Foundations of Fluids Numerical Exercises 1

Billy Hollis 201421513

October 2024

1 Question 1

A linear operator $\mathcal{L}$ is an operator for which $\mathcal{L}(u+v) = \mathcal{L}(u) + \mathcal{L}(v)$. We write equation (1a) in the form

$$\mathcal{L}u = 0 \tag{1}$$

where $\mathcal{L}$ is the linear differential operator

$$\mathcal{L} = \partial_t - a(t)\partial_x - \epsilon\partial_{xx} \tag{2}$$

Hence equation (1a) is linear. The term advection refers to the transport of a substance due to the bulk motion of a fluid and is demonstrated in the second term of equation (1a). The term diffusion refers to the random motion of molecules by which there is a net flow of matter from a region of high concentration to a region of low concentration. The other two terms in equation (1a) form the well known diffusion equation which describe this phenomenon. Hence, equation (1a) is a linear advection-diffusion equation.

2 Question 2

We will now prove equations (2.80) to (2.84) in M&M. We are given the following two equations

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

$$u_j^{n} = \left[u - \frac{1}{2}\Delta t u_t + \frac{1}{2}\left(\frac{1}{2}\Delta t\right)^2 u_{tt} - \frac{1}{6}\left(\frac{1}{2}\Delta t\right)^3 u_{ttt} + \cdots\right]_j^{n+\frac{1}{2}} \tag{4}$$

When taking the difference of these two equations, the odd terms will cancel and the even terms will be doubled which gives us

$$\delta_t u_j^{n+\frac{1}{2}} = u_j^{n+1} - u_j^n = \left[\Delta t u_t + \frac{1}{24}\left(\Delta t\right)^3 u_{ttt} + \cdots\right]_j^{n+\frac{1}{2}} \tag{5}$$

which is precisely equation (2.80). From equation (2.30), we have

$$\delta_x^2 u(x,t) = u_{xx}(\Delta x)^2 + \frac{1}{12}u_{xxxx}(\Delta x)^4 + \ldots \tag{6}$$

Evaluating this at position j and time step $n+1$ yields equation (2.81) which reads

$$\delta_x^2 u_j^{n+1} = \left[(\Delta x)^2 u_{xx} + \frac{1}{12}(\Delta x)^4 u_{xxxx} + \frac{2}{6!}(\Delta x)^6 u_{xxxxxx} + \ldots\right]_j^{n+1} \tag{7}$$

From M&M we have

$$\delta_x^2 u_j^{n+1} = \left[(\Delta x)^2 u_{xx} + \frac{1}{12}(\Delta x)^4 u_{xxxx} + \frac{2}{6!}(\Delta x)^6 u_{xxxxxx} + \dots\right]_j^{n+\frac{1}{2}} + \frac{1}{2}\Delta t\left[(\Delta x)^2 u_{xxt} + \frac{1}{12}(\Delta x)^4 u_{xxxxt} + \dots\right]_j^{n+\frac{1}{2}} + \frac{1}{2}\left(\frac{1}{2}\Delta t\right)^2 \left[(\Delta t)^2 u_{xxtt} + \dots\right]_j^{n+\frac{1}{2}} \quad (8)$$

We derive a similar expression for $\delta_x^2 u_j^n$ where we have

$$\delta_x^2 u_j^{n} = \left[(\Delta x)^2 u_{xx} + \frac{1}{12}(\Delta x)^4 u_{xxxx} + \frac{2}{6!}(\Delta x)^6 u_{xxxxxx} + \dots\right]_j^{n+\frac{1}{2}} - \frac{1}{2}\Delta t\left[(\Delta x)^2 u_{xxt} + \frac{1}{12}(\Delta x)^4 u_{xxxxt} + \dots\right]_j^{n+\frac{1}{2}} + \frac{1}{2}\left(\frac{1}{2}\Delta t\right)^2 \left[(\Delta t)^2 u_{xxtt} + \dots\right]_j^{n+\frac{1}{2}} \quad (9)$$

Equation (2.82) follows from some cumbersome re-arranging where we have

$$\begin{aligned}\theta\delta_x^2 u_j^{n+1} + (1-\theta)\delta_x^2 u_j^n = &\\ &\left[(\Delta x)^2 u_{xx} + \frac{1}{12}(\Delta x)^4 u_{xxxx} + \frac{2}{6!}(\Delta x)^6 u_{xxxxxx} + \dots\right]_j^{n+\frac{1}{2}}\\ &+ \left(\theta - \frac{1}{2}\right)\frac{1}{2}\Delta t\left[(\Delta x)^2 u_{xxt} + \frac{1}{12}(\Delta x)^4 u_{xxxxt} + \dots\right]_j^{n+\frac{1}{2}}\\ &+ \frac{1}{8}(\Delta t)^2(\Delta x)^2[u_{xxtt}]_j^{n+\frac{1}{2}} + \dots\end{aligned} \quad (10)$$

This is equation (2.82). Equation (2.83) is simply the definition of truncation error from earlier in the chapter and we have

$$T_j^{n+\frac{1}{2}} = \frac{\delta_t u_j^{n+\frac{1}{2}}}{\Delta t} - \frac{\theta\delta_x^2 u_j^{n+1} + (1-\theta)\delta_x^2 u_j^n}{(\Delta x)^2} \quad (11)$$

Substituting in the previous equations we obtained leads to equation (2.84) where

$$\begin{aligned}T_j^{n+\frac{1}{2}} = [u_t - u_{xx}]_j^{n+\frac{1}{2}} &+ \left[\left(\frac{1}{2} - \theta\right)\Delta t u_{xxt} - \frac{1}{12}(\Delta x)^2 u_{xxxx}\right]_j^{n+\frac{1}{2}}\\ &+ \left[\frac{1}{24}(\Delta t)^2 u_{ttt} - \frac{1}{8}(\Delta t)^2 u_{xxtt}\right]_j^{n+\frac{1}{2}}\\ &+ \left[\frac{1}{12}\left(\frac{1}{2} - \theta\right)\Delta t(\Delta x)^2 u_{xxxxt} - \frac{2}{6!}(\Delta x)^4 u_{xxxxxx}\right]_j^{n+\frac{1}{2}}\end{aligned} \quad (12)$$

3 Question 3

We incorporate the following differences

$$\frac{\partial u}{\partial t} \approx \frac{U_j^{n+1} - U_j^n}{\Delta t} \tag{13}$$

$$\frac{\partial u}{\partial x} \approx \frac{U_{j+1}^{n} - U_j^n}{\Delta x} \tag{14}$$

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{U_{j+1}^{n} - 2U_j^n + U_{j-1}^n}{(\Delta x)^2} \tag{15}$$

which allows us to form the explicit equation

$$U_j^{n+1} - U_j^n = a^n \gamma (U_{j+1}^n - U_j^n) + \epsilon\mu (U_{j+1}^n - 2U_j^n + U_{j-1}^n) \tag{16}$$

where $\gamma = \frac{\Delta t}{\Delta x}$ and $\mu = \frac{\Delta t}{(\Delta x)^2}$. Here, a^n denotes the value of $a(t)$ at $t = n\Delta t$, although we will choose $a(t)$ to be constant in the later simulations. At the time step $n+1$, we use a backwards difference for the spatial difference which is precisely equation (3). This gives an implicit equation

$$U_j^{n+1} - U_j^n = a^{n+1} \gamma (U_{j+1}^{n+1} - U_j^{n+1}) + \epsilon\mu (U_{j+1}^{n+1} - 2U_j^{n+1} + U_{j-1}^{n+1}) \tag{17}$$

We multiply equation (16) by $(1-\theta)$ then multiply equation (17) by θ and add them together. After some rearranging, we have the following θ-scheme

$$\begin{aligned}(1 + \theta(a^{n+1}\gamma + 2\epsilon\mu))U_j^{n+1} - \theta\epsilon\mu U_{j-1}^{n+1} - \theta(a^{n+1}\gamma + \epsilon\mu)U_{j+1}^{n+1} \\ = (1 + (\theta - 1)(a^n\gamma + 2\epsilon\mu))U_j^n + (1-\theta)\epsilon\mu U_{j-1}^n + (1-\theta)(a^n\gamma + \epsilon\mu)U_{j+1}^n\end{aligned} \tag{18}$$

This equation holds for $j = 1, 2, \ldots J-1$ and $n = 0, 1, 2, \ldots$ where $\Delta x = \frac{L - Lp}{J}$. We have the boundary conditions

$$U_0^n = U_J^n = 0 \quad \text{for} \quad n = 0, 1, 2, \ldots \tag{19}$$

To enforce these boundary conditions we add two extra rows to our matrix A in ThetaAdVec.py and RandomInitial.py. This is an effective but most likely way of ensuring compliance at the boundaries. We also have the initial condition

$$U_j^0 = u_0(x_j) \tag{20}$$

where $x_j = Lp + j\Delta x$.

4 Question 4

By running the code 1Dheat.py and altering the value of dt in line 8, we obtain similar figures to that presented in figure 2.2 of M&M, with the difference being that our plots show the evolution after certain times rather than a set number of iterations.

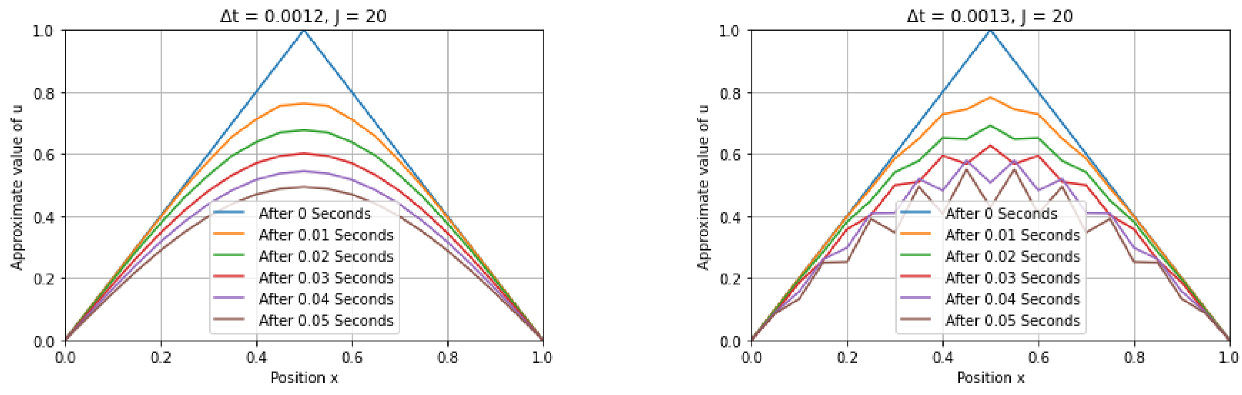


Figure 1: Replications of Figure 2.2 from M&M using 1Dheat.py

In the above figures, we have $J = 20$ and $\Delta x = 0.05$ and the explicit scheme exhibits the same behaviour that appears in M&M. From the Fourier analysis presented in Section 2.7 of M&M, we know that this explicit scheme is stable only when $\mu \leq \frac{1}{2}$. Hence, the above behaviour is explained by some simple calculations which show that $\mu = 0.48$ for $\Delta t = 0.0012$ and $\mu = 0.52$ for $\Delta t = 0.0013$, with our given spatial division.

We extend and implement the explicit scheme to equation (1a) in ExplicitAdVec.py where we have figures such as the following:

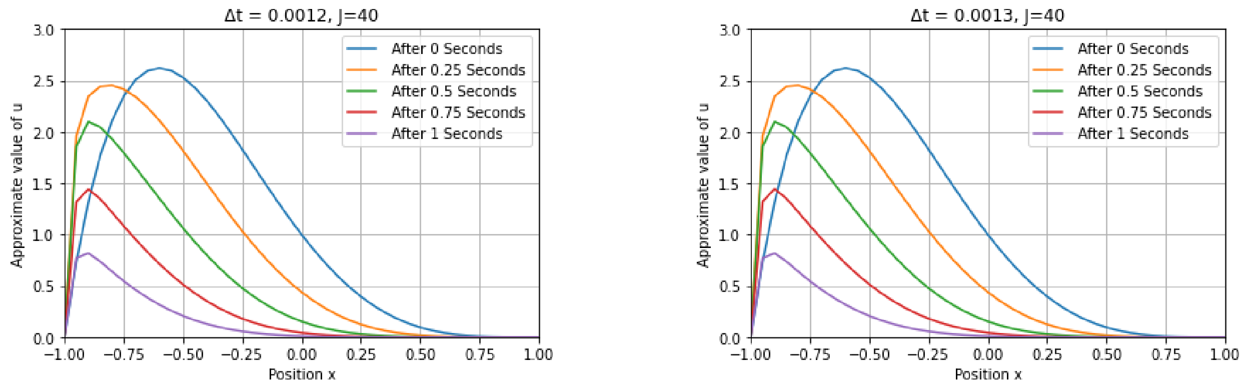


Figure 2: Example plots from the implementation of the explicit scheme for equation (1a) using ExplicitAdVec.py

For consistency, we have used $J = 40$ which gives $\Delta x = 0.05$ as before. From these plots, we see that this scheme does not require the bound $\mu \leq \frac{1}{2}$ which we had in the previous example.

5 Question 5

The θ-scheme is implemented in ThetaAdVec.py and returns a figure showing the evolution over time for chosen values of Δt and θ. These values can be suitably changed in lines 11 and 12 of ThetaAdVec.py. With $J = 400$, we have the following example output

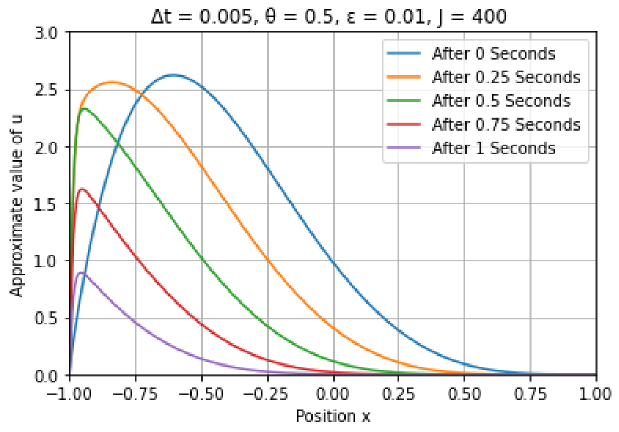


Figure 3: Example plot from the θ-scheme using ThetaAdVec.py. Note that from here and throughout, ThetaAdVec.py uses the initial condition (26). The remaining parameters are those used in the next section.

We will perform a Fourier analysis to examine the stability properties in very particular cases. We consider the case when $\epsilon = 0$ and $\theta = 0$ which is the explicit form of the pure advection equation. The remaining parameters are set as they are in the next section. With these considerations, equation (18) becomes

$$U_j^{n+1} = (1 - a^n\gamma)U_j^n + a^n\gamma U_{j+1}^n \tag{21}$$

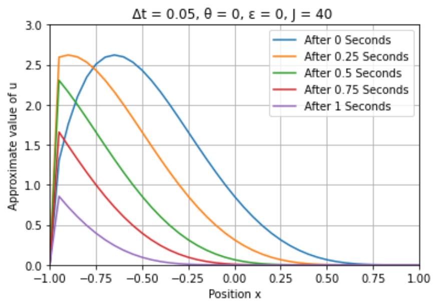
As usual, we seek a solution of the form

$$U_j^n = (\lambda)^n e^{ik(j\Delta x)} \tag{22}$$

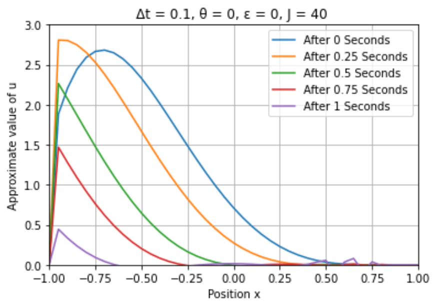
Following the working of M&M page 93, we obtain

$$|\lambda|^2 = 1 - 4a^n\gamma(1 - a^n\gamma)\sin^2\left(\frac{1}{2}k\Delta x\right) \tag{23}$$

In this case, we have stability so long as $0 \leq a^n\gamma \leq 1$. Indeed, with the parameters from the next section, we have the following plots



(a)



(b)

Figure 4: Implementation of ThetaAdVec.py with $\theta = 0$ and varying temporal resolutions to change the value of $a^n\gamma$.

In the second plot we have $a^n\gamma = 2$ and we see that instability appears. We keep $\epsilon = 1$ but now consider the case when $\theta = 1$ in which case equation (18) reduces to the fully implicit equation

$$(1 + a^{n+1}\gamma)U_j^{n+1} - a^{n+1}\gamma U_{j+1}^{n+1} = U_j^n \tag{24}$$

Upon seeking a solution of the form (22) and rearranging, we obtain the expression

$$\left|\frac{1}{\lambda}\right|^2 = 1 + 4a^{n+1}\gamma(1 + a^{n+1}\gamma)\sin^2\left(\frac{1}{2}k\Delta x\right) \tag{25}$$

For stability, the RHS of this equation must be greater than or equal to 1, which is indeed the case for all k provided that $a^{n+1}\gamma \geq 0$. We could perform similar analyses for other values of θ or we could set $a = 0$ for all time in which case we would have a pure diffusion equation. The mesh ratios obtained in doing so are presented in M&M so the details are omitted here.

6 Questions 6,7,8

We will now consider the model (1a)-(1c) with the initial condition

$$u(x,0) = (1-x)^4(1+x) \tag{26}$$

along with the parameters $a(t) = 1, L_p = -1, L = 1, \epsilon = 10^{-3}$ and $T = 1$ where $t \in [0, T]$. The θ-scheme for these parameters is implemented in ThetaAdvec.py and an example figure is shown below.

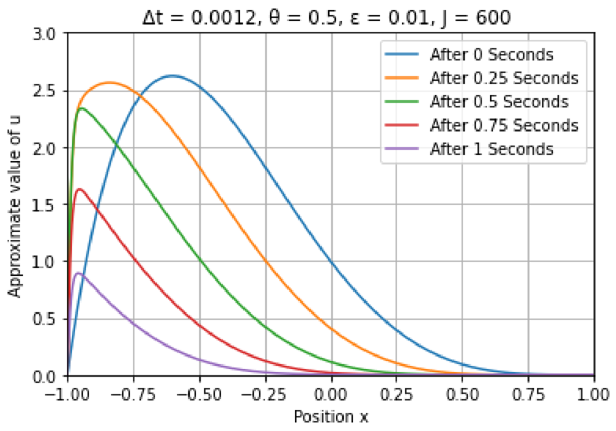


Figure 5: Implementation of θ-scheme using ThetaAdVec.py where the above parameters can be modified in lines 7,11,12 and 14.

As shown in the above derivations, setting $\theta = 0$ should yield the same outputs as the explicit scheme. We verify this by taking data from both ThetaAdVec.py with θ set to 0 in line 12 and ExplicitAdVec.py. Indeed, run both of these codes with $J = 20$ and observe the last printed array in the console on each occasion, which are equal.

Position j	θ-scheme with $\theta = 0$	Explicit Scheme
0	0.00000000e+00	0.00000000e+00
1	7.97231345e-01	7.97231345e-01
2	7.05738722e-01	7.05738722e-01
3	5.39696940e-01	5.39696940e-01
4	3.92136773e-01	3.92136773e-01
5	2.73383658e-01	2.73383658e-01
6	1.82667516e-01	1.82667516e-01
7	1.16607529e-01	1.16607529e-01
8	7.08309756e-02	7.08309756e-02
9	4.07480667e-02	4.07480667e-02
10	2.20820838e-02	2.20820838e-02
11	1.12042786e-02	1.12042786e-02
12	5.28668883e-03	5.28668883e-03
13	2.30213881e-03	2.30213881e-03
14	9.17207517e-04	9.17207517e-04
15	3.30907031e-04	3.30907031e-04
16	1.06621964e-04	1.06621964e-04
17	2.99931935e-05	2.99931935e-05
18	7.00708396e-06	7.00708396e-06
19	1.15261603e-06	1.15261603e-06
20	0.00000000e+00	0.00000000e+00

Now suppose that we have the more complex initial condition

$$u(x,0) = (1-x)^4(1+x)(\sum_{k=0}^{3} b_k\phi_k(x) + C) \tag{27}$$

Before implementing any of our finite difference schemes, we must obtain an array of random numbers which will be our values of b_k. We do this in a separate program as it is impractical to have new values every time we run one of our finite difference codes when we are trying to make comparisons. After running bkgenerator.py, input the array into line 54 of RandomInitial.py. This program then numerically finds the corresponding value of C before setting up the appropriate initial conditions and proceeding as usual. The draw we will use is

$$b_k = [0.26456307, 0.97212984, 0.35983848, 0.97892304] \tag{28}$$

for which we have $C = 1.3267999999998703$. The accuracy of C may be improved by reducing the implement in line 24.

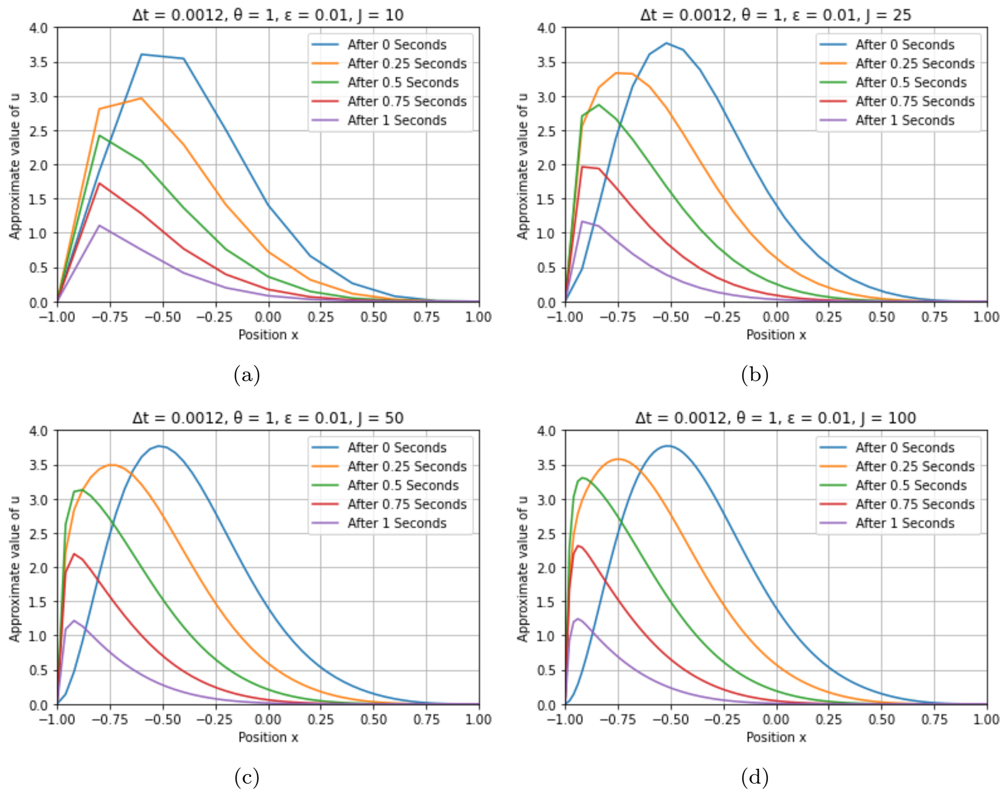
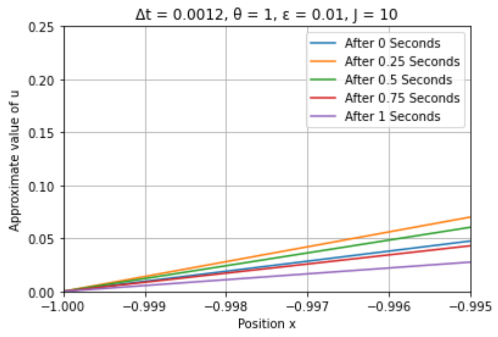
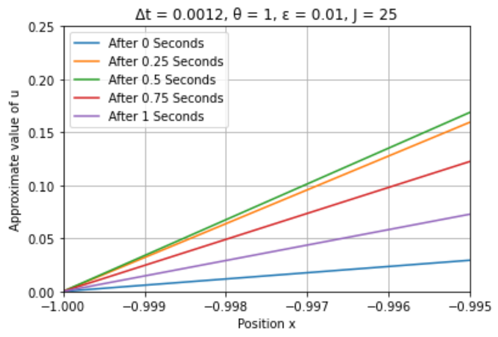


Figure 6: Implementation of θ-method for various spatial resolutions which indicate convergence to the true solution. All four plots come from RandomInitial.py with J changed accordingly.

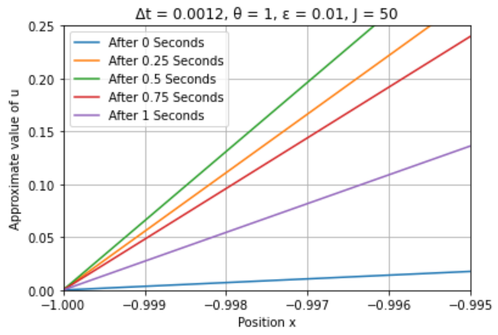
Having graphically confirmed convergence, we can now zoom in and examine the behaviour close to the point $x = -1$. This is done by altering the limits of the plot in RandomInitial.py. This gives the following plots which are analogous to Figure 6.



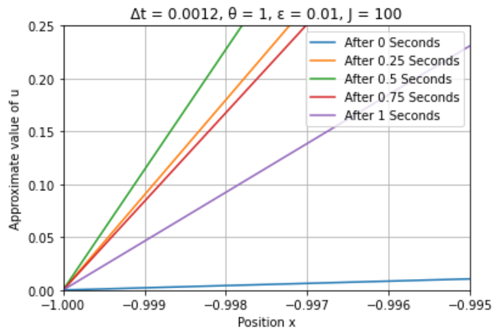
(a)



(b)



(c)

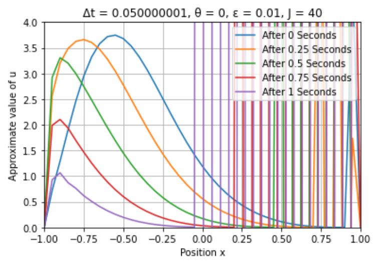


(d)

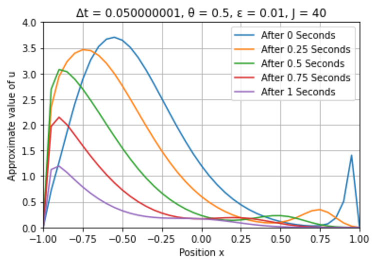
Figure 7: Same plots as Figure 6 zoomed in towards the point $x = -1$.

From Figure 7, we suggest the presence of a boundary layer at $x = -1$, hence the sharp velocity gradients.

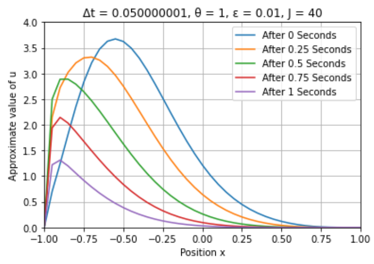
We have almost exclusively used $\theta = 1$ so we now present some plots in other cases, where the stability properties change.



(a)



(b)

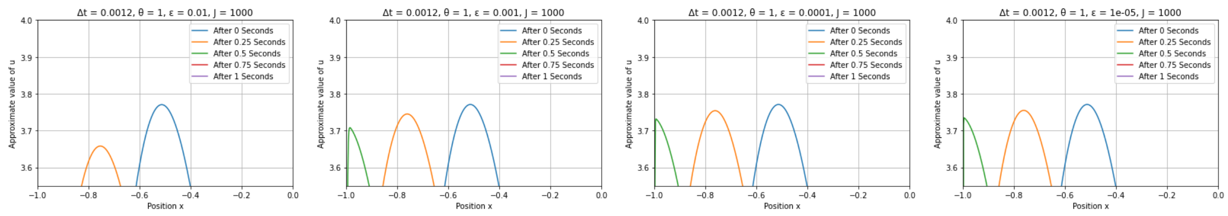


(c)

Figure 8: Comparison of stability for varying values of θ taken from RandomInitial.py

In Figure 8, we see that changing θ can have a drastic impact on the stability. Note we have chosen a time step which violates the CFL condition. Since ϵ is small, the mesh ratio restrictions that apply to the pure advection equation are more significant than those we would obtain from the pure diffusion equation. Of course, fully implicit schemes are typically unconditionally stable so would be unaffected. The details were provided in Chapter 5 with $\epsilon = 0$. For the other two cases, having a CFL number greater than 1 means that information propagates through more than one grid cell at each time step, leading to the observed results.

Our final task it to consider what happens if we choose increasingly small values for ϵ. We expect that decreasing ϵ will decrease the amount of diffusion in the system, meaning less decrease in the height of the wave crests when away from the boundary layer. This is indeed the case, as shown in the following figure, where we have used $J = 1000$ to provide greater detail..



(a) (b) (c) (d)

Figure 9

For the sake of interest, we could also investigate what happens for very small values of $a(t)$, where there would be less advection in the system. Suppose that $a(t) = 10^{-2}$, then we have the following figure.

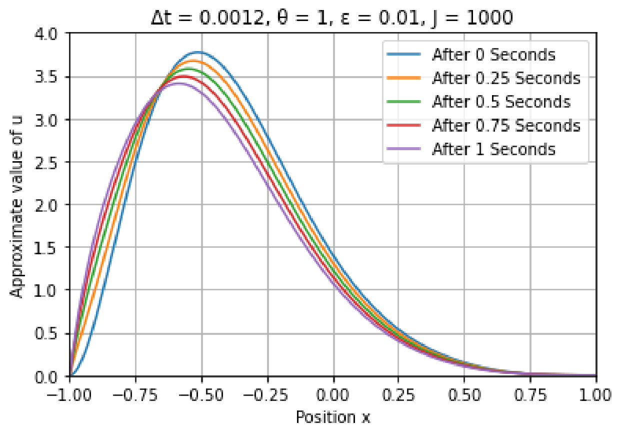


Figure 10: Plot using the θ-scheme where the advection term is greatly dampened

We note that, in many of the simulations, we have used varying spatial resolutions whilst keeping $\Delta t = 0.0012$. We have not given this much thought as $\theta = 1$ represents a fully implicit scheme and such schemes are typically undonditionally stable.