MATH 579, Solutions to Assignment 2,
by Alexandra Tcheng

1. Linear Advection Equation

Consider the linear advection equation:

$$\begin{aligned} u_t + u_x &= 0 \qquad x \in S^1 \quad t \in (0,T] && (1)\\ u(x,0) &= u_0(x) && (2) \end{aligned}$$

1) It is easy to verify that the solution to the PDE is: $u(x,t) = u_0(x-t)$, and therefore $u(x,1) = u_0(x-1)$.

2) We compute the GTE for each of the schemes, using the ideas presented in the solutions to Question 4) of Assignment 1:

a) Lax-Friedrich: The discretization of the advection equation using L-F gives:

$$\frac{u_j^{n+1} - \frac{1}{2}(u_{j+1}^n + u_{j-1}^n)}{\Delta t} = -\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} \tag{3}$$

We can verify:

$$\begin{aligned} \tau(\Delta x, \Delta t) &= \frac{u(x,t+\Delta t) - \frac{1}{2}(u(x+\Delta x,t) + u(x-\Delta x,t))}{\Delta t} + \frac{u(x+\Delta x,t) - u(x-\Delta x,t)}{2\Delta x} && (4)\\ &= O(\Delta t + (\Delta x)^2) && (5) \end{aligned}$$

and the CFL (=stability) condition is

$$\frac{\Delta t}{\Delta x} \leq 1 \qquad \Longleftrightarrow \qquad \Delta t = C\Delta x \tag{6}$$

Therefore (using the shortcut):

$$\mathrm{GTE}_{\mathrm{LF}} \leq \tau(\Delta x, \Delta x) = O(\Delta x + (\Delta x)^2) = O(\Delta x) \tag{7}$$

b) Lax-Wendroff: The discretization of the advection equation using L-W gives:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} + \left(\frac{\Delta t}{2}\right)\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2} \tag{8}$$

We can verify:

$$\begin{aligned} \tau(\Delta x, \Delta t) &= \frac{u(x,t+\Delta t) - u(x,t)}{\Delta t} + \frac{u(x+\Delta x,t) - u(x-\Delta x,t)}{2\Delta x} - \left(\frac{\Delta t}{2}\right)\frac{u(x+\Delta x,t) - 2u(x,t) + u(x-\Delta x,t)}{(\Delta x)^2}\\ &= O((\Delta t)^2 + (\Delta x)^2) && (9) \end{aligned}$$

and the CFL (=stability) condition is

$$\frac{\Delta t}{\Delta x} \leq 1 \qquad \Longleftrightarrow \qquad \Delta t = C\Delta x \tag{10}$$

Therefore (using the shortcut):

$$\mathrm{GTE}_{\mathrm{LW}} \leq \tau(\Delta x, \Delta x) = O((\Delta x)^2 + (\Delta x)^2) = O((\Delta x)^2) \tag{11}$$

c) RK3-TVD + WENO5: The one-step error for this scheme is:

$$\alpha(\Delta t, \Delta x) = O((\Delta t)^4 + (\Delta x)^6) \tag{12}$$

and the CFL (=stability) condition is

$$\frac{\Delta t}{\Delta x} \leq 1 \qquad \Longleftrightarrow \qquad \Delta t = C\Delta x \tag{13}$$

Therefore:

$$\mathrm{GTE}_{\mathrm{R3W5}} \leq \frac{1}{\Delta x}\ \alpha(\Delta x, \Delta x) = \frac{1}{\Delta x}\ O((\Delta x)^4 + (\Delta x)^6) = O((\Delta x)^3) \tag{14}$$

(See the appendices for details on how I implemented the schemes.)

Results for $u_0(x) = \cos x$:

a) Lax-Friedrich: The convergence results are shown on Figure (1). The grid resolution was varied from $dx = 2\pi/(2^5)$ to $dx = 2\pi/(2^{16})$. I chose not to go to lower resolutions to keep the computing time reasonable. Moreover, the expected orders of convergence were already visible at those resolutions. Since for each norm, all points seem to lie on the same line, they were all used in `polyfit` to compute the slope, which returned:
slope=0.9921 for the L^∞ norm,
slope=0.9945 for the L^1 norm,
slope=0.9930 for the L^2 norm.
The norms of the error behave as expected.

b) Lax-Wendroff: The convergence results are shown on Figure (2). As above, for the same reasons, the grid resolution was varied from $dx = 2\pi/(2^5)$ to $dx = 2\pi/(2^{11})$. All points were used in `polyfit` to compute the slope, which returned:
slope=2.0068 for the L^∞ norm,
slope=2.0103 for the L^1 norm,
slope=2.0087 for the L^2 norm.
The norms of the error behave as expected.

c) RK3-TVD + WENO5: The convergence results are shown on Figure (3). As above, for the same reasons, the grid resolution was varied from $dx = 2\pi/(2^4)$ to $dx = 2\pi/(2^{11})$. This time, not all points seem aligned. I only used the 4 left-most points to compute the slope in `polyfit` , which returned:
slope=3.0543 for the L^∞ norm,
slope=3.0545 for the L^1 norm,
slope=3.0541 for the L^2 norm.
The norms of the error behave as expected in the asymptotic behaviour. This example underlines the importance of letting the error go to $\epsilon_{\mathrm{machine}}$ whenever possible. Here, since the computer time became unreasonably long for high grid resolutions, this was not possible. However, as suggested in class, one way to observe the order of the GTE is to increase the final time (I picked $T = 5$) in order to let the error in Δt accumulate and become more significant than the error in Δx.

3) Results for $u_0(x) =$ box:

a) Lax-Friedrich: The convergence results are shown on Figure (4). The grid resolution was varied from $dx = 2\pi/(2^4)$ to $dx = 2\pi/(2^{16})$. I chose not to go to lower resolutions to keep the computing time reasonable. Moreover, the orders of convergence were already visible at those resolutions. Since not all points seem to lie on the same line, the 10 left-most points were used in `polyfit` to compute the slope of the L^1 and L^∞ norm of the error, and the 8 left-most points for the L^2 norm. This returned:
slope=0.0239 for the L^∞ norm,
slope=0.4949 for the L^1 norm,
slope=0.2528 for the L^2 norm.
Those norms do not agree with the ones expected from Part **2) a)**. However, this is not surprising, since in deriving the LTE and GTE, we used the Taylor series of the function $u(x,t)$. But the box function being only C^0, we cannot expect $u(x,t)$ to equal its Taylor series pointwise, and therefore the above reasoning does not apply.
Although this was not asked, it is interesting to watch the box evolve when advected with a Lax-Friedrich scheme. This is presented in a series of snapshots put together in Figure (5). Clearly, the scheme fails to capture the discontinuities of the function, and its diffusive nature eventually leads to large errors near the discontinuities. This explains why the L^∞ norm of the error cannot converge. This also explains why the L^1 norm, which is simply the area between the true solution and the numerical solution, improves with the grid resolution.

b) Lax-Wendroff: The convergence results are shown on Figure (6). As above, for the same reasons, the grid resolution was varied from $dx = 2\pi/(2^5)$ to $dx = 2\pi/(2^{16})$. All points were used in `polyfit` to compute the slope of the L^1 and L^2 norms of the errors. I did not compute the slope for the L^∞, since there clearly isn't convergence. This returned:

slope=0.5934 for the L^1 norm,

slope=0.3144 for the L^2 norm.

Again, this does not agree with the findings of Part **2) a)**, for the same reason as for Lax-Friedrich.

Again, it is interesting to watch the box evolve when advected by a Lax-Wendroff scheme. This is presented in a series of snapshots put together in Figure (7). This scheme also fails to capture the discontinuities of the function, and its dispersive nature eventually leads to large errors near the discontinuities, where wild oscillations appear. However, the overall shape of the box is (at least visually) better preserved than when advected with LF, which provides a simple explanation as to why the rate of convergence of the L^1 norm is slightly better.

c) RK3-TVD + WENO5: The convergence results are shown on Figure (8). As above, for the same reasons, the grid resolution was varied from $dx = 2\pi/(2^5)$ to $dx = 2\pi/(2^{12})$. The final time was increased to $T = 4$. Not all points seem aligned. I only used the 4 left-most points to compute the slope in `polyfit,` which returned:

slope=0.8345 for the L^1 norm,

slope=0.4291 for the L^2 norm.

We almost get first order convergence in the L^1 norm, which is much better than with the previous two schemes. However, convergence is slower in the L^2 norm, and absent in the L^∞ norm.

Again, it is interesting to watch the box evolve when advected by an RK3-TVD - WENO5 scheme. This is presented in a series of snapshots put together in Figure (9). Although this scheme also fails to capture the discontinuities of the function, the overall shape of the box is much better preserved than when advected with the previous two linear schemes. Although the L^∞ norm cannot converge because of the large errors near the discontinuities, those errors do not spread out and pollute the solution, which explains why the rate of convergence of the L^1 norm is almost 1.

2. Linear Advection Equation - Semi-Lagragian approach

1) Referring to Figure, the idea is to use various values of u_j^0 to build an interpolant $L(x)$, and then get u_2^1 from $u_2^1 = L(x_2 - c\Delta t)$. I'm going to use the formula for Lagrange interpolation in each case. I leave r unspecified until the end, where I then set $r = 0.5$ and $c = 1$.

Linear interpolation: Using the two neighbouring points (x_1, u_1^0) and (x_2, u_2^0):

$$L(x) = u_1^0\, l_1(x) + u_2^0\, l_2(x) = u_1^0 \frac{x - x_2}{x_1 - x_2} + u_2^0 \frac{x - x_1}{x_2 - x_1} = u_1^0 \frac{x - x_2}{-\Delta x} + u_2^0 \frac{x - x_1}{\Delta x} \tag{15}$$

$$\implies \quad u_2^1 \approx L(x_2 - c\Delta t) = u_1^0 \frac{-c\Delta t}{-\Delta x} + u_2^0 \frac{x_2 - x_1 - c\Delta t}{\Delta x} = u_1^0 \frac{c\Delta t}{\Delta x} + u_2^0 \left(1 - \frac{c\Delta t}{\Delta x}\right) \tag{16}$$

$$\implies \quad \frac{u_2^1 - u_2^0}{\Delta t} = c \frac{u_1^0 - u_2^0}{\Delta x} \qquad \text{Upwind method} \tag{17}$$

Or if we let $r = 0.5$ and $c = 1$: $u_2^1 - u_2^0 = -\frac{u_2^0 - u_1^0}{2}$.

Quadratic interpolation - Left: Using (x_0, u_0^0), (x_1, u_1^0) and (x_2, u_2^0):

$$L(x) = u_0^0\, l_0(x) + u_1^0\, l_1(x) + u_2^0\, l_2(x) \tag{18}$$

$$= u_0^0 \frac{(x - x_1)(x - x_2)}{2(\Delta x)^2} + u_1^0 \frac{(x - x_0)(x - x_2)}{-(\Delta x)^2} + u_2^0 \frac{(x - x_0)(x - x_1)}{2(\Delta x)^2} \tag{19}$$

$$\implies \quad u_2^1 \approx L(x_2 - c\Delta t) = u_2^0 + \frac{1}{2(\Delta x)^2}\left[c\Delta x \Delta t\left(-u_0^0 + 4u_1^0 - 3u_2^0\right) + c^2(\Delta t)^2\left(u_0^0 - 2u_1^0 + u_2^0\right)\right] \tag{20}$$

$$\implies \quad \frac{u_2^1 - u_2^0}{\Delta t} = -\frac{c}{2\Delta x}\left(u_0^0 - 4u_1^0 + 3u_2^0\right) + \frac{c^2 \Delta t}{2(\Delta x)^2}\left(u_0^0 - 2u_1^0 + u_2^0\right) \qquad \text{Beam} - \text{Warming method} \tag{21}$$

Or if we let $r = 0.5$ and $c = 1$: $u_2^1 - u_2^0 = -\frac{1}{4}\left(u_0^0 - 4u_1^0 + 3u_2^0\right) + \frac{1}{8}\left(u_0^0 - 2u_1^0 + u_2^0\right)$.

Quadratic interpolation - Right: Using (x_1, u_1^0), (x_2, u_2^0) and (x_3, u_3^0):

$$L(x) = u_1^0\, l_1(x) + u_2^0\, l_2(x) + u_3^0\, l_3(x) \tag{22}$$

$$= u_1^0 \frac{(x - x_2)(x - x_3)}{2(\Delta x)^2} + u_2^0 \frac{(x - x_1)(x - x_3)}{-(\Delta x)^2} + u_3^0 \frac{(x - x_1)(x - x_2)}{2(\Delta x)^2} \tag{23}$$

$$\implies \quad u_2^1 \approx L(x_2 - c\Delta t) = u_2^0 + \frac{c\Delta t}{2\Delta x}(u_1^0 - u_3^0) + \frac{c^2(\Delta t)^2}{2(\Delta x)^2}(u_3^0 - 2u_2^0 + u_1^0) \tag{24}$$

$$\implies \quad \frac{u_2^1 - u_2^0}{\Delta t} = \frac{c}{2\Delta x}(u_1^0 - u_3^0) + \frac{c^2 \Delta t}{2(\Delta x)^2}(u_3^0 - 2u_2^0 + u_1^0) \qquad \text{Lax} - \text{Wendroff method} \tag{25}$$

Or if we let $r = 0.5$ and $c = 1$: $u_2^1 - u_2^0 = -\frac{1}{4}(u_3^0 - u_1^0) + \frac{1}{8}(u_3^0 - 2u_2^0 + u_1^0)$.

2) I proceeded in the same way as in Part **1)** using 4 points, in order to get a cubic interpolant. Using (x_0, u_0^0), (x_1, u_1^0), (x_2, u_2^0) and (x_3, u_3^0):

$$L(x) = u_0^0\, l_0(x) + u_1^0\, l_1(x) + u_2^0\, l_2(x) + u_3^0\, l_3(x) \tag{26}$$

$$= u_0^0 \frac{(x - x_1)(x - x_2)(x - x_3)}{-6(\Delta x)^3} + u_1^0 \frac{(x - x_0)(x - x_2)(x - x_3)}{2(\Delta x)^3} \tag{27}$$

$$+ u_2^0 \frac{(x - x_0)(x - x_1)(x - x_3)}{-2(\Delta x)^3} + u_3^0 \frac{(x - x_0)(x - x_1)(x - x_2)}{6(\Delta x)^3} \tag{28}$$

$$\implies \quad u_2^1 \approx L(x_2 - c\Delta t) \quad = \quad u_2^0 + \frac{c\Delta t}{6\Delta x}(-u_0^0 + 6u_1^0 - 3u_2^0 - 2u_3^0) + \frac{c^2(\Delta t)^2}{6(\Delta x)^2}(3u_1^0 - 6u_2^0 + 3u_3^0) \tag{29}$$

$$+\frac{c^3(\Delta t)^3}{6(\Delta x)^3}(u_0^0 - 3u_1^0 + 3u_2^0 - u_3^0) \tag{30}$$

$$\implies \quad \frac{u_2^1 - u_2^0}{\Delta t} = \frac{c}{6\Delta x}(-u_0^0 + 6u_1^0 - 3u_2^0 - 2u_3^0) + \frac{c^2\Delta t}{6(\Delta x)^2}(3u_1^0 - 6u_2^0 + 3u_3^0) + \frac{c^3(\Delta t)^2}{6(\Delta x)^3}(u_0^0 - 3u_1^0 + 3u_2^0 - u_3^0) \tag{31}$$

Stability: Performing (quick) Von Neumann analysis gives the following:

$$G(k) = 1 + \frac{r}{6}(-e^{-2k\Delta x} + 6e^{-k\Delta x} - 3 - 2e^{k\Delta x}) + \frac{r^2}{6}(3e^{-k\Delta x} - 6 + 3e^{k\Delta x}) + \frac{r^3}{6}(e^{-2k\Delta x} - 3e^{-k\Delta x} + 3 - e^{k\Delta x}) \tag{32}$$

In order to find out for which values of r the scheme is stable, I plotted $|G(k)|$ for various values of r, ranging from $r = 0$ to $r = 1$ (since the CFL condition for the other schemes was $r \leq 1$, why not try this again!). The results shown on Figure (11) indicate that $r \leq 1$ ensures stability, since $|G(k)| \leq 1$ for all $k \in [0, 2\pi/(\Delta x)]$. (Note that there may be other possible intervals for r which ensure stability... However, I did not investigate this very much. All I know is that for r slightly greater than 1, the scheme is no longer stable, since $|G(k)| > 1$ for some values of k.)

Accuracy: We compute the LTE of the scheme:

$$\begin{aligned}
\tau(\Delta x, \Delta t) &= \frac{1}{\Delta t}\left(u(x, t+\Delta t) - u(x,t)\right) \\
&\quad - \Big[\frac{c}{6\Delta x}\left(-u(x-2\Delta x, t) + 6u(x-\Delta x, t) - 3u(x,t) - 2u(x+\Delta x, t)\right) \\
&\quad + \frac{c^2\Delta t}{6(\Delta x)^2}\left(3u(x-\Delta x, t) - 6u(x,t) + 3u(x+\Delta x, t)\right) \\
&\quad + \frac{c^3(\Delta t)^2}{6(\Delta x)^3}\left(u(x-2\Delta x, t) - 3u(x-\Delta x, t) + 3u(x,t) - u(x+\Delta x, t)\right)\Big] \qquad (33) \\
&= \frac{1}{\Delta t}\left((\Delta t)u_t + \frac{(\Delta t)^2}{2}u_{tt} + \frac{(\Delta t)^3}{6}u_{ttt} + \frac{(\Delta t)^4}{24}u_{tttt}\right) \\
&\quad - \Big[\frac{c}{6\Delta x}\left(-6(\Delta x)u_x - \frac{1}{2}(\Delta x)^4 u_{xxxx} + \frac{1}{5}(\Delta x)^5 u_{xxxxx} - \frac{1}{12}(\Delta x)^6 u_{xxxxxx}\right) \\
&\quad + \frac{c^2\Delta t}{6(\Delta x)^2}\left(3(\Delta x)^2 u_{xx} + \frac{1}{4}(\Delta x)^4 u_{xxxx} + \frac{1}{120}(\Delta x)^6 u_{xxxxxx}\right) \\
&\quad + \frac{c^3(\Delta t)^2}{6(\Delta x)^3}\left(-(\Delta x)^3 u_{xxx} + \frac{1}{2}(\Delta x)^4 u_{xxxx} - \frac{1}{40}(\Delta x)^5 u_{xxxxx} + \frac{1}{12}(\Delta x)^6 u_{xxxxxx}\right)\Big] \qquad (34) \\
&= \left(\underbrace{u_t + cu_x}_{=0}\right) + \frac{\Delta t}{2}\left(\underbrace{u_{tt} - c^2 u_{xx}}_{=0}\right) + \frac{(\Delta t)^2}{6}\left(\underbrace{u_{ttt} + c^3 u_{xxx}}_{=0}\right) \qquad (35) \\
&\quad + \frac{(\Delta t)^3}{24}u_{tttt} + \frac{c(\Delta x)^3}{12}\left(u_{xxxx} - \frac{c^2}{6}u_{xxxxxx}\right) \qquad (36) \\
&= O((\Delta t)^3 + (\Delta x)^3) \qquad (37)
\end{aligned}$$

Therefore, if we use the CFL condition $r \leq 1$, we expect the GTE to be $O((\Delta x)^3)$.

3) Results for $u_0(x) = \cos x$: The convergence results are shown on Figure (12). The grid resolution was varied from $dx = 2\pi/(2^5)$ to $dx = 2\pi/(2^{12})$. Since for each norm, all points seem to lie on the same line, they were all used in `polyfit` to compute the slope, which returned:

slope=3.0064 for the L^∞ norm,

slope=3.0113 for the L^1 norm,

slope=3.0083 for the L^2 norm.

The norms of the errors behave as expected.

Results for $u_0(x)$ =box$(x-t)$: The convergence results are shown on Figure (13). The grid resolution was varied from $dx = 2\pi/(2^4)$ to $dx = 2\pi/(2^{16})$. For the L^1 and L^2 norms, all points were used in `polyfit` to compute the slope. I didn't compute the slope for L^∞, since there clearly isn't convergence. This returned:

slope=0.7428 for the L^1 norm,

slope=0.3942 for the L^2 norm.

As for Lax-Friedrich and Lax-Wendroff, those results should not be surprising: we cannot expect a method based on the Taylor series of a function to work for a C^0 function.

Again, it is interesting to watch the box evolve when advected by a 3rd order scheme. This is presented in a series of snapshots put together in Figure (14). Although this scheme fails to capture the discontinuities of the function, just like the previous schemes, the oscillations created near the discontinuities don't seem to grow very fast. Moreover, the overall shape of the box is (at least visually) better preserved than when advected with LF or LW, which provides a simple explanation as to why the rate of convergence of the L^1 norm is better than with those two schemes.

Appendix

Question 1

For all schemes, I do the following:

- I build my spatial grid by letting $dx = 2\pi/(2^k)$ for some integer k, and define the vector

$$\begin{aligned} X &= \{0\ ,\ dx\ ,\ 2\cdot dx\ ,\ldots,\ (N-2)\cdot dx\ ,\ (N-1)\cdot dx = 2\pi\} && (38)\\ &= \{X(1), X(2), X(3), \ldots, X(N-1), X(N)\} && (39)\end{aligned}$$

which has N points. Since the boundary conditions are periodic, I will have to make the following identifications when necessary:

$$\begin{array}{rl} \vdots & X(1) = X(N) \\ X(-2) = X(N-3) & X(N+1) = X(2) \\ X(-1) = X(N-2) & X(N+2) = X(3) \\ X(0) = X(N-1) & X(N+3) = X(4) \\ & \vdots \end{array}$$

- To build my grid in time, I first fix $dt = r \cdot dx$ for some r satisfying the CFL condition. Then I define $M =$ceil(TF/dt) where TF is the final time. This way, I'm going to iterate until I reach the time $\tilde{T} = M * dt$. In general, $\tilde{T} \neq TF$, but at least, I am certain that all my time steps are of the same size dt. And when computing the GTE, the only important thing is to always compute it at the same time $\tilde{T}$ for all grid resolutions.

- Then, the scheme (LF, LW, or RK3-WENO5) is applied until $\tilde{T}$ is reached. The output is u_{num}. I record the grid resolution dx, calculate what the true solution is at time $\tilde{T}$, and compute the norm of the errors using the following formulae:

$$\begin{aligned} ||\text{Error}||_\infty &= \max_{i=1\ldots N} |u_{\text{num}}(i) - u_{\text{true}}(i)| && (40)\\ ||\text{Error}||_1 &= dx \cdot \sum_{i=1}^{N} |u_{\text{num}}(i) - u_{\text{true}}(i)| && (41)\\ ||\text{Error}||_2 &= \left(dx \cdot \sum_{i=1}^{N} |u_{\text{num}}(i) - u_{\text{true}}(i)|^2 \right)^{1/2} && (42)\end{aligned}$$

- I iterate over the resolution by varying k, in order to get a convergence plot of the various norms of the errors vs. the grid size.

a) Lax-Friedrich This scheme can be coded very efficiently using matrices. Notice that the scheme can be rewritten as:

$$u_j^{n+1} = \frac{1}{2}\left(\left(1 - \frac{\Delta t}{\Delta x}\right) u_{j+1}^n + \left(1 + \frac{\Delta t}{\Delta x}\right) u_{j-1}^n\right) = \frac{1}{2}\left((1-r)\, u_{j+1}^n + (1+r)\, u_{j-1}^n\right) \quad (43)$$

So that taking into account the periodic B.C.:

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ \vdots \\ u_{N-3} \\ u_{N-2} \\ u_{N-1} \\ u_N \end{pmatrix}^{n+1} = \frac{1}{2} \begin{pmatrix} 0 & 1+r & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 1-r & 0 \\ 1-r & 0 & 1+r & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-r & 0 & 1+r & 0 & \ldots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1-r & 0 & 1+r & \ldots & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \ldots & 1-r & 0 & 1+r & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 1-r & 0 & 1+r & 0 \\ 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 1-r & 0 & 1+r \\ 0 & 1+r & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 1-r & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ \vdots \\ u_{N-3} \\ u_{N-2} \\ u_{N-1} \\ u_N \end{pmatrix}^n$$

b) Lax-Wendroff Similarly, this can be implemented using:

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-2} \\ u_{N-1} \\ u_N \end{pmatrix}^{n+1} = \begin{pmatrix} 1-r^2 & -\frac{r}{2}(1-r) & 0 & 0 & 0\dots & 0 & 0 & \frac{r}{2}(1+r) & 0 \\ \frac{r}{2}(1+r) & 1-r^2 & -\frac{r}{2}(1-r) & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & \frac{r}{2}(1+r) & 1-r^2 & -\frac{r}{2}(1-r) & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \frac{r}{2}(1+r) & 1-r^2 & -\frac{r}{2}(1-r) & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & \frac{r}{2}(1+r) & 1-r^2 & -\frac{r}{2}(1-r) \\ 0 & -\frac{r}{2}(1-r) & 0 & 0 & \dots & 0 & 0 & \frac{r}{2}(1+r) & 1-r^2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-2} \\ u_{N-1} \\ u_N \end{pmatrix}^n$$

Note that a similar technique can be used for the 3rd order scheme presented in Question 2.

c) RK3-TVD - WENO 5 Let us first consider how RK3-TVD works. For each point on the grid, we need to compute two intermediate values as follows:

$$\psi_i^* = \mathrm{FE}(\psi_i^n) = \psi_i^n - \Delta t \cdot \mathrm{weno}(\psi_i^n) \tag{44}$$

$$\psi_i^{**} = \frac{3}{4}\psi_i^n + \frac{1}{4}\mathrm{FE}(\mathrm{FE}(\psi_i^n)) = \frac{3}{4}\psi_i^n + \frac{1}{4}\left(\psi_i^* - \Delta t \cdot \mathrm{weno}(\psi_i^*)\right) \tag{45}$$

$$\psi_i^{n+1} = \frac{1}{3}\psi_i^n + \frac{2}{3}\mathrm{FE}(\psi_i^{**}) = \frac{1}{3}\psi_i^n + \frac{2}{3}\left(\psi_i^{**} - \Delta t \cdot \mathrm{weno}(\psi_i^{**})\right) \tag{46}$$

Now here is the pseudo-code for WENO 5, when solving a conservative equation of the form:

$$u_t + \left(f(u)\right)_x = 0 \qquad \Longleftrightarrow \qquad u_t + f'(u)u_x = 0 \tag{47}$$

provided u and f are C^1. Then $c(u(x,t)) := f'(u)(x,t)$ acts as the speed function. Then at an interior point x_i:

- If $c(u_i^n) > 0$, define

$$(v_1, v_2, v_3, v_4, v_5) = \frac{1}{\Delta x}(u_{i-2} - u_{i-3}, u_{i-1} - u_{i-2}, u_i - u_{i-1}, u_{i+1} - u_i, u_{i+2} - u_{i+1}) \tag{48}$$

- If $c(u_i^n) \leq 0$, define

$$(v_1, v_2, v_3, v_4, v_5) = \frac{1}{\Delta x}(u_{i+3} - u_{i+2}, u_{i+2} - u_{i+1}, u_{i+1} - u_i, u_i - u_{i-1}, u_{i-1} - u_{i-2}) \tag{49}$$

Let $\epsilon = 10^{-6}$ (with no max taken!). Then, as given in class:

$$s_1 = \frac{13}{12}(v_1 - 2v_2 + v_3)^2 + \frac{1}{4}(v_1 - 4v_2 + 3v_3)^2 \tag{50}$$

$$s_2 = \frac{13}{12}(v_2 - 2v_3 + v_4)^2 + \frac{1}{4}(v_2 - v_4)^2 \tag{51}$$

$$s_3 = \frac{13}{12}(v_3 - 2v_4 + v_5)^2 + \frac{1}{4}(3v_3 - 4v_4 + v_5)^2) \tag{52}$$

$$(a_1, a_2, a_3) = \frac{1}{10}\left(\frac{1}{(\epsilon + s_1)^2}, \frac{6}{(\epsilon + s_2)^2}, \frac{3}{(\epsilon + s_3)^2}\right) \tag{53}$$

$$s_a = a_1 + a_2 + a_3 \tag{54}$$

$$(w_1, w_2, w_3) = \frac{1}{s_a}(a_1, a_2, a_3) \tag{55}$$

And finally:

$$\mathrm{weno}(u_i^n) = \frac{1}{6}\left(w_1(2v_1 - 7v_2 + 11v_3) + w_2(-v_2 + 5v_3 + 2v_4) + w_3(2v_3 + 5v_4 - v_5)\right) \tag{56}$$

Note that since in this assignment, $c = 1 > 0$, you could avoid checking the sign of the speed, and directly code the vector v as given by Equation (48). (That's what I did.)

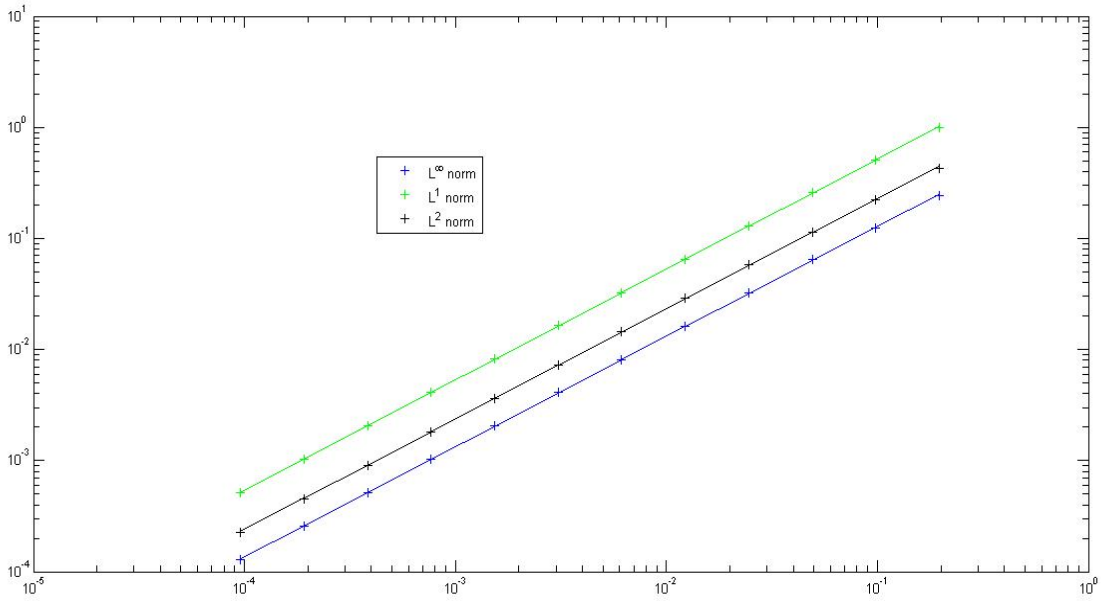

Figure 1: Convergence plot for $u(x,t) = \cos(x-t)$ in log-log scale, advected using Lax-Friedrich scheme. log(GTE) at $T \approx 1$ is plotted against $\log(h)$. For each norm, all points were used in the linear fit.

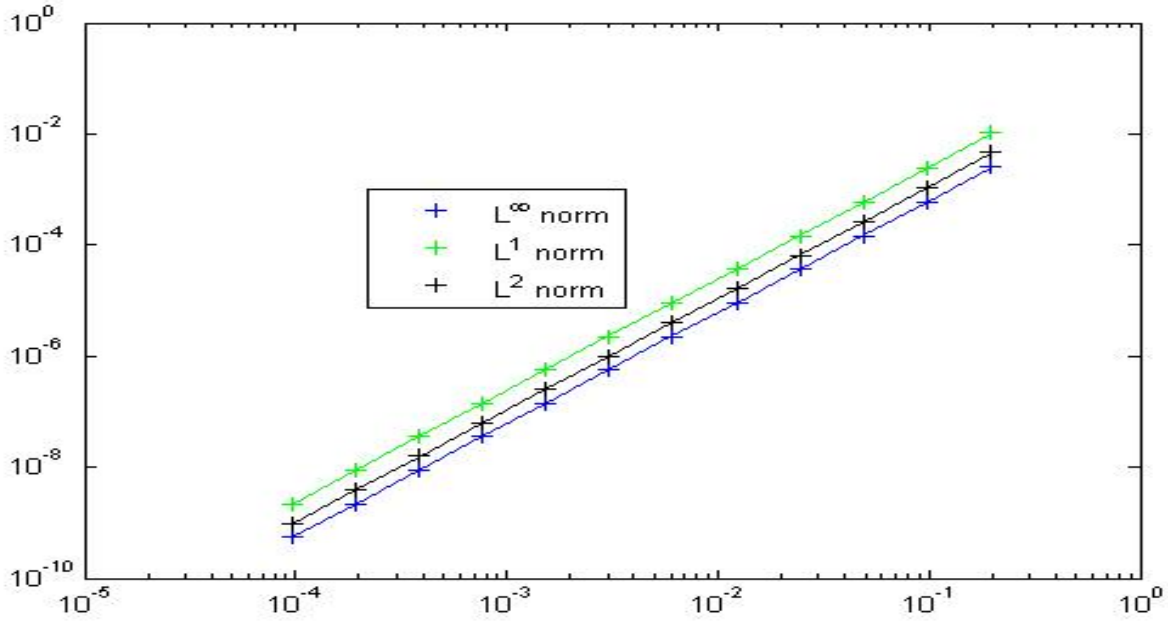

Figure 2: Convergence plot for $u(x,t) = \cos(x-t)$ in log-log scale, advected using Lax-Wendroff scheme. log(GTE) at $T \approx 1$ is plotted against $\log(h)$. For each norm, all points were used in the linear fit.

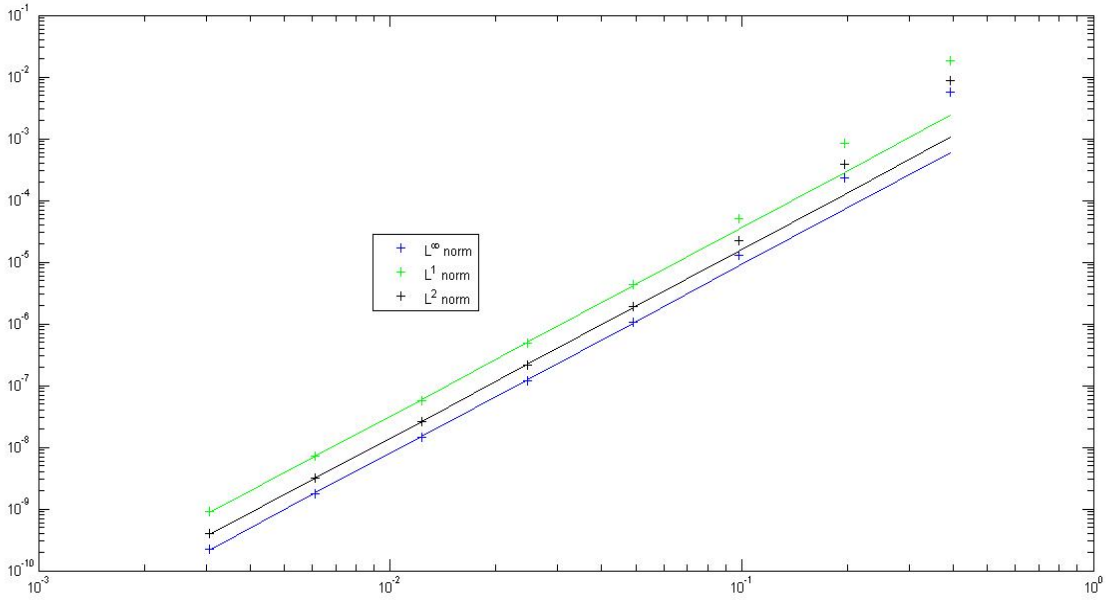

Figure 3: Convergence plot for $u(x,t) = \cos(x-t)$ in log-log scale, advected using RK3-TVD - WENO5 scheme. log(GTE) at $T \approx 5$ is plotted against $\log(h)$. For each norm, only the 4 left-most points were used in the linear fit.

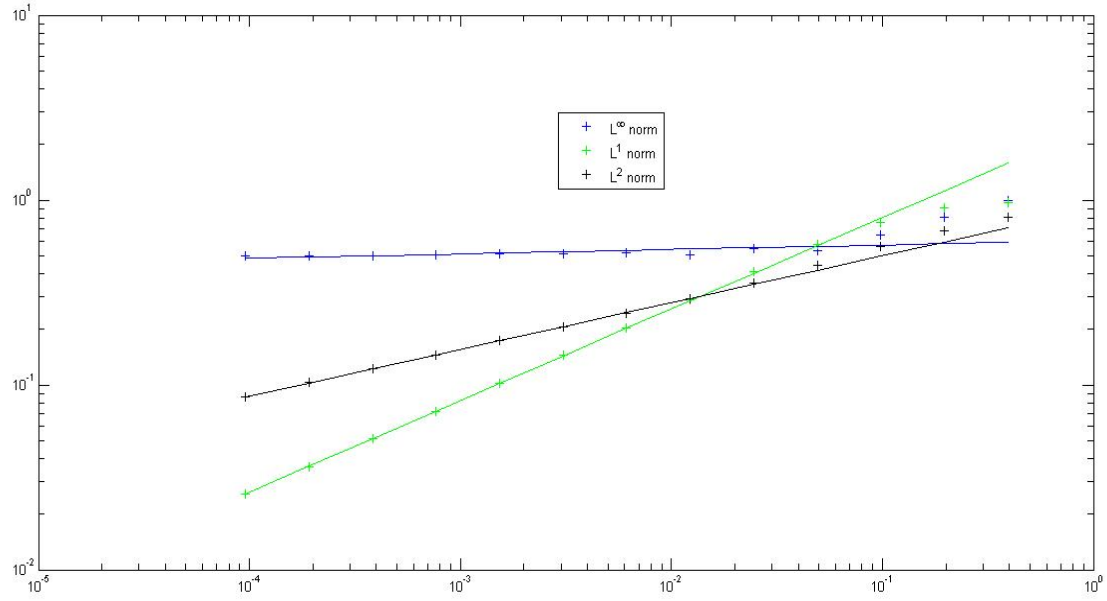

Figure 4: Convergence plot for $u(x,t) =$box$(x-t)$ in log-log scale, advected using Lax-Friedrich scheme. log(GTE) at $T \approx 1$ is plotted against $\log(h)$. The 10 left-most points were used in the fits of the L^1 and L^∞ norm of the error, and the 8 left-most points for the L^2 norm.

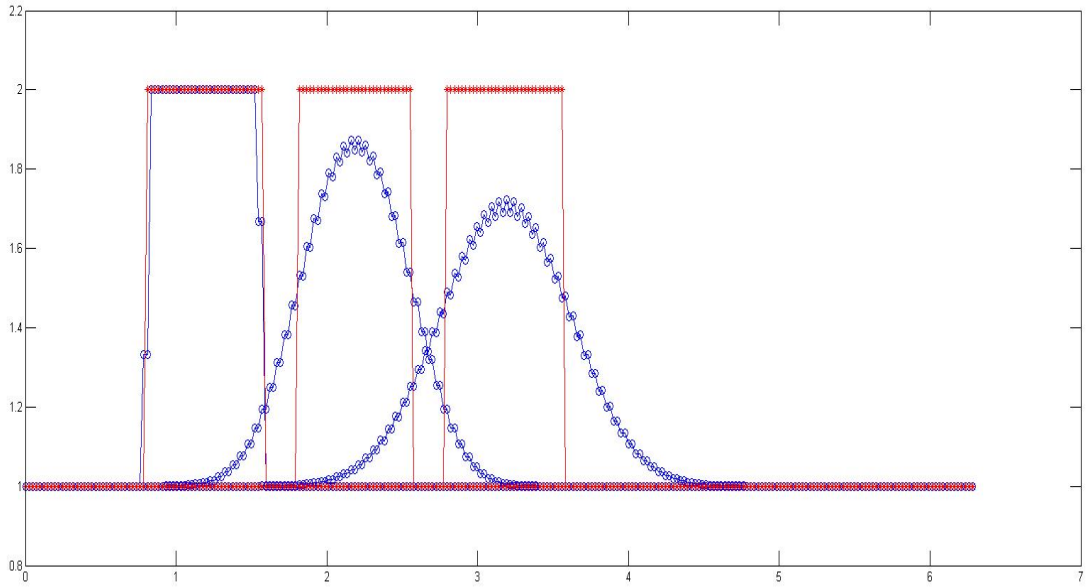

Figure 5: Snapshots of the box advected by a Lax-Friedrich scheme. The true solution is plotted in red, and the numerical solution is plotted in blue, at 3 different times (the box moves to the right). The grid resolution is $dx = 2\pi/(2^8)$.

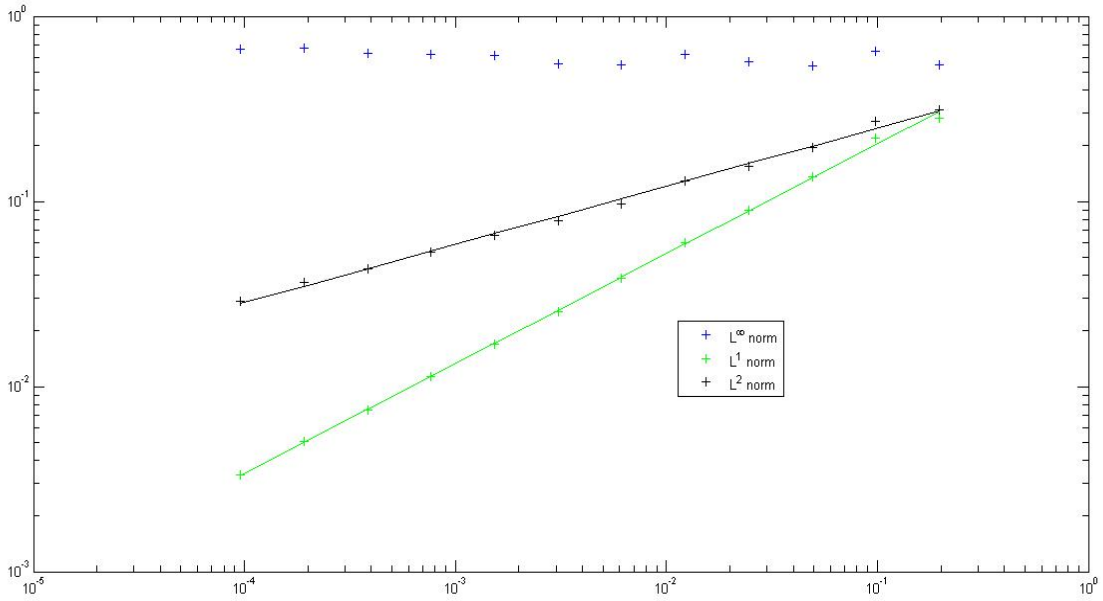

Figure 6: Convergence plot for $u(x,t) = \text{box}(x-t)$ in log-log scale, advected using Lax-Wendroff scheme. log(GTE) at $T \approx 1$ is plotted against $\log(h)$. For each norm, all points were used in the linear fit.

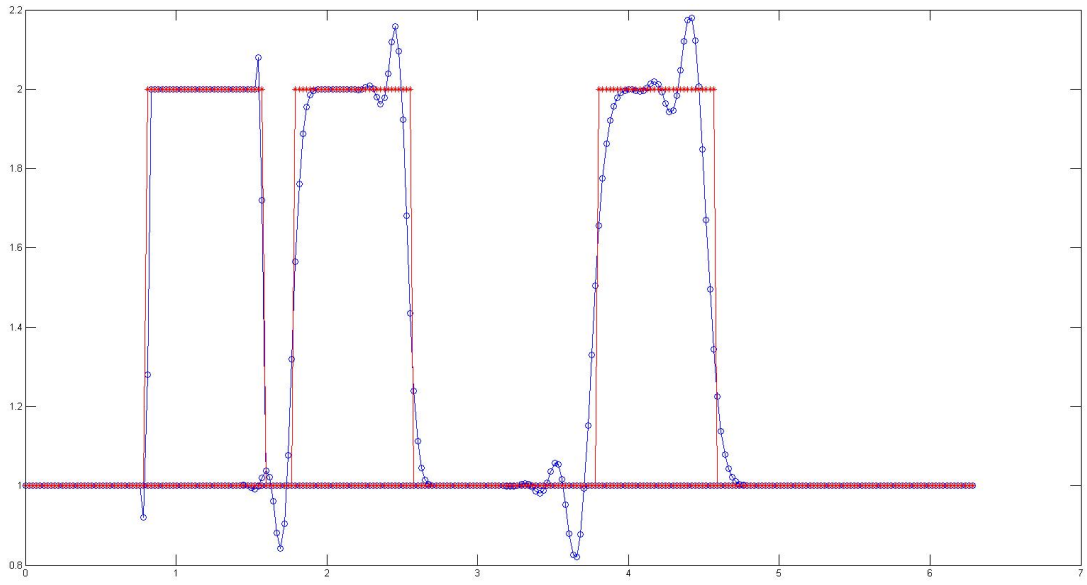

Figure 7: Snapshots of the box advected by a Lax-Wendroff scheme. The true solution is plotted in red, and the numerical solution is plotted in blue, at 3 different times (the box moves to the right). The grid resolution is $dx = 2\pi/(2^8)$.

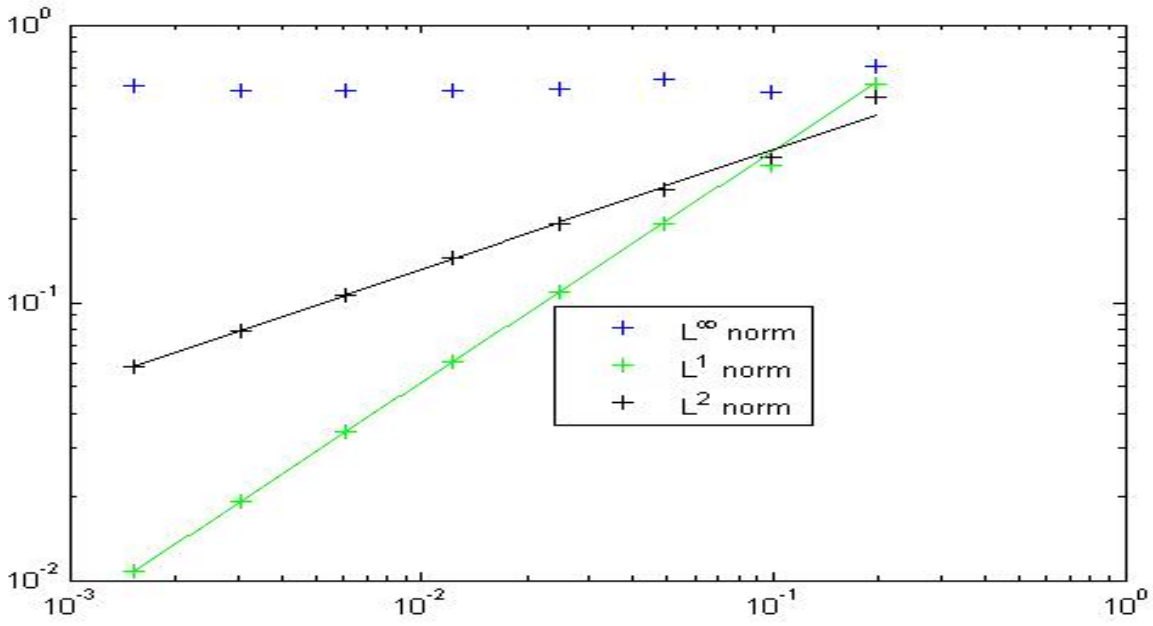

Figure 8: Convergence plot for $u(x,t) = \text{box}(x-t)$ in log-log scale, advected using RK3-TVD - WENO5 scheme. log(GTE) at $T \approx 5$ is plotted against $\log(h)$. For each norm, the 4 leftmost points were used in the linear fit.

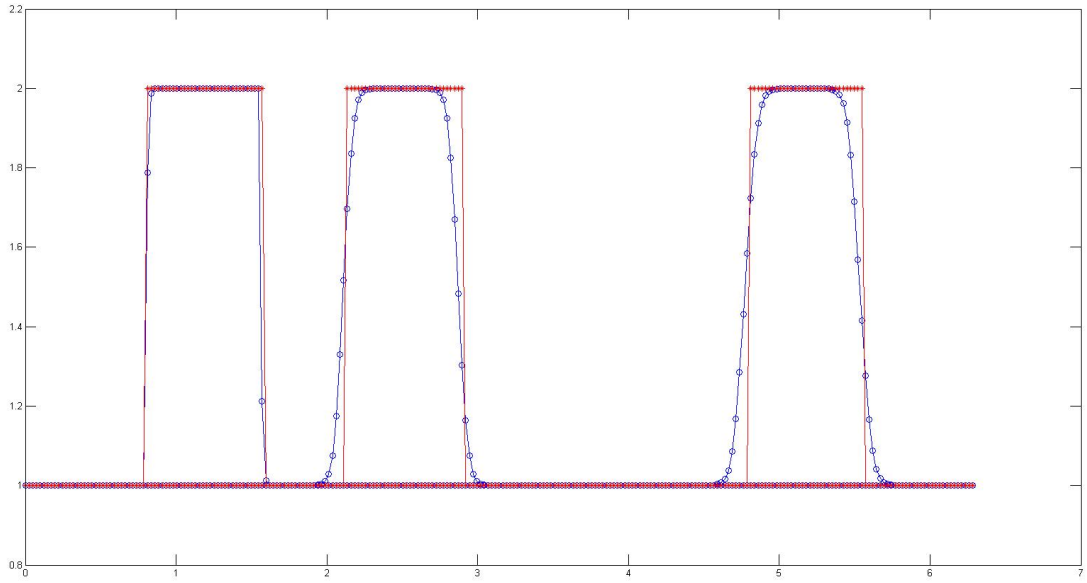

Figure 9: Snapshots of the box advected by an RK3-TVD - WENO5 scheme. The true solution is plotted in red, and the numerical solution is plotted in blue, at 3 different times (the box moves to the right). The grid resolution is $dx = 2\pi/(2^8)$.

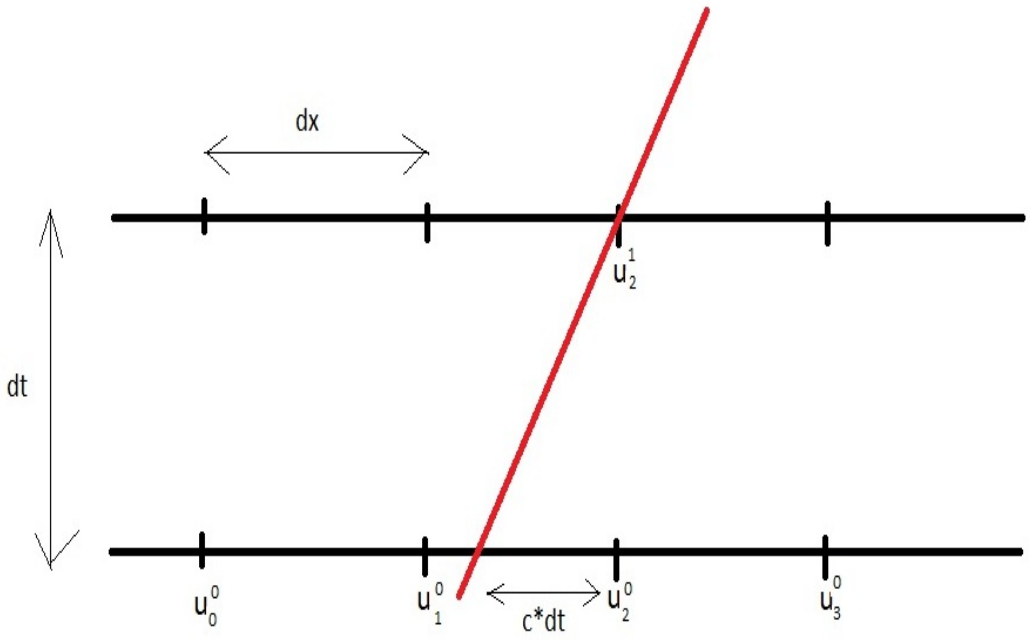

Figure 10: Points used in Question 2), Parts a) and b).

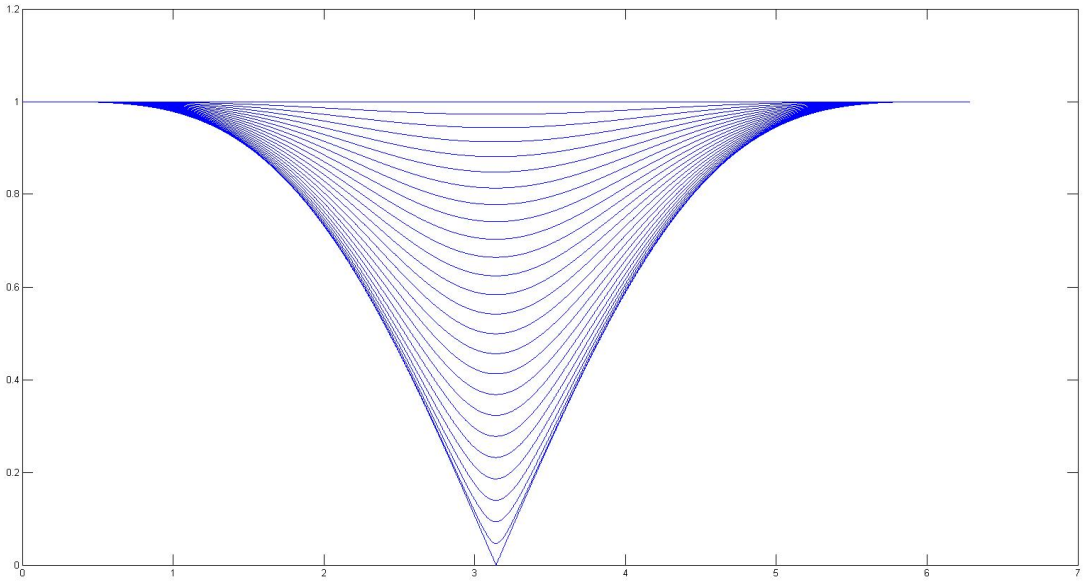

Figure 11: PLOT OF $|G(k)|$ AGAINST $k/(\Delta x)$ FOR $r = 0 : 0.02 : 1$. The top straight line corresponds to $r = 0$, and we have $|G(k)|(r_2) < |G(k)|(r_1)$ for $r_1 < r_2$, pointwise.

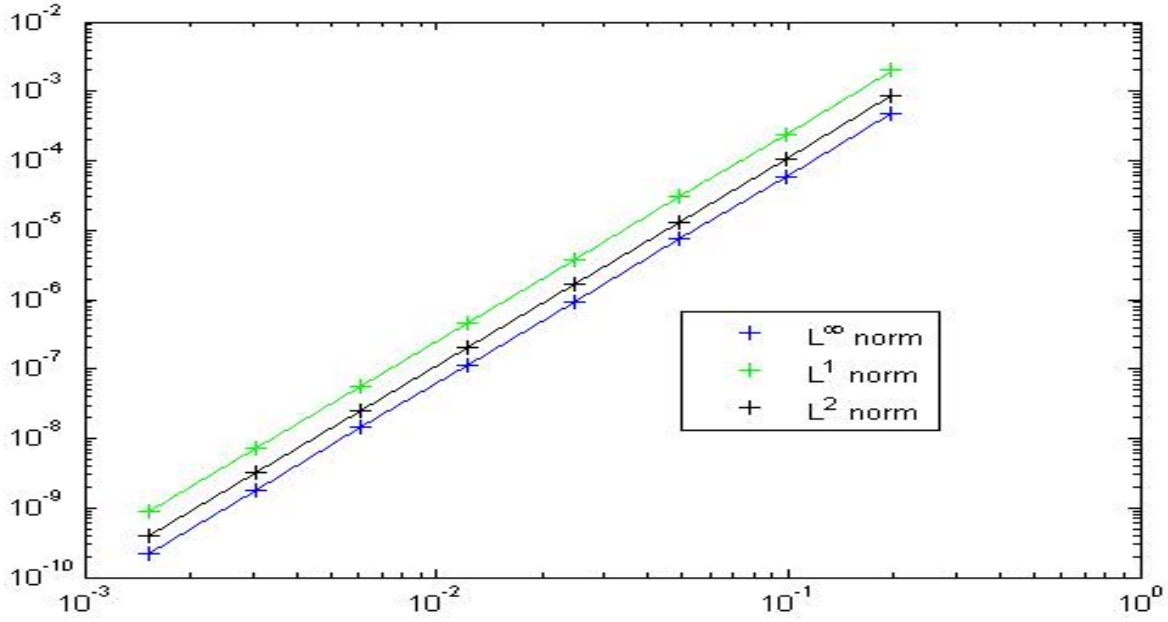

Figure 12: CONVERGENCE PLOT FOR $u(x,t) = \cos(x-t)$ IN LOG-LOG SCALE, ADVECTED USING A 3RD ORDER SCHEME. $\log(\text{GTE})$ at $T \approx 1$ is plotted against $\log(h)$. For each norm, all points were used in the linear fit.

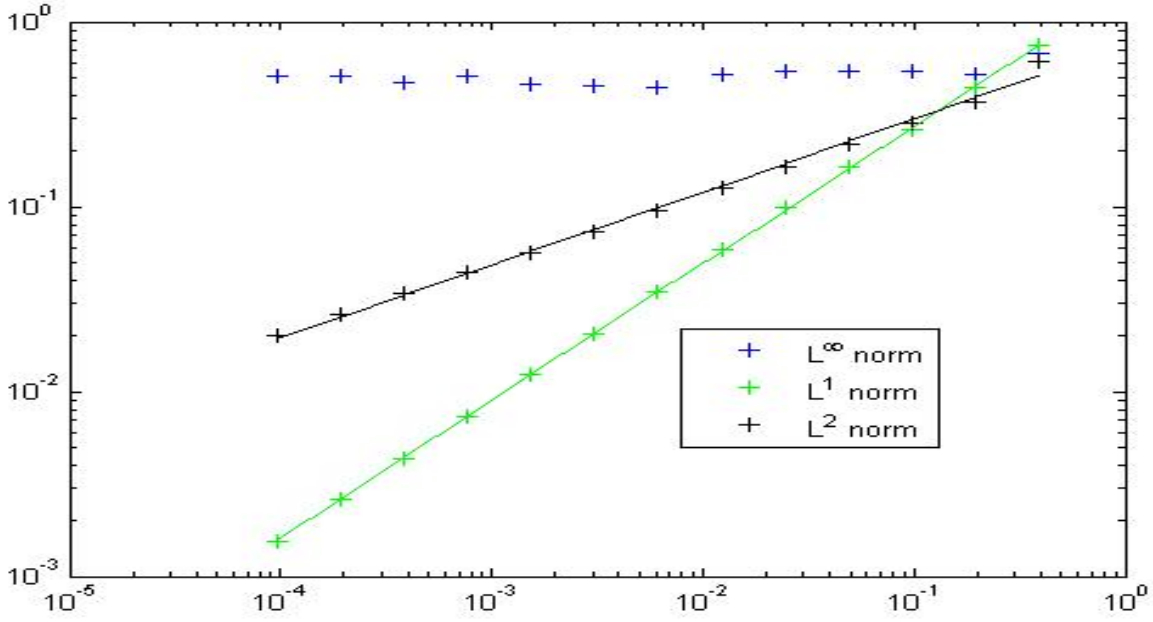

Figure 13: Convergence plot for $u(x,t) = \text{box}(x-t)$ in log-log scale, advected using a 3rd order scheme. log(GTE) at $T \approx 1$ is plotted against $\log(h)$. All points were used in the linear fit for the L^1 and the L^2 norm.

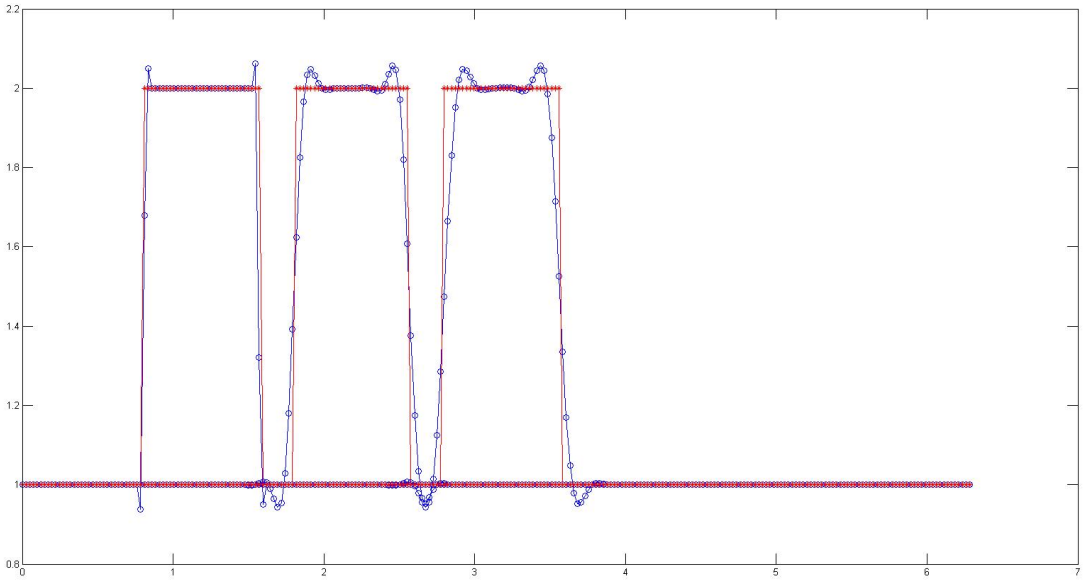

Figure 14: Snapshots of the box advected by a 3rd order scheme. The true solution is plotted in red, and the numerical solution is plotted in blue, at 3 different times (the box moves to the right). The grid resolution is $dx = 2\pi/(2^8)$.