Solving Possion equation for stream function

If you are really impatient and does not want to bother with the derivation of the solution procedure, you can jump to page 7 for the pseudo code.

The Possion 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

$$c_x=\frac{\Delta z^2}{2(\Delta x^2+\Delta z^2)},\ c_z=\frac{\Delta x^2}{2(\Delta x^2+\Delta z^2)},\ C=\frac{\Delta x^2\Delta z^2}{2(\Delta x^2+\Delta z^2)}, \tag{28}$$

Eq. (27) can be rewritten as

$$\psi_{i,k}=-C\eta_{i,k}^{n+1}+c_x(\psi_{i+1,k}+\psi_{i-1,k})+c_z(\psi_{i,k+1}+\psi_{i,k-1}). \tag{29}$$

We will use the iterative successive over-relaxation (SOR) method to solve it, which involves many iterations.

Iterative methods for solving Possion equation

Jacobi iteration method

Strategy: Make an initial guess for $\psi_{i,j}$ at the zeroth iteration, and then apply a correction to the guess so that local error or error of the equation at the grid point in question vanishes. Keeping on doing the correction for successive grid points, and come back again to do more sweeps (iterations) through the grid until the overall error is reduced to a small enough value. The iteration is then considered converged and the solution is found.

Let ψ^v be the solution to Eq.(29) after *v* iterations, whicvh is not completely accurate. Substituting ψ^v into (29) will yield a residual

$$R^v_{i,k} = c_x(\psi^v{}_{i+1,k} + \psi^v{}_{i-1,k}) + c_y(\psi^v{}_{i,k+1} + \psi^v{}_{i,k-1}) - C\eta_{i,k} - \psi^v{}_{i,k}. \qquad (30)$$

The solution is converged if $R^v_{i,j} = 0$. In practice, we can't make *R* exactly zero using iteration methods, but we want $\| R^v_{i,j}\| < \varepsilon$.

The iterative method works by making a change to $\psi^v_{i,j}$ in (30) so that the left hand side residual goes to zero. I.e., we find $\psi^{v+1}_{i,j}$ such that the residual is <u>temporarily</u> reduced to zero at (*i, j*):

$$c_x(\psi^v{}_{i+1,k} + \psi^v{}_{i-1,k}) + c_y(\psi^v{}_{i,k+1} + \psi^v{}_{i,k-1}) - C\eta_{i,k} - \psi^{v+1}{}_{i,k} = 0. \qquad (31)$$

To achieve that, we need, from (31) – (30):

$$\psi^{v+1}_{i,k} = \psi^v_{i,k} + R^v_{i,k} \qquad (32)$$

where $R^v_{i,j}$ is computed from (30).

Or

$$\psi^{v+1}_{i,k} = c_x(\psi^v{}_{i+1,k} + \psi^v{}_{i-1,k}) + c_y(\psi^v{}_{i,k+1} + \psi^v{}_{i,k-1}) - C\eta_{i,k}. \qquad (33)$$

(33) is the formula to obtain solution of ψ through iterations. It is calculated at all grid points.

The iterations are needed because even though the residual *R* has been reduced to zero by (33) at point (*i*, *k*), when ψ at the neighboring points are adjusted in a similar manner, the residual at (*i*, *k*) is messed up (because the equations at each point are not independent)! If one recompute *R* using new values of ψ at the neighboring points, it is no longer zero. But, we do this iteratively, i.e., perform the adjustment repeated, hopefully we can eventually reduce the residual at all points to below certain threshold!

This method can be shown to converge all the time, but unfortunately at a slow rate.

Gauss-Seidel or Sequential or Seccesive Iteration Method

This method is the same as the Jacobi method given above, except for the definition of the residual, which is now given by

$$R^{v}_{i,k} = c_x(\psi^{v}{}_{i+1,k} + \psi^{v+1}{}_{i-1,k}) + c_z(\psi^{v}{}_{i,k+1} + \psi^{v+1}{}_{i,k-1}) - C\mu_{i,k} - \psi^{v}{}_{i,k}. \quad (34)$$

The circled terms are different from the previous formula.

For loops of the form:

for *k* in range (0, nz)

for *i in range (0, nx)*

this essentially means that the newest values of ψ are used in the calculation of *R*.

Successive Over-relaxation (SOR)

SOR is essentially the Gauss-Seidel method augmented by a relaxation parameter α:

$$\psi_{i,k}^{\nu+1} = \psi_{i,k}^{\nu} + \alpha R_{i,k}^{\nu}. \qquad (35)$$

$\alpha > 1$ is usually used to improve convergence. It means that more correction than necessary to make error at the current point zero is made.

For large grids, the optimal value of α is given by

$$\alpha_{opt} = \frac{1}{2} - \frac{\pi}{2\sqrt{2}} \left(\frac{1}{N_x^2} + \frac{1}{N_z^2} \right)^{1/2}. \qquad (36)$$

So after all, ***the pseudo codes*** to solve the position equation of vorticity-stream function can be:

```
iter_max = 30 # maxmimm number of iterations
Calculate α according to (36).
Set ψ = first guess.
```

The first guess can be zero or equal to the value of previous time step (which should be a better guess – the closer it is to the final solution the fewer iterations will be needed to reach convergence)

```
Calculate cx, cy and C according to (28)
for iter in range(1, iter_max+1):
```

(Start iteration loops)

```
    max_change =0
    for k in range (1, nz-1):
```

(not including boundary points which are provided by the boundary conditions)

```
    for i in range (1, nx-1):
```

(not including boundary points which are provided by the boundary conditions)

```
        Calcualte  R^v_{i,k} = c_x(ψ^v_{i+1,k} + ψ^{v+1}_{i-1,k}) + c_z(ψ^v_{i,k+1} + ψ^{v+1}_{i,k-1}) - Cμ_{i,j} - ψ^v_{i,j}
        Calcuate  ψ^{v+1}_{i,k} = ψ^v_{i,k} + αR^v_{i,k}
    max_change = max( max_change,  ψ^{v+1}_{i,k} = ψ^v_{i,k} + αR^v_{i,k})
    If max_change < 10^-5, exit iteration loop.
```

With this algorithm, there is no need to define more than one version of ψ , since the updated value will be used next so the old version does not need to be kept. Therefore, superscript υ and υ+1 can be dropped.

Boundary conditions for the 2D cloud model

We assume a closed model domain of 2560 x 2560 m^2 in x and z directions.

The x boundaries are at $x = 0$ and $x =$ lenx.

The z boundaries are at $z = 0$ and $z =$ lenz.

The number of grid points in x and z are nx and nz, respectively.

$dx =$ lenx/(nx - 1), $dz =$ lenz/(nz - 1).

If using the FFT version of Possion solver, preferred values of nx and nz are $2^n + 1$.

Suggested choices for nx and nz are 65, 129 and 257 for low, medium and high resolution runs, corresponding to dx, dz = 40, 20 and 10 m. Remember, arrays of shape (nx, nz) have indices going from 0 to nx-1, and 0 to nz-1.

The domain has rigid wall boundaries, therefore velocity components normal to the boundaries are zero:

$u(x = 0) = u(x =$ lenx$) = 0$,

$w(z = 0) = w(z =$ lenz$) = 0$.

We also assume the velocity components parallel to the boundaries have zero normal gradient, therefore

$w(x = 0) = w(x = dx)$

$w(x =$ lenx$) = w(x =$ lenx $- dx)$

$u(z = 0) = u(z = dz)$

$u(z =$ lenz$) = u(z =$ lenz $- dz)$.

For stream function, the boundary condition is $\psi = 0$ at all boundaries.

For voriticty η and potential temperature θ', we apply zero normal gradient boundary conditions at all boundaries.