2D Boussinesq Equations

The equation of motion for 2D fluid with Boussinesq approximation:

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+w\frac{\partial u}{\partial z}=-\frac{1}{\rho_0}\frac{\partial p}{\partial x}, \qquad (1)$$

$$\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial x}+w\frac{\partial w}{\partial z}=-\frac{1}{\rho_0}\frac{\partial p}{\partial z}+g\frac{\theta'}{\theta_0}, \qquad (2)$$

where variables, *u*, *w, p, ρ, θ,* are wind components in x and z directions, *p* is pressure, *ρ, θ,* is potential temperature. ρ is constant base state air density, and θ is constant base state potential temperature. $\theta' = \theta - \theta_0$ is the perturbation potential temperature. The second term on the right hand side of (2) is the buoyancy term.

$$\frac{\partial \theta'}{\partial t}+u\frac{\partial \theta'}{\partial x}+w\frac{\partial \theta'}{\partial z}=0 \qquad (3)$$

is the potential temperature conservation equation.

$\frac{\partial \text{Eq.(1)}}{\partial z}-\frac{\partial \text{Eq.(2)}}{\partial z}$ gives the y-vorticity equation:

$$\frac{\partial \eta}{\partial t}+u\frac{\partial \eta}{\partial x}+w\frac{\partial \eta}{\partial z}=-\frac{g}{\theta_0}\frac{\partial \theta'}{\partial x}, \qquad (4)$$

where

$$\eta=\frac{\partial u}{\partial z}-\frac{\partial w}{\partial x}.$$

For this 2D non-divergent flow, the velocity components can be written in term of stream function ψ,

$$u=\frac{\partial \psi}{\partial z}, w=-\frac{\partial \psi}{\partial x}, \qquad (5)$$

so that vorticity

$$\eta=\frac{\partial^2 \psi}{\partial x^2}+\frac{\partial^2 \psi}{\partial z^2}=\nabla^2\psi, \qquad (6)$$

or $$\nabla^2\psi=\eta, \qquad (7)$$

which is an elliptic equation which can be solved for ψ when η is known. In this term project, you are to use equations (3), (4) and (7) to construct a 2D model where prognostic state variables are η and θ' , and ψ is a diagostic state variable.

Eq. (3) and (4) are first integrated forward to obtain η and θ' at the future time level, then elliptic Eq.(7) is solved to obtain ψ . *u* and *w* are then obtained from (5). One time step is then complete.

Term Project Assignment – Part 1: Solve for stream function and velocity given vorticity distribution. Advect the vorticity field with a constant horizontal flow.

Define a computational domain that is *H* deep, and *L* wide, and grid intervals *dx* and *dz*. Then $L = (nx - 1)\, dx$ and $H = (nz - 1)\, dz$ where *nx* and *nz* are the number of grid points defined from the left to right boundaries and from the bottom to top boundaries, respectively.

For the implementation lateral boundary conditions, it is convenient to define one extra column of grid points outside the left and right physical boundaries, so that the grid point index goes from -1 to *nx*+1. With Fortran, you can declare arrays like u(0:nx+1, nz) to match such index. With Python, all arrays have to start from 0 index.

For our term project, we will assume periodic lateral boundary conditions. For example, using Fortran, for *u*,

```
u(0,:) = u(nx-1,:)
u(nx+1,:) = u(2,:)
```

The above conditions should be set for all state variables. The top and bottom boundary conditions are rigid walls, therefore $w = 0$ at $z = 0$ and $z = H$.

Problem 1: Assume zero initial environmental flow, therefore initial stream function is zero everywhere. At the top and bottom boundaries, ψ remains zero always, which is the top and bottom boundary conditions for ψ .

For *H* and *L* = 10,000 m, and $\Delta x = \Delta z = 100$ m,

$$\eta = \eta_0 \cos^2\left(\frac{\pi}{2}\frac{r}{R}\right) \text{ for } r \le R \text{ otherwise } \eta = 0, \qquad (8)$$

where *r* is the radial distance from the domain center, and $R = \frac{1}{4} H$.

For the above vorticity distribution and η_0 = 0.01 s^{-1} solve for the stream function ψ using successive over-relaxation method. Confirm that your results are correct by plotting (within the physical domain without extra boundary points) the original η and η calculated from (6).

Calculate *u* and *w* from (5) and plot the wind vectors in the physical domain.

Problem 2: For constant flow *u* = 10 m/s, *w* = 0, and the initial vorticity given by (8), integrate (4) without the right hand side buoyancy gradient term for 2000 s, and plot your η solutions at 500, 1000, 1500, 2000 s.

Use forward-in-time, upstream-in-space scheme, and Courant number 0.25, 0.5, 0.75, and 1.0 to do the integration. Briefly discuss the results.

Term Project Assignment – Part 2. Integrate potential temperature equation (3) and vorticity equation (4) with an initial thermal bubble and zero vorticity, to simulate rising warm bubble and dropping cold bubble.

Problem 1: Rising warm bubble/dry thermal convection

Set initial η to zero (and therefore *u* and *w* also to zero).

Define an initial thermal bubble by setting initial perturbation potential temperature as

$$\theta' = \theta'_0 \cos^2\left(\frac{\pi}{2} r_T\right) \text{ for } r_T = \left[(x-x_c)^2/R_x^2 + (z-z_c)^2/R_z^2\right]^{1/2} \le 1 \text{ otherwise } \theta' = 0, \quad (9)$$

where (x_c, z_c) is the coordinate of the center of thermal bubble and R_x and R_z are the horizontal and vertical radius of the elliptic thermal bubble.

Set the bubble amplitude θ'_0 = 5 K. Set (x_c, z_c) = (*L*/2, 2000), i.e., center the initial bubble at 2 km above ground and in the center for horizontal domain. Set $R_x = R_z$ = 2000.0 meter. Set *H* = 12,800 m and *L* = 25,600 m, and $\Delta x = \Delta z$ = 100 m.

Use centered difference for the righthand side term of equation (4) with $g = 9.8\ m\ s^{-2}$, $\theta_0 = 300\ K$, and integrate equation (3) and equation (4) using forward-in-time, upstream-in-space scheme.

Step 1: Integrate equations (3) and (4) to obtain θ' and η at the next time level (in the interior domain). Assume that the maximum *u* and *w* will be 10 m s^{-1} to start with chose your time step size Δt that so that $\frac{\Delta t U_{\max}}{\Delta x} \le \frac{1}{\sqrt{2}}$. The $\sqrt{2}$ factor is due to 2D advection. If your integration becomes stable, check your maximum Courant number during the integration and reduce Δt appropriately.

Set the boundary conditions. For θ' and η, we assume zero gradient normal to the boundary for both. The following is a Python function that sets zero gradient condition on all boundaries.

```
def zero_gradient_bc(var,nx,nz):
    var[0,:] = var [1,:]
    var [nz-1,:] = var [nz-2,:]
    var [:,0] = var [:,1]
    var [:,nx-1] = var [:,nx-2]
    return 0
```

For velocity components *u* and *w*, we use the free-slip wall boundary conditions. That is, the normal wind component is zero, and the parallel wind component assumes zero normal gradient. For *u*, a python code would look like:

```
        u[t+1,   :,1] = 0
        u[t+1,:,nx-1] = 0
        u[t+1,   0,:] = u[t+1,   1,:]
        u[t+1,nz-1,:] = u[t+1,nz-2,:]
```

Step 2: Solve elliptic equation (7) to obtain ψ then calculate *u* and *w* as you did in Task 1, using the vorticity you obtained at the next time level. ψ at the boundaries should be set before the elliptic equation is solved. For the current problem, setting ψ = 0 at all boundries is appropriate.

This completes the full time step integration, and you have obtained all state variables at the next time level, from which you intergrate forward to more time steps until end time *T*=2000s.

Given Δt and *T,* the total number of time levels involved in the integration will be $N = T/\Delta t + 1$. For this 2D problem, memory is not an issue so you can declear arrays for all state variables like u(0:nx+1, nz, *N*) so that you have values at all time steps for output and plotting purpose.

For 3D NWP problems, we never save states at all time levels in memory (it will require too much memory). For the current problem, we only need to declare potential temperature and vorticity arrays at 2 time levels, like pt(0:nx+1, nz, *2*), vort(0:nx+1, nz, *2*), and the velocity arrays at one time level, like u(0:nx+1, nz), because within any time level, we don’t need to know the states at more than these number of time levels. In that case, state variables need to be written to disk files periodically, at specified intervals. Current NWP models often write output (forecasts) to disk every hour. Also, at the end if each time step, you need to transfer pt and vort at time level 2 to time level 1, i.e., pt(:, :, 1) = pt(:, :, 2), vort(:, :, 1) = vort(:, :, 2), so that the next time step integration will start again time level 1. You will need a time counter to keep track of the time.

Discuss your results. You should present your figures with captions, and reference your figures in your discussions.

Problem 2: Dropping cold bubble/density current along ground

Set θ'_0 = -15 K, (x_c, z_c) = (L/2, 3000) meter, R_x = 4000, R_z = 2000 meter in equation (9) for the initial cold thermal bubble. Set H = 6,400 m and L = 51,200 m, and Δx = Δz = 100 m. Simulate the dropping cold bubble/density current in Straka et al. (2003) (see https://twister.caps.ou.edu/CFD2023/Straka.pdf). Integrate the equations to 900 s, and output and plot the θ' and u fields using contours every 300s. Choose your time step size so that $\frac{\Delta t U_{\max}}{\Delta x} \le \frac{1}{\sqrt{2}}$. For this problem, the same boundary conditions as for the rising bubble can be used. The streamfunction at the boundaries can be set to zero.

Discuss your results in comparison with the 100 m grid spacing results in Fig. 3 of the Straka paper at 900 s, and the reference solutions using 25 m grid spacing at additional times in Fig.1 of the Straka paper. You should present your figures with captions, and reference your figures in your discussions. Discuss also the time step size you used, and the maximum horizontal and vertical Courant number during your time integration.

Given the large grid size (~ 512 x 64 grid points), the computation is likely slow, if you use python on a low-end laptop!

Term Project Assignment – Part 3. Repeat Part 2, but using third-order Rouge-Kutta time integration combined with second-order centered spatial difference, and include an additional numerical diffusion term in the vorticity and potential temperature equations as follows:

$$\frac{\partial \theta'}{\partial t} + u\frac{\partial \theta'}{\partial x} + w\frac{\partial \theta'}{\partial z} = K\left(\frac{\partial^2 \theta'}{\partial x^2} + \frac{\partial^2 \theta'}{\partial z^2}\right) \quad (10)$$

$$\frac{\partial \eta}{\partial t} + u\frac{\partial \eta}{\partial x} + w\frac{\partial \eta}{\partial z} = -\frac{g}{\theta_0}\frac{\partial \theta'}{\partial x} + K\left(\frac{\partial^2 \eta}{\partial x^2} + \frac{\partial^2 \eta}{\partial z^2}\right). \quad (11)$$

The numerical diffusion term is included to control numerical noise, since the centered difference scheme has large dispersion error.

As introduced in lecture notes, the third-order Rouge-Kutta scheme applied to 1D advection problem involves three steps, as follows.

$$u_i^* = u_i^n - \frac{\Delta t}{3} c \frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x},$$

$$u_i^{**} = u_i^n - \frac{\Delta t}{2} c \frac{u_{i+1}^{*} - u_{i-1}^{*}}{2\Delta x},$$

$$u_i^{n+1} = u_i^n - \Delta t c \frac{u_{i+1}^{**} - u_{i-1}^{**}}{2\Delta x}.$$

Apply the scheme to the 2D advection terms of Eqs. (10) and (11). In the first two steps, include only the advection terms in the time integration to obtain intermediate '**' solutions. In the third step, including the righthand side terms and evaluate them as time level *n*. For the diffusion terms, using centered spatial differencing. Please remember setting boundary conditions after each step. Tip: define separate arrays to store intermediate '*' and '**' solutions and use them in the follow-on steps. Avoid writing the intermediate solutions into the permanent state variable arrays.

Problem 1: Repeat rising warm bubble/dry thermal convection integration with K=75 and 25 $m^2 s^{-1}$, and discuss the results in comparison with those of upstream-forward scheme in relation to their error properties. Discuss the time step size you used, and the maximum horizontal and vertical Courant number during your time integration.

Problem 2: Repeat the dropping cold bubble/density current experiments with K=75 and 25 $m^2 s^{-1}$, and discuss the results in comparison with those of upstream-forward scheme in relation to their error properties. Discuss the time step size you used, and the maximum horizontal and vertical Courant number during your time integration.

Submit your final term project report in a single PDF file, together with your program/codes used to produce the results of Part 1, 2 and 3.