**MAE 250H, Spring 2019**
**J. D. Eldredge**
**Homework 3, Due Tuesday, April 23**

This homework is focused on the effect of differencing schemes on 1-d model problems.

The relevant code is on github at
`https://github.com/incompressible-flows-spring2019/mae250h_hw3.git`

1. **Dispersion and dissipation of a wave packet.** Consider the 1-d advection equation,

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

for the initial condition

$$u(x,0) = u_0(x) = \frac{1}{\sqrt{\pi\sigma^2}}\mathrm{e}^{-(x-x_0)^2/\sigma^2}\cos[k_0(x-x_0)]$$

In physical space, this is a sinusoid modulated by a Gaussian of width $\sim \sigma$ and centered at $x = x_0$. This function can be called a *wave packet*, because its spectrum is also a Gaussian, but of width $\sim 1/\sigma$ and centered at wavenumber $k_0$. That is, it contains content in several wavenumbers that are closely spaced in the spectrum.

Such a function is useful for observing the effects of dissipation and dispersion due to a finite difference operator because we can precisely target portions of the spectrum. Dispersion will cause the various wavenumber components of $u$ to travel at different speeds. Because they are so tightly packed in the spectrum, it can be shown that the entire wave packet travels approximately at the *group velocity*, $c_g = c\,\mathrm{d}k'/\mathrm{d}k$ evaluated at $k_0$, where $k'$ is the modified wavenumber. For the exact case, $c_g = c$. That is, the entire wave packet simply travels at speed $c$ without change of its shape. However, the actual group velocity is clearly different, as given by the slope of $k'$. In fact, for some values of $k_0\Delta x$, the entire packet might move in the wrong direction! Remember that we can only represent data on a grid for wavenumbers in the range $0 < k < \pi/\Delta x$.

In this problem, you are to explore the behavior of this wave packet in the context of a few spatial finite difference operators. We are not interested in the effects of boundaries here (which would simply complicate matters), you are to explore the behavior for the following conditions: on a periodic 1-d grid of $N$ cells extending from $x = 0$ to $x = 1$, with wave speed set at $c = 1$. For the wave packet, set $x_0 = 0.5$ and $\sigma = 0.25$. You will choose various values for $k_0$ in the problems below.

I have provided some Julia code that will help you attack this problem, including 1-d data types and operators, and a Runge-Kutta time marching. I provide a notebook (called `HW3`) that describes the basic aspects of this code. There is nothing for you to do in that notebook except explore the use of the code. The notebook called 1D Advection— is your starting point for this problem. But there is not much coding necessary (that is, if you work in Julia — you might want to use Julia for this problem even if your preferred language is something else). The notebook should be fairly self-explanatory.

(a) For three different choices of $k_0$ in the resolved spectrum (i.e., between $k_0 = 0$ and $k_0 = \pi/\Delta x$), compute the advection over 1/4 time unit (i.e. `tf=0.25`) using both 2nd-order central differencing and 1st-order backward differencing. Comment on the results. In particular, draw the connection with the spectrum. Set $N = 100$ for all cases. Also, for this part and the next one, we don't want the accuracy of the time marching to be of concern, so use 4th-order Runge-Kutta with a very small step size (Courant number, $c\Delta t/\Delta x$ of 0.01, for example.).

(b) Try some of the cases above with a larger number of points, say $N = 300$.

(c) Verify that, for *any* choice of finite difference scheme or time step size, the explicit Euler method is unstable for pure advection problems. You might have to run for longer time than in the previous parts to verify this. Comment on why this is the case.

2. **Solution of 1-d diffusion equation.** The objective of this problem is to solve the 1-d diffusion equation,

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2}$$

on a domain $x \in [0, L]$ with constant Dirichlet boundary conditions, $u(0, t) = 1$ and $u(L, t) = 2$, and initial condition $u(x, 0) = 0$. For this study, there is no loss in generality if we set $L = 1$ and $\nu = 1$. As in the previous problem, the domain is divided into $N$ cells (with $N + 1$ edges); two additional ghost cells are used to enforce boundary conditions. The purpose of this problem is two-fold:

- Learn how to set Dirichlet conditions for both edge-based and cell-based discretizations of the function $u(x, t)$.
- Apply both explicit (RK4) and semi-implicit (Crank-Nicolson) time marching and investigate the stability of each method.

Throughout this problem, it is sufficient to use standard 2nd-order central differencing of the Laplacian and set $N = 100$, since the focus is on time discretization. In the notebook called `1D Diffusion`, I have supplied a function called `diffuse1d_dirichlet` that takes in edge data and returns $\mathrm{d}u/\mathrm{d}t$. This function can be supplied to the RK integrator to advance the solution; I have provided an example of the time marching that goes with this.

**You** must construct *another* version of `diffuse1d_dirichlet` that accepts cell-centered data and enforces the same boundary conditions, using ghost cell values to enforce conditions by averaging, e.g., by setting $u_1 = -u_2 + 2u_L$. You must also implement a Crank–Nicolson solver for *both* data types. This solver should accept as inputs the values of $t$ and $u$ at time step $t^n$ and return $t$ and $u$ at time step $t^{n+1}$, just as the RK integrator does. It should use the `trisolve` routine you wrote in homework 1, in regular tridiagonal form. The cell-based case will be slightly more challenging for this case.

(a) Using a 4th-order Runge-Kutta and edge-based data, try different time-step sizes. Note that, in all of these diffusion cases, the time step size should be set via the grid Fourier number, $\mathrm{Fo} = \nu \Delta t / \Delta x^2$. Determine the stability boundary (that is, the critical value of the Fourier number) for this pure diffusion problem. Verify that it is consistent with the stability boundary we discussed in class.

(b) Explore the same with cell-based data. Verify your solution.

(c) Now use the Crank-Nicolson method on both edge-based and cell-based data. Verify your solutions again. Confirm that the stability boundary has been increased. (But that time discretization error becomes the main issue for large Fo).