e-ISSN 2757-5195

Çanakkale Onsekiz Mart University
Journal of Advanced Research in Natural and Applied Sciences
Open Access

doi.org/10.28979/jarnas.985946 2022, Cilt 8, Sayı 1, Sayfa: 45-53 dergipark.org.tr/tr/pub/jarnas

A Stable and High Accurate Numerical Simulator for Convection-Diffusion Equation

Osman Ünal[1],*

[1]Department of Mechanical Engineering, Graduate Education Institute, Sakarya University of Applied Sciences, Sakarya, Turkey

Article History

Received: 22.08.2021
Accepted: 22.11.2021
Published: 10.03.2022

Research Article

Abstract – This study presents a simulator to obtain numerical solution of convection-diffusion equation. It includes explicit, fully implicit and semi-implicit time discretization techniques. In addition to time discretization techniques, this simulator contains several space discretization methods such as first-order upstream and UMIST (University of Manchester Institute of Science and Technology) techniques. It is observed that the use of UMIST or semi-implicit techniques in the different numerical simulator decreases numerical errors. However, the combination of UMIST and semi-implicit methods is not available in literature. The proposed numerical simulator is suitable for easily using the different combinations of time and space discretization methods. Second objective of this study is to present a novel combination that includes both semi-implicit time discretization technique and UMIST space discretization method to minimize numerical errors namely numerical dispersion and unphysical oscillation. Alt-hough UMIST method suppresses unphysical oscillation, it causes a small and undesired oscillation at flood front for very large Courant number. Thirdly, this study proposes a minor modification on the UMIST method to elimi-nate this unphysical oscillation. The novel combination of modified UMIST method and semi-implicit technique decreases numerical dispersion significantly and suppress unphysical oscillation effectively. Moreover, the numeri-cal result of proposed model is very close to analytical solution.

Keywords – *Finite difference method, high-resolution scheme, explicit method, fully implicit method, semi-implicit method*

1. Introduction

A great number of physical phenomena and many real processes are effectively modeled by partial differential equations. One of the most widely used types of partial differential equations for describing these physical processes is the unsteady convection-diffusion equation. The convection–diffusion equation is used in simulating transport process (Peng et. al., 2013), flow in oil and gas reservoirs (Kurganov & Tadmor, 2000), non-isothermal injection techniques, chemical displacement, miscible displacement and immiscible displacement (Kamalyar et. al., 2014), pollutant dispersion in a river estuary (Morton, 2019), etc. Following Equation 1.1 shows one-dimensional form of convection-diffusion equation.

$$D\frac{\partial^2 U}{\partial x^2} - v\frac{\partial U}{\partial x} = \frac{\partial U}{\partial t} \tag{1.1}$$

There are two basic way to solve convection-diffusion equation. While one of them is analytical solution, the other is numerical techniques. Nowadays, there is no any analytical solution of multi-dimensional and complex physical systems. Therefore, it is necessary to use numerical techniques for these complicated problems. The finite difference method is one of the most commonly used numerical techniques for solution of convection-diffusion equation. This method has been used for many years due to its simplicity, prac-

[1] osman92unal@gmail.com
*Sorumlu Yazar / Corresponding Author

ticality and effectiveness. The finite difference method is applied for time discretization or space discretization. There are a number of time and space discretization techniques in literature. However, some of them stand out such as explicit, fully implicit and semi-implicit techniques for time discretization and first-order upstream and UMIST (University of Manchester Institute of Science and Technology) methods for space discretization. Although the explicit method is efficient for some simple conduction problems, it needs stability criteria. Otherwise, it is unstable especially for high Courant number and large time step size. When using the explicit method, the time step size is chosen with care to get well-posed numerical solution. This requirement sets a serious constraint for the explicit method because choosing small time step size causes the simulation time to become quite long. Even though the implicit method is stable for large time step size and high Courant number, it leads to numerical dispersion like the explicit method. On the other hand, second-order accurate semi-implicit method significantly reduces numerical dispersion. One another method to decrease numerical dispersion is to use higher-order space discretization techniques for example UMIST method. However, the combination of semi-implicit and UMIST methods is not available in literature. The objective of this study is to combine second-order accurate semi-implicit method and UMIST technique in order to minimize numerical errors.

The analytical solutions of some one-dimensional systems exist in literature. They are used to validate the reliability of numerical methods. After verification of numerical techniques with analytical solution for simple and one-dimensional problem, proposed numerical techniques may be used for sophisticated and multi-dimensional problems. All numerical methods used in this study are checked with the following analytical solutions.

$$U(x,t) = \frac{1}{2} erfc\left(\frac{x - v\,t}{2\sqrt{D\,t}}\right) + \frac{1}{2} exp\left(\frac{v\,x}{D}\right) erfc\left(\frac{x + v\,t}{2\sqrt{D\,t}}\right) \quad (1.2)$$

If physical dispersion coefficient is zero, equation 1.2 (Peaceman, 2000) is undefined. The convection-diffusion equation transforms to the transport equation for that cases (cancelling of first term in Equation 1.1). The transport equation is used for convection-dominated fluid flow. The analytical solution of transport or advection equation is described in following Equation 1.3 (Sarra, 2002).

$$U(x,t) = G(x_0) = G(x - v\,t) \quad (1.3)$$

In Equation 1.3, $G(x_0)$ refers to initial condition. The exact solution of transport equation depends mainly on initial condition and it may be defined as propagation of all points on the initial condition with the same speed that is constant velocity (Sarra, 2002).

2. Materials and Methods

The convection-diffusion equation is solved using time discretization and space discretization techniques. While accumulation term (last term in Equation 1.1) is approximated by time discretization methods, convection term (second term in Equation 1.1) and diffusion term (first term in Equation 1.1) are estimated by space discretization techniques. Although there are a number of space and time discretization methods, some of them lead to unacceptable numerical errors. One of the aims of this study is to select best combination of space and time discretization techniques in order to minimize numerical dispersion and suppress unphysical oscillation.

2.1. Time Discretization

Time discretization is applied to transient or unsteady problems and it is also called as temporal discretization. Temporal discretization is the integration of diffusion and convection terms over each time steps. Explicit, implicit and semi-implicit methods are the most widely used approaches to calculate the integral over a time step. The explicit and the fully implicit methods are first-order time discretization

methods and they leads to significant numerical errors namely numerical dispersion. The explicit method gives stable results with numerical dispersion for small Courant number. However, it is not stable for large Courant number. The coefficient of interested grid block central value for previous time step in numerical solution of explicit method must be positive (Versteeg & Malalasekera, 2007) because next time step values are calculated by adding to previous time step values. Following stability criteria must be provided for numerical solution of convection-dominated fluid flow equation using explicit time scheme and first-order upstream space discretization method.

$$U_i^{n+1} = U_i^n - \Delta t\left(v\frac{U_i^n - U_{i-1}^n}{\Delta x}\right) \tag{2.1}$$

The coefficient of at right-hand side in Equation 2.1 must be positive. The following inequality (Equation 2.2) is obtained to provide positive coefficient of.

$$\Delta t < \frac{\Delta x}{v} \tag{2.2}$$

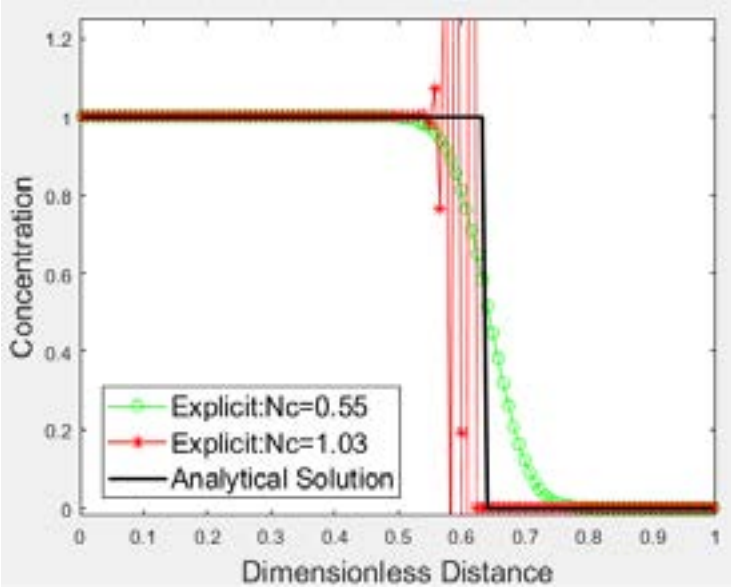

Figure 1. Explicit scheme for first-order upstream method (dx=0.1).

Figure 1 shows numerical solutions of advection equation (physical dispersion is zero, D=0) using the explicit time discretization method and first-order upstream space discretization technique for 0.55 and 1.03 Courant number (for both cases space interval are same, dx=0.1 and time step sizes are 0.055 and 0.103 for green line with circle and red line with star, respectively). According to Figure 1, the explicit method for small Courant number (Nc=0.55) is stable (there is no oscillation) with large numerical dispersion (green line with circle). However, the explicit method is unstable (it has large unphysical oscillation) when Courant number is greater than 1 (red line with star) due to the large time step size (dt=0.103). This numerical result is far away from analytical solution and it's not acceptable to model any physical system. This is the main disadvantage of explicit time discretization methods that are not convenient for large time step sizes or large Courant number. The implicit time discretization methods should be used to suppress unphysical oscillation (red line) for large Courant number and higher-order space discretization method should be used to diminish numerical dispersion (green line).

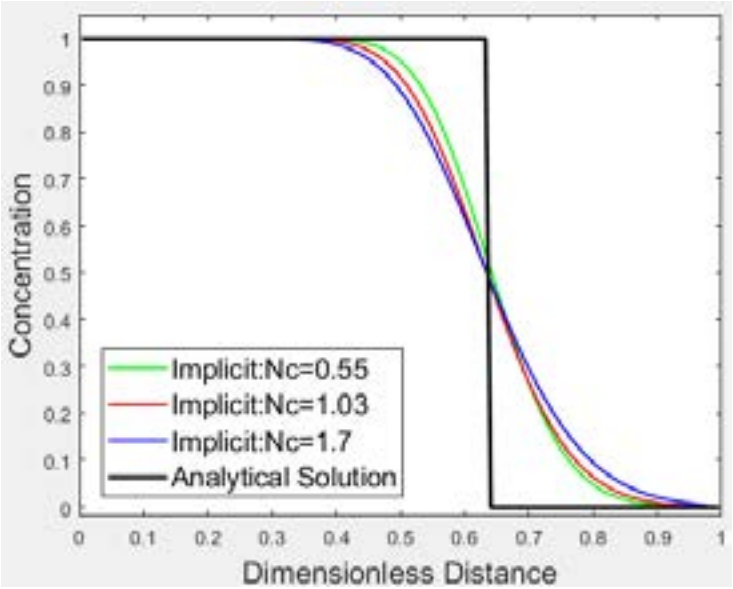

Figure 2. Implicit scheme for first-order upstream method (dx=0.1).

Figure 2 indicates numerical results of fully-implicit scheme and first-order upstream space discretization technique for 0.55, 1.03 and 1.7 Courant number (for both cases space interval are same, dx=0.1 and time step sizes are 0.055, 0.103 and 0.17 for green, red and blue lines, respectively). According to numerical results of Figure 2, increasing of the time step sizes leads to numerical dispersion without any oscillation. The main advantage of fully-implicit method over the explicit scheme is stability. The implicit method is unconditional stable and it can be used even for large time step size and high Courant number. The small time step interval must be selected for explicit method in order to provide stability requirement and obtain meaningful numerical results. The small time step size causes the numerical solution to take quite long. Therefore, the fully-implicit method is used to reduce total simulation time. In spite of the stability advantage of fully-implicit technique, the numerical results in Figure 2 are far from analytical solution due to the numerical dispersion. When time step size or Courant number increases, the numerical dispersion of implicit method becomes greater. Hence, higher-order methods are used to minimize numerical dispersion like semi-implicit time discretization method.

The semi-implicit method called also as Crank-Nicolson technique (Crank & Nicolson, 1947) is second-order accurate approximation and it reduces numerical errors effectively. Following equations show full discretization of convection-diffusion equation using explicit, fully-implicit and Crank-Nicolson method respectively. In Equation 2.3, superscripts n and n+1 refer to current time step and next time step. Subscripts i, i-1/2, i+1/2 are index for nodes, left face value and right face value, respectively. The face values are used to numerically calculate the first derivative.

$$D\frac{U_{i+1}^{n}-2U_{i}^{n}+U_{i-1}^{n}}{\Delta x^{2}}-v\frac{U_{i+\frac{1}{2}}^{n}-U_{i-\frac{1}{2}}^{n}}{\Delta x}=\frac{U_{i}^{n+1}-U_{i}^{n}}{\Delta t} \quad (2.3)$$

$$D\frac{U_{i+1}^{n+1}-2U_{i}^{n+1}+U_{i-1}^{n+1}}{\Delta x^{2}}-v\frac{U_{i+\frac{1}{2}}^{n+1}-U_{i-\frac{1}{2}}^{n+1}}{\Delta x}=\frac{U_{i}^{n+1}-U_{i}^{n}}{\Delta t} \quad (2.4)$$

$$0.5\left(D\frac{U_{i+1}^{n}-2U_{i}^{n}+U_{i-1}^{n}}{\Delta x^{2}}-v\frac{U_{i+\frac{1}{2}}^{n}-U_{i-\frac{1}{2}}^{n}}{\Delta x}\right)+0.5\left(D\frac{U_{i+1}^{n+1}-2U_{i}^{n+1}+U_{i-1}^{n+1}}{\Delta x^{2}}-v\frac{U_{i+\frac{1}{2}}^{n+1}-U_{i-\frac{1}{2}}^{n+1}}{\Delta x}\right)=\frac{U_{i}^{n+1}-U_{i}^{n}}{\Delta t} \quad (2.5)$$

The left-hand side terms in Equations 2.3, 2.4 and 2.5 can be expressed as a function of time namely F(t). The Figure 3 shows numerical integral of Equations 2.3, 2.4 and 2.5 respectively.

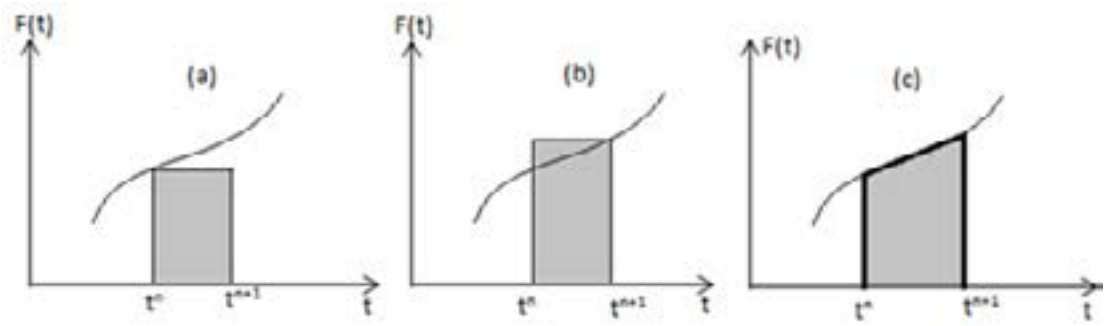

Figure 3. Integral approximations for explicit method (a), implicit method (b) and semi-implicit method (c).

In Figure 3, it's assumed that black curved line is a continuous function. The objective in numerical integration is to determine the area under the continuous function between t^n and t^{n+1} time intervals. According to Figure 3, the integral approximations of explicit and fully-implicit methods have significant numerical errors at lower side and upper side of the function. Therefore, these techniques lead to numerical dispersion. However, numerical integration of semi-implicit method is closer to analytical integration. Hence, it reduces numerical dispersion seriously (see the pink line in Figure 4). The numerical results in Figure 4 are designed for large Courant number (Nc=1.7) in order to indicate the effect of large time step sizes. It's assumed that space (dx=0.1) and time (dt=0.17) intervals are same for both cases to compare different types of time discretization. In Figure 4, blue line is first-order time discretization method and it causes large numerical dispersion. It has large numerical errors due to ineffective numerical integration technique. However, the pink line is second-order method and it decreases numerical errors significantly. That's why it's closer to analytical solution compared to first-order technique.

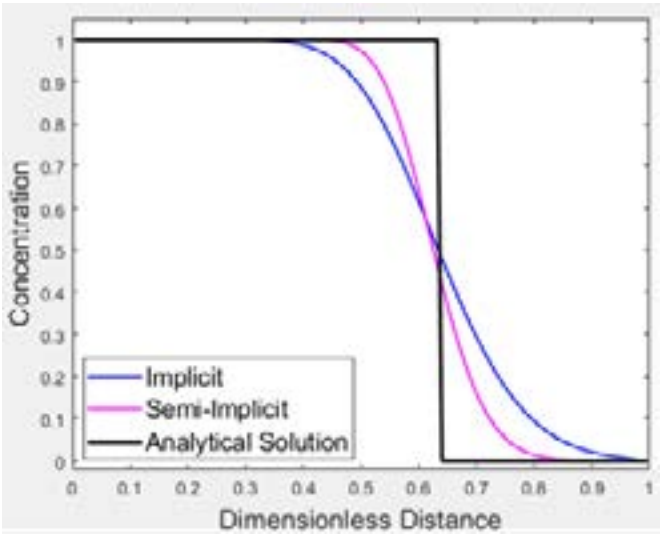

Figure 4. Implicit and semi-implicit scheme for first-order space method (Nc=1.7, dx=0.1 and dt=0.17).

2.2. Space Discretization

The space discretization techniques are applied to convection and diffusion terms. The diffusion term is discretized using second-order central difference method and only grid block or node central values are required to calculate numerical result of diffusion term. On the other hand, space discretization of convection term depends on grid block face values. However, there is no any exact knowledge about face values in numerical calculation. It may be predicted using some techniques. Therefore, the numerical errors arise from these approximations. Equations 2.6 and 2.7 show first-order upstream method (Ertekin et. al., 2001) and TVD (Total Variation Diminishing) technique (Harten, 1984, Sweby, 1984) respectively.

$$U^n_{i+\frac{1}{2}} \approx U^n_i \tag{2.6}$$

$$U^n_{i+\frac{1}{2}} \approx U^n_i + \frac{1}{2}\varphi(r)[U^n_i - U^n_{i-1}] \tag{2.7}$$

The second term at right-hand side in Equation 2.7 is anti-diffusive term. It decreases numerical dispersion. The flux limiter for UMIST (University of Manchester Institute of Science and Technology) technique is described by following Equation 2.8 that is an extension of the Quick method (Leonard, 1979).

$$\varphi(r) = min\left(2, 2r, \frac{3r+1}{4}, \frac{r+3}{4}\right) \tag{2.8}$$

In Equation 2.8, r is gradient ratio and it is defined as following Equation 2.9 (Wolcott et. al., 1996).

$$r = \frac{U^n_{i+1} - U^n_i}{U^n_i - U^n_{i-1}} \tag{2.9}$$

The UMIST technique reduces numerical dispersion effectively because it's higher-order space discretization method. The first-order upstream space discretization method and the UMIST space discretization technique with the implicit time discretization method has been used in Figure 5. It is assumed that space interval is 0.15 (dx=0.15), time interval is 0.1 (dt=0.1) and Courant number is 0.67 (Nc=0.67). According to Figure 5, the UMIST method gives sharper flood front than first-order upstream technique and it is closer to analytical solution. The use of higher-order technique (UMIST) decreases the space discretization errors. However, the numerical solution using UMIST method (third-order in space discretization) and implicit scheme (first-order in time discretization) still has numerical dispersion due to the time discretization errors. The combination of the semi-implicit method (second-order in time discretization) and the UMIST technique may bring the numerical solution closer to the analytical solution.

3. Results and Discussion

In this study, the importance of the use of higher-order finite difference technique is investigated. Nowadays, the explicit scheme (Lundgren & Mattsson, 2020) is still used to solve partial differential equation. However, it is unstable for large Courant number and large time step sizes (see red line in Figure 1). The small time step sizes must be used for the explicit scheme to obtain stable numerical results (see

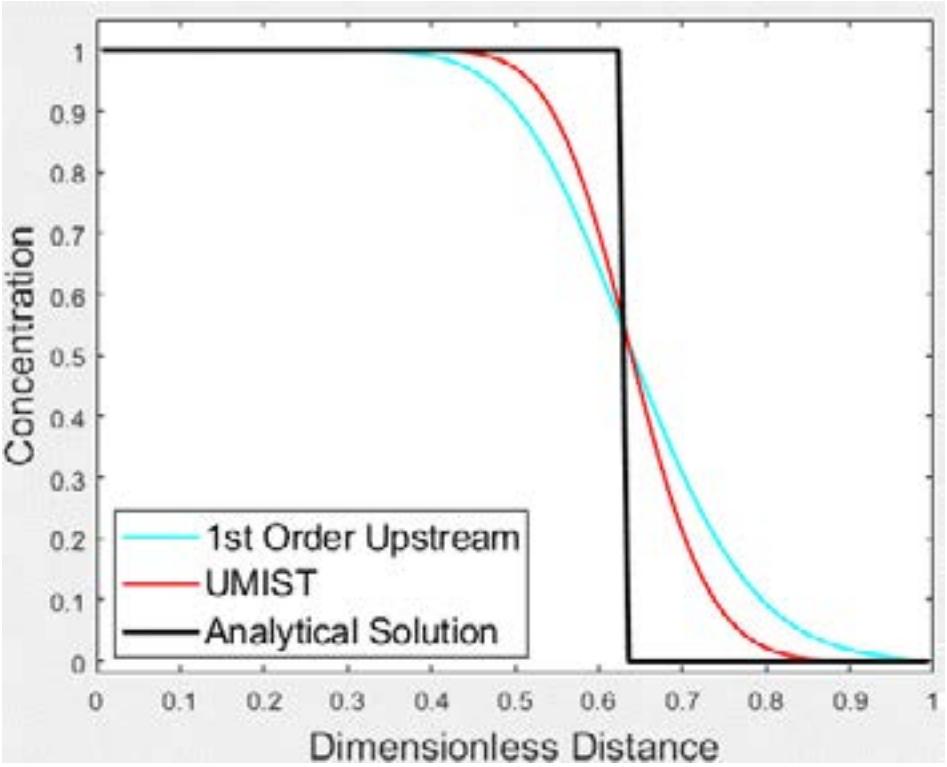

Figure 5. First-order upstream and UMIST methods for implicit scheme (Nc=0.67, dx=0.15 and dt=0.1).

green line in Figure 1). The use of small time step sizes causes quite long simulation time. Therefore, the higher-order techniques in space and time discretization for large time step sizes and large space intervals must be used to decrease total simulation time and obtain more accurate numerical results. It is observed that the application of UMIST space discretization method (third-order in space discretization) to the semi-implicit time discretization technique reduces numerical errors because semi-implicit method is second-order scheme. Moreover, proposed model is very stable and it is very close to analytical solution (see red line in Figure 6). This proposed model works properly for moderate Courant number (lesser than 1) and it diminish numerical dispersion impressively. Nevertheless, the novel combination of the UMIST method with the semi-implicit technique causes to a small unphysical oscillation at the flood front (see red line in Figure 6). In this study, it was observed that reducing the upper limit of the UMIST flux limiter function suppresses these undesired oscillations. In Figure 6, it is assumed that space interval is 0.1 (dx=0.1), time interval is 0.13 (dt=0.13) and Courant number is 1.3 (Nc=1.3). Figure 6 shows UMIST method with 2.0 and 1.3 upper limits for semi-implicit time discretization method. The proposed model (blue circles) minimizes numerical dispersion without any unphysical oscillation even for large Courant number.

It is important note that please use Supplementary material 1 and Supplementary material 2 or Supplementary material 3 in order to obtain all figures in this study.

4. Conclusion

This study has presented a numerical simulator for convection-diffusion equation. This simulator includes explicit, fully-implicit and semi-implicit time discretization techniques. It has been observed that explicit method requires the stability criteria and it is unstable for large Courant number. The time step size must be reduced to provide stability criteria for explicit scheme. However, decreasing time step size causes long simulation time. The fully-implicit method has been used because it is stable even for large Courant number. Despite the stability advantage of fully-implicit method, it is first-order accurate scheme like explicit method. Thus, the implicit method leads to numerical dispersion for large time step size. The second-order accurate semi-implicit or Crank-Nicolson time discretization method has been selected to minimize numerical dispersion. The proposed numerical simulator has also several space discretization

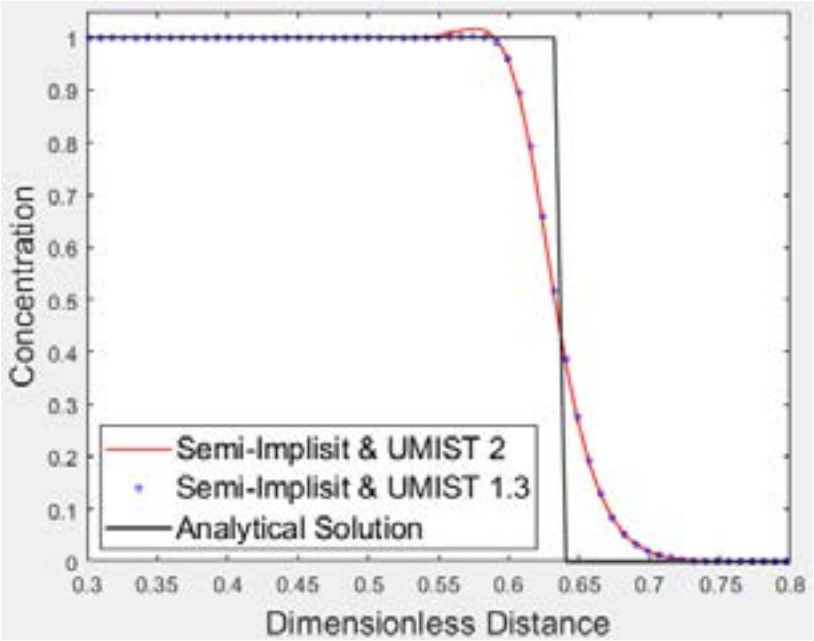

Figure 6. Application of UMIST methods to semi-implicit scheme (Nc=1.3, dx=0.1 and dt=0.13).

methods. The first-order upstream method is a scheme without any unphysical oscillation. Nonetheless, it is first-order method and it has large numerical dispersion. Although the higher-order UMIST method decreases numerical dispersion successfully, it has been noticed that the combination of UMIST method with semi-implicit technique causes a small unphysical oscillation at flood front. In this study, it is proposed to decrease upper limit of UMIST flux limiter function in order to suppress this undesired oscillation. The proposed model has minimized numerical dispersion without any unphysical oscillation. In this study, it is observed that this novel combination of modified UMIST method and semi-implicit technique minimizes the most important numerical errors namely numerical dispersion and unphysical oscillation and it is very close to analytical solution. Moreover, the novel combination method can be easily applied to all type of convection-diffusion problem especially for most recent physical problems such as the simulating miscible displacement, immiscible displacement, chemical displacement and non-isothermal injection, the contaminant and sediment movement in the rivers, lakes and groundwater aquifers, simulating transport and flow in oil and gas reservoirs, particularly in two-phase flow. Finally, all Matlab codes related to the numerical simulator have been added to Supplementary Materials in order to facilitate other researchers' work.

Author Contributions

Author O.Ü.: Performed all the study and has currently written the paper.

Conflicts of Interest

The author declares no conflict of interest.

Symbols

U = concentration or interested value

D = physical dispersion

Nc = courant number

dt = timestep

dx = space interval

v = velocity

Subscripts

i = index for nodes in the x direction

i-1/2 = index for left face values

i+1/2= index for right face values

Superscripts

n = old timestep

n+1 = current timestep

References

Crank, J., & Nicolson, P. (1947, January). A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. In Mathematical Proceedings of the Cambridge Philosophical Society (Vol. 43, No. 1, pp. 50-67). Cambridge University Press. DOI: https://doi.org/10.1017/S0305004100023197

Ertekin, T., Abou-Kassem, J. H., & King, G. R. (2001). *Basic applied reservoir simulation* (Vol. 7). Richardson, TX: Society of Petroleum Engineers. Retrieved from: https://store.spe.org/Basic-Applied-Reservoir-Simulation--P12.aspx

Harten, A. (1984). On a class of high resolution total-variation-stable finite-difference schemes. *SIAM Journal on Numerical Analysis, 21*(1), 1-23. DOI: https://doi.org/10.1137/0721001

Kamalyar, K., Kharrat, R., & Nikbakht, M. (2014). Numerical Aspects of the Convection–Dispersion Equation. *Petroleum science and technology, 32*(14), 1729-1762. DOI: https://doi.org/10.1080/10916466.2010.490802

Kurganov, A., & Tadmor, E. (2000). New high-resolution central schemes for nonlinear conservation laws and convection–diffusion equations. *Journal of Computational Physics, 160*(1), 241-282. DOI: https://doi.org/10.1006/jcph.2000.6459

Leonard, B. P. (1979). A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computer methods in applied mechanics and engineering, 19*(1), 59-98. DOI: https://doi.org/10.1016/0045-7825(79)90034-3

Lundgren, L., & Mattsson, K. (2020). An efficient finite difference method for the shallow water equations. Journal of Computational Physics, 422, 109784. DOI: https://doi.org/10.1016/j.jcp.2020.109784

Morton, K. W. (2019). Numerical solution of convection-diffusion problems. CRC Press. DOI: https://doi.org/10.1201/9780203711194

Peaceman, D. W. (2000). *Fundamentals of numerical reservoir simulation.* Elsevier. DOI: https://doi.org/10.1137/1021068

Peng, Y., Liu, C., & Shi, L. (2013, August). Soution of Convection-Diffusion Equations. *In International Conference on Information Computing and Applications* (pp. 546-555). Springer, Berlin, Heidelberg. DOI: https://doi.org/10.1007/978-3-642-53703-5_56

Sarra, S. A. (2003). The method of characteristics with applications to conservation laws. *Journal of Online mathematics and its Applications, 3*, 1-16. Retrieved from: http://www.scottsarra.org/math/papers/characteristicsSarra.pdf

Sweby, P. K. (1984). High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM journal on numerical analysis, 21*(5), 995-1011. DOI: https://doi.org/10.1137/0721062

Versteeg, H. K., & Malalasekera, W. (2007). An introduction to computational fluid dynamics: the finite volume method. Pearson education. Retrieved from: https://www.pearson.com/store/p/an-introduction-to-computational-fluid-dynamics-the-finite-volume-method/P100001392465/9780131274983

Wolcott, D. S., Kazemi, H., & Dean, R. H. (1996, October). A practical method for minimizing the grid orientation effect in reservoir simulation. In *SPE annual technical conference and exhibition.* OnePetro. DOI: https://doi.org/10.2118/36723-MS

Supplementary Materials

Supplementary material 1. Sub-function to run the numerical simulator.

```
function y=TVD(x)
% Select Space Discretization Method (SDM)
%For First Order Upstream Method->SDO=1
%For UMIST->SDO=30
SDO=30;
if SDO==1%First Order Upstream
y=0;
elseif SDO==30%UMIST 1.3
y=max(0,min([2,2*x,(3*x+1)/4,(x+3)/4]));
end
end
```

Supplementary material 2. The numerical simulator.

Note: In order to run Matlab codes in *Supplementary material 2*, it's required to get ''TVD.m'' Matlab file. It can be obtained using *Supplementary material 1* The name of the Matlab file must be ''TVD'' without quotes. Secondly, designed TVD.m Matlab file and the numerical simulator mfile (in *Supplementary material 2*) must be at the same path. Thirdly, if you don't want to use *Supplementary material 1* and *Supplementary material 2* in order to run Matlab file, you can use *Supplementary material 3*.

```
tic; clc; clearvars;
%%INPUT DATA
t=7.7;%Total simulation time
dt=0.13;%Time interval
I=120+1;%Number of points at i-direction
dx=0.1;%Space interval
X=dx/2:dx:I*dx-dx/2;%Distance
W=1;%W=1(upstream) W=0.5(mid-point) W=0(downstream)
Q=0.5;%Q=1(implicit) Q=0.5(CN) Q=0(explicit)
vf=1;%Velocity*(df/du)
L=vf*dt/dx;%Courant Number
D=0;%Physical dispersion
K=D*dt/2/dx^2;%Diffusion term coefficient
U00=0.5; Ui0=0;%Initial Condition
U0n=1; UIn=0;%Boundary Condition
Ucc=0.00001;%Convergence Criteria
dU=10^-8;%NRI interval
%%OUTPUT DATA
Up(1)=U00;
Up(2:I)=Ui0;
Un(1)=U0n; Un(2:I-1)=NaN; Un(I)=UIn;
for n=1:t/dt%Time iteration
if Q==0%Explicit solution
for i=2:I-1
if i==2
rf=Up(i); lf=Up(i-1);
else
rf=Up(i)+0.5*max(0,TVD((Up(i+1)-Up(i))/(Up(i)-Up(i-1))))*(Up(i)-Up(i-1));
lf=Up(i-1)+0.5*max(0,TVD((Up(i)-Up(i-1))/(Up(i-1)-Up(i-2))))*(Up(i-1)-Up(i-2));
end
Un(i)=dt*(D*(Up(i+1)-2*Up(i)+Up(i-1))/dx^2-vf*(rf-lf)/dx)+Up(i);
end
Up=Un;%End of explicit solution
else%Implicit solution
Uv=Up;
it=1;
```

```
vUv=Ucc;
while vUv>=Ucc%NR iteration
iU=Uv+dU;
%Determination of A and B Matrices
%A matrix
f(1:I)=NaN;
for ii=1:I
if ii==1
f(ii)=K*(Uv(ii+1)-2*U0n+U0n+Up(ii+1)-2*Up(ii)+Up(ii))-L*(Q*(U0n-U0n)+(1-Q)*(Up(ii)-
Up(ii)))-U0n+Up(ii);
elseif ii==2
f(ii)=K*(Uv(ii+1)-2*Uv(ii)+U0n+Up(ii+1)-2*Up(ii)+Up(ii-1))-L*(Q*(Uv(ii)-U0n)+(1-
Q)*(Up(ii)-Up(ii-1)))-Uv(ii)+Up(ii);
elseif ii==3
rvi=(Uv(ii+1)-Uv(ii))/(Uv(ii)-Uv(ii-1)); lfvi=max(0,TVD(rvi));
ADTvi=0.5*lfvi*(Uv(ii)-Uv(ii-1)); rvi_1=(Uv(ii)-Uv(ii-1))/(Uv(ii-1)-U0n);
lfvi_1=max(0,TVD(rvi_1)); ADTvi_1=0.5*lfvi_1*(Uv(ii-1)-U0n);
rpi=(Up(ii+1)-Up(ii))/(Up(ii)-Up(ii-1)); lfpi=max(0,TVD(rpi));
ADTpi=0.5*lfpi*(Up(ii)-Up(ii-1)); rpi_1=(Up(ii)-Up(ii-1))/(Up(ii-1)-Up(ii-2));
lfpi_1=max(0,TVD(rpi_1)); ADTpi_1=0.5*lfpi_1*(Up(ii-1)-Up(ii-2));
f(ii)=K*(Uv(ii+1)-2*Uv(ii)+Uv(ii-1)+Up(ii+1)-2*Up(ii)+Up(ii-1))-L*(Q*(Uv(ii)+ADTvi-
Uv(ii-1)-ADTvi_1)+(1-Q)*(Up(ii)+ADTpi-Up(ii-1)-ADTpi_1))-Uv(ii)+Up(ii);
elseif 3<ii && ii<I-1
rvi=(Uv(ii+1)-Uv(ii))/(Uv(ii)-Uv(ii-1)); lfvi=max(0,TVD(rvi));
ADTvi=0.5*lfvi*(Uv(ii)-Uv(ii-1)); rvi_1=(Uv(ii)-Uv(ii-1))/(Uv(ii-1)-Uv(ii-2));
lfvi_1=max(0,TVD(rvi_1)); ADTvi_1=0.5*lfvi_1*(Uv(ii-1)-Uv(ii-2));
rpi=(Up(ii+1)-Up(ii))/(Up(ii)-Up(ii-1)); lfpi=max(0,TVD(rpi));
ADTpi=0.5*lfpi*(Up(ii)-Up(ii-1)); rpi_1=(Up(ii)-Up(ii-1))/(Up(ii-1)-Up(ii-2));
lfpi_1=max(0,TVD(rpi_1)); ADTpi_1=0.5*lfpi_1*(Up(ii-1)-Up(ii-2));
f(ii)=K*(Uv(ii+1)-2*Uv(ii)+Uv(ii-1)+Up(ii+1)-2*Up(ii)+Up(ii-1))-L*(Q*(Uv(ii)+ADTvi-
Uv(ii-1)-ADTvi_1)+(1-Q)*(Up(ii)+ADTpi-Up(ii-1)-ADTpi_1))-Uv(ii)+Up(ii);
elseif ii==I-1
rvi=(UIn-Uv(ii))/(Uv(ii)-Uv(ii-1)); lfvi=max(0,TVD(rvi));
ADTvi=0.5*lfvi*(Uv(ii)-Uv(ii-1)); rvi_1=(Uv(ii)-Uv(ii-1))/(Uv(ii-1)-Uv(ii-2));
lfvi_1=max(0,TVD(rvi_1)); ADTvi_1=0.5*lfvi_1*(Uv(ii-1)-Uv(ii-2));
rpi=(Up(ii+1)-Up(ii))/(Up(ii)-Up(ii-1)); lfpi=max(0,TVD(rpi));
ADTpi=0.5*lfpi*(Up(ii)-Up(ii-1)); rpi_1=(Up(ii)-Up(ii-1))/(Up(ii-1)-Up(ii-2));
lfpi_1=max(0,TVD(rpi_1)); ADTpi_1=0.5*lfpi_1*(Up(ii-1)-Up(ii-2));
f(ii)=K*(UIn-2*Uv(ii)+Uv(ii-1)+Up(ii+1)-2*Up(ii)+Up(ii-1))-L*(Q*(Uv(ii)+ADTvi-Uv(ii-
1)-ADTvi_1)+(1-Q)*(Up(ii)+ADTpi-Up(ii-1)-ADTpi_1))-Uv(ii)+Up(ii);
elseif ii==I
f(ii)=K*(UIn-2*UIn+Uv(ii-1)+Up(ii)-2*Up(ii)+Up(ii-1))-L*(Q*(UIn-Uv(ii-1))+(1-
Q)*(Up(ii)-Up(ii-1)))-UIn+Up(ii);
end
end
A=f';
%B matrix
B=zeros(I,I);
fiU(1:I)=NaN;
for ii=1:I
if ii==1
%for east
fiU(ii)=K*(iU(ii+1)-2*U0n+U0n+Up(ii+1)-2*Up(ii)+Up(ii))-L*(Q*(U0n-U0n)+(1-Q)*(Up(ii)-
Up(ii)))-U0n+Up(ii);
B(ii,ii+1)=(fiU(ii)-f(ii))/dU;
elseif ii==2
%for center
fiU(ii)=K*(Uv(ii+1)-2*iU(ii)+U0n+Up(ii+1)-2*Up(ii)+Up(ii-1))-L*(Q*(iU(ii)-U0n)+(1-
Q)*(Up(ii)-Up(ii-1)))-iU(ii)+Up(ii);
B(ii,ii)=(fiU(ii)-f(ii))/dU;
%for east
fiU(ii)=K*(iU(ii+1)-2*Uv(ii)+U0n+Up(ii+1)-2*Up(ii)+Up(ii-1))-L*(Q*(Uv(ii)-U0n)+(1-
Q)*(Up(ii)-Up(ii-1)))-Uv(ii)+Up(ii);
```

```
B(ii,ii+1)=(fiU(ii)-f(ii))/dU;
elseif ii==3
%for west
rvi=(Uv(ii+1)-Uv(ii))/(Uv(ii)-iU(ii-1)); lfvi=max(0,TVD(rvi));
ADTvi=0.5*lfvi*(Uv(ii)-iU(ii-1)); rvi_1=(Uv(ii)-iU(ii-1))/(iU(ii-1)-U0n);
lfvi_1=max(0,TVD(rvi_1)); ADTvi_1=0.5*lfvi_1*(iU(ii-1)-U0n);
rpi=(Up(ii+1)-Up(ii))/(Up(ii)-Up(ii-1)); lfpi=max(0,TVD(rpi));
ADTpi=0.5*lfpi*(Up(ii)-Up(ii-1)); rpi_1=(Up(ii)-Up(ii-1))/(Up(ii-1)-Up(ii-2));
lfpi_1=max(0,TVD(rpi_1)); ADTpi_1=0.5*lfpi_1*(Up(ii-1)-Up(ii-2));
fiU(ii)=K*(Uv(ii+1)-2*Uv(ii)+iU(ii-1)+Up(ii+1)-2*Up(ii)+Up(ii-1))-L*(Q*(Uv(ii)+ADTvi-
iU(ii-1)-ADTvi_1)+(1-Q)*(Up(ii)+ADTpi-Up(ii-1)-ADTpi_1))-Uv(ii)+Up(ii);
B(ii,ii-1)=(fiU(ii)-f(ii))/dU;
%for center
rvi=(Uv(ii+1)-iU(ii))/(iU(ii)-Uv(ii-1)); lfvi=max(0,TVD(rvi));
ADTvi=0.5*lfvi*(iU(ii)-Uv(ii-1)); rvi_1=(iU(ii)-Uv(ii-1))/(Uv(ii-1)-U0n);
lfvi_1=max(0,TVD(rvi_1)); ADTvi_1=0.5*lfvi_1*(Uv(ii-1)-U0n);
rpi=(Up(ii+1)-Up(ii))/(Up(ii)-Up(ii-1)); lfpi=max(0,TVD(rpi));
ADTpi=0.5*lfpi*(Up(ii)-Up(ii-1)); rpi_1=(Up(ii)-Up(ii-1))/(Up(ii-1)-Up(ii-2));
lfpi_1=max(0,TVD(rpi_1)); ADTpi_1=0.5*lfpi_1*(Up(ii-1)-Up(ii-2));
fiU(ii)=K*(Uv(ii+1)-2*iU(ii)+Uv(ii-1)+Up(ii+1)-2*Up(ii)+Up(ii-1))-L*(Q*(iU(ii)+ADTvi-
Uv(ii-1)-ADTvi_1)+(1-Q)*(Up(ii)+ADTpi-Up(ii-1)-ADTpi_1))-iU(ii)+Up(ii);
B(ii,ii)=(fiU(ii)-f(ii))/dU;
%for east
rvi=(iU(ii+1)-Uv(ii))/(Uv(ii)-Uv(ii-1)); lfvi=max(0,TVD(rvi));
ADTvi=0.5*lfvi*(Uv(ii)-Uv(ii-1)); rvi_1=(Uv(ii)-Uv(ii-1))/(Uv(ii-1)-U0n);
lfvi_1=max(0,TVD(rvi_1)); ADTvi_1=0.5*lfvi_1*(Uv(ii-1)-U0n);
rpi=(Up(ii+1)-Up(ii))/(Up(ii)-Up(ii-1)); lfpi=max(0,TVD(rpi));
ADTpi=0.5*lfpi*(Up(ii)-Up(ii-1)); rpi_1=(Up(ii)-Up(ii-1))/(Up(ii-1)-Up(ii-2));
lfpi_1=max(0,TVD(rpi_1)); ADTpi_1=0.5*lfpi_1*(Up(ii-1)-Up(ii-2));
fiU(ii)=K*(iU(ii+1)-2*Uv(ii)+Uv(ii-1)+Up(ii+1)-2*Up(ii)+Up(ii-1))-L*(Q*(Uv(ii)+ADTvi-
Uv(ii-1)-ADTvi_1)+(1-Q)*(Up(ii)+ADTpi-Up(ii-1)-ADTpi_1))-Uv(ii)+Up(ii);
B(ii,ii+1)=(fiU(ii)-f(ii))/dU;
elseif 3<ii && ii<I-1
%for west-west
rvi=(Uv(ii+1)-Uv(ii))/(Uv(ii)-Uv(ii-1)); lfvi=max(0,TVD(rvi));
ADTvi=0.5*lfvi*(Uv(ii)-Uv(ii-1)); rvi_1=(Uv(ii)-Uv(ii-1))/(Uv(ii-1)-iU(ii-2));
lfvi_1=max(0,TVD(rvi_1)); ADTvi_1=0.5*lfvi_1*(Uv(ii-1)-iU(ii-2));
rpi=(Up(ii+1)-Up(ii))/(Up(ii)-Up(ii-1)); lfpi=max(0,TVD(rpi));
ADTpi=0.5*lfpi*(Up(ii)-Up(ii-1)); rpi_1=(Up(ii)-Up(ii-1))/(Up(ii-1)-Up(ii-2));
lfpi_1=max(0,TVD(rpi_1)); ADTpi_1=0.5*lfpi_1*(Up(ii-1)-Up(ii-2));
fiU(ii)=K*(Uv(ii+1)-2*Uv(ii)+Uv(ii-1)+Up(ii+1)-2*Up(ii)+Up(ii-1))-L*(Q*(Uv(ii)+ADTvi-
Uv(ii-1)-ADTvi_1)+(1-Q)*(Up(ii)+ADTpi-Up(ii-1)-ADTpi_1))-Uv(ii)+Up(ii);
B(ii,ii-2)=(fiU(ii)-f(ii))/dU;
%for west
rvi=(Uv(ii+1)-Uv(ii))/(Uv(ii)-iU(ii-1)); lfvi=max(0,TVD(rvi));
ADTvi=0.5*lfvi*(Uv(ii)-iU(ii-1)); rvi_1=(Uv(ii)-iU(ii-1))/(iU(ii-1)-Uv(ii-2));
lfvi_1=max(0,TVD(rvi_1)); ADTvi_1=0.5*lfvi_1*(iU(ii-1)-Uv(ii-2));
rpi=(Up(ii+1)-Up(ii))/(Up(ii)-Up(ii-1)); lfpi=max(0,TVD(rpi));
ADTpi=0.5*lfpi*(Up(ii)-Up(ii-1)); rpi_1=(Up(ii)-Up(ii-1))/(Up(ii-1)-Up(ii-2));
lfpi_1=max(0,TVD(rpi_1)); ADTpi_1=0.5*lfpi_1*(Up(ii-1)-Up(ii-2));
fiU(ii)=K*(Uv(ii+1)-2*Uv(ii)+iU(ii-1)+Up(ii+1)-2*Up(ii)+Up(ii-1))-L*(Q*(Uv(ii)+ADTvi-
iU(ii-1)-ADTvi_1)+(1-Q)*(Up(ii)+ADTpi-Up(ii-1)-ADTpi_1))-Uv(ii)+Up(ii);
B(ii,ii-1)=(fiU(ii)-f(ii))/dU;
%for center
rvi=(Uv(ii+1)-iU(ii))/(iU(ii)-Uv(ii-1)); lfvi=max(0,TVD(rvi));
ADTvi=0.5*lfvi*(iU(ii)-Uv(ii-1)); rvi_1=(iU(ii)-Uv(ii-1))/(Uv(ii-1)-Uv(ii-2));
lfvi_1=max(0,TVD(rvi_1)); ADTvi_1=0.5*lfvi_1*(Uv(ii-1)-Uv(ii-2));
rpi=(Up(ii+1)-Up(ii))/(Up(ii)-Up(ii-1)); lfpi=max(0,TVD(rpi));
ADTpi=0.5*lfpi*(Up(ii)-Up(ii-1)); rpi_1=(Up(ii)-Up(ii-1))/(Up(ii-1)-Up(ii-2));
lfpi_1=max(0,TVD(rpi_1)); ADTpi_1=0.5*lfpi_1*(Up(ii-1)-Up(ii-2));
fiU(ii)=K*(Uv(ii+1)-2*iU(ii)+Uv(ii-1)+Up(ii+1)-2*Up(ii)+Up(ii-1))-L*(Q*(iU(ii)+ADTvi-
Uv(ii-1)-ADTvi_1)+(1-Q)*(Up(ii)+ADTpi-Up(ii-1)-ADTpi_1))-iU(ii)+Up(ii);
B(ii,ii)=(fiU(ii)-f(ii))/dU;
```

```
%for east
rvi=(iU(ii+1)-Uv(ii))/(Uv(ii)-Uv(ii-1)); lfvi=max(0,TVD(rvi));
ADTvi=0.5*lfvi*(Uv(ii)-Uv(ii-1)); rvi_1=(Uv(ii)-Uv(ii-1))/(Uv(ii-1)-Uv(ii-2));
lfvi_1=max(0,TVD(rvi_1)); ADTvi_1=0.5*lfvi_1*(Uv(ii-1)-Uv(ii-2));
rpi=(Up(ii+1)-Up(ii))/(Up(ii)-Up(ii-1)); lfpi=max(0,TVD(rpi));
ADTpi=0.5*lfpi*(Up(ii)-Up(ii-1)); rpi_1=(Up(ii)-Up(ii-1))/(Up(ii-1)-Up(ii-2));
lfpi_1=max(0,TVD(rpi_1)); ADTpi_1=0.5*lfpi_1*(Up(ii-1)-Up(ii-2));
fiU(ii)=K*(iU(ii+1)-2*Uv(ii)+Uv(ii-1)+Up(ii+1)-2*Up(ii)+Up(ii-1))-L*(Q*(Uv(ii)+ADTvi-
Uv(ii-1)-ADTvi_1)+(1-Q)*(Up(ii)+ADTpi-Up(ii-1)-ADTpi_1))-Uv(ii)+Up(ii);
B(ii,ii+1)=(fiU(ii)-f(ii))/dU;
elseif ii==I-1
%for west-west
rvi=(UIn-Uv(ii))/(Uv(ii)-Uv(ii-1)); lfvi=max(0,TVD(rvi));
ADTvi=0.5*lfvi*(Uv(ii)-Uv(ii-1)); rvi_1=(Uv(ii)-Uv(ii-1))/(Uv(ii-1)-iU(ii-2));
lfvi_1=max(0,TVD(rvi_1)); ADTvi_1=0.5*lfvi_1*(Uv(ii-1)-iU(ii-2));
rpi=(Up(ii+1)-Up(ii))/(Up(ii)-Up(ii-1)); lfpi=max(0,TVD(rpi));
ADTpi=0.5*lfpi*(Up(ii)-Up(ii-1)); rpi_1=(Up(ii)-Up(ii-1))/(Up(ii-1)-Up(ii-2));
lfpi_1=max(0,TVD(rpi_1)); ADTpi_1=0.5*lfpi_1*(Up(ii-1)-Up(ii-2));
fiU(ii)=K*(UIn-2*Uv(ii)+Uv(ii-1)+Up(ii+1)-2*Up(ii)+Up(ii-1))-L*(Q*(Uv(ii)+ADTvi-Uv(ii-
1)-ADTvi_1)+(1-Q)*(Up(ii)+ADTpi-Up(ii-1)-ADTpi_1))-Uv(ii)+Up(ii);
B(ii,ii-2)=(fiU(ii)-f(ii))/dU;
%for west
rvi=(UIn-Uv(ii))/(Uv(ii)-iU(ii-1)); lfvi=max(0,TVD(rvi));
ADTvi=0.5*lfvi*(Uv(ii)-iU(ii-1)); rvi_1=(Uv(ii)-iU(ii-1))/(iU(ii-1)-Uv(ii-2));
lfvi_1=max(0,TVD(rvi_1)); ADTvi_1=0.5*lfvi_1*(iU(ii-1)-Uv(ii-2));
rpi=(Up(ii+1)-Up(ii))/(Up(ii)-Up(ii-1)); lfpi=max(0,TVD(rpi));
ADTpi=0.5*lfpi*(Up(ii)-Up(ii-1)); rpi_1=(Up(ii)-Up(ii-1))/(Up(ii-1)-Up(ii-2));
lfpi_1=max(0,TVD(rpi_1)); ADTpi_1=0.5*lfpi_1*(Up(ii-1)-Up(ii-2));
fiU(ii)=K*(UIn-2*Uv(ii)+iU(ii-1)+Up(ii+1)-2*Up(ii)+Up(ii-1))-L*(Q*(Uv(ii)+ADTvi-iU(ii-
1)-ADTvi_1)+(1-Q)*(Up(ii)+ADTpi-Up(ii-1)-ADTpi_1))-Uv(ii)+Up(ii);
B(ii,ii-1)=(fiU(ii)-f(ii))/dU;
%for center
rvi=(UIn-iU(ii))/(iU(ii)-Uv(ii-1)); lfvi=max(0,TVD(rvi));
ADTvi=0.5*lfvi*(iU(ii)-Uv(ii-1)); rvi_1=(iU(ii)-Uv(ii-1))/(Uv(ii-1)-Uv(ii-2));
lfvi_1=max(0,TVD(rvi_1)); ADTvi_1=0.5*lfvi_1*(Uv(ii-1)-Uv(ii-2));
rpi=(Up(ii+1)-Up(ii))/(Up(ii)-Up(ii-1)); lfpi=max(0,TVD(rpi));
ADTpi=0.5*lfpi*(Up(ii)-Up(ii-1)); rpi_1=(Up(ii)-Up(ii-1))/(Up(ii-1)-Up(ii-2));
lfpi_1=max(0,TVD(rpi_1)); ADTpi_1=0.5*lfpi_1*(Up(ii-1)-Up(ii-2));
fiU(ii)=K*(UIn-2*iU(ii)+Uv(ii-1)+Up(ii+1)-2*Up(ii)+Up(ii-1))-L*(Q*(iU(ii)+ADTvi-Uv(ii-
1)-ADTvi_1)+(1-Q)*(Up(ii)+ADTpi-Up(ii-1)-ADTpi_1))-iU(ii)+Up(ii);
B(ii,ii+1)=(fiU(ii)-f(ii))/dU;
elseif ii==I
%for west
fiU(ii)=K*(UIn-2*UIn+iU(ii-1)+Up(ii)-2*Up(ii)+Up(ii-1))-L*(Q*(UIn-iU(ii-1))+(1-
Q)*(Up(ii)-Up(ii-1)))-UIn+Up(ii);
B(ii,ii-1)=(fiU(ii)-f(ii))/dU;
end
end
B(:,1)=[]; B(:,end)=[];%End of the Determination of A and B Matrices
v=sparse(B)\sparse(A); vX=Uv(2:end-1)'-v; Uv=[U0n vX' UIn];
vUv=max(abs(v)); it=it+1;
end %End of NR iteration
Un=Uv; Up=Un;%End of the implicit solution
end%End of the numerical solution
%Analytical Solution
i=1; Ua(1:I)=NaN;
for x=X
if D==0
Ua(i)=1-heaviside(x-vf*dt*n); %Analytical Solution with Method of Characteristics
else
Ua(i)=0.5*erfc((x-
vf*dt*n)/(2*(D*dt*n)^0.5))+0.5*exp(vf*x/D)*erfc((x+vf*dt*n)/(2*(D*dt*n)^0.5));
end
```

```
i=i+1;
end
%Plot Solution
set(gcf, 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]);
plot(X/I/dx,Un,'gv-','markerfacecolor','g');
hold on
plot(X/I/dx,Ua,'k-','linewidth',2);
xlabel('Dimensionless Distance','fontsize',12)
ylabel('Concentration','fontsize',12)
legend('Numerical Solution','Analytical Solution')
hold off
pause(0.001)
end
toc
```

Supplementary material 3. Google Drive link.

In order to reach Matlab files, please use following Google Drive link:

https://drive.google.com/drive/folders/1JCkZqNRVqzAFGR8M3wqQMsGMDtqKuamx?usp=sharing