**AOS 180 – Modeling Task 4**
2D advection-diffusion equation
Due date: May 3rd 11:00AM

---

## Problem description

We are interested in the solution of a simple two-dimensional advection-diffusion equation given by

$$\frac{\partial u}{\partial t} + c_x \frac{\partial u}{\partial x} + c_y \frac{\partial u}{\partial y} = K_x \frac{\partial^2 u}{\partial x^2} + K_y \frac{\partial^2 u}{\partial y^2}, \tag{1}$$

on the domain $0 \leq x \leq L_x$, $0 \leq y \leq L_y$ for $t \geq 0$. Here $c_x > 0$ and $c_y > 0$ are constant propagation velocities, $K_x = K_y = K > 0$ is a constant diffusivity, and the quantity $u$ is arbitrary and it has no units.

We will use the following initial condition

$$u(x, y, t = 0) = u_0(x, y) = \frac{1}{2} A \left[\cos\left(d\pi\right) + 1\right], \tag{2}$$

where $A$ is the amplitude and $d(x, y)$ is given by

$$d = \min\left\{1, \frac{1}{r}\sqrt{\left(x - x_0\right)^2 + \left(y - y_0\right)^2}\right\}. \tag{3}$$

Here, $r$ is the radius of the disturbance centered at $(x_0, y_0)$. This is a smart way to define a localized initial condition without the need of using an "IF statement" inside a double loop. As for the boundary conditions, we will use cyclic (or periodic) boundary conditions in both directions. This means that $u(x = 0, y, t) = u(x = L_x, y, t)$ and $u(x, y = 0, t) = u(x, y = L_y, t)$.

For this problem the analytical solution for the case with $K = 0$ is given by

$$u(x, y, t) = u_0(x - c_x t, y - c_y t). \tag{4}$$

You can also write an analytical solution for a case with constant $K$, but we will not discuss that here. Instead, we will focus on other properties of the equation. More specifically, this is a conservation equation, which means that the total amount of $u$ in the domain

$$\mathcal{U}(t) = \int_0^{L_x} \int_0^{L_y} u(x, y, t)\, dy dx \tag{5}$$

should be constant in time. You will use this property of the solution to check your code.

## Numerical approach

In larger numerical projects it is useful to break your code development and debugging into phases. Here it is a good idea to use 3 phases: (i) advection, (ii) diffusion, and (iii) advection-diffusion. You will use the simple operator splitting scheme with leapfrog for advection and

Euler forward (FTCS) with a time step equal to $2\Delta t$ for diffusion. This is not a very good scheme for diffusion and you should really not use it if you are interested in getting diffusion right. But it will serve for our purposes here, as our main goal for the next projects is to get advection right and use diffusion as a way to avoid numerical instabilities and damp small numerical errors. In this project, I recommend that you design your algorithm before you start writing your code. Sketch your domain, number your indices, decide how you will treat the boundary conditions, etc. It will pay off to be organized from the beginning.

For the present solution you will use the following basic configuration: $L_x = L_y = 50\,\mathrm{m}$, $c_x = c_y = 1.0\,\mathrm{m/s}$, and $K = 0.1\,\mathrm{m}^2/\mathrm{s}$. For the initial disturbance you will adopt $A = 1$ and $r = 12.5\,\mathrm{m}$ and start the disturbance at the center of the domain $(x_0, y_0) = (25\,\mathrm{m}, 25\,\mathrm{m})$. You will integrate the model for a total time of $T_{tot} = 400\,\mathrm{s}$, which corresponds to 8 full cycles of the solution (i.e., the solution will pass 10 times through its initial position). For your base simulation, use 51 grid points in each direction. The choice of $\Delta t$ is yours.

Even though you can track the analytical solution at all times (with and without diffusion), this is neither straightforward nor needed. For the case without diffusion, we know that every 50 s the analytical solution is identical to the initial condition (for the basic configuration given above). So you will compare the numerical solutions every 50 s with the initial condition. I suggest you produce 3 types of outputs: (i) the control output, every single time step, containing the maximum value of $u$ in the domain and the quantity $\mathcal{U}$; (ii) the complete two-dimensional field at spaced time steps to compare with the initial condition and to produce animations, and (iii) "cuts" of the solution through the two diagonals of the domain to make some line plots for visualization (i.e., the solution along the lines $y = x$ and $y = L_y - x$ (the reason for these two lines is that the former is the streamwise direction and the latter the crosswise direction for the advection velocities indicated above).

**Advection in 2D**

I suggest you start from your code for Modeling Task 3, and simply extend it to two dimensions. Change the initial condition to reflect the problem above and run your code. This is just advection with leapfrog, so based on your choice for $\Delta t$, you should know what to expect. Make sure your code behaves properly for other choice of $c_x$ and $c_y$, including cases in which one of the components is zero, before you move on to the next stage.

**Diffusion in 2D**

Next you should turn off the advection (by setting $c_x = c_y = 0$) and code the diffusion. It is fairly easy to test your diffusion code, as you know what to expect from a simple diffusive process. Play with the diffusion coefficient to make sure your code is behaving as expected (i.e., make sure you get what you expect when you increase or decrease $K$). If you want to be fancy, play also with cases in which $K_x \neq K_y$.

**Advection-diffusion in 2D**

Now that both parts of your code work independently, you can go ahead and run both together. I suggest you always keep track of the Courant and von Neumann numbers (defined for a 2D problem), and maybe the grid Peclet number as well (even though it is not as important for the scheme we chose).

### Report

Please include in your report:

**i.** Figures: Figures that briefly document your code with the basic configuration and support your discussion (see below).

**ii.** Documentation: Please include in your report the recursive equations used and how you implemented the boundary conditions. Also include a brief rationale for your choice of time step.

**iii.** Discussion: Please briefly discuss your observations for the basic configuration with (i) only advection, (ii) only diffusion, and (iii) advection and diffusion. For consistency, I suggest you use the same $\Delta t$ for all 3 simulations. It may be worthwhile monitoring the amplitude of the solution (both the maximum and minimum values) and the conservation property given by Eq. (5).

**iv.** Challenge question: For the case with no diffusion, how come we have loss in amplitude if $|\lambda| = 1$?

Please upload a **.pdf** file with your report and **.pdf** file with your code to the dropbox that will be available on the CCLE website for the course. If you also want to upload your original code, that is fine. Please, do not upload word files or other formats, as they do not work well in other operational systems.

### Additional suggestions

I am asking you to report and discuss results for only 3 simulations. I suggest you also explore a few more things, even if you choose not to include them in your report.

- Change the shape and or the radius of the disturbance and note its effect on the results (e.g. something like the square wave would be interesting).
- Read on the textbook about the Robert-Asselin filter for the leapfrog scheme. Try to implement it and check if it helps damping the computational mode as expected.
- Try to run the simulation at higher resolutions (say 201 grid points in each direction). How does that increase your computational cost? How much do you have to decrease $\Delta t$? Can you find ways to document computational cost?
- Modify your integration scheme. You could either try the FTCS with diffusion stabilizing advection (as discussed in class) or change the entire integration into RK3 or AB3 (to be honest, I have not tried AB3 in this problem yet, so I am curious about its performance).