2D Governing Equations

Governing equations for 2D atmospheric flow (momentum conservation, heat energy conservation and mass continuity) with incompressible Boussinesq approximation:

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+w\frac{\partial u}{\partial z}=-\frac{1}{\rho_0}\frac{\partial p'}{\partial x}+K\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial z^2}\right) \tag{1}$$

$$\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial x}+w\frac{\partial w}{\partial z}=-\frac{1}{\rho_0}\frac{\partial p'}{\partial z}+g\frac{\theta'}{\theta_0}+K\left(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial z^2}\right) \tag{2}$$

$$\frac{\partial \theta'}{\partial t}+u\frac{\partial \theta'}{\partial x}+w\frac{\partial \theta'}{\partial z}=K\left(\frac{\partial^2 \theta'}{\partial x^2}+\frac{\partial^2 \theta'}{\partial z^2}\right) \tag{3}$$

$$D=\frac{\partial u}{\partial x}+\frac{\partial w}{\partial z}=0 \tag{4}$$

where u and w are velocity components in x and z directions, p' is perturbation pressure from a mean state pressure p_0 that is assumed to be in hydrostatic balance with mean state potential temperature θ_0. θ' is perturbation potential temperature from mean state θ_0. D is divergence, and K is mixing coefficient. The second term on the right hand side of (2) is the buoyancy term.

With Boussinesq approximation, ρ_0 is assumed to be constant, and divergence D is zero (essentially the flow is incompressible).

Derive equation for horizontal vorticity η, by taking $\frac{\partial Eq.(1)}{\partial z} - \frac{\partial Eq.(2)}{\partial x}$. The pressure terms then disappear, yielding

$$\frac{\partial \eta}{\partial t} + u\frac{\partial \eta}{\partial x} + w\frac{\partial \eta}{\partial z} + \frac{\partial u}{\partial z}\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z}\frac{\partial u}{\partial z} - \frac{\partial u}{\partial x}\frac{\partial w}{\partial x} - \frac{\partial w}{\partial x}\frac{\partial w}{\partial z} = -\frac{g}{\theta_0}\frac{\partial \theta'}{\partial x} + K\nabla^2\eta \quad (5)$$

Because $D = 0$, the above becomes

$$\frac{\partial \eta}{\partial t} + u\frac{\partial \eta}{\partial x} + w\frac{\partial \eta}{\partial z} = -\frac{g}{\theta_0}\frac{\partial \theta'}{\partial x} + K\nabla^2\eta \,. \quad (6)$$

For 2D incompressible flow, the velocity component can be expressed in terms of stream function ψ ,

$$u = \frac{\partial \psi}{\partial z} \text{ and } w = -\frac{\partial \psi}{\partial x}. \quad (7)$$

Plug them into the vorticity definition:

$$\eta = \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x} = \frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial z^2} \equiv \nabla^2\psi \,, \quad (8)$$

$\nabla^2()$ is call Laplace operator. ***The contours of stream function are parallel to velocity vectors everywhere.***

Equation (8) is an elliptic equation that can be solved given boundary conditions of ψ and the right hand side η (which is the vorticity predicted by (6)).

Reproducing potential temperature Eq.(3) below as

$$\frac{\partial \theta'}{\partial t} + u\frac{\partial \theta'}{\partial x} + w\frac{\partial \theta'}{\partial z} = K\nabla^2\theta' \quad (9)$$

Equations (6), (9) and (8) form a closed system and can be used to formulate a 2D numerical model that predicts future state of flow and temperature distribution.

Numerical solutions

- The left hand side of both vorticity equation (6) and potential temperature equation (9) represent advection processes.
- ***Advection*** is one of the most difficult processes to represent accurately within numerical models. Many research and numerical schemes have been devoted to the advection processes.
- The equations we have are partial differential equations (PDEs) that involve many partial derivatives.
- Being able to calculate these derivatives as accurately as possible is important to reduce numerical model errors. The component of numerical model that solves the dynamic equations is often called the ***dynamic core***, while packages that represent processes that cannot be explicitly represented by the model grid are usually called physics ***parameterizations*** or ***model physics***.

Finite Difference as Approximation to Derivatives

- In numerical models, the most common way to represent continuous fields is to define their values at finite number of grid points, typically arranged regularly in space (in three directions). Some models do use irregularly spacing grid points.
- Such models are called grid point models or finite difference models.
- Models that represent fields using spectral coefficients are called spectral models.
- In the time dimension, we need to do the same discretization, and represent continuous fields in time as a finitely number of time levels.

First, we lay down a convention for notion:

Time level - superscript n - $\rho^n \sim \rho$ at time level n

Δt = time interval = $t^{n+1} - t^n$.

Most times, we use constant Δt. Occasionally, Δt changes with time.

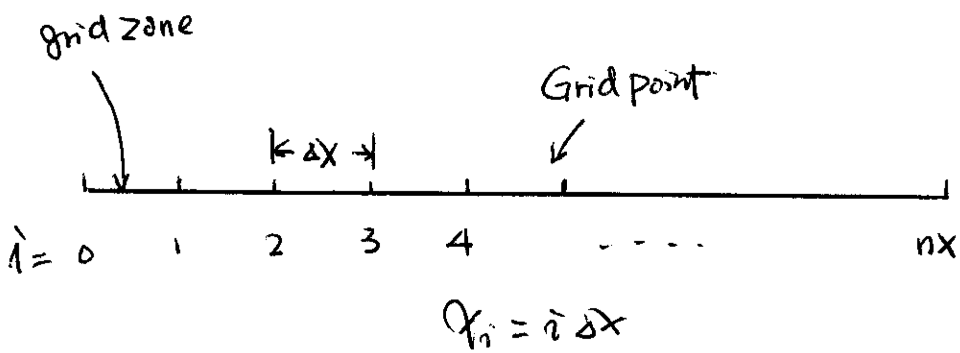
n ~ present

n+1 ~ future

$t = n\, \Delta t$ where n = number of time steps = 0, 1, 2, 3, ….., N

$T = N\, \Delta t$ = final time.

Spatial Location – subscript i, j, k, for x, y, and z.



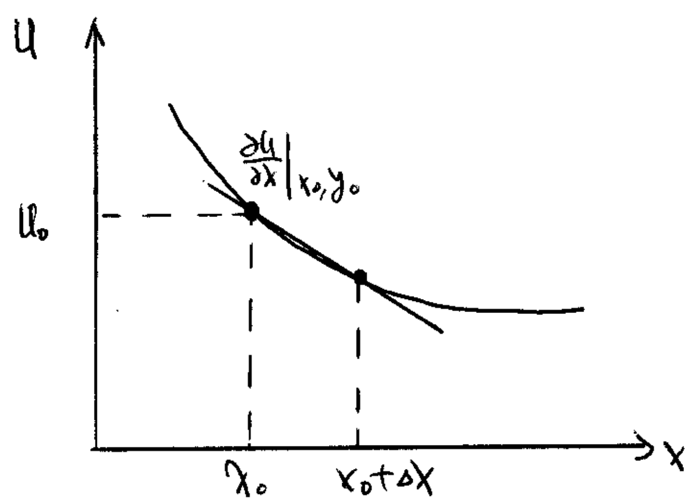
Δx – constant grid interval

$x_i = i\, \Delta x$

Recall the definition of a derivative:

$$\left.\frac{\partial u}{\partial x}\right|_{x_0,y_0} = \lim_{\Delta x\to 0}\frac{u(x_0+\Delta x, y_0)-u(x_0,y_0)}{\Delta x} \qquad (10)$$

For a finite Δx,



the derivative is approximated as

$$\left.\frac{\partial u}{\partial x}\right|_{x_0,y_0} \approx \frac{u(x_0+\Delta x, y_0)-u(x_0,y_0)}{\Delta x}. \qquad (11)$$

One can also use the value at $x_0 - \Delta x$ instead to get

$$\left.\frac{\partial u}{\partial x}\right|_{x_0, y_0} \approx \frac{u(x_0, y_0) - u(x_0 - \Delta x, y_0)}{\Delta x}. \quad (12)$$

Both (1) and (2) provide an expression for $\partial u/\partial x$, but numerically the answers will be different.

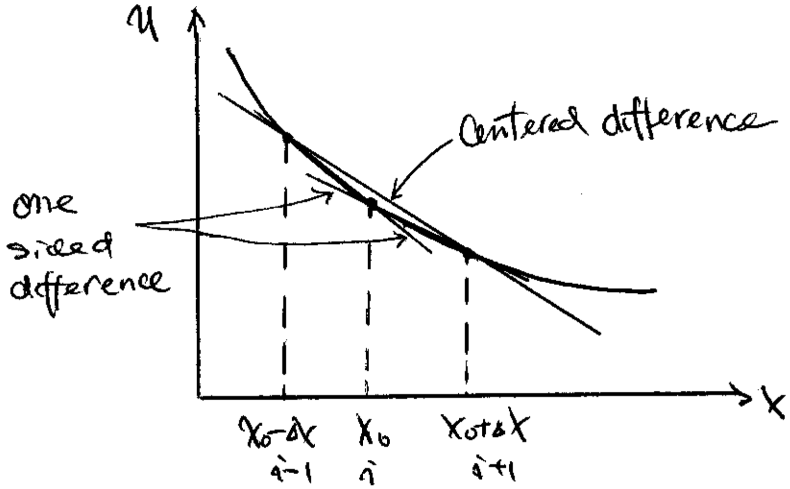
(11) is called a <u>forward finite difference</u>

(12) is called a <u>backward finite difference</u>

We can get another discrete approximation to $\partial u/\partial x$ by adding (11) and (12):

$$\left.\frac{\partial u}{\partial x}\right|_{x_0, y_0} \approx \frac{u(x_0 + \Delta x, y_0) - u(x_0 - \Delta x, y_0)}{2\Delta x} = \frac{u_{i+1} - u_{i-1}}{2\Delta x} \qquad (13)$$

This is called <u>centered difference</u>. Note, it doesn't even use value of u at the current point i. It approximates the <u>slope</u> using two neighboring points:



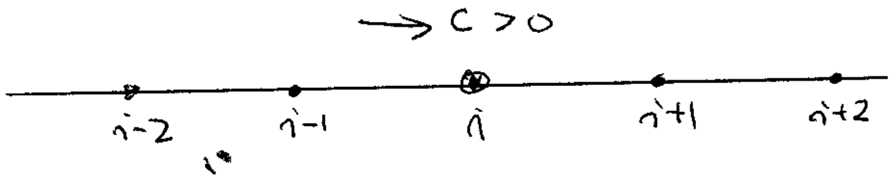
Most times, the center difference is more accurate than one sided difference. Actually the numerical order of accuracy of center difference is 2, higher than order 1 of one-sided difference.

The second derivatives in the Laplace terms of equations (6), (7) and (9), such as $\partial^2 u/\partial x^2$ can be discretized using center difference as:

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u(x+\Delta x, y) - 2u(x, y) + u(x-\Delta x, y)}{(\Delta x)^2} = \frac{u_{i-1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2}\,. \qquad (14)$$

Now let's consider simple ***1-D advection equation***

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0, \quad \text{where } c > 0, \qquad (15)$$



we might want to use

$$\frac{\partial u}{\partial t} + c\frac{u_i - u_{i-1}}{\Delta x} = 0, \qquad c > 0, \ (16)$$

which is called ***upwind or upstream difference*** because the point upstream of *i* is used in the advection term.

Alternatively, in

$$\frac{\partial u}{\partial t} + c\frac{u_{i+1} - u_i}{\Delta x} = 0, \qquad c > 0, \qquad (17)$$

downstream difference is used.

Upstream difference is better than downstream difference for this problem, because for this pure advection problem, signals move from upstream (left) to downstream (right). The value of *u* at point *i* at a future time should be influenced by the values of *u* upstream, not downstream. For this reason, using downstream stream can lead to unphysical solution.

When combined with forward-in-time time difference, the above two equations become

$$\frac{u_i^{n+1}-u_i^n}{\Delta t}+c\frac{u_i^n-u_{i-1}^n}{\Delta x}=0, \qquad c>0, \qquad (18)$$

$$\frac{u_i^{n+1}-u_i^n}{\Delta t}+c\frac{u_{i+1}^n-u_i^n}{\Delta x}=0. \qquad c>0. \qquad (19)$$

Numerical stability shows that downstream advection scheme in (19) is ***absolutely unstable***.

The forward-in-time, upstream-in-space scheme in (18) is ***conditionally stable***. The stability condition is

$$\frac{c\Delta t}{\Delta x}\le 1 \text{ or } \Delta t \le \Delta x/c. \qquad (20)$$

$$\sigma=\frac{c\Delta t}{\Delta x} \text{ is called } \textbf{\textit{Courant number}} \qquad (21)$$

This condition is called the ***CFL criteria***, after three scientists Courant, Friedrichs and Lewy.

Physically, the ***CFL criteria says that within one time step, the signal should not travel by more than grid spacing***.

The actual equation that can be used to code time integration for u can be written as

$$\begin{aligned} u_i^{n+1}&=u_i^n-\sigma(u_i^n-u_{i-1}^n), \quad c\ge 0 \\ u_i^{n+1}&=u_i^n-\sigma(u_{i+1}^n-u_i^n). \quad c<0 \end{aligned} \qquad (22)$$

The two different formula are used for postive and negative *c* ensures that 'upstream' spatial difference is always used.

Eq. (22) should be calcualted or looped over all grid points where it can be calcluated (it means from second point on the left to the last grid point).

The 2D version of (18) in *x* and *z* directions is

$$\frac{u_{i,k}^{n+1} - u_{i,k}^{n}}{\Delta t} + c_x \frac{u_{i,k}^{n} - u_{i-1,k}^{n}}{\Delta x} + c_z \frac{u_{i,j}^{n} - u_{i,k-1}^{n}}{\Delta z} = 0, \text{ for } c_x \geq 0, c_z \geq 0. \tag{23}$$

or

$$u_{i,k}^{n+1} = u_{i,k}^{n} - \sigma_x (u_{i,k}^{n} - u_{i-1,k}^{n}) - \sigma_z (u_{i,k}^{n} - u_{i,k-1}^{n}) \text{ for } c_x \geq 0, c_z \geq 0. \tag{24}$$

When equation grid spacing in the two directions, the stability condition is

$$\frac{c\Delta t}{\Delta x} \leq \frac{1}{\sqrt{2}} \text{ or } \Delta t \leq \frac{\Delta x}{\sqrt{2}c} . \tag{25}$$

In models where the advection speed is not constant, we need to ensure that the condition is not violated anywhere within the model, so we need to use maximum velocity to estimate the maximum Δt that can be used.

The forward-in-time, upstream-in-space scheme is ***very diffusive*** – it tends to damp the amplitude of signals too much. It does keep positive values positive (does not create artificial negative values when advecting positive fields such as moisture, hydrometeors or chemical species). It also has smaller phase or position error.

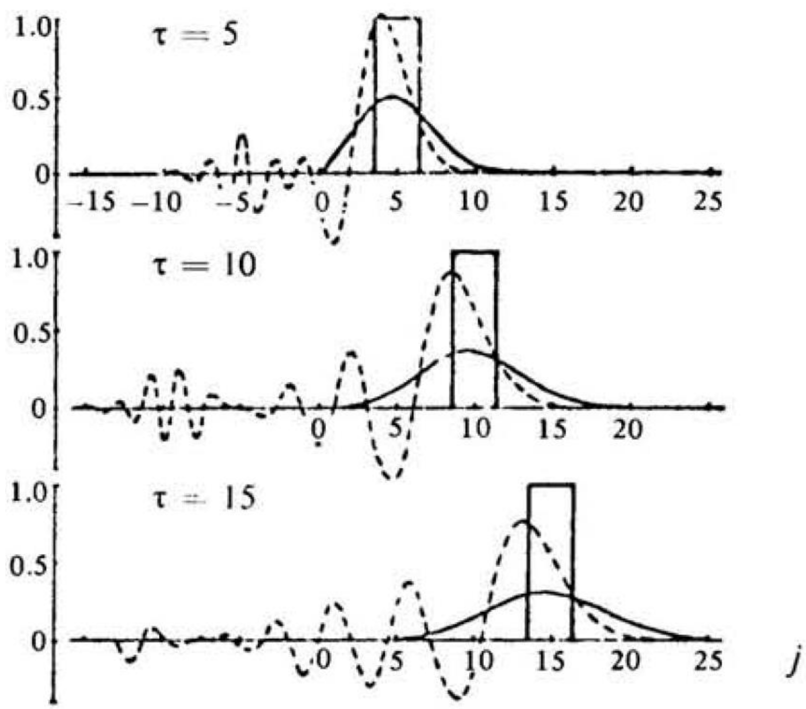


Figure 3.4 Analytic solutions of the exact advection equation (heavy solid line), of the equation using centered differencing (dashed line), and of the equation using upstream differencing (thin solid line), for three different values of the non-dimensional time τ (Wurtele, 1961).

Centered difference maintains the amplitude better, but has large dispersion/phase error (signal lagging behind the correct solution).

Smolarkiewicz (MWR 1983) developed a scheme that corrects error of the upstream stream in additional steps while maintaining positivity:

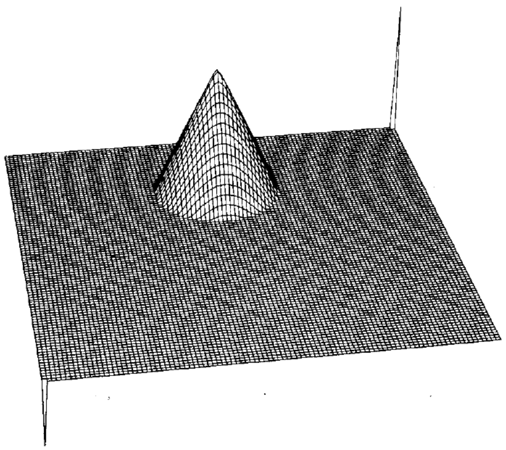


FIG. 1. Initial condition for all tests. Scale values in left-front and right-back corners are −2 and 4, respectively. The scale values are the same in all figures.

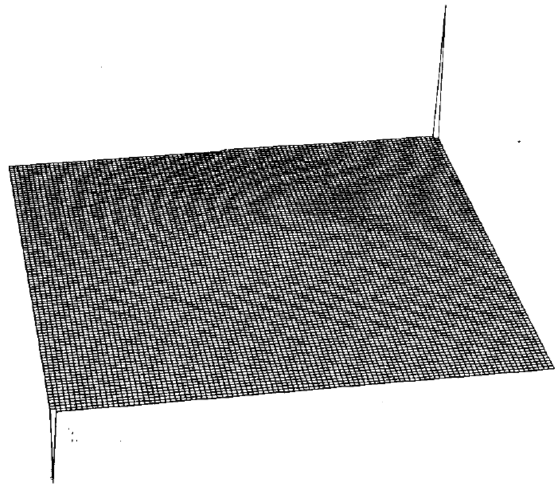


FIG. 2. Solution for "upstream" scheme after six full rotations (3768 iterations).

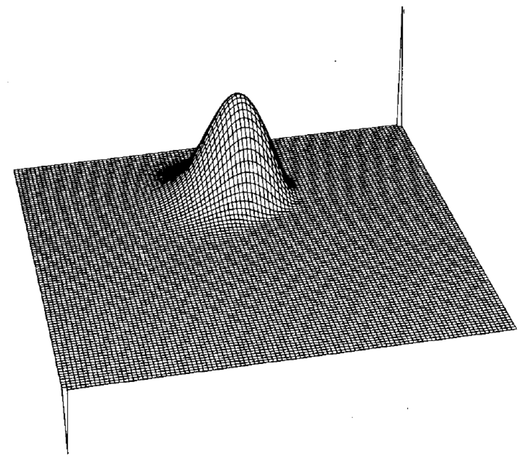


FIG. 15. As in Fig. 13, but with twice repeated corrective step, $[(13), (14)^3]_{TS}$.

From Smolarkiewicz, P. K., 1983: A simple positive definite advection scheme with small implicit diffusion. *Mon. Wea. Rev.*, **111**, 479-486.

The scheme is attractive for advecting positive fields like water vapor.

Numerical solutions of 2D vorticity and potential temperature equations

We will use forward-in-time upstream-in-space advection time integration scheme, sometimes just called upwind scheme, for the advection term, and forward-in-time, centered difference for the buoyancy gradient term. The upstream-in-space advection means that the actual formula depends on the sign of wind speed *u* and *w* at the grid point. When *u* is positive, we will use $\eta_{i,j} - \eta_{i-1,j}$ in advection in *x* direction while when *u* is negative we will use $\eta_{i+1,j} - \eta_{i,j}$, i.e., ***always use the value upstream and the value at the current location*** to calculate the finite difference in *x* direction. Similarly in the *z* direction. Stability analysis shows that you need to use upstream finite difference for the time integration to be stable when forward time difference is used. Therefore, when *u* and *w* are positive, the formula is:

$$\frac{\eta^{n+1}_{i,k} - \eta^{n}_{i,k}}{\Delta t} = -u^n_{i,k}\frac{\eta^{n}_{i,k} - \eta^{n}_{i-1,k}}{\Delta x} - w^n_{i,k}\frac{\eta^{n}_{i,k} - \eta^{n}_{i,k-1}}{\Delta z} - \frac{g}{\theta_0}\frac{\theta'^{n}_{i+1,k} - \theta'^{n}_{i-1,k}}{2\Delta x}, \text{ when } u \geq 0, w \geq 0, \tag{25a}$$

$$\frac{\eta^{n+1}_{i,k} - \eta^{n}_{i,k}}{\Delta t} = -u^n_{i,k}\frac{\eta^{n}_{i+1,k} - \eta^{n}_{i,k}}{\Delta x} - w^n_{i,k}\frac{\eta^{n}_{i,k} - \eta^{n}_{i,k-1}}{\Delta z} - \frac{g}{\theta_0}\frac{\theta'^{n}_{i+1,k} - \theta'^{n}_{i-1,k}}{2\Delta x}, \text{ when } u < 0, w \geq 0, \tag{26b}$$

$$\frac{\eta^{n+1}_{i,k} - \eta^{n}_{i,k}}{\Delta t} = -u^n_{i,k}\frac{\eta^{n}_{i,k} - \eta^{n}_{i-1,k}}{\Delta x} - w^n_{i,k}\frac{\eta^{n}_{i,k+1} - \eta^{n}_{i,k}}{\Delta z} - \frac{g}{\theta_0}\frac{\theta'^{n}_{i+1,k} - \theta'^{n}_{i-1,k}}{2\Delta x}, \text{ when } u \geq 0, w < 0, \tag{26c}$$

$$\frac{\eta^{n+1}_{i,k} - \eta^{n}_{i,k}}{\Delta t} = -u^n_{i,k}\frac{\eta^{n}_{i+1,k} - \eta^{n}_{i,k}}{\Delta x} - w^n_{i,k}\frac{\eta^{n}_{i,k+1} - \eta^{n}_{i,k}}{\Delta z} - \frac{g}{\theta_0}\frac{\theta'^{n}_{i+1,k} - \theta'^{n}_{i-1,k}}{2\Delta x}, \text{ when } u < 0, w < 0. \tag{26d}$$

As pointed out earlier, forward-in-time upstream-in-space advection is very diffusive. Because of that, we don't need to include the explicit diffusion term (the *K* term) in the equations.

We can calculate the advection terms first, like the following ***pseudo codes:***

```
For all interior k points,
For all interior i points,
    if $u^n_{i,k} \geq 0$ then

        $ADVX_{ij} = -u^n_{i,k} \frac{\eta^n_{i,k} - \eta^n_{i-1,k}}{\Delta x}$

    else

        $ADVX_{ij} = -u^n_{i,k} \frac{\eta^n_{i+1,k} - \eta^n_{i,k}}{\Delta x}$

    endif
    if $w^n_{i,k} \geq 0$ then

        $ADVZ_{ij} = -w^n_{i,k} \frac{\eta^n_{i,k} - \eta^n_{i,k-1}}{\Delta z}$

    else

        $ADVZ_{ij} = -w^n_{i,k} \frac{\eta^n_{i,k+1} - \eta^n_{i,k}}{\Delta z}$

    endif
    $\eta^{n+1}_{i,k} = \eta^n_{i,k} + \Delta t(ADVX_{i,k} + ADVZ_{i,k}) - \Delta t g(\theta'^n_{i+1,k} - \theta'^n_{i-1,k}) / (2\Delta x \theta_0)$
end i loop
end k loop
```

Almost identical codes can be used for θ'

```
For all interior k points,
For all interior i points,
    if u^n_{i,k} ≥ 0 then
```

$$ADVX_{ij} = -u^n_{i,k}\frac{\theta'^n_{i,k} - \theta'^n_{i-1,k}}{\Delta x}$$

```
    else
```

$$ADVX_{ij} = -u^n_{i,k}\frac{\theta'^n_{i+1,k} - \theta'^n_{i,k}}{\Delta x}$$

```
    endif
    if w^n_{i,k} ≥ 0 then
```

$$ADVZ_{ij} = -w^n_{i,k}\frac{\theta'^n_{i,k} - \theta'^n_{i,k-1}}{\Delta z}$$

```
    else
```

$$ADVZ_{ij} = -w^n_{i,k}\frac{\theta'^n_{i,k+1} - \theta'^n_{i,k}}{\Delta z}$$

```
    endif
```

$$\theta'^{n+1}_{i,k} = \theta'^n_{i,k} + \Delta t(ADVX_{i,k} + ADVZ_{i,k})$$

```
end i loop
end k loop
```

Equations (26) can be combined into one as below,

$$\eta_{i,k}^{n+1} = \eta_{i,k}^{n} - 0.5\Delta t(u_{i,j}^{n} + |u_{i,j}^{n}|)\frac{\eta_{i,k}^{n} - \eta_{i-1,k}^{n}}{\Delta x} - 0.5\Delta t(u_{i,j}^{n} - |u_{i,j}^{n}|)u_{i,j}^{n}\frac{\eta_{i+1,k}^{n} - \eta_{i,k}^{n}}{\Delta x}$$

$$-0.5\Delta t(w_{i,j}^{n} + |w_{i,j}^{n}|)\frac{\eta_{i,k}^{n} - \eta_{i,k-1}^{n}}{\Delta z} - 0.5\Delta t(w_{i,j}^{n} - |w_{i,j}^{n}|)\frac{\eta_{i,k+1}^{n} - \eta_{i,k}^{n}}{\Delta z} - \Delta t\frac{g}{\theta_0}\frac{\theta'^{n}_{i+1,k} - \theta'^{n}_{i-1,k}}{2\Delta x}. \quad (27)$$

The same advection scheme is applied to θ':

$$\theta'^{n+1}_{i,k} = \theta'^{n}_{i,k} - 0.5\Delta t(u_{i,j}^{n} + |u_{i,j}^{n}|)\frac{\theta'^{n}_{i,k} - \theta'^{n}_{i-1,k}}{\Delta x} - 0.5\Delta t(u_{i,j}^{n} - |u_{i,j}^{n}|)u_{i,j}^{n}\frac{\theta'^{n}_{i+1,k} - \theta'^{n}_{i,k}}{\Delta x}$$

$$-0.5\Delta t(w_{i,j}^{n} + |w_{i,j}^{n}|)\frac{\theta'^{n}_{i,k} - \theta'^{n}_{i,k-1}}{\Delta z} - 0.5\Delta t(w_{i,j}^{n} - |w_{i,j}^{n}|)\frac{\theta'^{n}_{i,k+1} - \theta'^{n}_{i,k}}{\Delta z}. \quad (28)$$

This is best that you create a function for the 2D advection, and call it for both η and θ' equations. Let's say the function is ADV(u, w, ϕ),

$$\mathrm{ADV}(u,w,\phi) = -0.5(u_{i,j}^{n} + |u_{i,j}^{n}|)(\phi_{i,k}^{n} - \phi_{i-1,k}^{n})/\Delta x - 0.5(u_{i,j}^{n} - |u_{i,j}^{n}|)(\phi_{i+1,k}^{n} - \phi_{i,k}^{n})/\Delta x$$

$$-0.5(w_{i,j}^{n} + |w_{i,j}^{n}|)(\phi_{i,k}^{n} - \phi_{i,k-1}^{n})/\Delta z - 0.5(w_{i,j}^{n} - |w_{i,j}^{n}|)(\phi_{i,k+1}^{n} - \phi_{i,k}^{n})/\Delta z. \quad (29)$$

then Eqs. (27) and (28) can be written as

$$\eta_{i,k}^{n+1} = \eta_{i,k}^{n} + \Delta t \mathrm{ADV}(u,w,\eta) - \Delta t g(\theta'^{n}_{i+1,k} - \theta'^{n}_{i-1,k}) / (2\Delta x \theta_0), \tag{30}$$

$$\theta'^{n+1}_{i,k} = \theta'^{n}_{i,k} + \Delta t \mathrm{ADV}(u,w,\theta'). \tag{31}$$

Given initial conditions of u, w and θ, vorticity η is calculated from u and w, and equations (6) and (7) are integrated forward by one time step to obtain η and θ' are the next time level. Possion equation (8) is then solved from the new η^{n+1} to obtain stream function ψ^{n+1}. The solution of the Possion equation will be discussed in the next section.

u and w at the next time level are the calculated from (9) to obtain u and w at the next time level. The finite difference formulations are:

$$u_{i,k}^{n+1} = (\psi_{i,k+1}^{n+1} - \psi_{i,k-1}^{n+1}) / (2\Delta \mathrm{z}) \text{ and } w_{i,k}^{n+1} = -(\psi_{i+1,k}^{n+1} - \psi_{i-1,k}^{n+1}) / (2\Delta x). \tag{32}$$

By this time, all variables are the next time level have been obtained. The above steps will be repeat once again, and for a number of steps so that the intended final integration time T is reached.

Solving Poisson equation for streamfunction

The Poisson equation (7) for stream function (contours of stream function are parallel to the velocity vectors everywhere so stream function contours are streamlines!)

$$\frac{\partial^2\psi}{\partial x^2}+\frac{\partial^2\psi}{\partial z^2}=\eta$$

does not contain any time derivative, we call such equations diagnostic equations. Mathematically, it belongs to the class of elliptic PDEs. Elliptic PDEs are boundary value problems (while advection equations are initial value problems), i.e., solutions can be found appropriate boundary conditions.

After vorticity equation (30) has been integrated to the next time level, the right hand side of vorticity at next time step, η^{n+1} is known. The finite difference form of the equation using center-in-space finite differencing is

$$\frac{\psi_{i+1,k}-2\psi_{i,k}+\psi_{i-1,k}}{\Delta x^2}+\frac{\psi_{i,k+1}-2\psi_{i,k}+\psi_{i,k-1}}{\Delta z^2}=\eta_{i,k}^{n+1}. \qquad (27)$$

Notes: The finite difference approximation to the second derivative can be obtained by applying finite difference of first derivative twice:

$$\frac{\partial^2\psi}{\partial x^2}=\frac{\partial}{\partial x}\left(\frac{\partial\,\psi}{\partial x}\right)\approx\frac{1}{\Delta x}\left[\frac{\psi_{i+1,k}-\psi_{i,k}}{\Delta x}-\frac{\psi_{i,k}-\psi_{i-1,k}}{\Delta x}\right]=\frac{\psi_{i+1,k}-2\psi_{i,k}+\psi_{i-1,k}}{\Delta x^2},$$

where $\frac{\psi_{i+1,k} - \psi_{i,k}}{\Delta x}$ is a finite difference to $\frac{\partial \psi}{\partial x}$ defined at (i+1/2, k) point. At that point, it uses ψ half a grid point to the right, and half a grid point to the left, so it is effectively centered difference. The finite difference of $\frac{\partial \psi}{\partial x}$ in the second step is effectively centered difference also.

Eq. (27) can rewritten as

$$\Delta z^2 \psi_{i+1,k} - 2\Delta z^2 \psi_{i,k} + \Delta z^2 \psi_{i-1,k} + \Delta x^2 \psi_{i,k+1} - 2\Delta x^2 \psi_{i,k} + \Delta x^2 \psi_{i,k-1} = \Delta x^2 \Delta z^2 \eta_{i,k}^{n+1},$$

or $$\Delta z^2 \psi_{i+1,k} + \Delta z^2 \psi_{i-1,k} + \Delta x^2 \psi_{i,k+1} + \Delta x^2 \psi_{i,k-1} - 2(\Delta x^2 + \Delta z^2)\psi_{i,k} = \Delta x^2 \Delta z^2 \eta_{i,k}^{n+1}.$$

Let $c_x = \frac{\Delta z^2}{2(\Delta x^2 + \Delta z^2)}$, $c_z = \frac{\Delta x^2}{2(\Delta x^2 + \Delta z^2)}$, $C = \frac{\Delta x^2 \Delta z^2}{2(\Delta x^2 + \Delta z^2)}$, Eq. (27) can be rewritten as

$$\psi_{i,k} = -C\eta_{i,k}^{n+1} + c_x(\psi_{i+1,k} + \psi_{i-1,k}) + c_z(\psi_{i,k+1} + \psi_{i,k-1}). \tag{28}$$

We will use the iterative successive over-relaxation (SOR) method to solve it, which involves many iterations. It will be described in a separate document.