Report 7: Rising Thermal

Name: Adrian Lam Ling Ho
UID: 105211252

Documentation:

I. Task Goal:

In this modelling task, the goal is to solve the 2-D Boussinesq equation system numerically and simulate the evolving behavior of a 2D thermal bubble.

1. Approach:

The basic outline of the complete program is as follows: At first, a potential temperature field θ with a spherical disturbance initialized over the domain. At the beginning of each time step, 2 Jacobians will be calculated (for temperature θ and vertical velocity ω respectively)from the previous vertical velocity, temperature and stream-function fields. These Jacobians will be calculated using the Arakawa Discretization method.

Then, the vertical velocity and temperature fields are marched in time. Afterwards, the solutions will again be advanced in time to include an artificial/numerical diffusion component. For the next step, the vertical velocity and temperature will be used to compute a stream-function by a Poisson SOR solver. At the end of each time step, the final stream function will then have its signs reversed. This process repeats at every time step.

2. Vertical velocity & Temperature Equations:

To do so, the equation system is solved numerically by simultaneously evolving one vertical velocity and one temperature equation in time. **AB3** scheme is used to advance the solutions in time, with the exception of the first 2 time steps, which is carried out by Euler Forward scheme. In each time step, the vertical velocity (denoted by ω) and temperature (denoted by θ) solutions will be used to solve for a stream-function via a SOR Poisson solver.

3. Artificial Viscosity:

To suppress numerical errors, artificial viscosity is also added in the code with a simple operator splitting scheme. The artificial viscosity term is marched in time by the FTCS (forward in time, centered in space) scheme with time step size = 2dT. Both equations use the same diffusivity in both X and Z directions.

4. Recursive Equations:

The recursive equations for advancing θ & ω in time are as follows:

Euler Forward

$$w_g^{n+1} = w_g^n - J_{i,j}^n(\psi, w) \times dT - \frac{g}{\theta_0} \frac{\left(\theta_{i+1,j}^n - \theta_{i-1,j}^n\right)}{2dx} dT$$

$$\theta_{i,j}^{n+1} = \theta_{i,j}^n - J_{i,j}^n(\psi, \theta) \times dT$$

$$AB3: (Stability\ Condition: Courant\ Number \leqslant 0.72)$$

$$w_{g\,(i,j)}^{n+1} = w_{g\,(i,j)}^n - \frac{23}{12}\left(J_{i,j}^n(\psi, w) \times dT + \frac{g}{\theta_0} \frac{\left(\theta_{i+1,j}^n - \theta_{i-1,j}^n\right)}{2dx} dT\right) + \frac{16}{12}\left(J_{i,j}^{n-1}(\psi, w) \times dT + \frac{g}{\theta_0} \frac{\left(\theta_{i+1,j}^{n-1} - \theta_{i-1,j}^{n-1}\right)}{2dx} dT\right) - \frac{5}{12}\left(J_{i,j}^{n-2}(\psi, w) \times dT + \frac{g}{\theta_0} \frac{\left(\theta_{i+1,j}^{n-2} - \theta_{i-1,j}^{n-2}\right)}{2dx} dT\right)$$

$$\theta_{i,j}^{n+1} = \theta_{i,j}^n - \frac{23}{12}\left(J_{i,j}^n(\psi, \theta) \times dT\right) + \frac{16}{12}\left(J_{i,j}^{n-1}(\psi, \theta) \times dT\right) - \frac{5}{12}\left(J_{i,j}^{n-2}(\psi, \theta) \times dT\right)$$

The recursive equations for advancing numerical diffusion in time are as follows:

$$Diffusion: (Forward\ in\ time\ Centered\ in\ Space,\ Step\ size = 2\ dT)$$

$$w_g^{n+1} = w_g^{n-1} + K_x \frac{2dT}{\Delta x^2}\left(w_{i+1,j}^{n-1} - 2w_{i,j}^{n-1} + w_{i-1,j}^{n-1}\right) + K_z \frac{2dT}{\Delta z^2}\left(w_{i,j+1}^{n-1} - 2w_{i,j}^{n-1} + w_{i,j-1}^{n-1}\right)$$

$$\theta_{i,j}^{n+1} = \theta_{i,j}^{n-1} + K_x \frac{2dT}{\Delta x^2}\left(\theta_{i+1,j}^{n-1} - 2\theta_{i,j}^{n-1} + \theta_{i-1,j}^{n-1}\right) + K_z \frac{2dT}{\Delta z^2}\left(\theta_{i,j+1}^{n-1} - 2\theta_{i,j}^{n-1} + \theta\,{}_{i,j-1}^{n-1}\right)$$

5. Boundary Conditions:

In our experiment, the vertical (left-right) boundaries are periodic while the horizontal (up-down) boundaries are walls (Dirichlet boundaries), represented by the equations below:

Dirichlet Boundary Condition (*Horizontal*) + *Periodic Boundary Condition*(*Vertical*) :

$$w(x=0,z)=w\,(x=L_x,z) \quad and \quad w(x,z\ =\ 0)=w\,(x,z\ =\ L_z)=\ 0$$

$$\theta(x=0,z)=\theta\,(x=L_x,z) \quad and \quad \theta(x,z\ =\ 0)=\theta\,(x,z\ =\ L_z)\ =\ \theta_0$$

$$\psi(x=0,z)=\psi\,(x=L_x,z) \quad and \quad \psi(x,z\ =\ 0)=\psi\,(x,z\ =\ L_z)\ =\ 0$$

6. Choice of Time Step:

The maximum Cournant Number and Von Neumann number are calculated as follows :
The choice of time step is contrained by these 2 *quantities in order to obtain stable, non – oscillating solutions.*

$$Maximum\ Courant\ Number\ =\ max\left(\frac{|u_g|\,dt}{dx}+\frac{|v_g|\,dt}{dz}\right)\leqslant 0.72\ (stability\ limit\ for\ AB3)$$

$$Von\ Neumann\ Number\ =\ K_x\frac{2dT}{\Delta x^2}+K_z\frac{2dT}{\Delta z^2}\leqslant\frac{1}{4}\ (no-oscillation-limit)$$

Time Step Size was chosen to be 0.5 *s to satisfy both constraits.*

---

7. Conversion of Time Scale and Velocity Scale from the paper:

Scale conversions from the paper are as follows:

$$U\ =\ \sqrt{2\,r_0\,g\,\frac{d\theta}{\theta_0}}\ =\ \sim 2.859$$

$$Time\ Scale:\ T\ =\frac{r_0}{U}\ =\ 87.337$$

$$\tau\ =\ 2;\ T\ =\ 168.87s$$
$$\tau\ =\ 5;\ T\ =\ 437s$$
$$\tau\ =\ 7;\ T\ =\ 612s$$

The solutions at T = 168.87s, 437s and 612s will be compared with the literature results. Also, the literature uses a time step of 0.06T = ~5.22s, which is not used in our case to prevent instability (Courant Number > 1). In addition, this larger time step might lead to a larger numerical truncation error compared to our run.

8. **Choice of Diffusivity:**

To ensure similarity in setup, the diffusivity used in the numerical scheme is calculated identically to the paper. The exact viscosity coefficient (denoted in the report as v or K_x) was calculated as follows:

Reynolds number:
$Re = 2r_0U/v = 1500$

Artificial Viscosity (*diffusion*)
$v = \sim 0.952999$

The choice for diffusivity value is identical for both vertical velocity and temperature equations.

---

9. **Conservation of Total Temperature:**

$$\iint_{0,0}^{Lx,Lz} \theta \, dxdz = constant$$

In practice, total temperature might not be conserved due to truncation errors of our space discretization scheme

---

Discussion of Results

Summary:

In short, 3 runs were carried out overall to compare with the paper's results.

- **RUN A:** Grid Resolution **201X201**, without diffusion, **Ttot = 620s**
  - Goal: Compare with Fig 3C in paper
- **RUN B**: Grid Resolution **201X201**, with diffusion, **Ttot = 620s**
  - Goal: Compare with Fig 4C in paper
- **RUN C**: Grid Resolution **51x51,** with diffusion, **Ttot = 620s**
  - Goal: Investigate impact of grid resolution on the solution

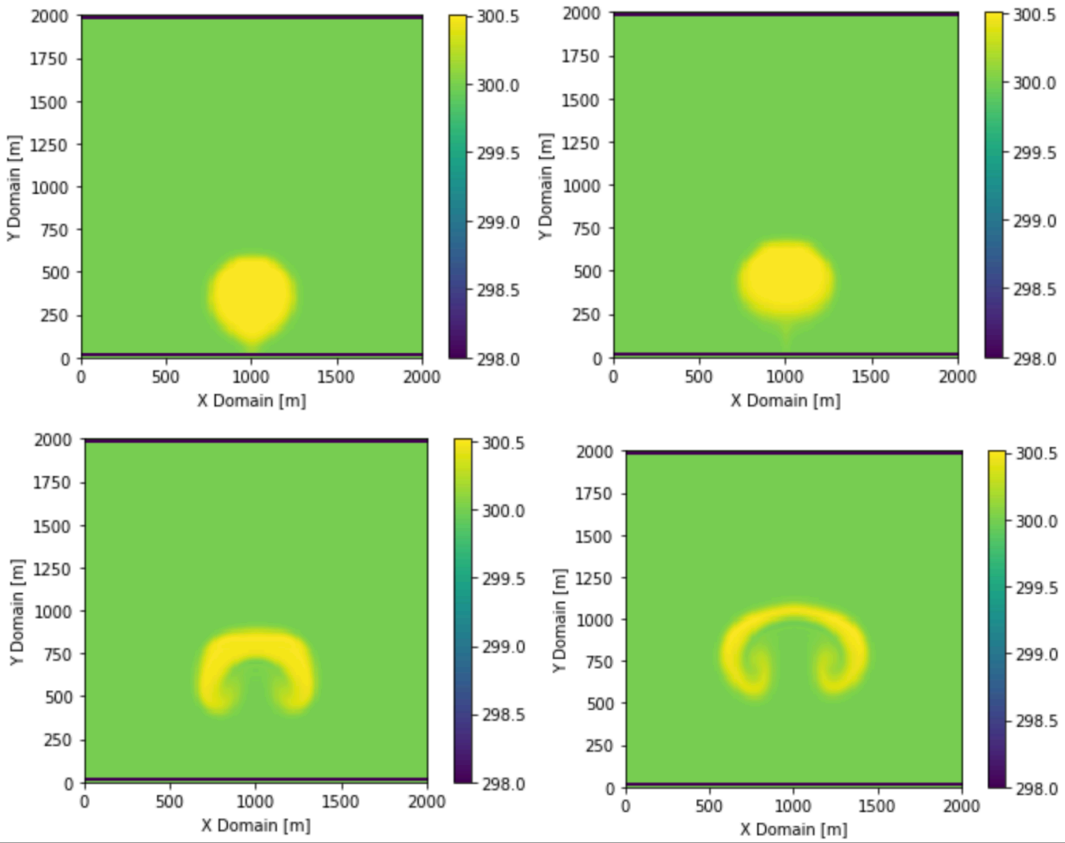
Unfortunately, since the program's runtime is extremely long (6Hrs+), I didn't compute the solutions in 404x404 grid resolutions, nor extend the total integration time beyond 620s.

Basic Observations:
In summary, a thermal bubble was observed to rise (along Z axis) in the domain for all runs, deforming and spreading in both axis over time. However, the presence/absence of diffusion and grid resolution made considerable differences in the quality of the solution. One interesting observation was that across all runs, the total temperature would all slowly drop over time. This is expected since space and time discretization carry small truncation errors that would "overcount" or "undercount" the temperature within the domain throughout the entire integration time. However, this drop would be less than 1% of the total temperature value, thus it would not strongly affect the solution.

It was also observed that there is a layer of strange values along the physical wall boundary only. This was caused by a bug in the enforcement of boundary conditions, which caused numerical errors when summing large arrays in its calculations. However, it appears to be local to the boundary and the solutions still look very similar to the paper.

RUN A: Grid Resolution 201X201, with diffusion, Ttot = 620s

Temperature-Domain Color Plot at T = 170s, 250s, 440s, 615s



In the above graphs, we can see the time evolution for of the thermal bubble with numerical diffusion. Due to its high spatial resolution, one could see the small smooth perturbations within the bubble and the formation of a cavity within the mushroom like bubble. Compared to Figure 4C of the paper (see below), the solution at T = 615s shows a much more uniform temperature distribution within the domain. In addition, regarding the exterior appearance of the bubble, it also appears to be very smooth compared to the Fig 4C. In this solution at T = 615s, there is no central spike at the top of the bubble observable in Fig 4C.

Temperature-Domain Contour Plot at T = 615s

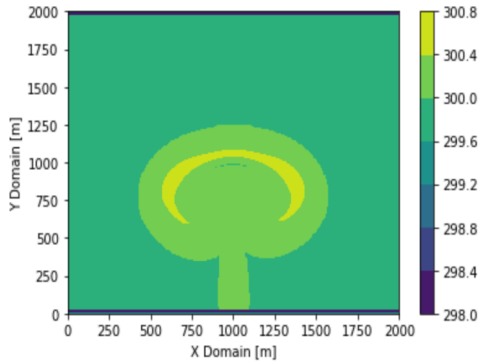
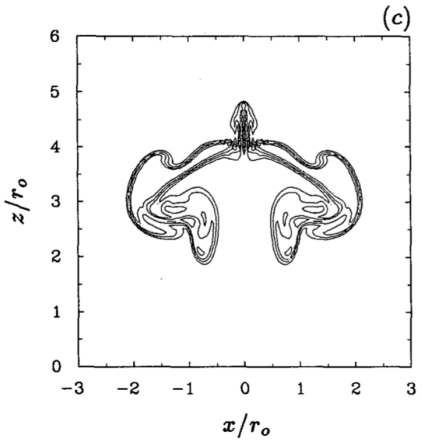


Figure 4C. in Paper



In order to make a better comparison and showing fine ripples in the solution, the temperature field of the solution at T = 615 is plotted again to the right. One can see no observable ripples were travelling behind the rising bubble. This shows that the computational mode is damped rather strongly.

The bottom graph shows the result that we are comparing our solutions with. The solution is also damped with a roughly equal amount of artificial viscosity. We can see considerable temperature heterogeneity within the bubble itself. Also, the upper outer edge of the bubble is rather unsmooth and irregular, which is not the case for the above solutions. Thus, we can postulate that these irregularities are non-physical artefacts (numerical errors, computational mode etc) of the numerical approach used.

RUN B: Grid Resolution 201X201, without diffusion, Ttot = 620s

Temperature-Domain Color Plot at T = 170s, 250s, 440s, 615s

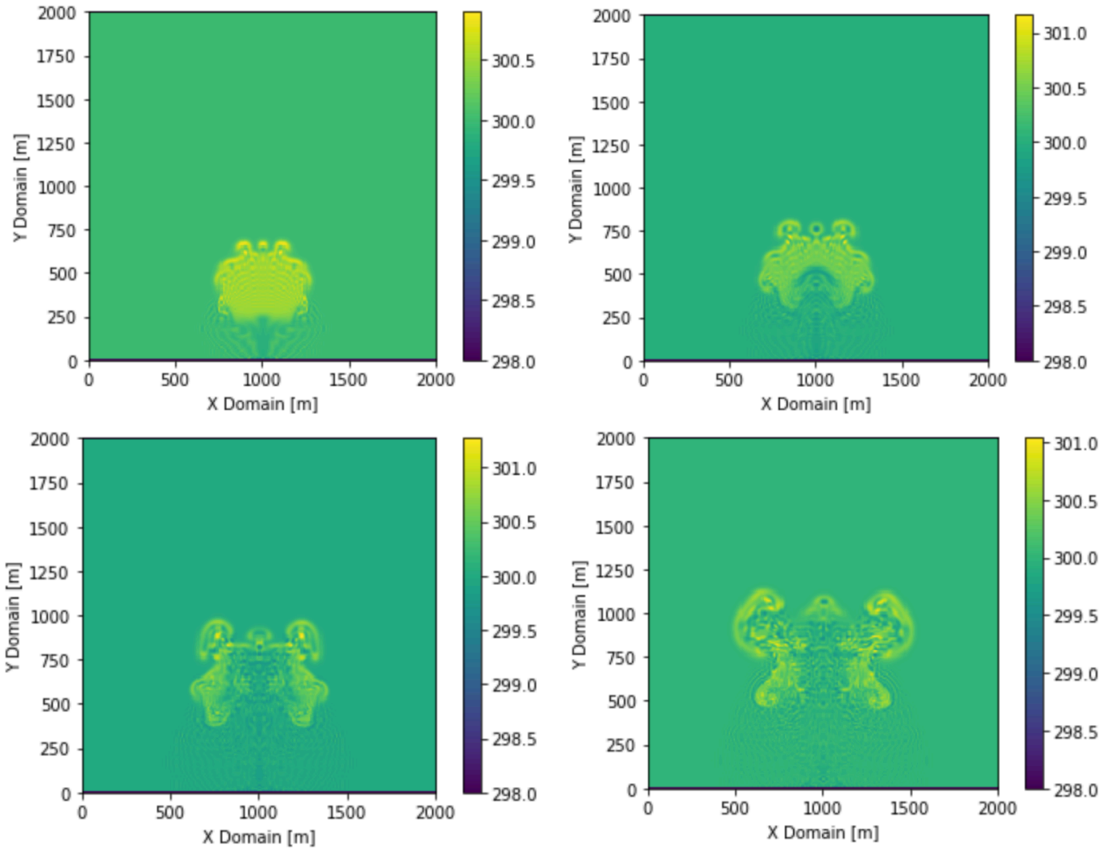
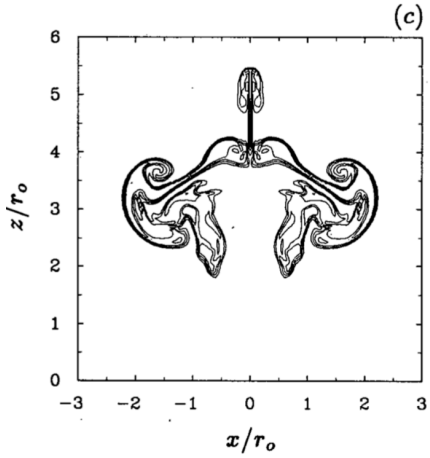


Figure 3C

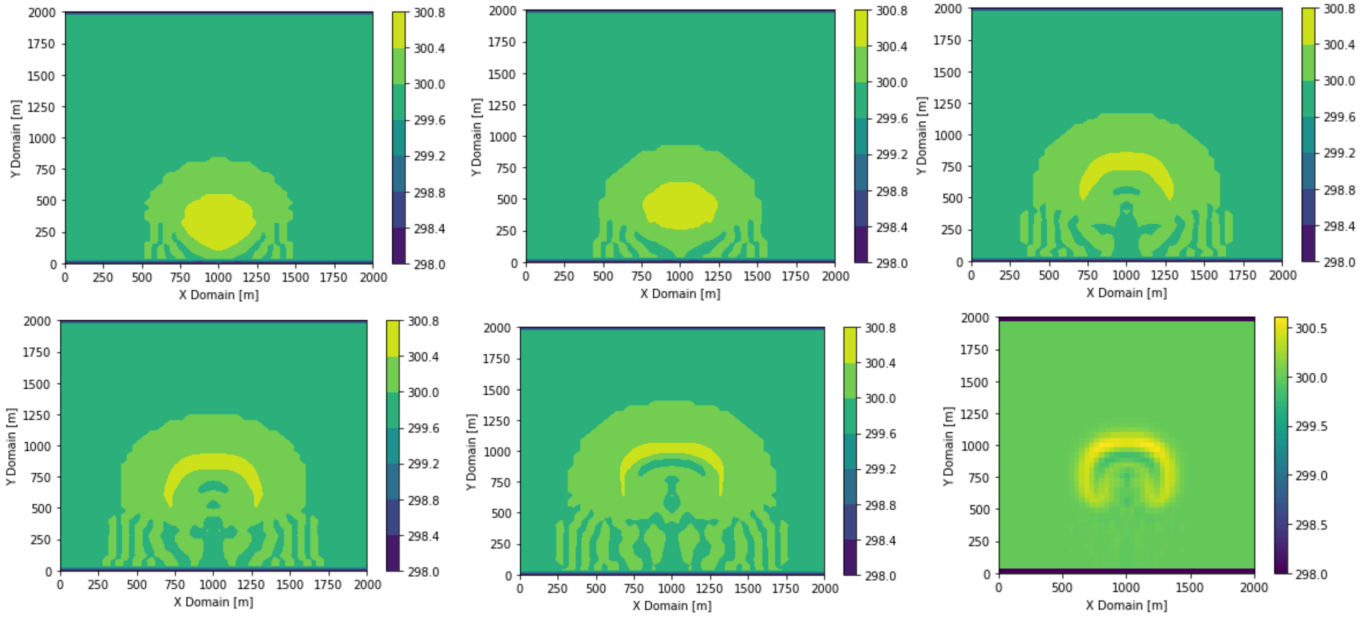


The above graphs show the time evolution of the base case solution without artificial viscosity. At T = 615s can immediately see the irregularities in the exterior of the bubble comparable to Figure 3C, which shows the solution at the same time (after converting time scales). Quantitatively, we can see the irregularity and roughness in both the exterior and interior of the bubble. Also, we can see that all regions of the bubble are heterogenous in temperature in our solution without numerical diffusion. Our solution is mostly consistent with figure 3C with the exception of the absence of Fig 3C's large protruding spike at the top center.

Comparing the solutions with and without diffusion, we can see a fair comparison that shows the effect of numerical diffusion. Thus, we can conclude that numerical diffusion is highly effective in removing local temperature heterogeneities in our solution at a particular grid resolution.

RUN C: Grid Resolution 51X51, with diffusion, Ttot = 620s

Temperature-Domain Color Plot at T = 170s, 250s, 440s, 500s, 615s, and Contour Plot at T = 615s



The above contour plots show the time evolution of the bubble throughout the entire integration time for grid resolution of 51x51. For this run, the previously-used color-pixel plot was considered to show the solution, but due to the lack of resolution, contour plot was used to better demonstrate the behavior of the solution.

Firstly, we can immediately observe that the solution contains ripples-like perturbations behind as it advects upwards. One potential explanation for these “ripples” was that they’re caused by computational modes of AB3. This is because the numerical values of these “ripples” were oscillating with an interval roughly equal to the spatial resolution. Also, it appears to be travelling opposite to main bubble’s advection direction. Hence, this fits both the oscillating and travelling behavior of the computational mode as we know it.

The solution at T = 615s is also plotted with color-pixel plot at the end, from which we can also roughly observe the temperature heterogeneity at the bottom of the bubble.

An important comparison is that even with a very poor grid resolution, we can see the addition numerical diffusion helps considerably in removing most irregularities of the bubble’s shape.

Thus, regardless of grid resolution, we can see that adding numerical diffusion plays a huge role in removing major numerical artefacts in the solution.

Capabilities:

The advantages of the current simulation over the literature are summarized by the following points:

- Comparing to the numerical scheme in the literature, we can see that the AB3 solutions has much less prominent numerical errors. This is either due to AB3's damping of high frequency computational mode or its higher order truncation error compared to literature.

- The inclusion of numerical diffusion visibly damped some numerical errors and removed some of the non-physical features of the bubble.

Limitations:

- **Runtime**

  Most likely due to being coded in python, the major limitation for this program is its very long runtime (~6 hrs without diffusion, +8hrs with diffusion) even with the basic grid resolution (201x201). Thus, I didn't have the opportunity to run repeated runs/finetune the program.

  In addition, in this task, it was not possible for me to speed up the SOR process by not overwriting stream-function with 0 at every time step. This is because after exiting the SOR loop, the stream-function's signs were reversed to satisfy the current equation. Thus, the error of the first guess stream-function rises again at the start of every time step. In my case, the number of Poisson iterations per time step was roughly similar (~200) either with or without resetting the stream-function to 0.

- **Numerical errors:**

  Despite using a higher order time discretization scheme, this program still shows some numerical errors due to the nonlinear advection errors (aliasing) and computational modes from AB3. In addition, this also manifests as a net change in conserved quantities like temperature. This becomes more obvious as grid resolution decreases, for instance, the computational modes became the most noticeable in 51x51 grid resolutions.