Scientific Research Publishing

Higher Order Approximation of Advection Diffusion Equation by Semi-Discretization Method

Khandoker Nasrin Ismet Ara, Md. Sabbir Alam, Laek Sazzad Andallah

Department of Mathematics, Jahangirnagar University, Savar, Bangladesh
Email: s14dh051@gmail.com

Abstract

A system of ordinary differential equations (ODEs) is produced by the semi-discretize method of discretizing the advection diffusion equation (ADE). Runge-Kutta methods of the second and fourth orders are used to solve the system of ODEs. We compute the ADE numerically for initial and boundary conditions, for which the exact solution is known. In the semi-discretization approach, we estimate the error for both the second and fourth-order Runge-Kutta schemes. The semi-discretization method's outcome is contrasted with the ADE's numerical solution derived from the complete discretization explicit centered difference scheme.

Keywords

Advection Diffusion Equation, Semi-Discretization, Finite Difference Scheme, Rate of Convergence, System of ODE's

1. Introduction

The semi-discretization method (SDM) is used to derive numerical solutions for time-dependent partial differential equations (PDEs). The numerical solution procedure is known as space semi-discretization, in which the time variable is left continuous while the space variable is discretised and time integration is applied to transform the PDEs into a system of ordinary differential equations (ODEs). We look at the mathematical model of advection diffusion, which may be used to simulate natural processes in a variety of natural science and engineering applications. Many scholars have tried to solve this equation and apply it in their simulations [1]-[8]. A finite difference approach is used to solve the advection diffusion problem. It is well known that the optimal approach to solving the advection dif-

fusion problem is usually to use an FDS (finite difference scheme) [7]-[10]. The researcher explored the operator splitting approach for the numerical solution of the Advection Diffusion Equation [2]. The Crank-Nicolson (CN) finite difference scheme and the characteristic method with cubic spline interpolation (MOC-CS) were used to implement these techniques [2]. The results were compared to the analytical solution. It is evident that even with large time steps; the applied method yields correct results and has a smaller error than other methods. Explicit finite difference techniques are used in this study to solve the advection diffusion problem [8]-[10]. We look into an alternative method that produces a system of ODEs with a time independent variable: the semi-discretization method on spatial variables [1]. We use Runge-Kutta's second and fourth order methods as well as Euler's method to solve this system of ODEs. In order to calculate the unknown concentration $u(x,t)$, we create an algorithm that combines the Euler and Runge Kutta methods for the system of ODEs.

2. The Semi-Discretization Techniques

2.1. Advection Diffusion Equations (ADE)

Advection Diffusion equations have both advective and diffusive terms together, represented as a partial differential equation below:

$$\frac{\partial u(x,t)}{\partial t} + c\frac{\partial u(x,t)}{\partial x} = D\frac{\partial^2 u(x,t)}{\partial x^2} \tag{1}$$

with initial condition

$$u(x,t_0) = f(x); a < x < b;$$

and boundary condition

$$u(a,t) = u_a(t); \; t_0 < t < T$$
$$u(b,t) = u_b(t); \; t_0 < t < T$$

where, c is the velocity of the medium in the x direction, D is the diffusion coefficient. An advection diffusion problem whose general solution [1] is given in **Figure 1**.
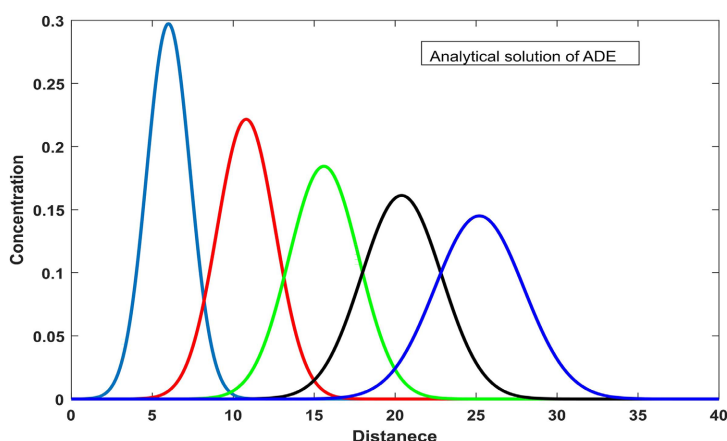

Figure 1. Analytical solution of advection diffusion equation.

2.2. Finite Difference Discretization

Among the discretization methods for partial differential equations, the finite difference method (FDM) is the most traditional. Numerous contemporary numerical methods for transport phenomena have their roots in finite difference approximations that were created between the late 1950s and the early 1980s. FDM is especially easy to derive and apply to structured meshes, which are topologically equal to a uniform Cartesian grid. The nodal value of the approximate solution at node m.

$$u_m(t) \cong u(x_m, t) \tag{2}$$

is a pointwise approximation to the true solution of the partial differential equation.

3. Numerical Method

3.1. Semi-Discretization Euler Methods

Our aim is to introduce several numerical methods of solving ODEs to obtain approximate solutions [1]. We consider some simple space discretization on a uniform grid. We divide the spatial interval $[0, L]$ into $M+1$ equal sub-interval such that $x_1 < x_2 < x_3 < \cdots, < L$ with $x_m = (m-1)\Delta x$, $m = 1, 2, 3, \cdots, M+1$ and $\Delta x = \dfrac{L}{M}$. Approximations $u(t) \approx u(x_m, t)_0$ are found by replacing the spatial derivatives by difference quotients. This gives a finite difference discretization in space. Setting

$$u(t) = (u_1(t), \cdots, u_m(t))^{\mathrm{T}}.$$

Therefore, we get a system of ordinary difference equations (ODEs) of (1)

$$u'(t) = F(t, u(t)), t > 0, \ u(0) = u_0 \tag{3}$$

with a given initial value $u(0)$. To approximate (1.1) with we also divide the time interval $[0, T]$ into $N+1$ equal sub-interval such that $t_1 < t_2 < t_3 < \cdots < T$ with $t_n = (n-1)\Delta t$, $n = 1, 2, 3, \cdots, N+1$, and $\Delta t = \dfrac{T}{N}$. For purpose of the notation $\Delta t = k$. So, the following approximations are constructed $u^0, u^1, \cdots, u^n, u^{n+1}$.

Where u^n is an intended approximation of $u(t_n)$. Now, with time step Δt for the numerical solution of advection diffusion equation with the Euler method is-

$$u^{n+1} = u^n + \Delta t \, F(t_n, u^n) \tag{4}$$

3.2. Full Discretization of Explicit Centered Difference Scheme (ECDS)

To approximate the solution to Equation (1) using the Explicit Centered Difference Scheme, we use the following approximations

$$u_t\left(x_m,t^n\right) \cong \frac{u_m^{n+1}-u_m^n}{\Delta t}$$

$$u_x\left(x_m,t^n\right) \cong \frac{u_{m+1}^n-u_{m-1}^n}{2\Delta x}$$

$$u_{xx}\left(x_m,t^n\right) \cong \frac{u_{m+1}^{n+1}-2u_m^n+u_{m-1}^n}{\left(\Delta x\right)^2}$$

where, Δx is the spatial step, Δt is the time step, m and n is spatial and temporal node respectively. Substituting these above equations in (1) and solving for unknown u_m^{n+1}. We obtain, $u_m^{n+1} = \left(\frac{\alpha}{2}+\gamma\right)u_{m-1}^n + \left(1-2\gamma\right)u_m^n + \left(\gamma-\frac{\alpha}{2}\right)u_{m+1}^n$

Where, $\frac{c\Delta t}{\Delta x} = \alpha$ and $\frac{D\Delta t}{\Delta x^2} = \gamma$. Stability condition $\frac{c\Delta t}{\Delta x} \le 1$ & $\frac{D\Delta t}{\Delta x^2} \le \frac{1}{2}$.

3.3. Semi-Discretization Runge-Kutta Methods

Euler method and ECDS both are first order methods. In order to develop efficient, highly accurate approximation algorithm, higher-order difference methods are designed. Assume $t^{n+1} = t^n + \Delta t$, then we use the Runge-Kutta 2nd order method and obtain

$$u^{n+1} = u^n + \frac{\Delta t}{2}\left(F\left(t_n,u^n\right) + F\left(t_{n+1},u^{n+1}\right)\right) \tag{5}$$

This is called the Runge-Kutta 2nd order for ADE.

The formula of fourth-order Runge-Kutta method is given by where

$$t^{n+1} = t^n + \frac{\Delta t}{2}$$

$$K_1 = F\left(t_n,u^n\right)$$

$$K_2 = F\left(t_{n+1},u^{n+1}+hK_1\right)$$

$$K_3 = F\left(t_{n+1},u^{n+1}+hK_2\right)$$

$$K_4 = F\left(t_{n+1},u^n+K_3\right)$$

$$u^{n+1} = u^n + \frac{h}{6}\left(K_1+2K_2+2K_3+K_4\right) \tag{6}$$

Which are two simplest methods for the semi-discretization of $F\left(t,u\right) = A\dot{u} + b\left(t\right)$.

4. Numerical Results and Discussion

4.1. Discussion Semi-Discretization Methods

The general solution for the advection diffusion Equation (1) this study, we assume that spatial length, $l = 50\,\text{m}$, $c = 0.6\,\text{m/s}$, $D = 0.09\,\text{m}^2/\text{s}$, $\Delta x = 0.1$, $\Delta t = 0.08$ total time $t = 10\,\text{sec}$ to $t = 50\,\text{sec}$. The advection diffusion equation for this problem $u_t + cu_x = Du_{xx}$ is to be solved with initial and boundary conditions

$$u(x,0) = f(x).$$

$$u(x = 0, t) = f_0(x)$$

$$u(x = l, t) = f_l(x); 0 \le t \le T$$

A computer programing code and various values of time steps are to be used to investigate the numerical schemes [9]-[12] (**Table 1**).

Table 1. Elapsed time with different temporal grid point.

Δt	Δx	nt	nx	ECDS (E.T.sec)	Euler (E.T. sec)	Runge Kutta	
						2nd order (E.T)	4th order (E.T)
0.0020	0.0667	20000	600	1.742224	20.768796	43.015423	341.596993
0.0027	0.0667	15000	600	1.454636	28.784721	77.852146	256.830589
0.0040	0.0667	10000	600	1.397545	18.871882	53.445049	152.704436
0.0050	0.0800	8000	600	1.475086	14.689404	28.975703	118.589866
0.0080	0.0800	5000	600	1.010213	9.434639	25.819919	61.982884

4.1.1. Case 1: Euler Method

For this case the time step is increased to $\Delta t = 0.008$, $\Delta x =$, $\dfrac{c\Delta t}{\Delta x} = \alpha$ known as advection equation number and $\dfrac{D\Delta t}{(\Delta x)^2} = \gamma$ known as diffusion term.

Numerical implementation of semi-discretization method our solving Equation (1):

$$u_t + cu_x = Du_{xx}$$

Initial condition: $u_m^0 = f(x_m) = u(x, t = 0)$.

Boundary condition: $u(x = 0, t) = u_0^n = f_1(t_n)$; $u(x = b, t) = u_b^n = f_b(t_n)$

We can write out the matrix system of equation we will solve numerically for the concentration u. Suppose we use 4 grid points $x_1, x_2, x_3, x_4 = x_{m+1}$ *i.e.* $m = 3$ in this example. $\boldsymbol{u}^n = \begin{pmatrix} u_2^n \\ u_3^n \end{pmatrix}$, solution for concentration vector $\boldsymbol{u}^n$ at time t_n. The boundary condition gives $u_1^n = u(x = 0, t_n)$ and $u_{m+1}^n = u_4^n = u(x = b, t_b)$.

We can rewrite general nth term grid point in equation as

$$\boldsymbol{u}^{n+1} = \begin{pmatrix} u_2^n \\ u_3^n \end{pmatrix} + \alpha \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} u_2^n \\ u_3^n \end{pmatrix} + \lambda \begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix} \begin{pmatrix} u_2^n \\ u_3^n \end{pmatrix} - \alpha \begin{pmatrix} u_1^n \\ u_4^n \end{pmatrix} + \lambda \begin{pmatrix} u_1^n \\ u_4^n \end{pmatrix}$$

$$\boldsymbol{u}^{n+1} = \begin{pmatrix} u_2^n \\ u_3^n \end{pmatrix} + \Delta t \left\{ -\alpha A \begin{pmatrix} u_2^n \\ u_3^n \end{pmatrix} + \lambda B \begin{pmatrix} u_2^n \\ u_3^n \end{pmatrix} - \alpha \begin{pmatrix} u_1^n \\ u_4^n \end{pmatrix} + \lambda \begin{pmatrix} u_1^n \\ u_4^n \end{pmatrix} \right\}$$

where, $A = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$ and $B = \begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix}$

$$u^{n+1} = \begin{pmatrix} u_2^n \\ u_3^n \end{pmatrix} + \Delta t \boldsymbol{F}; \ \alpha = \frac{c}{2\Delta x} \ \& \ \gamma = \frac{D}{(\Delta x)^2}$$

$$u_m^{n+1} = u_m^n + \Delta t \left\{ \alpha \left(u_{m+1}^n - u_{m-1}^n \right) + \gamma \left(u_{m+1}^n - 2u_m^n + u_{m-1}^n \right) \right\};$$

In the matrix form

$$\therefore u_m^{n+1} = u_m^n + \Delta t \, F_{m,n}\left(t_n, u^n\right) \tag{7}$$

4.1.2. Case II: Runge-Kutta Method

We consider the semi-discretize form in Equation (4), so we can write from the **case 1**:

$$\boldsymbol{K}_1 = F\left(t_n, u^n\right) = \left(A\alpha + B\gamma\right)\boldsymbol{u} + \begin{bmatrix} (\alpha + \gamma)u_1 \\ 0 \\ (\alpha + \gamma)u_{M+1} \end{bmatrix}$$

$$\boldsymbol{K}_2 = F\left(t_{n+1}, u^{n+1}\right) = \left(A\alpha + B\gamma\right)\left(\boldsymbol{u} + h\boldsymbol{K}_1\right) + \begin{bmatrix} (\alpha + \gamma)\left(\boldsymbol{u} + h\boldsymbol{K}_1\right) \\ 0 \\ (\alpha + \gamma)\left(u_{M+1} + h\boldsymbol{K}_1\right) \end{bmatrix}$$

$$\boldsymbol{K}_1 = \left\{ \alpha \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} + \lambda \begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix} \right\} \begin{pmatrix} u_2^n \\ u_3^n \end{pmatrix} + \begin{bmatrix} (\alpha + \lambda)u_1^n \\ 0 \\ (-\alpha + \lambda)u_4^n \end{bmatrix}$$

$$\boldsymbol{K}_2 = \left\{ \alpha \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} + \lambda \begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix} \right\} \begin{pmatrix} u_2^n + hK_1 \\ u_3^n + hK_1 \end{pmatrix} + \begin{bmatrix} (\alpha + \lambda)\left(u_1^n + hK_1\right) \\ 0 \\ (-\alpha + \lambda)\left(u_4^n + hK_1\right) \end{bmatrix}$$

where, $A = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$ and $B = \begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix}$

Now we get,

$$u^{n+1} = u^n + \frac{\Delta t}{2}\left(K_1 + K_2\right) \tag{8}$$

Equation (8) is the semi-discretized Runge-Kutta 2nd order method of advection diffusion equation. Similarly, we calculate K_3 and K_4 Runge Kutta 4th order method of advection diffusion equation.

5. Error Analysis and Convergence

5.1. Error Analysis

Figure 2 shows that the comparison of relative error for two finite difference schemes. The relative error for ECDS which remains below 0.0010 and for Runge-Kutta 4th order the relative error is 0.0004.

Figure 3 shows that the comparison of relative error for two finite difference schemes. The relative error for RK2 and Rk4 which remains below 0.00041 and for Runge-Kutta 4th order the relative error is 0.0004.
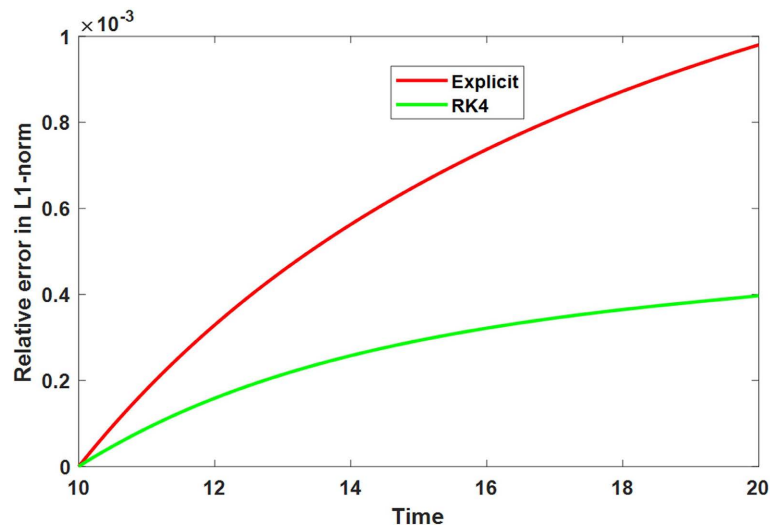
Figure 2. Relative error of 1D ADE for ECDS and RK4 scheme.
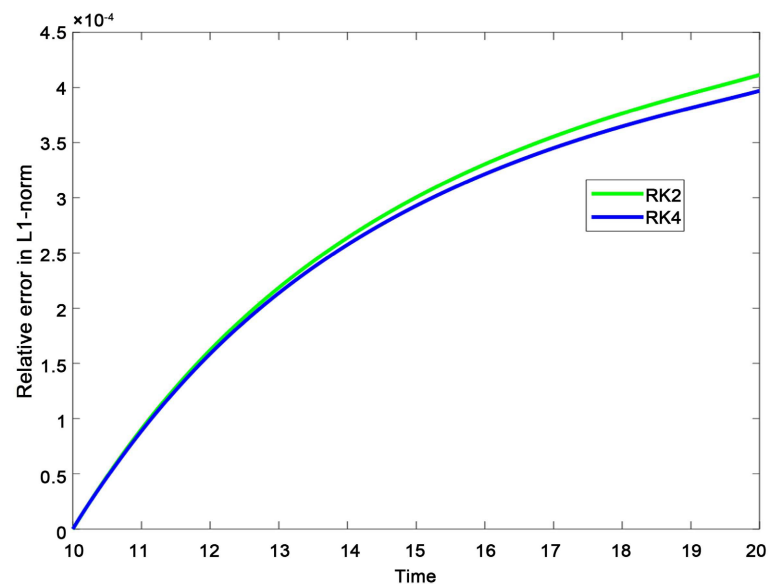

Figure 3. Relative error of 1D ADE for Runge-Kutta 2nd and 4th order scheme.

The following **Figure 4** shows that the comparison of relative error with full discretization and semi-discretization in finite difference schemes. The relative error for ECDS which remains below 0.00001. The relative error for Euler and 4th order the relative error is 0.00009 and 0.000003 respectively.

5.2. Convergence

5.2.1. Convergence of Relative Error for ECDS Method
The following **Figure 5** shows the convergence of relative error by the ECDS. This figure shows a very good rate of convergence of full discretizing method.

5.2.2. Convergence of Relative Error for Euler Method
The following **Figure 6** shows the convergence of relative errors by the Euler

method. This figure shows a very good rate of convergence of semi-discretizing method.
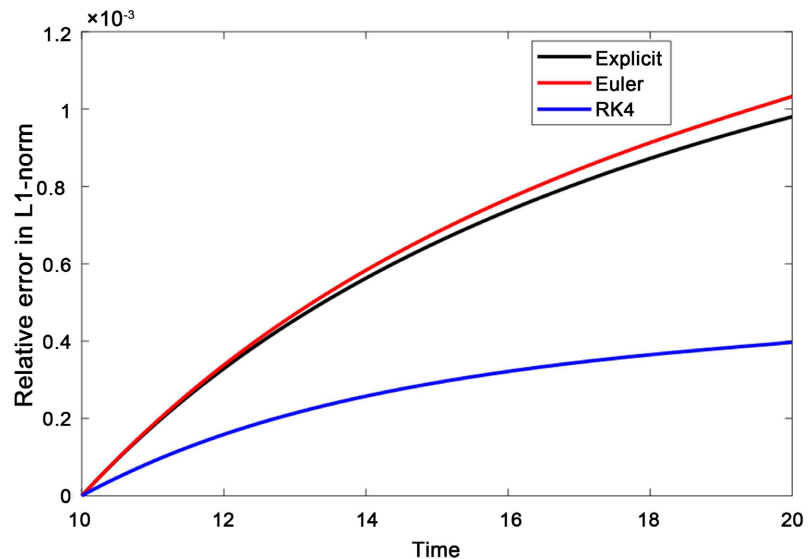

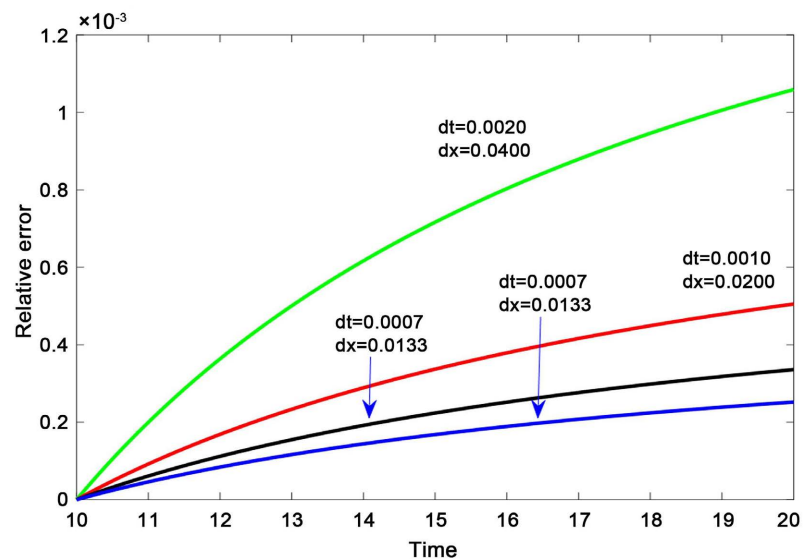**Figure 4.** Relative error of 1D ADE for ECDS, Euler, Runge-Kutta 4th order scheme.

Figure 5. Convergence of relative error ECDS.

5.2.3. Convergence of Relative Error for Runge-Kutta 2nd Order Method

The following **Figure 7** shows the convergence of relative error by the 2nd order Runge-Kutta method. This figure shows a very good rate of convergence of semi-discretizing method.
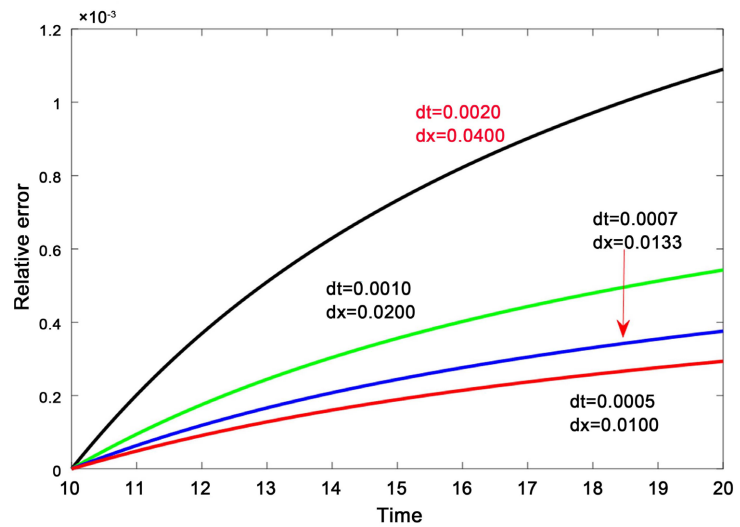
5.2.4. Convergence of Relative Error for Runge-Kutta 4th Order Method

The following **Figure 8**, **Table 2** show the convergence of relative error by the 4th order Runge-Kutta method. This figure shows a very good rate of convergence of semi-discretizing method.

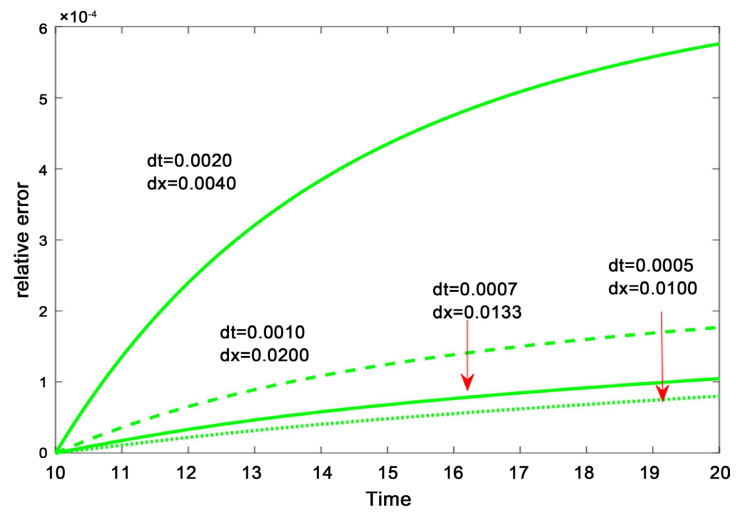**Figure 6.** Convergence of relative error Euler.

Figure 7. Convergence of relative error Runge-Kutta 2nd order method.
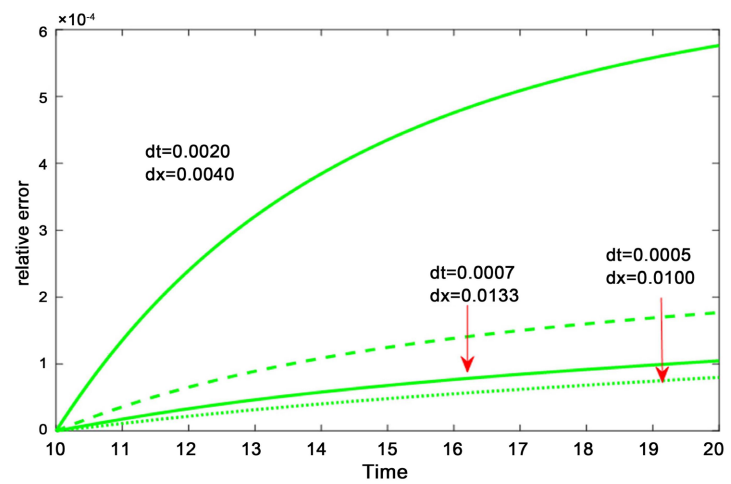

Figure 8. Convergence of relative error Runge-Kutta 4th order method.

Table 2. Error analysis of the three different schemes at different spaces and time steps size.

Δt	Δx	Error for ECDS	Error for Euler	Error for Runge Kutta	
				RK 2nd order	RK 4th order
0.0020	0.0400	1.15×10^{-3}	1.16×10^{-3}	0.058×10^{-3}	0.056×10^{-3}
0.0027	0.0667	2.4×10^{-3}	2.4×10^{-3}	2.5×10^{-3}	1.2×10^{-3}
0.0040	0.0667	3.2×10^{-3}	3.4×10^{-3}	1.6×10^{-3}	1.2×10^{-3}
0.0050	0.0800	4×10^{-3}	4.4×10^{-3}	1.5×10^{-3}	1.4×10^{-3}
0.0080	0.0800	6.4×10^{-3}	6.5×10^{-3}	1.7×10^{-3}	1.6×10^{-3}

5.3. Problem Discussion

Here we can see in **Figure 9** that the semi-discretization method Runge-Kutta 4th order method is the better one. The following **Figure 10** shows concentration distribution by semi-discretization method for different velocities and for different diffusion coefficients. **Figure 11** shows concentration distribution for varying diffusion rate a time $t = 30$ sec.
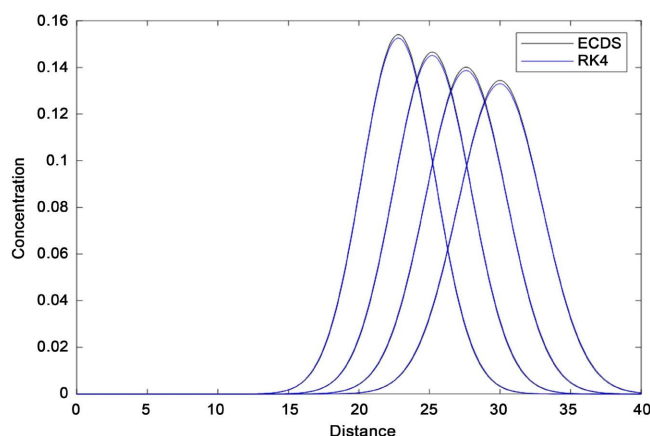
Figure 9. Concentration distribution of full discretization ECDS and semi-discretization Runge-Kutta 4th order method.
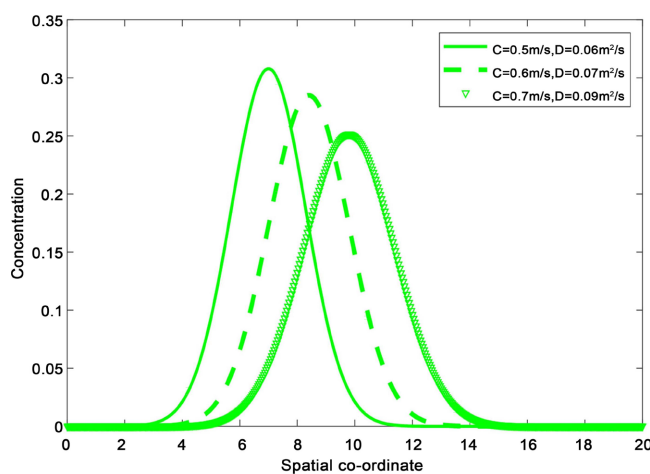
Figure 10. The figure demonstrated varying advection & diffusion rate a time $t = 20$ secs.
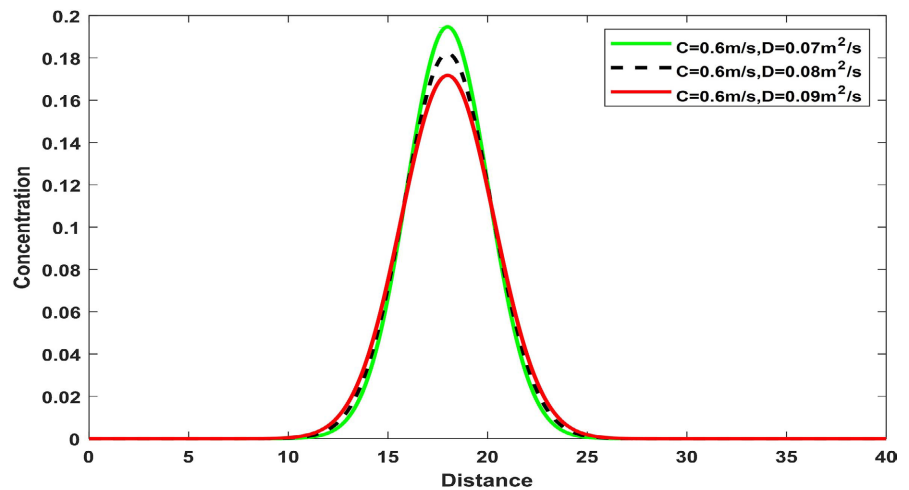
Figure 11. The plot demonstrated varying diffusion rate a time *t* = 30 sec.

The "red curved" in **Figure 12** below indicates a high concentration for 20 seconds, whereas the green curve indicates a concentration for 30 seconds. The "blue" curve indicates that concentration is high for 40 seconds. The figure indicated by "black" shows that concentration is high for 50 seconds, and as time increases, we can observe that the concentration profile decreases.
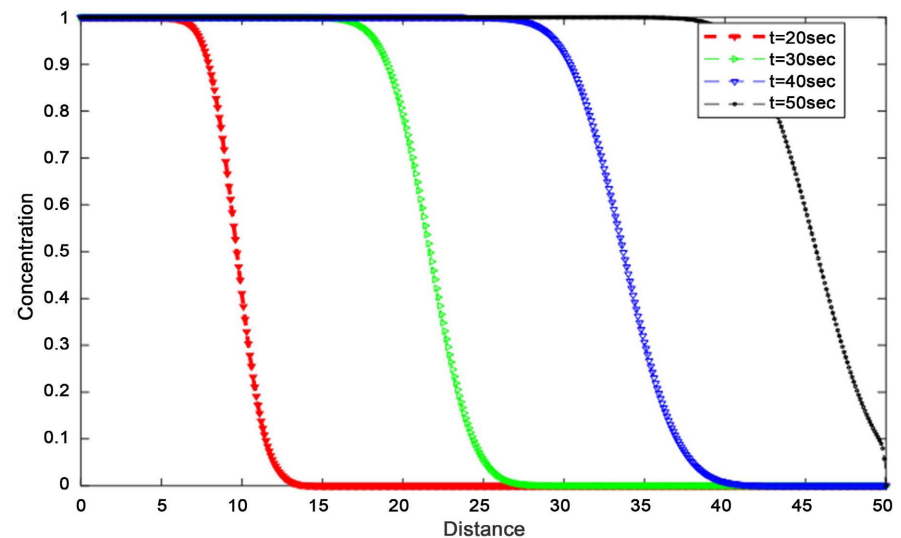

Figure 12. The figure demonstrated different time.

In **Figure 13**, the "black curved" indicates that the concentration is high for 8 meters, while the green curve indicates that the concentration is lower for 16 meters. The red curve indicates 40 m, the blue curve indicates 24 m, and the plot indicated by "deep red" indicates 32 m of high concentration. With regard to time, we may observe that when focus is raised while in a motionless position.

Figure 14, Concentration distributions for varying diffusion rates and velocities over time It is evident that the concentration distribution varies with velocity and diffusion rate. The "Green" curve, for instance, indicates that concentration be-

gins at 0.85 and drops to zero at 2 meters. Comparably, the "Black" curve begins at about 0.82 and drops to zero and 4 m, while the "Red curve" begins at concentration 1 and drops directly to less than 0.8 concentration at 0 m before beginning to decrease to zero at 6 m.
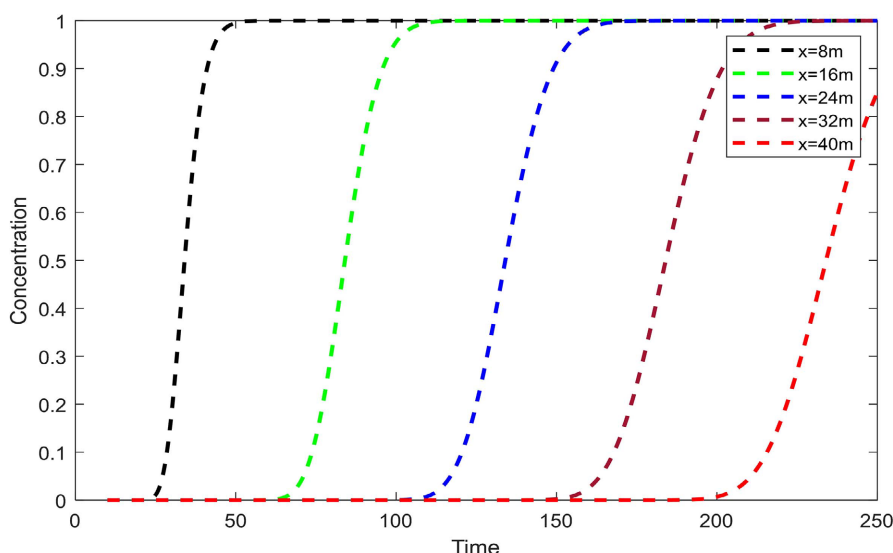

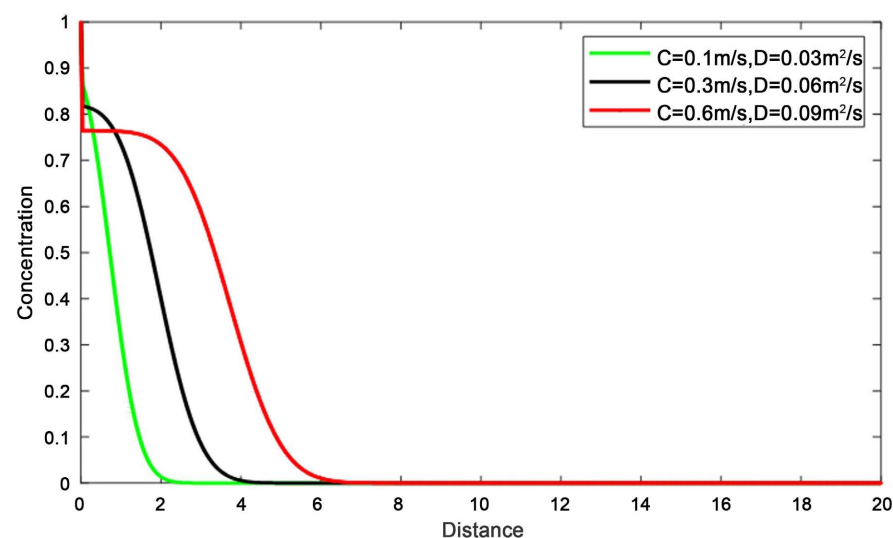**Figure 13.** Concentration distribution for different time for RK4.

Figure 14. The figure demonstrated a different position.

6. Conclusion

In this work, we investigate higher-order approximations of ADE by a semi-discretization method with initial and boundary conditions. Numerical results of ADE are briefly discussed, along with the relative error for both semi-discretization and full discretization methods. Both schemes show a good rate of convergence. We observe that the higher-order semi-discretization method gives better results than ECDS. The 4th order R-K method gives less error than the 2nd order

R-K method.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Ara, K.N.I., Rahaman, M.M. and Alam, M.S. (2021) Numerical Solution of Advection Diffusion Equation Using Semi-Discretization Scheme. *Applied Mathematics*, **12**, 1236-1247. https://doi.org/10.4236/am.2021.1212079

[2] Bahar, E. and Gürarslan, G. (2017) Numerical Solution of Advection-Diffusion Equation Using Operator Splitting Method. *International Journal of Engineering & Applied Sciences*, **9**, 76-88. https://doi.org/10.24107/ijeas.357237

[3] Dehghan, M. (2005) On the Numerical Solution of the One-Dimensional Convection-Diffusion Equation. *Mathematical Problems in Engineering*, **2005**, 61-74. https://doi.org/10.1155/mpe.2005.61

[4] Ahmed, G.S. (2012) A Numerical Algorithm for Solving Advection-Diffusion Equation with Constant and Variable Coefficients. *The Open Numerical Methods Journal*, **4**, 1-7. https://doi.org/10.2174/1876389801204010001

[5] Mojtabi, A. and Deville, M.O. (2015) One-Dimensional Linear Advection-Diffusion Equation: Analytical and Finite Element Solutions. *Computers & Fluids*, **107**, 189-195. https://doi.org/10.1016/j.compfluid.2014.11.006

[6] Tinega, A.K. and Ndede, C.O. (2016) Stability and Consistency Analysis of Central Difference Scheme for Advection Diffusion Partial Differential Equation. *International Journal of Science and Research*, **7**, 1046-1049.

[7] Azad, T. and Andallah, L. (2017) Stability Analysis of Finite Difference Schemes for an Advection Diffusion Equation. *Bangladesh Journal of Scientific Research*, **29**, 143-151. https://doi.org/10.3329/bjsr.v29i2.32331

[8] Appadu, A.R. (2013) Numerical Solution of the 1D Advection-Diffusion Equation Using Standard and Nonstandard Finite Difference Schemes. *Journal of Applied Mathematics*, **2013**, Article ID 734374. https://doi.org/10.1155/2013/734374

[9] Djordjevich, A., Savović, S. and Janićijević, A. (2017) Explicit Finite-Difference Solution of Two-Dimensional Solute Transport with Periodic Flow in Homogenous Porous Media. *Journal of Hydrology and Hydromechanics*, **65**, 426-432. https://doi.org/10.1515/johh-2017-0040

[10] Iliev, O.P., *et al.* (2013) Numerical Solution of Partial Differential Equations: Theory, Algorithms, and Their Applications. Springer Science & Business Media.

[11] Rahaman, M.M. and Andallah, L.S. (2014) Simulation of Water Pollution by Finite Difference Method. *International Journal of Research in Information Technology*, **2**, 17-24.

[12] Mathews, J.H. and Fink, K.K. (2004) Numerical Methods Using Matlab. 4th Edition, Prentice Hall.