Computer Problem 3
2D Advection, Deformational Flow

Due: 2:00 PM Friday March 1
Turn in: **your code and plotted results** (*all* submitted via Moodle)

Problem being solved: 2-D linear advection with fractional step (directional) splitting.
Boundary condition: 0-gradient (as in program #2)
Numerical methods: Lax-Wendroff, 6[th]-order Crowley, and Takacs.

1. Crowley 2[nd]-order *(**same as** Lax-Wendroff)*	$s_j^{n+1} = s_j^n - \frac{\nu}{2}\left(s_{j+1}^n - s_{j-1}^n\right) + \frac{\nu^2}{2}\left(s_{j+1}^n - 2s_j^n + s_{j-1}^n\right)$
2. Crowley 6th-order	*See Tremback p. 542, ORD=6 (advective form)*
3. Takacs (1985) *(as in program 2)*	$\nu \ge 0: \begin{cases} s_j^{n+1} = s_j^n - \frac{\nu}{2}\left(s_{j+1}^n - s_{j-1}^n\right) + \frac{\nu^2}{2}\left(s_{j+1}^n - 2s_j^n + s_{j-1}^n\right) \\ \qquad -\left(\frac{1+\nu}{6}\right)\nu(\nu-1)\left(s_{j+1}^n - 3s_j^n + 3s_{j-1}^n - s_{j-2}^n\right) \end{cases}$ $\nu < 0: \begin{cases} s_j^{n+1} = s_j^n - \frac{\nu}{2}\left(s_{j+1}^n - s_{j-1}^n\right) + \frac{\nu^2}{2}\left(s_{j+1}^n - 2s_j^n + s_{j-1}^n\right) \\ \qquad -\left(\frac{1+\lvert\nu\rvert}{6}\right)\nu(\nu+1)\left(s_{j-1}^n - 3s_j^n + 3s_{j+1}^n - s_{j+2}^n\right) \end{cases}$

Domain: The domain size/layout are the same as program #2. However, ***u***, ***v*** *differ from the last problem*, as does the initial position/size of the cone. Be careful: ***u(i,j)*** (½ grid length to the left of ***s***) and ***v(i,j)*** (½ grid length below ***s***) are now functions of x *and* y.

If you see asymmetry (discussed below) in your solutions, the #1 most likely cause is a problem in the initial conditions – probably the X and Y coordinates used in creating the initial conditions. All you need is for the cone or the U or V velocity components to be incorrectly located by dx/2 or dy/2 to result in erroneous behavior. Symmetry tests are great at identifying problems in the initial or boundary condition or advection scheme.

Advection method: Lax-Wendroff, 6[th]-order Crowley, Takacs are directionally split, and are unaware of C-grid staggering so you must average velocity to the scalar point in your 1-D advection. Takacs needs **2** ghost points, and 6[th]-order Crowley requires **3**.

Settings: nx, ny, cone center, cone radius, time step, # of steps – *see program 3 page.*
Read In: the numerical method to use • number of steps to run • how often to plot

Initial conditions: Define *s* as before, though the cone radius and center are changed.

Wind field – deformation	$u(x,y) = sin(4\pi x) \times sin(4\pi y)$ $v(x,y) = cos(4\pi x) \times cos(4\pi y)$	

Code layout *requirements*:

1. you **<u>must</u>** use and call *separate advection (2-D)* and *advect1d (1-D) routines*.
   - Do the 1-D advection step fully in *a separate* **advect1d** routine, where your 1-D methods reside.
   - Do **not** combine 2D, 1D steps or embed integration code in the main program <u>*or*</u> in your 2-D advection routine.
2. do not "hard-code" your program for any scheme! So your code must be set up for the maximum number of ghost points (3) you need, and to run any scheme.
3. pass the *staggered* [not averaged] u or v data to *advect1d*.
   - the unstaggering of the velocities is done inside *advect1d* when you e.g. compute the Courant number (in Fortran: dt/dx*0.5*(u1d(i)+u1d(i+1)))
4. do not hard-code the program's (maximum) grid dimensions except at the <u>start</u> of the main program, in (Fortran) a module routine, or in an include file.
5. **do not** (in C) use point 0 as always a single ghost point, 1 as the first physical point, etc; you *must* use the I1,I2,J1,J2 (etc.) notation for handling ghost points.
6. code generally! *See class content page for full code rules.*

Submit online:

- Contour plots of the initial u, v, and s field. And, for each method, create *contr* **and** *sfc* plots of the solutions at 125, 250, 750 steps.
- Smin(t) & Smax(t) plots are *not* necessary, but **do** use my *contr* and *sfc* routines.