Pre-class assignment #9

Brian O'Shea,
PHY-905-003, Computational Astrophysics and Astrostatistics
Spring 2023

This assignment is due the evening of Wednesday Feb. 8, 2023. Turn in all materials via GitHub.

Reading:

1. Chapter 4 ("Advection Basics") of Mike Zingale's Computational Hydrodynamics Tutorial (also included in this repository).

Your assignment:

You're going to solve the linear advection equation (Equation 4.1) for "top hat" and Gaussian wave pulses in a 1D domain, and experiment with two different first-order methods of solution – upwinding and forward-time, centered-space (FTCS) – and compare how they behave. We will do this by creating a set of functions that do the following (and which we will re-use both in class and later in the semester):

1. A function that takes in several arguments and generates a set of initial conditions comprised of 1-D numpy arrays. The input arguments should be the number of points in the domain, N, and the type of initial condition ("top hat" wave or Gaussian wave). The function will **generate and return** a 1D array of point locations in the domain $x = 0$ to $x = 1$, assuming N equidstant points, as well as an array of a values. For a top hat wave, set $a = 1.0$ for $0.35 < x < 0.65$ and $a = 0.0$ outside of that, and for a Gaussian wave set a max value of $a = 1.0$ at $x = 0.5$ and a full-width, half-max of 0.1. You should make both arrays the same length (N elements).

2. A function that evolves the system one time step (from step n to $n + 1$). The input arguments should be the a array described above (at time n), the wave speed u, the CFL number C, and the method that you wish to use (either upwind or FTCS). The returned value should be a new array corresponding to the a array at time $n + 1$. Assume that the domain is periodic – in other words, think of your a array as looping around such that $a[N - 1]$ is next to $a[0]$.

3. A function to make a plot of the simulation at some state, compared to the $t = 0$ dataset, and save it to a file with a unique name. You should take as arguments the x and a arrays, plus whatever else you need to generate a unique file name and title.

You may find the numpy array and Python function tutorial notebooks included in this Git repository to be helpful in constructing your functions!

Using the functions you created above, evolve at top-hat wave with a velocity of $u = 0.1$ from $t = 0$ to $t = 1/u = 10$, which is exactly one period through the domain. Use $N = 100$ grid points and $C = 0.5$ and verify that your method works by running the upwind method (we will call this particular combination of method, initial condtions, N, and C the "standard model"). Make plots at the beginning of the evolution ($t = 0$), 1/10 of a period ($t = 1$) and a full period ($t = 10$). Then, do some experiments to answer the following questions:

1. Experiment with the forward-time, center-space (FTCS) method. Is there any combination of N (varied from, say, 10 to 1,000) and C (varied from $0.1 - 2$) that is stable for an entire period (i.e., until $t = 1/u = 10$)?

2. For the upwind method, if you vary C from $0.1 - 2$ for your $N = 100$ top-hat model, how does the behavior vary as compared to the standard model?

3. Using the upwind method, $N = 100$, and $C = 0.5$, compare the behavior of the top-hat and Gaussian initial conditions after one period for $C = 0.1, 0.5, 0.9, 1, 1.1, 1.5, 2$. How do the results differ?

4. Qualitatively, how would you describe the overall evolution in the solutions between the two methods and two initial conditions over time?

Answer these questions in the `ANSWERS.md` file, and include any plots that are needed to demonstrate your results. Please also include any questions you have about the reading!