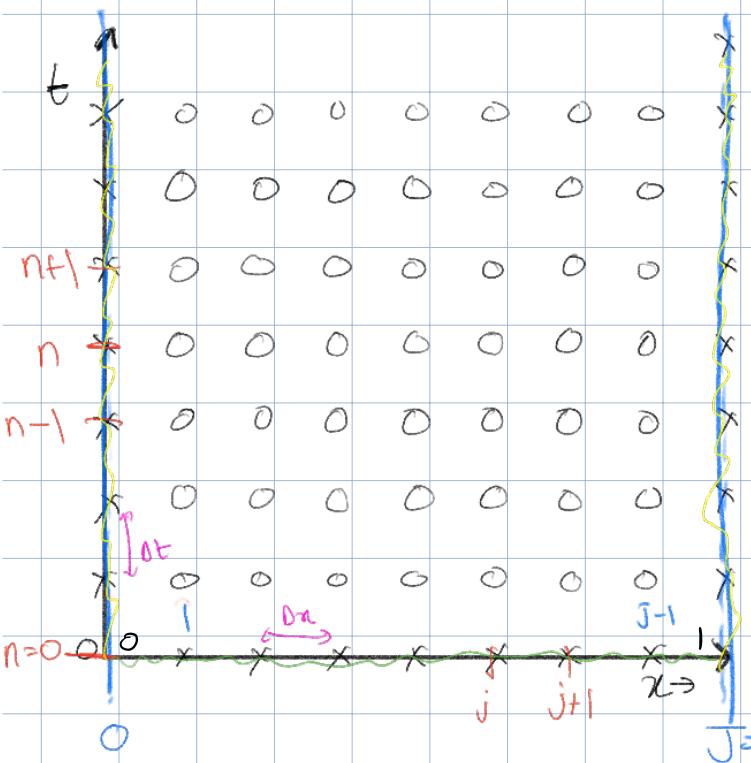
Question 4
(explicit scheme)



there are J+1 grid points

x = knowns from BCs

o = unknown vals

$x_j = j\Delta x \quad ; \quad j = 0, 1, 2 \ldots J$

$t_n = n\Delta t \quad ; \quad n = 0, 1 \ldots$

$u_j^n = u(x_j, t_n)$

Boundary conditions

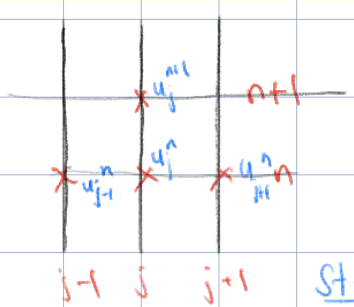
$u(x,0) = u_0(x)$

$u(0,t) = u(J,t) = 0$

for the explicit method

$$u_j^{n+1} = u_j^n + \nu\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right) \quad (*)$$

$\nu = \frac{\Delta t}{\Delta x^2}$



Stencil

for this scheme,

$J = 20 \qquad \Delta x = 0.05$

$\Delta t_1 = 0.0012 \qquad \Delta t_2 = 0.0013$

$$u^0(x) = \begin{cases} 2x & 0 \le x \le 1/2 \\ 2-2x & 1/2 \le x \le 1 \end{cases}$$

The code for this thus use a vector set up in Python, initially binding the vector U_0 using the given initial condition. Then, this vector is used to find the next timestep vector using (1).

ie $U_j^1 = U_j^0 + \mu\left(U_{j+1}^0 - 2U_j^0 + U_{j-1}^0\right)$

$\vdots$

$U_j^{50} = U_j^{49} + \mu\left(U_{j+1}^{49} - 2U_j^{49} + U_{j-1}^{49}\right)$

$\Rightarrow$ The code is written in file "gulana_94.py"

⇒ Figure 2.2 is proved and saved in the github folder as "gulana_q4_dt_0.002.jpeg" for dt=0.002 and "gulana_q4_dt_0.0013.jpeg" for dt=0.003

If the fourier analysis is preformed on the explicit scheme dealt with above, as shown in M&M 2.7, we find that the stability criteria is

$$N \leq \frac{1}{2}$$

and thus,

$$N = \frac{\Delta t}{\Delta x^2} \Rightarrow \frac{\Delta t}{\Delta x^2} \leq \frac{1}{2}$$

$$\Rightarrow \Delta t \leq \frac{1}{2}\Delta x^2 \quad (++)$$

if we calculate with $\frac{1}{2}\Delta x^2 = \frac{1}{2}\Delta(0.05)^2 = 1.25\times10^{-3}$

condition 1 $\Delta t = 0.0012$

$0.0012 \leq 0.00125$ (satisfied ++)

→ stable as shown in "gulana-q4_at_0.0012.jpg"

condition 2 $\Delta t = 0.0013$

$0.0013 > 0.0012$

↑ does not satisfy (++)

→ thus this solution is unstable, as shown the figure in file "gulana-q4_at-0.0013.jpg"

extending Explicit scheme to linear advection diffusion:

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} - \alpha(t) \frac{U_{j+1}^n - U_j^n}{\Delta x} + \left(\frac{\epsilon U_j^n - 2U_j^n + U_{j-1}^n}{\Delta x^2} \right) = 0$$

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = +\alpha(t) \frac{U_{j+1}^n - U_j^n}{\Delta x} + \epsilon \left(\frac{U_j^n - 2U_j^n + U_{j-1}^n}{\Delta x^2} \right)$$

$$U_j^{n+1} = U_j^n + \frac{\Delta t}{\Delta x} \alpha(t) U_{j+1}^n - U_j^n + \epsilon \frac{\Delta t}{\Delta x^2} \left(U_j^n - 2U_j^n + U_{j-1}^n \right)$$

↑ in my code I will take $\epsilon = 0.001$ and $\alpha(t) = 1$

The implementation is shown in code:

"exercise 1_q4_advection.py"

and the graph output is shown in

"guiana_q4_advection_diffusion.jpeg"