully developed a versatile MATLAB app for simulating reaction kinetics.

Key findings include:

- RK4 provides the best balance between accuracy and stability.
- Explicit Euler is suitable for quick approximations with small time steps.
- The GUI simplifies parameter input and visualization, making it accessible for educational and industrial use.

Future work could expand the app to handle complex reaction networks and temperature-dependent rate constants.

Thank You!