Fluid Dynamics — Numerical Techniques

MATH5453M Numerical Exercises 1, 2024

Aly Ilyas
mmai@leeds.ac.uk

Due date: Friday October 18th

Sources: Chapter 2 of Morton and Mayers (2005, M&M), Internet.

Problem Statement

Consider the non-dimensional linear advection-diffusion equation for the variable/unknown $u = u(x,t)$, with an initial condition and boundary conditions:

$$u_t - a(t)u_x - \epsilon u_{xx} = 0, \quad x \in [L_p, L] \tag{1a}$$

$$u(x,0) = u_0(x) \tag{1b}$$

$$u(L_p,t) = u(L,t) = 0, \tag{1c}$$

where ϵ is a small constant diffusion and $a(t)$ is a given function. The boundary conditions are classical homogeneous Dirichlet conditions. The above system arose from the research on machine learning of Choi et al. (2022). In the end, we will use $L_p = -1$ and $a(t) = 1$.

Tasks

Task 1:

Equation (1) is referred to as a linear advection-diffusion equation because it combines the effects of both advection (transport by a flow) and diffusion (spreading due to gradients), and its structure is linear in the unknown $u(x,t)$. Let's break down the three key terms:

The term **linear** means that the equation is linear in the unknown variable $u(x,t)$ and its derivatives. Specifically, the equation is of the form:

$$u_t - a(t)u_x - \epsilon u_{xx} = 0. \tag{1}$$

The unknown function u and its derivatives u_x, u_{xx}, and u_t all appear with constant coefficients (or coefficients that are independent of u), and there are no terms like u^2, $\sin(u)$, or any other nonlinear functions of u. In this case, the coefficient of u_x is $a(t)$, which is a function of time, but this does not depend on u itself, making the equation linear.

The term **advection** refers to the transport of a quantity (in this case, u) due to the movement of a medium. In the equation, this is represented by the term:

$$-a(t)u_x. \tag{2}$$

The derivative u_x describes how u changes with respect to space, and $a(t)$ represents the velocity at which u is transported in space. When $a(t) = 1$, it means the quantity u is advected or transported at a constant speed of 1 in the x-direction. The negative sign indicates that the dirrection of advection is opposite to the dirrection of incresing x (if $a(t)$ is positive, the flow is in the positive x dirrection. Thus the term (2) describes how the quantity u is carried along by the flow or shift of the solution over time.

The term **diffusion** refers to the spreading of a quantity due to gradients, which is typically modeled by a second-order spatial derivative. In the equation, this is represented by:

$$-\epsilon u_{xx}. \tag{3}$$

The second derivative u_{xx} describes how the gradient of u changes in space, and ϵ is a small constant representing the strength of the diffusion. This term causes u to "smooth out" or diffuse over time. The larger the value of ϵ, the faster the diffusion. In the case of small ϵ, diffusion is weak.

Task 2

Given the equation (1) at a specific time point (x_j, t_n), the time derivative u_t can be approximated using a central difference between t_n and t_{n+1} as:

$$\delta_t u_j^{n+\frac{1}{2}} = u_j^{n+1} - u_j^n. \tag{4}$$

Then, we expand both term u_j^{n+1} and u_j^n in Taylor series around mid point $(x_j, t_{n+\frac{1}{2}})$. We get:

$$u_j^{n+1} = \left[u + \frac{1}{2}\Delta t u_t + \frac{1}{2}\left(\frac{1}{2}\Delta t\right)^2 u_{tt} + \frac{1}{6}\left(\frac{1}{2}\Delta t\right)^3 u_{ttt} + \cdots \right]_j^{n+\frac{1}{2}}, \tag{5}$$

and

$$u_j^{n} = \left[u - \frac{1}{2}\Delta t u_t + \frac{1}{2}\left(\frac{1}{2}\Delta t\right)^2 u_{tt} - \frac{1}{6}\left(\frac{1}{2}\Delta t\right)^3 u_{ttt} + \cdots \right]_j^{n+\frac{1}{2}}. \tag{6}$$

Subtracting (5) to (6) and we obtain equation 2.80 in M&M:

$$\delta_t u_j^{n+\frac{1}{2}} = u_j^{n+1} - u_j^n = \left[\Delta t u_t + \frac{1}{24}(\Delta t)^3 u_{ttt} + \cdots \right]_j^{n+\frac{1}{2}}. \tag{7}$$

Next, the spatial second derivative u_{xx} at point x_j based on equation 2.30 in M&M, for $n+1$:

$$\delta_x^2 u_j^{n+1} = u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}. \tag{8}$$

For (8), we expand both term u_{j+1}^n and u_{j-1}^n. We get:

$$u_{j+1}^{n+1} = u_j^{n+1} + \Delta x u_x^{n+1} + \frac{(\Delta x)^2}{2} u_{xx}^{n+1} + \frac{(\Delta x)^3}{6} u_{xxx}^{n+1} + \frac{(\Delta x)^4}{24} u_{xxxx}^{n+1} + \cdots, \tag{9}$$

and

$$u_{j-1}^{n+1} = u_j^{n+1} - \Delta x u_x^{n+1} + \frac{(\Delta x)^2}{2} u_{xx}^{n+1} - \frac{(\Delta x)^3}{6} u_{xxx}^{n+1} + \frac{(\Delta x)^4}{24} u_{xxxx}^{n+1} + \cdots. \tag{10}$$

Subtituting (9) and (10) into (8) and we get the equation 2.81 from M&M as:

$$\delta_x^2 u_j^{n+1} = \left[(\Delta x)^2 u_{xx} + \frac{1}{12}(\Delta x)^4 u_{xxxx} + \frac{2}{6!}(\Delta x)^6 u_{xxxxxx} + \cdots \right]_j^{n+1}. \tag{11}$$

Then we expand each term in (11) in powers of Δt, about the point $(x_j, t_{n+\frac{1}{2}})$. The Taylor series expansion for a function $u(x,t)$ is given by:

$$\begin{aligned} u(x,t) = u(x_j, t_{n+\frac{1}{2}}) \\ &+ (x - x_j) u_x(x_j, t_{n+\frac{1}{2}}) \\ &+ (t - t_{n+\frac{1}{2}}) u_t(x_j, t_{n+\frac{1}{2}}) \\ &+ \frac{1}{2}(x - x_j)^2 u_{xx}(x_j, t_{n+\frac{1}{2}}) \\ &+ \frac{1}{2}(t - t_{n+\frac{1}{2}})^2 u_{tt}(x_j, t_{n+\frac{1}{2}}) + \cdots. \end{aligned} \tag{12}$$

Now we apply (12) to the second derivative u_{xx} and higher derivatives, for u_{xx}:

$$u_{xx} = u_{xx}(x_j, t_{n+\frac{1}{2}}) + \Delta t\, u_{xxt}(x_j, t_{n+\frac{1}{2}}) + \frac{1}{2}(\Delta t)^2 u_{xxtt}(x_j, t_{n+\frac{1}{2}}) + \cdots\,, \quad (13)$$

for u_{xxxx}:

$$u_{xxxx} = u_{xxxx}(x_j, t_{n+\frac{1}{2}}) + \Delta t\, u_{xxxxt}(x_j, t_{n+\frac{1}{2}}) + \cdots\,, \quad (14)$$

for u_{xxxxxx}:

$$u_{xxxxxx} = u_{xxxxxx}(x_j, t_{n+\frac{1}{2}}) + \Delta t\, u_{xxxxxxxt}(x_j, t_{n+\frac{1}{2}}) + \cdots\,. \quad (15)$$

Then we subtitute (13), (14), and (15) (PS. let's just drop $(x_j, t_{n+\frac{1}{2}})$ for simplicity) into (11) and get:

$$\begin{aligned}\delta_x^2 u_j^{n+1} = &\left[(\Delta x)^2 u_{xx} + \frac{1}{12}(\Delta x)^4 u_{xxxx} + \frac{2}{6!}(\Delta x)^6 u_{xxxxxx} + \cdots\right] \\ &+ \left[\frac{1}{2}\Delta t(\Delta x)^2 u_{xxt} + \frac{1}{12}(\Delta x)^4 u_{xxxxt} + \cdots\right] \\ &+ \left[\frac{1}{2}(\frac{1}{2}\Delta t)^2(\Delta x)^2 u_{xxtt} + \cdots\right] + \cdots\,.\end{aligned} \quad (16)$$

There is similar expansion for $\delta_x^2 u_j^n$ and combining it with (16), we get the equation 2.82 in M&M:

$$\begin{aligned}\theta\delta_x^2 u_j^{n+1} + (1-\theta)\delta_x^2 u_j^n = &\left[(\Delta x)^2 u_{xx} + \frac{1}{12}(\Delta x)^4 u_{xxxx} + \frac{2}{6!}(\Delta x)^6 u_{xxxxxx} + \cdots\right] \\ &+ \left[\left(\theta - \frac{1}{2}\right)\Delta t(\Delta x)^2 u_{xxt} + \frac{1}{12}(\Delta x)^4 u_{xxxxt} + \cdots\right] \\ &+ \frac{1}{8}(\Delta t)^2(\Delta x)^2 u_{xxtt} + \cdots\,.\end{aligned} \quad (17)$$

We already got (7) and (17), and the form of truncation error is given by equation 2.83 in the M&M as:

$$T_j^{n+\frac{1}{2}} := \frac{\delta_t u_j^{n+\frac{1}{2}}}{\Delta t} - \frac{\theta\delta_x^2 u_j^{n+1} + (1-\theta)\delta_x^2 u_j^n}{(\Delta x)^2}. \quad (18)$$

Finally, we get the equation 2.84 in M&M as:

$$T_j^{n+\frac{1}{2}} := \left[\left(\frac{1}{2}-\theta\right)\Delta t\, u_{xxt} - \frac{1}{12}(\Delta x)^2 u_{xxxx}\right] + \left[\frac{1}{24}(\Delta t)^2 u_{ttt} - \frac{1}{8}(\Delta t)^2 u_{xxtt}\right] + \left[\frac{1}{12}\left(\frac{1}{2}-\theta\right)\Delta t(\Delta x)^2 u_{xxxxt} - \frac{2}{6!}(\Delta x)^4 u_{xxxxxx}\right]. \tag{19}$$

Task 3

Given the non-dimensional linear advection-diffusion equation (1a) with initial and boundary conditions (1b) and (1c), we will let the domain $[L_p, L]$ be divided into J points.

$$\Delta x = \frac{L-L_p}{J}. \tag{20}$$

The spatial grid points are denoted as x_j for $j = 0, 1, 2, \ldots, J$, where:

$$x_j = L_p + j\Delta x. \tag{21}$$

Next, we discretize the time domain into N points, with time step size Δt, such that the time grid points are t_n for $n = 0, 1, 2, \ldots, N-1$, where:

$$t_n = n\Delta t. \tag{22}$$

Then, the time derivative u_t is discretized using the θ-method:

$$u_t \approx \frac{u_j^{n+1} - u_j^n}{\Delta t}. \tag{23}$$

The θ-method approximates the equation at time level $n+1$ as:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \theta\left(a(t)u_x^{n+1} + \epsilon u_{xx}^{n+1}\right) + (1-\theta)\left(a(t)u_x^n + \epsilon u_{xx}^n\right). \tag{24}$$

Here, $\theta \in [0,1]$ controls the scheme: $\theta = 0$ gives a fully explicit scheme, $\theta = 1$ gives a fully implicit scheme, and $\theta = \frac{1}{2}$ gives the Crank-Nicolson scheme (midpoint).

The advection term $a(t)u_x$ is discretized using the first-order upwind scheme. The direction of the upwinding is determined by the sign of $a(t)$. Since we assume $a(t) = 1$ (positive), we use the leftward upwind stencil, thus, the advection term becomes:
for $n+1$:

$$u_x^{n+1} \approx \frac{u_{j+1}^{n+1} - u_j^{n+1}}{\Delta x}, \tag{25}$$

and for n:

$$u_x^n \approx \frac{u_{j+1}^n - u_j^n}{\Delta x}. \tag{26}$$

The diffusion term ϵu_{xx} is discretized using the central difference scheme: for $n+1$:

$$u_{xx}^{n+1} \approx \frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{\Delta x^2}, \tag{27}$$

and for n:

$$u_{xx}^n \approx \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2}. \tag{28}$$

The boundary conditions are:

$$u(L_p, t) = u(L, t) = 0. \tag{29}$$

At grid points corresponding to $j = 0$ and $j = J$, we enforce:

$$u_0^{n+1} = u_0^n = 0, \quad u_J^{n+1} = u_J^n = 0. \tag{30}$$

First, we arrange the equation 1a and we get:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} - a(t)\frac{u_{j+1}^n - u_j^n}{\Delta x} - \epsilon\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2} = 0.$$

Putting the advection and diffusion term to the right hand side we get:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = a(t)\frac{u_{j+1}^n - u_j^n}{\Delta x} + \epsilon\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2},$$

$$u_j^{n+1} = u_j^n + \Delta t\left(a(t)\frac{u_{j+1}^n - u_j^n}{\Delta x} + \epsilon\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2}\right).$$

Then, we let ν be $\frac{\Delta t}{\Delta x}$ and μ be $\frac{\Delta t}{\Delta x^2}$ and we get:

$$u_j^{n+1} = u_j^n + \nu a(t)\left(u_{j+1}^n - u_j^n\right) + \mu\epsilon\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right).$$

We can do the same for the implicit scheme and we got:

$$u_j^{n+1} = u_j^n + \nu a(t)\left(u_{j+1}^{n+1} - u_j^{n+1}\right) + \mu\epsilon\left(u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}\right).$$

For near the boundary scenario, we rewrite the θ-methods with the internal points $j = 1, 2, \ldots, J-1$, the discretization becomes:

$$u_j^{n+1} = u_j^n + \theta\left(\nu a(t)\left(u_{j+1}^{n+1} - u_j^{n+1}\right) + \mu\epsilon\left(u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}\right)\right)$$

$$+(1-\theta)\left(\nu a(t)\left(u_{j+1}^n-u_j^n\right)+\mu\epsilon\left(u_{j+1}^n-2u_j^n+u_{j-1}^n\right)\right). \tag{31}$$

Then we rearrange terms to isolate u_j^{n+1} (unknown) on the left-hand side and u_j^n (known) on the right-hand side:

$$\begin{aligned}
&(1+\theta\nu a(t)+2\theta\mu\epsilon)\,u_j^{n+1}+(-\theta\mu\epsilon)\,u_{j-1}^{n+1}\\
&\qquad\qquad+(-\theta(\nu a(t)+\mu\epsilon))\,u_{j+1}^{n+1}\\
&=(1-\theta\nu a(t)-2\theta\mu\epsilon)\,u_j^n-((1-\theta)\mu\epsilon)\,u_{j-1}^n\\
&\qquad\qquad+(1-\theta)(\nu a(t)+\theta\mu\epsilon))u_{j+1}^n.
\end{aligned} \tag{32}$$

The system of (32) can also be written in matrix form:

$$Au^{n+1}=Bu^n, \tag{33}$$

where A and B are matrices containing the coefficients from the discretization of u_j^{n+1} and u_j^n, respectively.

When solving the advection-diffusion equation for near-boundary scenarios, three distinct cases arise, each requiring modifications to the discretized equation based on the grid point's position.

The first case occurs at the grid point $j=1$ (near the left boundary). Here, the boundary term involving u_0^{n+1} is known from the boundary condition, which in many cases equals zero. This simplifies the discretized equation, and the resulting system accounts for the known boundary term when solving for u_1^{n+1}.

The second case applies to internal grid points, $j=2,3,\dots,J-2$, away from the boundaries. At these points, all terms on the left-hand side of the discretized equation are unknowns, requiring a fully implicit or semi-implicit scheme without any simplifications.

The third case arises at $j=J-1$ (near the right boundary). Similar to the first case, a boundary term involving u_J^{n+1} is known from the boundary condition. This term is incorporated into the discretized equation for u_{J-1}^{n+1}, ensuring the boundary condition is satisfied while solving the system.

Each of these cases modifies the general discretized equation to incorporate the effects of boundary conditions while maintaining consistency in the numerical scheme.

Task 4

In order to reproduce Fig 2.2 in M&M, we need to solve a one-dimensional partial differential equation (PDE) of the form:

$$u_t=u_{xx}, \tag{34}$$

where $u(x,t)$ represents a quantity (e.g., temperature or concentration) at position x and time t. The boundary conditions are homogeneous Dirichlet conditions:

$$u(0,t) = u(1,t) = 0, \tag{35}$$

and the initial condition is given by a "hat" function:

$$u_0(x) = \begin{cases} 2x, & 0 \leq x \leq 0.5, \\ 2(1-x), & 0.5 \leq x \leq 1. \end{cases} \tag{36}$$

The explicit finite difference scheme for this PDE is written as:

$$u_j^{n+1} = u_j^n + \mu \left(u_{j+1}^n - 2u_j^n + u_{j-1}^n \right), \tag{37}$$

where u_j^n is the value of $u(x_j, t_n)$ at the j-th spatial point and the n-th time step, and $\mu = \frac{\Delta t}{\Delta x^2}$. The boundary conditions ensure that $u_0^n = u_J^n = 0$ for all n.

Using **number4.py** we simulate both case. For the stable case, the time step $\Delta t = 0.0012$ results in $\mu = 0.48$, which satisfies the stability condition:

$$\mu = \frac{\Delta t}{\Delta x^2} \leq 0.5. \tag{38}$$

The solution remains stable without oscillations or unbounded growth, as shown in the Figure 1a. The solution evolves smoothly over time, reflecting a well-behaved diffusion process.

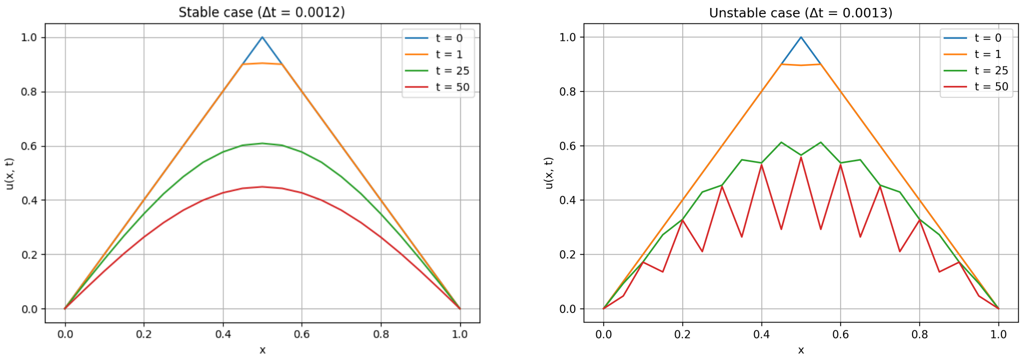
In the unstable case, the time step $\Delta t = 0.0013$ leads to $\mu = 0.52$, which exceeds the stability threshold of 0.5. As a result, the solution becomes unstable, and oscillations grow over time, as shown in the Figure 1b.

The amplification factor for the Fourier mode with wavenumber k in the explicit scheme is:

$$\lambda(k) = 1 - 4\mu \sin^2\left(\frac{k\Delta x}{2}\right). \tag{39}$$

Stability requires that $|\lambda(k)| \leq 1$ for all modes k. If $\mu > 0.5$, some Fourier modes grow exponentially, leading to instability. For $\mu = 0.52$, the amplification factor exceeds 1 for certain wavenumbers, causing the unbounded oscillations observed in the unstable case.

The explicit scheme satisfies the maximum principle when $\mu \leq 0.5$. This ensures that the solution $u(x,t)$ remains bounded within the initial and boundary conditions. When $\mu > 0.5$, this principle is violated, leading to the unbounded growth of oscillations, as observed in the unstable case.



(a) (b)

Figure 1: Stability and unstability comparison (reproduce from Fig 2.2 in M&M using **number4.py**) (a) stable case where $\mu = 0.48$ using the time step $\Delta t = 0.0012$ (b) unstable case where $\mu = 0.52$ using the time step $\Delta t = 0.0013$.

Next, we extend the explicit scheme $\theta = 0$ to the previous statement problem (1) using **number4b.py**. In this case, we are considering the advection-diffusion equation with the following parameters: $L = 1.0$ (right boundary), $L_p = -1.0$ (left boundary), $J = 40$ (number of spatial points), $\Delta x = \frac{L-L_p}{J} = 0.05$ (spatial step size), $\epsilon = 1 \times 10^{-3}$ (diffusion constant), and $a = 1.0$ (advection speed). The initial condition is given by function:

$$u(x, 0) = (1 - x)^4(1 + x). \tag{40}$$

The explicit scheme is subject to stability conditions for both advection and diffusion. The stability condition for diffusion is determined by:

$$\epsilon \frac{\Delta t}{\Delta x^2} < \frac{1}{2}. \tag{41}$$

Solving for Δt, we obtain:

$$\Delta t_{\text{diffusion}} < \frac{1}{2} \times \frac{\Delta x^2}{\epsilon} = 1.25. \tag{42}$$

Similarly, the stability condition for advection is given by:

$$\frac{a\Delta t}{\Delta x} < 1. \tag{43}$$

Solving for Δt, we obtain:

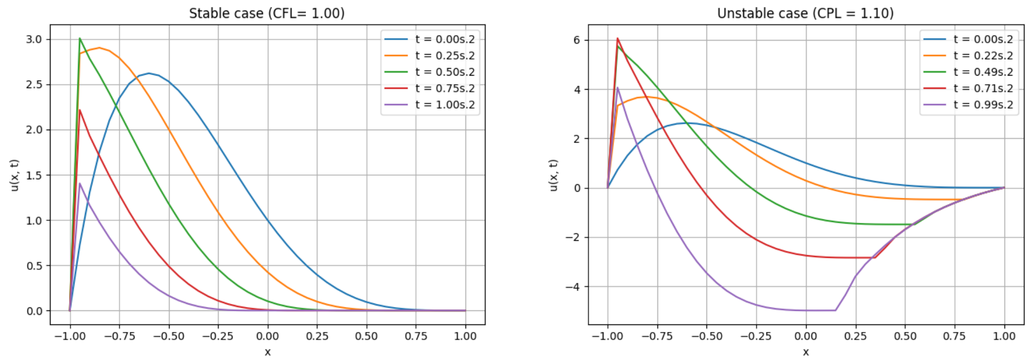
$$\Delta t_{\text{advection}} < \frac{0.05}{1.0} = 0.05. \tag{44}$$

Since both diffusion and advection must be stable, the critical time step $\Delta t_{\text{critical}}$ is determined as the smaller of the two values of (42) and (44):

$$\Delta t_{\text{critical}} = 0.05. \tag{45}$$

Thus, for stability, we use $\Delta t_1 = 0.05$ for the stable case. To illustrate an unstable case, we select $\Delta t_2 = 0.051$, which exceeds the critical value.

Figure 2 demonstrates the behavior of the system using these two time steps. Figure 2a shows the stable case with $\Delta t_1 = 0.05$, where the solution becomes stable by time step 2. Meanwhile, Figure 2b illustrates the unstable case with $\Delta t_2 = 0.051$, where oscillations begin to appear.



(a) (b)

Figure 2: Stability and unstability on explicit scheme $\theta = 0$ comparison from problem statement (1) using **number4b.py** (a) stable case where $\Delta t = 0.05$ (b) unstable case where $\Delta t = 0.051$.

Task 5

We have already presented the non-dimensional linear advection-diffusion equation (1a) along with the initial and boundary conditions (1b) and (1c). Additionally, we have the θ-scheme to discretize the time derivative from (32). From equations (25), (26), (27), and (28), the discretization for the advection term at time step $n+1$ is given by:

$$a(t)\frac{u_{j+1}^{n+1} - u_j^{n+1}}{\Delta x}. \tag{46}$$

Similarly, at time step n, it is:

$$a(t)\frac{u_{j+1}^{n} - u_j^{n}}{\Delta x}. \tag{47}$$

Next, we can derive the diffusion term. At time step $n+1$, the diffusion term becomes:

$$\epsilon\frac{u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1}}{\Delta x^2}. \tag{48}$$

Similarly, at time step n, it is:

$$\epsilon\frac{u_{j+1}^{n}-2u_j^{n}+u_{j-1}^{n}}{\Delta x^2}. \tag{49}$$

We already got the equation in matrix form in (33). For the matrix A, we use the following relationships for the coefficients of u_j^{n+1}:

$$A[j,j-1]=\left(-\theta\mu\epsilon\right), \tag{50}$$

$$A[j,j]=\left(1+\theta\nu a(t)+2\theta\mu\epsilon\right), \tag{51}$$

$$A[j,j+1]=\left(-\theta(\nu a(t)+\mu\epsilon)\right). \tag{52}$$

For the matrix B, we use the following relationships for the coefficients of u_j^n:

$$B[j,j-1]=\left((1-\theta)\mu\epsilon\right), \tag{53}$$

$$B[j,j]=\left(1-\theta\nu a(t)-2\theta\mu\epsilon\right), \tag{54}$$

$$B[j,j+1]=(1-\theta)(\nu a(t)+\theta\mu\epsilon)). \tag{55}$$

The boundary conditions are applied explicitly, ensuring $u_0=u_{N-1}=0$ for Dirichlet boundary conditions.

Considering the case where $a>0$:

$$u_j^{n+1}=u_j^n+\theta\nu a_{n+1}\left(u_{j+1}^{n+1}-u_j^{n+1}\right)+\epsilon\mu\left(u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1}\right) \tag{56}$$

$$+(1-\theta)\nu a_n\left(u_{j+1}^n-u_j^n\right)+\epsilon\mu\left(u_{j+1}^n-2u_j^n+u_{j-1}^n\right). \tag{57}$$

Now, we assume the solution takes the form $u_j^n=\lambda^n e^{ikj\Delta x}$, which ensures stability when $|\lambda|\leq 1$. Substituting this into the equation above yields:

$$\lambda^{n+1}e^{ik(j)\Delta x}=\lambda^n e^{ikj\Delta x}+\theta\nu a_{n+1}\left(\lambda^{n+1}e^{ik(j+1)\Delta x}-\lambda^{n+1}e^{ikj\Delta x}\right) \tag{58}$$

$$+\epsilon\mu\left(\lambda^{n+1}e^{ik(j+1)\Delta x}-2\lambda^{n+1}e^{ikj\Delta x}+\lambda^{n+1}e^{ik(j-1)\Delta x}\right) \tag{59}$$

$$+(1-\theta)\nu a_n\left(\lambda^n e^{ik(j+1)\Delta x}-\lambda^n e^{ikj\Delta x}\right) \tag{60}$$

$$+\epsilon\mu\left(\lambda^n e^{ik(j+1)\Delta x}-2\lambda^n e^{ikj\Delta x}+\lambda^n e^{ik(j-1)\Delta x}\right). \tag{61}$$

After some algebra, we can rewrite the equation as:

$$\lambda=1+\lambda\theta\nu a_{n+1}\left(e^{ik\Delta x}-1\right)+\epsilon\mu\left(e^{ik\Delta x}-2+e^{-ik\Delta x}\right) \tag{62}$$

$$+(1-\theta)\nu a_n\left(e^{ik\Delta x}-1\right)+\epsilon\mu\left(e^{ik\Delta x}-2+e^{-ik\Delta x}\right). \tag{63}$$

Finally, simplifying and rearranging the equation, we get:

$$\lambda\left(1-\theta\nu a_{n+1}\left(e^{ik\Delta x}-1\right)+\epsilon\mu\left(e^{ik\Delta x}-2+e^{-ik\Delta x}\right)\right)=1+(1-\theta)\nu a_n\left(e^{ik\Delta x}-1\right) \tag{64}$$

$$+\epsilon\mu\left(e^{ik\Delta x}-2+e^{-ik\Delta x}\right). \tag{65}$$

Thus, the solution for λ becomes:

$$\lambda=1+(1-\theta)\nu a_n\left(e^{ik\Delta x}-1\right)+\epsilon\mu\left(e^{ik\Delta x}-2+e^{-ik\Delta x}\right). \tag{66}$$

Next, for the diffusion part when $a=0$, we observe a similar case where stability occurs when:

$$\mu\leq\frac{1}{2\epsilon(1-2\theta)}. \tag{67}$$

For the pure advection case, where $\epsilon=0$, Equation (66) simplifies to:

$$\lambda=1+(1-\theta)\nu a_n\left(e^{ik\Delta x}-1\right). \tag{68}$$

This leads to the expression:

$$\lambda=\frac{1+(1-\theta)\nu a\left(e^{ik\Delta x}-1\right)}{1-\theta\nu a_{n+1}\left(e^{ik\Delta x}-1\right)}. \tag{69}$$

Assuming that a is constant, we have $a_{n+1}=a_n$, and using Euler's formula $e^{i\theta}=\cos(\theta)+i\sin(\theta)$, we get:

$$\lambda=\frac{1+(1-\theta)\nu a\left(\cos(k\Delta x)+i\sin(k\Delta x)-1\right)}{1-\theta\nu a\left(\cos(k\Delta x)+i\sin(k\Delta x)-1\right)}. \tag{70}$$

Next, we use the identity $\cos(2\theta)-1=-2\sin^2(\theta)$ and separate the real and imaginary parts, which leads to:

$$\lambda=\left[1-2(1-\theta)\nu a\sin^2\left(\frac{k\Delta x}{2}\right)\right]+i(1-\theta)\nu a\sin(k\Delta x) \tag{71}$$

$$\left[1+2\theta\nu a\sin^2\left(\frac{k\Delta x}{2}\right)\right]-i\theta\nu a\sin(k\Delta x). \tag{72}$$

To ensure stability, we require $|\lambda|\leq 1$. Thus, we must calculate $|\lambda|^2$, which involves multiplying by the complex conjugate:

$$|\lambda|^2=\left[1-2(1-\theta)\nu a\sin^2\left(\frac{k\Delta x}{2}\right)\right]^2+\left((1-\theta)\nu a\sin(k\Delta x)\right)^2 \tag{73}$$

$$\left[1 + 2\theta\nu a \sin^2\left(\frac{k\Delta x}{2}\right)\right]^2 + \left(\theta\nu a \sin(k\Delta x)\right)^2. \tag{74}$$

Expanding the squared terms and simplifying, we get the condition for stability:

$$-4\sin^2\left(\frac{k\Delta x}{2}\right) + 4(1-2\theta)(\nu a)\sin^4\left(\frac{k\Delta x}{2}\right) + (1-2\theta)(\nu a)\sin^2(k\Delta x) \leq 0. \tag{75}$$

This can be rewritten as:

$$-4\sin^2\left(\frac{k\Delta x}{2}\right) + (1-2\theta)\nu a\left[4\sin^4\left(\frac{k\Delta x}{2}\right) + \sin^2(k\Delta x)\right] \leq 0. \tag{76}$$

Next, using the identity $\sin(2\theta) = 2\sin(\theta)\cos(\theta)$, we get:

$$-4\sin^2\left(\frac{k\Delta x}{2}\right) + (1-2\theta)\nu a\left[4\sin^4\left(\frac{k\Delta x}{2}\right) + 4\sin^2\left(\frac{k\Delta x}{2}\right)\cos^2\left(\frac{k\Delta x}{2}\right)\right] \leq 0. \tag{77}$$

Factoring out the terms, we obtain:

$$-4\sin^2\left(\frac{k\Delta x}{2}\right) + 4(1-2\theta)\nu a \sin^2\left(\frac{k\Delta x}{2}\right)\left[\sin^2\left(\frac{k\Delta x}{2}\right) + \cos^2\left(\frac{k\Delta x}{2}\right)\right] \leq 0. \tag{78}$$

Since $\sin^2(\theta) + \cos^2(\theta) = 1$, we simplify the expression to:

$$-4\sin^2\left(\frac{k\Delta x}{2}\right) + 4(1-2\theta)\nu a \sin^2\left(\frac{k\Delta x}{2}\right) \leq 0. \tag{79}$$

Thus, we find the stability condition:

$$(1-2\theta)\nu a \leq 1. \tag{80}$$

This gives the final stability criterion for small θ, valid for $a > 0$.

Given the parameters $L = 1.0$, $L_p = -1.0$, $J = 100$, $\Delta x = 0.02$, $\Delta t = 0.05$, $a = 1.0$, $\epsilon = 0.001$, and $\theta = 0.5$, we can assess the stability of the scheme using the derived stability condition:

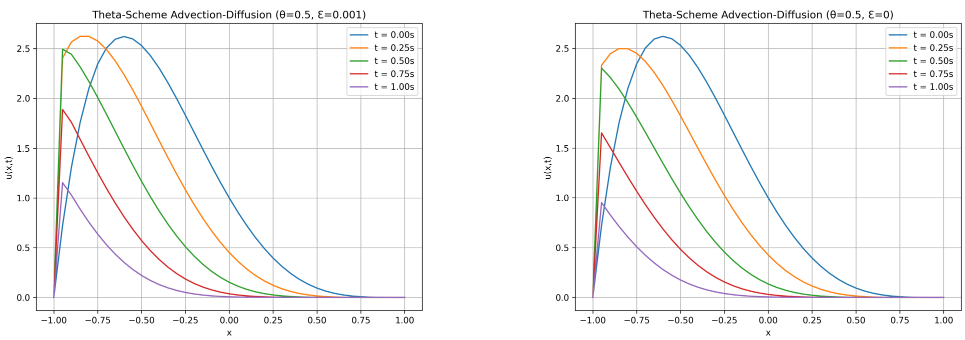
$$(1-2\theta)\nu a \leq 1,$$

where ν is the Courant number, defined as $\nu = \frac{\Delta t}{\Delta x}$. Substituting the values of $\Delta t = 0.05$ and $\Delta x = 0.02$ into this expression, we calculate $\nu = 2.5$. Substituting $\nu = 2.5$, $a = 1.0$, and $\theta = 0.5$ into the stability condition results in:

$$(1 - 2(0.5)) \cdot 2.5 \cdot 1.0 \leq 1,$$

which simplifies to $0 \leq 1$. Since this inequality is satisfied, the stability condition holds, and the scheme is stable for these parameters. Therefore, the chosen time step $\Delta t = 0.05$, grid spacing $\Delta x = 0.02$, and advection speed $a = 1.0$ ensure the stability of the numerical scheme under the given conditions.

This indicate that the θ-scheme remains stable under the given conditions for the selected parameters. These results affirm that for appropriate choices of Δt, Δx, and the advection speed a, the numerical solution will accurately reflect the underlying physical behavior of the advection-diffusion problem without instabilities or violations of the maximum principle. We implemented the θ-scheme in **number5.py** using linear algebra routines to solve the matrix system and set $CFL = 1$. The result is shown in Figure 3a as we can see the amplitude error is zero. For comparison, We set $\epsilon = 0$ in Figure 3b. We can see the amplitude error is also zero but the plot converge at a bit faster rate as the diffusion term disappears.



(a) (b)

Figure 3: Numerical solution of the non-dimensional linear advection-diffusion equation obtained using the θ-scheme in **number5.py**. The solution is computed under the specified initial and boundary conditions, as outlined in equations (1b) and (1c). (a) using $\theta = 0.5$ and $\epsilon = 0.001$. (b) using $\theta = 0.5$ and $\epsilon = 0$

Task 6

Now, we investigate the case where $a(t) = 1$ and the initial condition is given by

$$u(x, 0) = (1-x)^4(1+x)\left(\sum_{k=0}^{3} b_k\phi_k(x) + C\right), \tag{81}$$

with the boundaries set at $L_p = -1$ and $L = 1$. The parameters for the simulation are chosen as $\epsilon = 10^{-3}$, $T = 1$ and $t \in [0, T]$. The Legendre polynomials used are defined as:

$$\begin{aligned}
\phi_0(x) &= 1,\\
\phi_1(x) &= x,\\
\phi_2(x) &= \frac{3}{2}x^2 - \frac{1}{2},\\
\phi_3(x) &= \frac{5}{2}x^3 - \frac{3}{2}x.
\end{aligned}$$

The coefficients b_k are randomly selected from a uniform distribution in the interval $(0, 1)$ for $k = 0, 1, 2, 3$. A constant $C \geq 0$ is determined numerically such that

$$\sum_{k=0}^{3} b_k\phi_k(x) + C \geq 0. \tag{82}$$

In our simulations, the random coefficients were reported as:

$$b_k = [0.37454012, 0.95071431, 0.73199394, 0.59865848].$$

For the numerical implementation, we utilize a while-loop with discrete time steps rather than fixed iterations. This approach ensures that the computed time profiles reflect the time-dependent behavior of the solution.

To validate our results, we explore various values of θ and CFL. The stability and potential violation of the maximum principle are examined through a CFL–θ parameter plot. Specifically, we select three combinations of CFL and θ corresponding to the cases of $\theta = 0$, $\frac{1}{2}$, and 1.

The function implementing the explicit calculates the solution iteratively and captures the results at set time intervals. The choice of θ significantly impacts stability; for instance, when $\theta = 1$, the CFL condition is satisfied, ensuring stability.

Figure 4 shows the results for different θ combinations concerning different stability criteria based on the maximum principle. The maximum principle guarantees that the numerical solution remains bounded under certain conditions. For the advection term, the maximum principle requires:

$$a\Delta t \leq \Delta x \quad \text{(advection stability condition).} \tag{83}$$

For the diffusion term, the maximum principle is satisfied if:

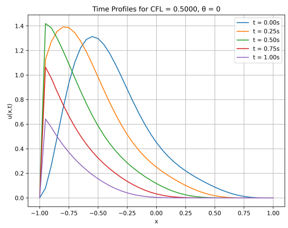
$$\Delta t \leq \frac{\Delta x^2}{2\epsilon} \quad \text{(diffusion stability condition).} \tag{84}$$

If these conditions are violated, the explicit scheme becomes unstable, and the solution may diverge or exhibit oscillations. But the fully implicit scheme remains stable under all condition.

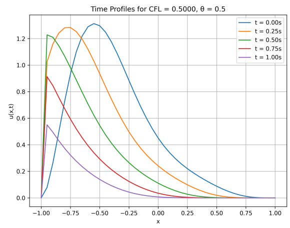
In the code of **number6.py**, we use three different time steps: $\Delta t = 0.025$, $\Delta t = 0.05$, and $\Delta t = 0.1$, along with a spatial step size $\Delta x = 0.05$. These values lead to the following conditions:

- For $\Delta t = 0.025$ and $CFL = 0.5$, which satisfies the stability condition for all scheme,
- for $\Delta t = 0.05$ and $CFL = 1$, which also satisfies the stability condition for all scheme, and
- for $\Delta t = 0.1$ and $CFL = 2$, which exceeds the stability threshold of $CFL \leq 1$ for explicit scheme.

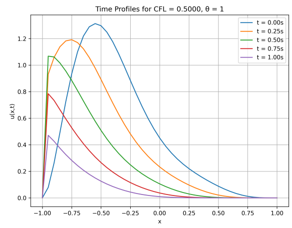
From figures 4, it can be observed that both the the Crank-Nicolson scheme and fully implicit scheme show a good stability and coverge to exact solution, but the implicit scheme show a fastest convergence rate compared to the other scheme. On the other hand, the explicit scheme only show stability when the threshold $CFL \leq 1$ is achieved, as we can see, instability begin to appear when the $CFL > 1$ as shown in figure (4g). In Figure (4f), where we used fully implicit scheme with $CFL = 1$, we can observe that the converge rate is faster than the other implicit scheme (In Figure (4c) and (4i)). Thus, setting $\theta = 1$ with $CFL = 1$ is not only practical but essential for achieving stable, accurate results in advection-diffusion simulations.



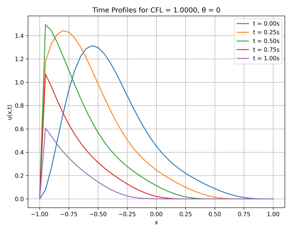
(a)
$CFL = 0.5$ and
$\theta = 0$



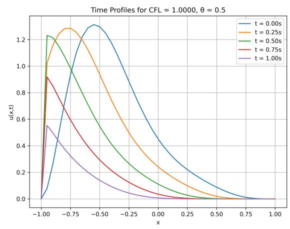
(b)
$CFL = 0.5$ and
$\theta = 0.5$



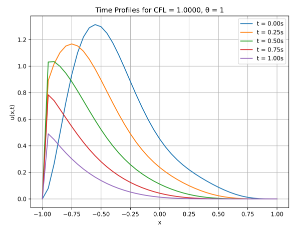
(c)
$CFL = 0.5$ and
$\theta = 1$



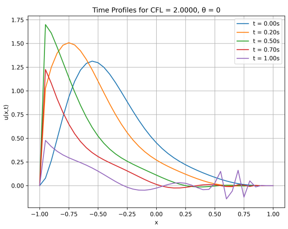
(d)
$CFL = 1$ and
$\theta = 0$



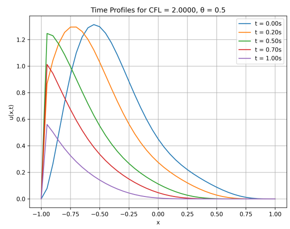
(e)
$CFL = 1$ and
$\theta = 0.5$



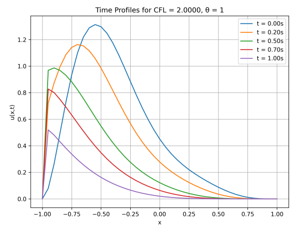
(f)
$CFL = 1$ and
$\theta = 1$



(g)
$CFL = 2$ and
$\theta = 0$



(h)
$CFL = 2$ and
$\theta = 0.5$



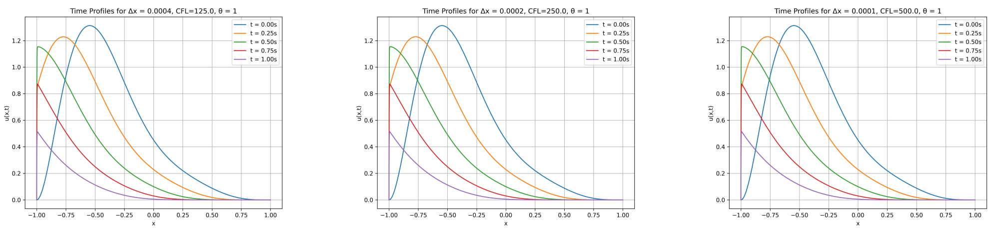
(i)
$CFL = 2$ and
$\theta = 1$

Figure 4: the results for different θ combinations concerning different stability criteria based on the maximum principle in **number6.py**.

Task 7

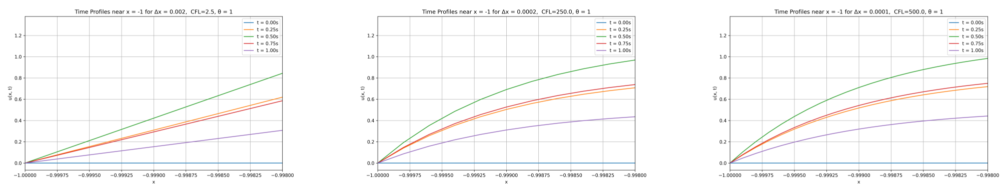
We consider the case where the advection term $a(t) = 1$ and the initial condition is defined in (64), where $L_p = -1$, $L = 1$, $\epsilon = 10^{-3}$, $\theta = 1$ and $T = 1$ with $t \in [0, T]$. In Figure 5, the simulation results from **number7.py** for three different spatial resolutions ($\Delta x = 0.002$, $\Delta x = 0.0002$, and $\Delta x = 0.0001$) are presented. We observe that the fully implicit scheme remains unconditionally stable, even with a high CFL value. This stability is a significant advantage of the fully implicit approach, allowing for larger spatial grids without causing numerical instabilities.

As Δt remains constant, refining the spatial grid (i.e., decreasing Δx) improves the resolution of the spatial domain. This increased resolution enables better convergence of the numerical solution to the expected physical behavior (without knowing exact solution). Figure 6 shows that for $\Delta x = 0.0002$, the solution converges clearly at $x = -0.998$, capturing the details of the solution more accurately. In contrast, for a coarser grid ($\Delta x = 0.02$), it is still far from convergence, illustrating the limitations of lower spatial resolutions.



(a) $\Delta x = 0.002$ (b) $\Delta x = 0.0002$ (c) $\Delta x = 0.0001$

Figure 5: Simulation results for different spatial resolutions: (a) $\Delta x = 0.002$, (b) $\Delta x = 0.0002$, and (c) $\Delta x = 0.0001$.



(a) $\Delta x = 0.002$ (b) $\Delta x = 0.0002$ (c) $\Delta x = 0.0001$

Figure 6: Zoomed-in view of Figure 5, focusing on the range near $x = -1$. The refinement in Δx leads to progressively smoother solutions.

To better understand convergence, we plot the solution at the final time iteration ($t = 1.0s$) for different values of Δx in Figure 7. Ideally, convergence implies that the numerical solution overlaps with the exact solution, resulting in a true error of zero. However, in practice, if the exact solution were known, there would be no need for a numerical solution. In our case, the exact solution is not provided and remains unknown. So, how can we confirm that the numerical solution has converged using a graphical approach?

Convergence can be verified by refining Δx to achieve finer spatial resolution. As we use smaller values of Δx, the numerical solution approaches the converged solution (essentially the exact solution), though it will never perfectly match it. Graphically, convergence can be observed by focusing on a zoomed-in region near $x = -1$. From Figure 7, we see that convergence begins at $\Delta x = 0.00015$ and improves further as Δx decreases to 0.0001. Within this range, the numerical solutions for successive values of $\Delta x = 0.00015$ nearly overlap, indicating that the estimated error (numerical error) is becoming smaller. Thus, we can confirm graphically that the numerical solution has effectively reached convergence.

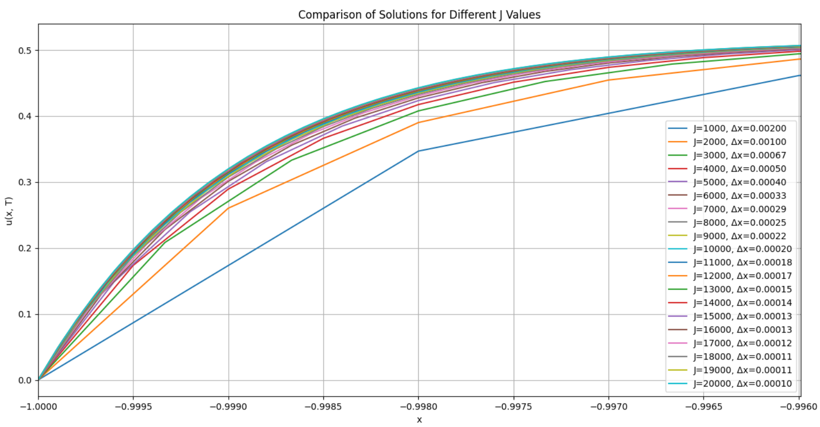
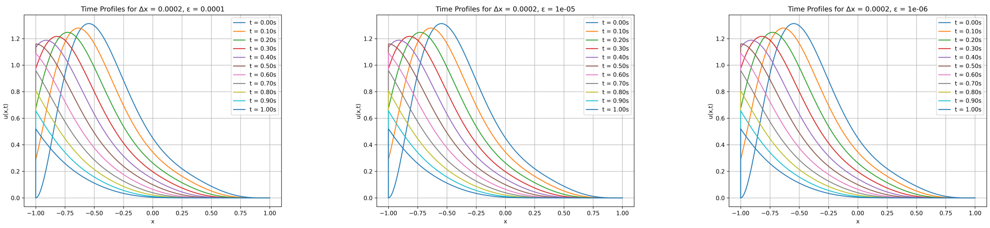


Figure 7: The comparison numerical solution with different Δx at zoomed-in region near $x = -1$ and at final time iteration $t = 1.0s$)

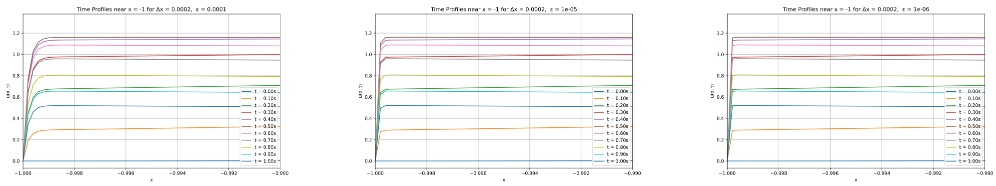
Task 8

Lastly, we explore the impact of smaller values of the diffusion coefficient ϵ in **number8.py**, specifically $\epsilon = 10^{-4}, 10^{-5}, 10^{-6}$, on the convergence behavior of the solution. The results from Figure 8 for different values of ϵ look very similar.



(a) $\epsilon = 10^{-4}$ (b) $\epsilon = 10^{-}5$ (c) $\epsilon = 10^{-6}$

Figure 8: the result of smaller values of the diffusion coefficient ϵ in **number8.py**.



(a) $\epsilon = 10^{-4}$ (b) $\epsilon = 10^{-5}$ (c) $\epsilon = 10^{-6}$

Figure 9: Zoomed-in view of Figure 7, focusing on the range $x = -1$ to $x = -0.99$ in **number8.py**.

In figure 9 the behavior of the linear advection-diffusion equation changes significantly as the diffusion coefficient ϵ decreases, transitioning from smooth to sharp profiles. For $\epsilon = 10^{-4}$, the solution remains relatively smooth because the diffusion term still plays a significant role in dissipating gradients. This smoothing effect slightly reduces the amplitude of the solution, as diffusion spreads the energy or concentration over a wider area. In contrast, at $\epsilon = 10^{-5}$, the solution becomes sharper, with steeper gradients forming due to the reduced influence of diffusion. The advection term begins to dominate, and the solution exhibits less smoothing, leading to more distinct fronts and a relatively stable amplitude. Finally, at $\epsilon = 10^{-6}$, the solution approaches near-discontinuous behavior, as diffusion becomes almost negligible. The advection term entirely dominates, preserving sharp fronts with minimal smoothing, which results in steep transitions or shock-like profiles.