Question 5

The theta scheme was defined in eq. 3 as shown:

$$U_j^{n+1}\left(1+A\theta+2B\right)-U_{j-1}^{n+1}\left(B\theta\right)-U_{j+1}^{n+1}\left(A\theta+B\theta\right)$$

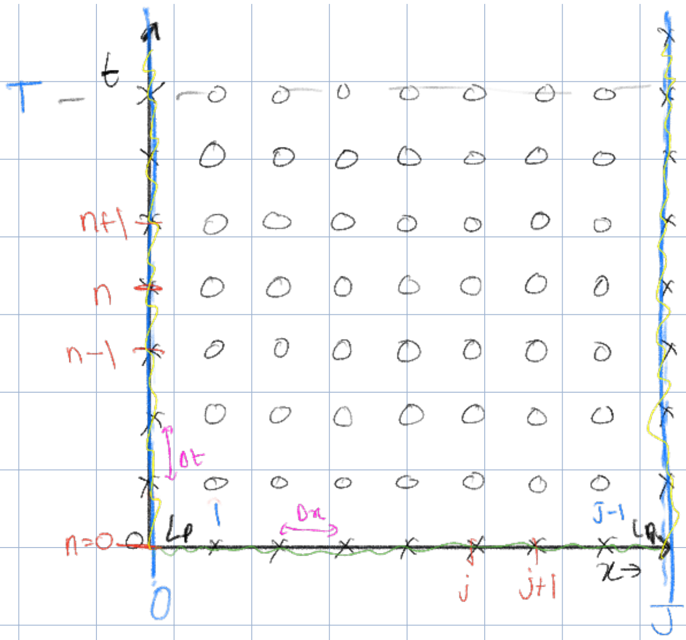
$$=$$

$$U_j^n\left(1-A(1-\theta)-2B(1-\theta)\right)+U_{j+1}^n\left(A(1-\theta)+B(1-\theta)\right)$$

$$+U_{j-1}^n\left(B(1-\theta)\right) \quad (\text{eq } 3)$$

where $A=\dfrac{\alpha(t)\Delta t}{\Delta x}$ $\quad B=\dfrac{\epsilon\,\Delta t}{\Delta x}$

$\Rightarrow$ The grid indexing is shown below



x = knowns from BCs and ICs

o = unknown vals

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

$t_n = n\Delta t$; $n = 0, 1, \dots$

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

Boundary conditions

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

$u(L_p,t) = u(L,t) = 0$

To implement this equation in code, a matrix set up in the form:

$$Ax = b$$

known terms

tridiagonal matrix

u_j^n terms

A can be taken from the LHS of the equation as the coefficients of the unknown values.

To deal with Boundary conditions, we set the first index in the Matrix A to be 1 and the rest to be zero as U_0^n is known to always be zero. We also set the last index in the matrix to be 1, with all other columns in the J+1 column of the matrix to be zero, as shown below:

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ (-B\Theta) & (1+A\Theta+2B) & (A\Theta+B\Theta) & 0 & \cdots & 0 \\ 0 & (-B\Theta) & (1+A\Theta+2B) & (A\Theta+B\Theta) & \cdots & 0 \\ 0 & 0 & -(B\Theta) & \cdots & & \\ & & \vdots & & & \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

← boundary conds

rows = (J+1)

columns = (J+1)

tridiagonal matrix continued.

← boundary

$$x = \begin{pmatrix} U_0^{n+1} \\ U_1^{n+1} \\ U_2^{n+1} \\ \vdots \\ \vdots \\ U_J^{n+1} \end{pmatrix}$$

rows = (J+1)

colums = (1)

The matrix b is computed using the RHS of eq. 3, with the first and last values also being 0 as this satisfies our boundary conditions.

$$b = \begin{pmatrix} 0 \\ U_0^n(B-(1-\Theta))+ \\ U_1^n(1+A\Theta+B\Theta) \\ +\; U_2^n((A+B)(1-\Theta)) \\ \vdots \\ 0 \end{pmatrix} \begin{matrix} \leftarrow \text{boundary conds} \\ \\ \\ \\ \\ \leftarrow \text{boundary cond.} \end{matrix}$$

$\Rightarrow$ U_j^0 can be calculated from the known initial conditions as $U(x,0) = U_0(x)$

$\Rightarrow$ the domain is divided into J points by $L - L_p / \Delta x$

⇒ The number of time steps is given by T.

⇒ once the U_j^0 vector is found, these values become known and can be used in the b vector. This can be used to find U_j^{n+1}.

⇒ linalg.solve is used to solve for x, and this vector is then used to update the value of b to find the U values at the next time step.

⇒ This is continued until the desired timesteps (T) have been reached.

The implementation of the theta scheme is shown in the file

"gulana-q5.py"

saved in my folder.

Fourier Analysis of the error (right biased upwind case only)

advection term: $u_t - \alpha(t) u_x = 0$

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \alpha(t) \left[\theta \left(\frac{u_{j+1}^{n+1} - u_j^{n+1}}{\Delta x} \right) + (1-\theta) \left(\frac{u_{j+1}^n - u_j^n}{\Delta x} \right) \right]$$

substitute $u_j^n = \lambda^n e^{ijk\Delta x}$ into ƺ

where $\mu = \Delta t / \Delta x$ and always greater than zero

$$u_j^{n+1} = u_j^n - \mu \alpha(t) \left[\theta \left(\lambda h^n e^{i(j+1)k\Delta x} - \lambda u_j^n \right] + (1-\theta) \left(h^n e^{i(j+1)k\Delta x} - h^n e^{ij\Delta x} \right) \right]$$

$\div\ u_j^n$

$$h = 1 - \mu \alpha(t) \left[\theta \left(h e^{ik\Delta x} - h \right) + (1-\theta) \left(e^{ik\Delta x} - 1 \right) \right]$$

$\therefore$

$$\lambda - 1 = -\mu\,\alpha(t)\left[\left(\theta\lambda + (1-\theta)\right)\left(e^{ik\Delta x} - 1\right)\right]$$

$$\lambda - 1 = -\mu\,\alpha(t)\left[\left(\theta\lambda + 1 - \theta\right)\left(\cos k\Delta x + i\sin k\Delta x - 1\right)\right]$$

$$\therefore\ \lambda = \frac{1 - \mu\,\alpha(t)(1-\theta)(\cos k\Delta x - 1 + i\sin k\Delta x)}{1 + \theta\,\mu(t)\left(\cos k\Delta x - 1 + i\sin k\Delta x\right)}$$

getting rid of imaginary terms:

$$|\lambda|^2 = \frac{\left(1 - \mu\,\alpha(t)(1-\theta)(\cos k\Delta x - 1)\right)^2 + \left(-\mu\alpha(t)(1-\theta)\sin k\Delta x\right)^2}{\left(1 + \theta\,\alpha(t)\,\mu\underbrace{(\cos k\Delta x - 1)}_{<0}\right)^2 + \left(\mu\,\theta\,\alpha(t)\sin k\Delta x\right)^2}$$

since $\mu > 0$ and $0 < \theta < 1$
and knowing that $(\cos k\Delta x - 1) < 0$,
the condition $\lambda \leq 1$ is only
valid if $\alpha(t)$ has a positive
sign for this right biased upwind
scheme.

$\Rightarrow$ Thus if $\alpha(t) > 0$ solution is stable

$\Rightarrow$ if $\alpha(t) < 0$ solution is unstable

$\Rightarrow$ It can thus be seen that advection term can thus add stability to system.

Diffusion term

$\Rightarrow$ The diffusion term will follow the same fourier analysis as in Morton and Mayers section 2.10, with an added ϵ term. such that:

$$\lambda = \frac{1 - 4(1-\theta)\mu \sin^2 \frac{1}{2} k \Delta x}{1 + 4\theta\mu \sin^2 k \Delta x} \quad \left(\begin{matrix} 2.77 \\ M\&M \end{matrix} \right)$$

becomes $\Rightarrow$

$$h = \frac{1 - 4(1-\theta)\epsilon\mu \sin^2 \tfrac{1}{2} k \Delta x}{1 + 4\theta\epsilon\mu \sin^2 \tfrac{1}{2} k \Delta x}$$

→ which is only unstable when
$h < -1$

$$\boxed{\mu\epsilon(1-2\theta) > \frac{1}{2}}$$

Stability condition for diffusion term.

if $\epsilon = 0$ the equation becomes simply the linear advection equation. the CFL condition states that the numerical domain of dependance must cover the analytical / physical domain of dependance. (D.O.D)

thus slope of numerical D.O.D $= \dfrac{\Delta t}{\Delta x}$

slope of physical D.O.D $= \dfrac{1}{\alpha(t)}$

for $\epsilon = 0$

CFL condition:

$$\left|\frac{\Delta t}{\Delta x}\right| \leq \left|\frac{1}{\alpha(t)}\right|$$

$$\therefore \quad \Delta t \leq \frac{\Delta x}{|\alpha(t)|}$$

$$\text{CFL number} = \frac{\Delta t}{\Delta x}|\alpha(t)|$$

thus if the CFL = 1

$$1 = \frac{\Delta t}{\Delta x}|\alpha(t)|$$

$\Rightarrow$ assuming $\alpha(t) = 1$, then for CFL = 1:

$$\boxed{\Delta t = \Delta x \quad \text{for} \quad \text{CFL} = 1}$$

The code is given parameters:

$L_p = 0$, $L = 1$, $dt = 0.05$, $dx = 0.05$, $\alpha(t) = 1$, $T = 50$, $\theta = 0$, $\epsilon = 0$

The graph plotted for this condition is saved in the folder as "gulana-q5.jpeg"

which shows pure advection with an error of zero as the peak height is the same as it travels upwind. The boundary conditions are the same as those implemented for Question 4.

* To reproduce the figure, run the file as it is.