

MEA 712: (An Introduction to) Mesoscale Atmospheric Modeling
Fourth mini computing assignment

Due at the start of class on Tuesday 8 September

Hopefully, you've gotten some FORTRAN under your belt, and you should now know how to make plots of your output. So, the fourth mini assignment is to actually run a simple advection model! This should build on our existing code. We will be coding a finite difference approximation to the equation for advection by a constant current (c):

$$\frac{\partial \psi}{\partial t} = -c \frac{\partial \psi}{\partial x}.$$

The finite difference scheme will be forward in time and backward in space; this is referred to as the *upstream scheme* for advection. The finite difference formulation is:

$$\frac{\psi_{n+1} - \psi_n}{\Delta t} = -c \frac{\psi_j - \psi_{j-1}}{\Delta x},$$

or:

$$\psi_{n+1} = \psi_n - c\Delta t \frac{\psi_j - \psi_{j-1}}{\Delta x}.$$

1. We will need to declare a few more variables.
 - a. In addition to NX, we also now need to declare NT, the number of time steps we wish to run. Set this to 10 with a PARAMETER statement.
 - b. In addition to DX, we also now need to declare DT, the length of our time steps in seconds. Set this to 10.0 with a PARAMETER statement.
 - c. In addition to I, we also now need to declare N, the index for the time dimension.
 - d. We already have PSI. In our model, we will treat PSI as the current/present value. Declare another variable, PSIF, which has the same dimensions as PSI. This will represent our future/forecast value during each time step.
 - e. We will be considering advection by a current of constant speed. Declare the variable C to represent this speed, and set it to 10.0 with a PARAMETER statement.
2. You already have code to define the initial values of PSI. For each I value within that DO loop, set each PSIF to be equal to PSI. In other words, at the start of the model run, the two fields should be identical.
3. Now you are ready to add the actual forecast loop into the code. Do the following below the point in your code where you write out the initial values of PSI (i.e. below what you wrote for assignment 3).
 - a. Create a new DO loop over N for the time range from 1 to NT.
 - b. Within this loop, four things need to happen (in order):
 - i. This will be redundant for our very first trip through the loop, but it is needed every time through thereafter. If we have just forecast a new value of PSIF during the previous timestep (previous trip through the loop), we now want to set what used to be the "future" value to be the new "current" value. In other words, before going further, we need to perform $PSI(I) = PSIF(I)$ (be sure to do this for all values of I).
 - ii. The next part of the loop actually makes the forecast of the new PSIF values. For the upstream scheme, this forecast has the form:
$$PSIF(I) = PSI(I) - C*DT*(PSI(I) - PSI(I-1))/DX$$
You should perform this calculation for I values from 2 through NX. Why don't we perform it for I=1?

iii. We can't forecast for $I=1$ because there is no value of PSI that is defined upstream! So, we have to apply what is called a *lateral boundary condition* in order to update the edge point. For this model, we will use a *periodic* lateral boundary condition, which is to say that whatever passes off one edge of the domain will then come back on the other. The code needed to implement this simple boundary condition for the upstream scheme is:

$$PSIF(1) = PSIF(NX)$$

iv. You are already writing out the initial values of PSI . We would also like to write out and be able to plot each forecast value. The final piece of your timestep loop is to write out your newly forecast values. Use the same form for the `WRITE` statement that you learned in assignment 3, except use it to write out $PSIF$ at the end of each forecast loop. **Note:** you need to increment `IREC` by 1 after each time you write out your data (including your initial data), as explained in the handout for assignment 3.

4. The coding is now complete. Compile and run your model. To summarize, we are running a model of advection by a constant current, using the upstream scheme. The timestep is 10.0s, and we are running the model for 10 timesteps (100.0s). At a current speed of 10.0m/s, this should advect the wave pattern eastward by 1000.0m, or exactly one wavelength.

5. Make some plots of your data, using GrADS or your own software. The following instructions are for GrADS users.

a. If you're using GrADS, you will need to modify the `tdef` line of your `output.ctl` file, since you are now writing out multiple times. Adjust it so that the number of times is 11 (ten time steps plus the initial time), and so that the time interval is 5mn (remember, we 'bogus' seconds as minutes for a work-around).

b. Go into GrADS, and make a plot of your `psi` field a times 1, 2, 3, and 10. How? After you have plotted the first output time using the procedure from assignment 3, issue the command `set t 2` and then `display psi` again (and so on). If you do not clear the display window (which, by the way, is accomplished with the `clear` command), then each time you issue a `display` command, the new data will simply be plotted on top of your old data (in a new color).

6. Print out your combined plot of times 1, 2, 3, and 10, along with a printout of your code, and bring them to class for credit.

GrADS bonus info

If you wanted to animate your data (i.e. see the pattern translate), you would use the following:

```
ga-> set t 1 11
ga-> set looping on
ga-> display psi
```

If, instead, you wanted to view your output as a Hovmoller diagram (i.e. value versus time plot), you would use the following:

```
ga-> set t 1 11
ga-> set looping off
ga-> display psi
```

Both of these approaches will be useful for assignments in this course!