CEE 6510 – Fall 2004 – Assignment No. 6

[1]. Stability analysis of upwind difference scheme for pure advection

Consider the linear advection equation, namely,

$$\frac{\partial c(x,t)}{\partial t} + u \frac{\partial c(x,t)}{\partial t} = 0$$

were u is a constant. Suppose that we want to solve this equation for $c(x,t)$ in the domain $0 < x < L$, $0 < t < t_{max}$ with boundary condition $c(0,t) = c_1(t)$, and initial condition $c(x,0) = c_0(x)$. The numerical solution will utilize the so-called *upwind finite difference approximations* for the derivatives, namely,

$$\frac{\partial c}{\partial x} \approx \frac{c_i^j - c_{i-1}^j}{\Delta x_i}, \text{ and } \frac{\partial c}{\partial t} \approx \frac{c_i^{j+1} - c_i^j}{\Delta t_j}.$$

(a) Perform a stability analysis (von Neumann's analysis) on the upwind difference scheme that results from using the finite difference approximations shown above in the linear advection equation.

(b) Obtain expression for the amplification and phase parameters, $R_1(r, Ls)$ and $R_2(r, Ls)$, respectively, where $L_s = \frac{L}{\Delta x}$ and $r = \frac{u \cdot \Delta t}{\Delta x}$ (NOTE: *r* is referred to as the Courant number of the computational grid).

(c) Using MATLAB, plot amplitude and phase portraits for Courant numbers $r = 0.25, 0.5, 0.75, 1.0, 2.0$, and for $0 < \frac{L}{\Delta x} < 30$.

(d) Based on the results shown in your amplitude and phase portraits discuss the stability of the upwind method.

[2]. Modeling pure advection with the upwind difference scheme

Consider a Gaussian distribution of contaminant given by the equation

$$c_0(x) = 10 \cdot \exp\left(-\frac{(x - x_o)^2}{2\sigma_x^2} \right),$$

with σ_x = *264 m*. Use this equation to provide the initial and boundary conditions for a pure-advection simulation in a channel that is *10-km* long (L = 10 km) if the simulation lasts for t_{max} = *160 minutes*. The water in the channel is moving with a constant velocity *u* =*0.5 m*/s. At time *t* = *0*, the curve is centered at $x_0 = 0$ (initial conditions), and, for *t*>*0*, the lagging half of the distribution enters the model as the boundary condition $c_1(t)$.

(a) Write a *Matlab* function to implement the numerical solution of pure advection using the upwind algorithm, i.e.,

$$\frac{\partial c}{\partial x} \approx \frac{c_i^j - c_{i-1}^j}{\Delta x_i}, \quad \frac{\partial c}{\partial t} \approx \frac{c_i^{j+1} - c_i^j}{\Delta t_j},$$

in the pure-advection equation:

$$\frac{\partial c}{\partial t} + u\frac{\partial c}{\partial t} = 0.$$

(b) Use a script to activate the function and perform the pure-advection simulation with Δx = *200 m*, and Δt = *100 s, 200 s, 300 s, 400 s,* and *800 s*. Let the script also plot solution curves c vs. t for the different values of Δt at t = *800 s, 1600 s, 2400 s*, and so on, until reaching t_{max}. Also, plot the exact solution (i.e., the advection of the Gaussian curve at a constant speed u, for the times indicated above, in the same graphs.

(c) Interpret your results in the context of your amplitude and phase portraits of problem [1]. Could you have predicted the error in the numerical solution using those portraits? Why should the solutions with Δt = *100 s* and Δt = *300 s,* not have the same error at t_{max} = *160 min ?* If you had to use this scheme for a practical problem, how would you attempt to minimize the error?

[3] Modeling pure advection with characteristics and the Holly-Preissmann scheme

Consider a Gaussian distribution of contaminant given by the equation

$$c_0(x) = 10 \cdot \exp\left(-\frac{(x-x_o)^2}{2\sigma_x^2}\right),$$

with σ_x = *264 m*. Use this equation to provide the initial and boundary conditions for a pure-advection simulation in a channel that is *10-km* long (L = *10 km*) if the simulation lasts for t_{max} = *160 minutes*. The water in the channel is moving with a constant velocity u =*0.5 m*/s. At time t = *0*, the curve is centered at x_0 = *0* (initial conditions), and, for $t>0$, the lagging half of the distribution enters the model as the boundary condition $c_1(t)$.

(a) Write a *Matlab* function to implement the numerical solution of pure advection using the Holly-Preissmann algorithm.

(b) Use a script to perform to activate the function and perform the pure-advection simulation with Δx = *200 m*, and Δt = *100 s, 200 s, 300 s, 400 s,* and *800 s*. Let the script also plot solution curves c vs. t for the different values of Δt at t = *800 s, 1600 s, 2400 s*, and so on, until reaching t_{max}. Also, plot the exact solution (i.e., the advection of the Gaussian curve at a constant speed u) for the times indicated above in the same graphs.

(c) Repeat one of the calculations in (b), but this time set = *0* at time t = *0*. Discuss the effect of this inconsistency in the results.

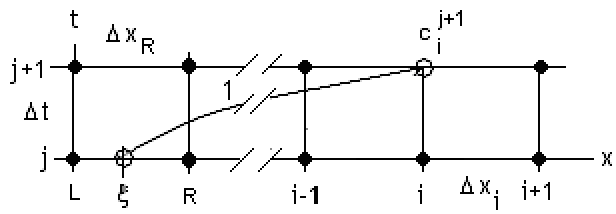
[4] Advection with decay – implementation of characteristics method

Adding a decay term to the equation of pure advection results in the equation

$$\frac{\partial c(x,t)}{\partial t} + u\frac{\partial c(x,t)}{\partial t} + \frac{c(x,t)}{T} = 0,$$

were T is a decay time constant.

(a) Write the corresponding characteristic equation along the characteristic line $dx/dt = u(x,t)$, and integrate the characteristic equation in the grid shown below, i.e., find c_i^{j+1}, given c_ξ.



(b) By taking the derivative of the previous equation with respect to x, and defining $cx = \partial c/\partial x$, obtain a partial differential equation for cx. Then, determine the corresponding ordinary differential equation along the characteristic line.

(c) Integrate the characteristic equation for cx in the grid shown above, i.e., find cx_i^{j+1}, given cx_ξ.

(d) Modify the function developed in problem [3] to include the decay term using the results of parts (a), (b), and (c).

(e) Repeat the simulations of problem [3] using the decay term with $T = t_{max}/10$, $t_{max}/5$, and $ts_{max}/2$.

Submit:

- Word document showing problem statement, functions or scripts used to solve problem, and output showing solution. Include your name in this document.
- M-files with Matlab scripts and functions.
- Input or output text files.

How to submit your assignment:

Place the Word document, *m*-files, and input and output text files in a zip file. E-mail the zip file to: gurro@cc.usu.edu