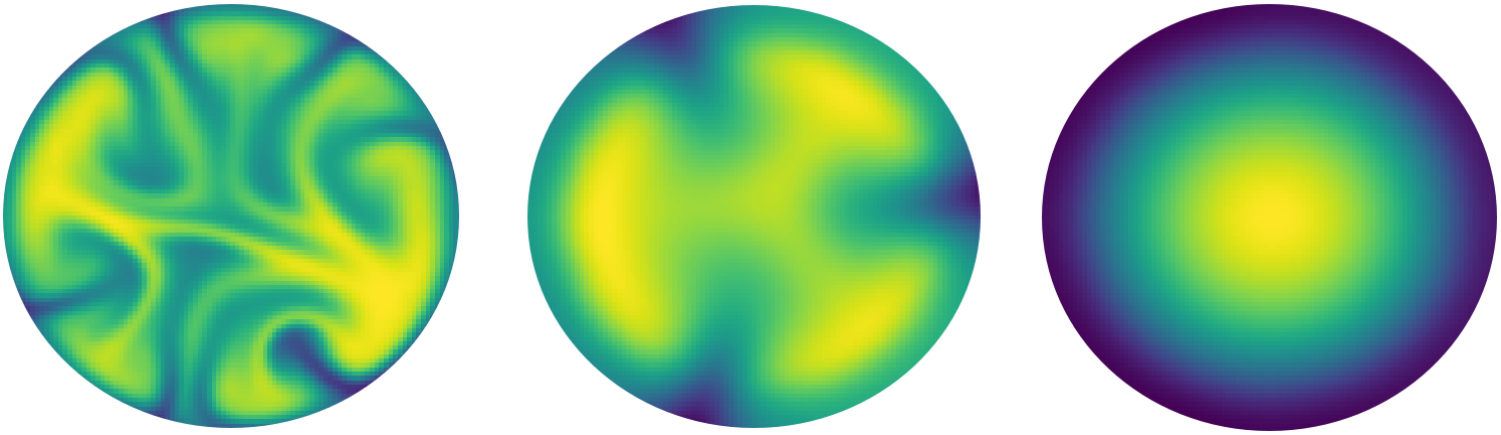
A Numerical Analysis of Convection in the Inner Core

Third Year Physics ASC
Supervised by Professor Louis Moresi
Start Date: 16th of August 2021 End Date: 29th of October 2021

Maximilian Williams (u6338634)

October 2021



Abstract

Whether The Earth's inner core convects or not has been a contentious topic in geoscience. Here we numerically model the Earth's inner core as a self-gravitating internally heated fluid in 2-dimensions using a streamfunction-vorticity method and a Thermal Lattice Boltzmann approach coded from scratch in Python3 and goLang. By comparing results of two simple streamfunction-vorticity codes we conclude that the polar geometry of our 2-dimensional inner core model and the central region are important in modelling convection. Using a Thermal Lattice Boltzmann method (TLBM) we simulate the entire domain while respecting the geometry. Our TLBM simulations are in the high Prandtl number limit with $Pr = 5000$ and give an upper bound on the critical internally heated Rayleigh number of $Ra_H = 10^5$ which agrees with other work [2].

Introduction

The Earth's inner core is a particularly interesting subterranean region as it helps maintain Earth's magnetic field which shields life on Earth from solar radiation. Yet this critical region of the Earth is relatively poorly understood. It sits within the liquid outer core, which shares a boundary with a sharp density change with the lower mantle and is opaque to S-waves [6]. This makes seismic signals which pass through the inner core such as PKJKP and PKIKP waves difficult to detect [17, 3]. The inner core is also one of the most extreme environments, with temperatures exceeding 5000°C and pressures exceeding 3 million atmospheres which are largely outside of experimental material physics, making composition and physical properties difficult to obtain. Besides these difficulties, modelling convection in the inner core is also non-trivial due to internal heating and non constant gravity. Emulating varying gravitation and internal heating directly in experiment is difficult. For this reason, the problem of convection in the inner core is well suited to numerical methods.
In this report, a simplified model of the inner core and accompanying theory is presented. Two numerical methods for simulating this model, the streamfunction-vorticity method and Thermal Lattice Boltzmann method (TLBM) are introduced and their implementations described. An advection test is performed on the streamfunction-vorticity code while a full convection simulation comparison is done between the TLBM implemented here and other work [12]. Results from the streamfunction-vorticity code are presented showing the central region and geometry of the inner core model are important. Finally, the TLBM code is used to produce high Prandtl number simulations with varying internally heated Rayleigh numbers to determine an upper bound on the critical Rayleigh number for the model. The report concludes with a discussion of difficulties in using the TLBM, possible improvements and extensions for future work.

All codes produced for this report are available here or at the url: `https://github.com/maxzwilliams/ASC2021`

Physical Picture

Here I talk about what I am modelling and my assumptions about the inner core.

Throughout this report, we approximate the Earth's inner core as a perfect liquid sphere and model a 2-dimensional cross-section. We take the inner core as being internally heated except for a thin outer layer that is cooled. We ignore all affects of rotation or precession, assuming the core is in an inertial reference frame. We assume that the earth above the inner core is spherically symmetric in all respects and so model no variations in pressure and take the core to be self gravitating.
To simulate this model, we investigate two geometries. The first is a polar geometry shown in figure 1 (top) and is a more accurate depiction of the model. The second, is a Cartesian depiction of the model (figure 1 (bottom)) which is easier to simulate, but does not respect the geometry. It can be thought of as cutting the polar model at the $0, 2\pi$ azimuthal boundary and stretching it out.

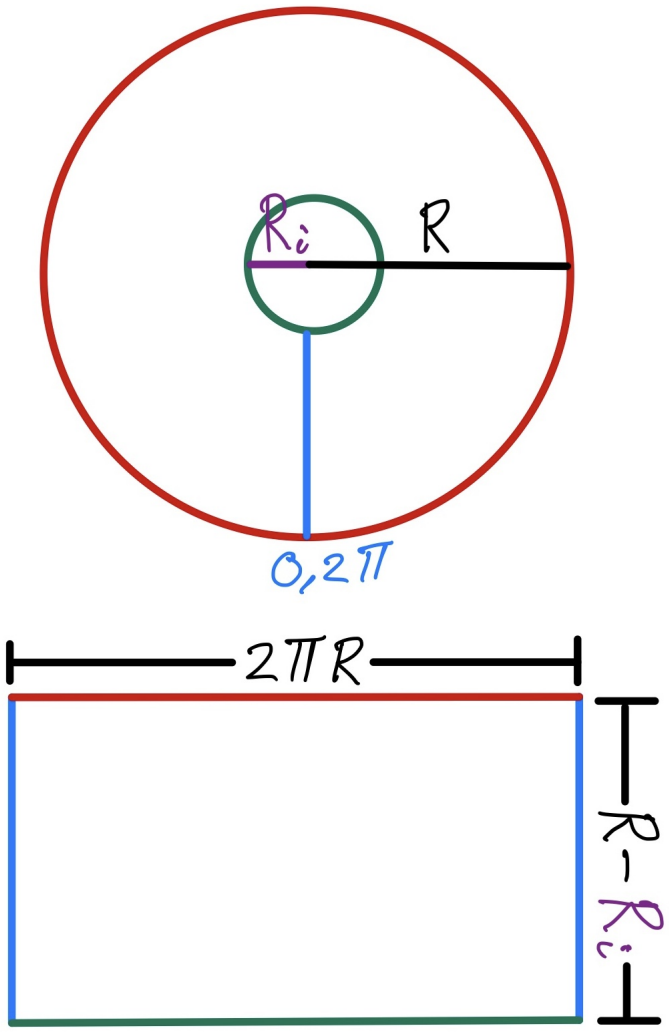


Figure 1: Figure showing how the 2-dimensional polar domain (top) is cut at the azimuthal angle $0, 2\pi$ boundary and stretched into the Cartesian geometry (bottom). Sides are color coded. R and R_i are the outer and inner radii of the polar domain.

Governing Equations

In this section I introduce the physics of the problem; the governing equations that we wish to numerically solve.

We describe the fluid in the inner core in the Eulerian frame under a graviational acceleration $\vec{g}$ which varies in space. We give each location in the fluid a time and space dependent velocity $\vec{u}$ and density ρ. We assume that the fluid is incompressible and its viscosity μ and thermal diffusivity κ are constant. By conserving fluid momentum, we produce the Navier-Stokes equation:

$$\rho \frac{D\vec{u}}{Dt} = \rho\vec{g} - \nabla p + \mu\nabla^2\vec{u}, \tag{1}$$

where $\frac{D}{Dt} = \frac{\partial}{\partial t} + (\vec{u} \cdot \nabla)$ is the material derivative, p the pressure and ∇ the del operator. The dynamics of fluid temperature T are described the inhomogenous advection diffusion equation:

$$\frac{DT}{Dt} = \kappa \nabla^2 T + H, \tag{2}$$

where H is internal heating [8]. The density of the fluid ρ is assumed to vary linearly in temperature according to the equation of state:

$$\rho = \rho_0(1 - \alpha(T - T_0)), \tag{3}$$

where α is the volumetric expansion coefficient and ρ_0 the density at a reference temperature T_0. We assume the density variations are small and that the inner core is evolving on geologic timescales, allowing us to take the velocities $\vec{u}$ to be small. These assumptions give the slow flow Boussinesq approximation [8, 18] to equation 1:

$$\frac{\partial \vec{u}}{\partial t} = \frac{\rho'}{\rho}\vec{g} - \frac{\nabla p'}{\rho_0} + \nu \nabla^2 \vec{u}, \tag{4}$$

where $\rho' = -\alpha(T - T_0)$, ν the kinematic viscosity $\nu = \frac{\mu}{\rho_0}$ and p' a first order perturbation to the background pressure p_0.

$$Pr = \frac{\nu}{\kappa} \tag{5}$$

We use two numbers to characterise the problem, the Prandtl number and the Rayleigh number [8]. The Prandtl number Pr (equation 5) is the ratio of momentum and thermal diffusivity. The Rayleigh number (Ra) (equation 6) is the ratio of timescales for thermal diffusion and thermal convection.

$$Ra = \frac{g\alpha \Delta T d^3}{\nu \kappa}, \tag{6}$$

Here ΔT is the temperature variation over the length scale of the problem d. Rayleigh numbers higher than the critical Rayleigh number imply thermal convection. For internally heated problems such as ours, the temperature variation is poorly defined and so we use $\Delta T = \frac{Hd^2}{\kappa}$ instead, giving the internally heated Rayleigh number [8]:

$$Ra_H = \frac{g\alpha H d^5}{\nu \kappa^2}. \tag{7}$$

Numerical Methods

Two numerical methods are introduced for solving the thermal convection problem, the Lattice Boltzmann method and the streamfunction vorticity formulation.

Streamfunction-Vorticity Formulation

Here I introduce the streamfunction-vorticity method for use in 2 dimensions and use it to eliminate pressure terms in 4 and 2 in Cartesian and polar geometries giving a set of numerically solvable equations.

The streamfunction-vorticity formulation is a popular method for analytical and simple numerical analysis of incompressible fluids in two dimensions. Its main advantage is its elimination of all pressure terms, which would typically require iterative techniques to solve, an example being the SIMPLE algorithm and its derivatives [5]. However, the streamfunction-vorticity method is limited to 2-dimensional and 3-dimensional symmetric flows and so has limited applications.

We define scalar vorticity ω by:

$$\omega = (\nabla \times \vec{u})_z, \tag{8}$$

and streamfunction ψ by:

$$\omega = -\nabla^2 \psi, \tag{9}$$

where $\nabla\times$ is the curl and subscript z denotes component out of the simulation plane. Given a coordinate system, and a clever definition of velocities $\vec{u}$ we can rewrite equations 2 and 4 in terms of ω and ψ rather than $\vec{u}$ and p [18]. In Cartesian coordinates (x, y) with $\vec{u} = u\hat{x} + v\hat{y}$ we pick

$$u = \frac{\partial \psi}{\partial y}, v = -\frac{\partial \psi}{\partial x}, \tag{10}$$

allowing us to write equations 2 and 4 as:

$$\frac{\partial T}{\partial t} + \frac{\partial \psi}{\partial y}\frac{\partial T}{\partial x} - \frac{\partial \psi}{\partial x}\frac{\partial T}{\partial y} = \kappa \nabla^2 T + H \tag{11}$$

$$\frac{\partial \omega}{\partial t} = -\frac{g_y}{\rho_0}\frac{\partial \rho'}{\partial x} + \nu \nabla^2 \omega \tag{12}$$

Similarly, in polar coordinates (r, θ) with $\vec{u} = u\hat{r} + v\hat{\theta}$ we pick:

$$u = \frac{1}{r}\frac{\partial \psi}{\partial \theta}, v = -\frac{\partial \psi}{\partial r}, \tag{13}$$

giving:

$$\frac{\partial T}{\partial t}+\frac{1}{r}\frac{\partial \psi}{\partial \theta}\frac{\partial T}{\partial r}-\frac{1}{r}\frac{\partial \psi}{\partial r}\frac{\partial T}{\partial \theta}=\kappa\nabla^2 T+H \qquad (14)$$

$$\frac{\partial \omega}{\partial t}=-\frac{g_r}{\rho_0 r}\frac{\partial \rho'}{\partial \theta}+\nu\nabla^2\omega. \qquad (15)$$

Importantly, our definitions of u and v in equations 10 and 13 satisfy equation 8 and 9. Equations 9, 11, 12 for the Cartesian case and 9, 14, 15 for the polar case can be directly solved.

Solving The Streamfunction-Vorticity Equations

The finite difference method used to solve the streamfunction-vorticity-formulated governing equations in my streamfunction-vorticity codes for Cartesian and polar geometries is described.

We first discretize our domain $\mathcal{D}$. In the Cartesian case, we use $(x_i, y_j) = (i\Delta x, j\Delta y)$ with integers i and j satisfying $0 \leq i < N_x$ $0 \leq j < N_y$. In the polar case, we use $(r_i, \theta_j) = (R_0 + i\Delta r, j\Delta\theta)$ again with $0 \leq i < N_r$ and $0 \leq j < N_\theta$. We impose $\Delta\theta = \frac{2\pi}{N_\theta - 1}$ for consistency with θ-periodic boundary conditions and an inner radius R_0 in polar coordinates to avoid singularities at $r = 0$. This inability to model the innermost region is an important deficiency of the streamfunction-vorticity method in polar coordinates. We discretize time t by $t_n = n\Delta t$. For a function f, we use $f^n_{i,j}$ to denote f evaluated at time n at position (x_i, y_j) in Cartesian coordinates or (r_i, θ_j) in polar coordinates.
To approximate derivatives we use a finite difference approach. All time derivatives are approximated by forward difference [13]:

$$\frac{\partial f}{\partial t}=\frac{f^{n+1}-f^n}{\Delta t} \qquad (16)$$

Second order space derivatives are approximated by a central difference [13]:

$$\frac{\partial^2 f_{i,j}}{\partial {x_1}^2}=\frac{f_{i+1,j}-2f_{i,j}+f_{i-1,j}}{\Delta {x_1}^2}, \qquad (17)$$

$$\frac{\partial^2 f_{i,j}}{\partial {x_2}^2}=\frac{f_{i,j+1}-2f_{i,j}+f_{i,j-1}}{\Delta {x_2}^2}, \qquad (18)$$

where x_1 is the first coordinate and x_2 is the second coordinate. For example, in Cartesian coordinates (x, y), we would have $x_1 = x$ and $x_2 = y$. For non advection terms, we approximate first order spatial derivatives by [13]:

$$\frac{\partial f_{i,j}}{\partial x_1}=\frac{f_{i+1,j}-f_{i-1,j}}{2\Delta x_1}, \qquad (19)$$

and

$$\frac{\partial f_{i,j}}{\partial x_2}=\frac{f_{i,j+1}-f_{i,j+1}}{2\Delta x_2}. \qquad (20)$$

For advection terms, of the form $a\frac{\partial f_{i,j}}{\partial x_1}$ we employ a first order Godunov scheme [7, 9]:

$$a\frac{\partial f_{i,j}}{\partial x_1}=\frac{1}{\Delta x}(\mid a \mid (\frac{1}{2}f_{i+1,j}-\frac{1}{2}f_{i-1,j})-a(\frac{1}{2}f_{i+1,j}-f_{i,j}-\frac{1}{2}f_{i-1,j})). \qquad (21)$$

This scheme is always upstream, regardless of the direction of the advecting field a and is stable so long as $\mid a \mid \frac{\Delta t}{\Delta x} \leq 1$. Using this Godunov scheme, the modified equivalent partial differential equation for the advection diffusion equation shows a numerical diffusion with constant $\kappa' = \mid a \mid \frac{\Delta x}{2}(1 - \mid a \mid \frac{\Delta t}{\Delta x})$ which can be made small relative to the diffusion constant κ by taking a fine mesh ($\Delta x \to 0$).
A lot of time in this project was spent getting a stable and accurate advection scheme. Simple schemes for advection were tried such as Lax-Wendroff, central difference and forwards/backwards Euler. These schemes either introduced large numerical diffusion or were unconditionally unstable. Other more accurate, but substantially more complex methods for solving these equations, particularly the advection equation, exists such as the Semi-Lagrange Crank-Nicholson scheme [16].
To solve the streamfunction-vorticity equations we assume a starting vorticity ω on our domain $\mathcal{D}$. We then apply the Jacobi method (given in appendix) to solve equation 9 for ψ on the interior of the domain which we call $\mathcal{D}'$. Using ψ we update T on $\mathcal{D}'$ using equation 11 (or 14 for polar). Finally, ω is updated on $\mathcal{D}'$ using equation 12 (15 for polar) [1]. This process is repeated.
Using this streamfunction-vorticity method we simulate the polar (figure 1 (top)) and Cartesian (figure 1 (bottom)) model.

Lattice Boltzmann Method

In this section I give a brief introduction to the Lattice Boltzmann Method and why its different from traditional techniques. I introduce a 2-dimensional lattice D2Q9 and describe the Thermal Lattice Boltzmann method (TLBM) which is employed in my code to simulate convection. I then show how the Lattice Boltzmann method can be used to simulate the polar model while still maintaining the inner region which the streamfunction-vorticity polar model lacks.

The Lattice Boltzmann Method (LBM) is a generalization of a Lattice Gas Automata (LGA), which are themselves a specialized Automata for simulating fluid flows [15]. These Automata methods, like common fluid simulation techniques, discretize space and time. They directly simulate the state of particles or their distributions and evolve in time according to simple rules which give

the desired macroscopic fluid properties as an emergent effect [19]. This is fundamentally different to typical approaches which amount to directly numerically solving a set of partial differential equations. In the Thermal Lattice Boltzmann Method (TLBM) the internal energy is also simulated and used to simulate thermal affects which couple to the motion of the particles.
To discretize space, we place nodes at locations $(x_i, y_j) = (i, j)$ with i, j integers. Each node has attached to it a lattice, here the D2Q9 lattice shown in figure 2. The lattice defines unit vectors $\vec{e}_i$, $i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$ [12]. In addition, each direction e_i gets a weight w_i [12]. In the D2Q9 lattice these are:

$$w_i = \begin{cases} \frac{4}{9} & \text{if } i = 0 \\ \frac{1}{9} & \text{if } i = 1, 2, 3, 4 \\ \frac{1}{36} & \text{if } i = 5, 6, 7, 8 \end{cases}.$$

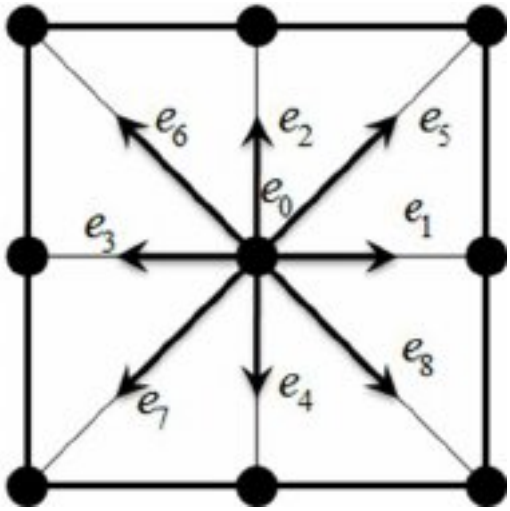


Figure 2: D2Q9 Lattice. Black nodes represent lattice points, vectors $e_0, .., e_8$ are the lattice vectors. Image sourced from [10]

We wish to simulate convection. For this, we require the particle motion and the internal energy throughout the lattice. We define two distribution functions $f_\alpha(\vec{x}, t)$ and $g_\alpha(\vec{x}, t)$ denoting the particle and internal energy distributions along direction α at lattice points $\vec{x}$ and times t. The direction can be thought of as the direction of flow for particles or energy. Each timestep there are two steps to updating f and g, a streaming step:

$$f_\alpha(\vec{x} + \vec{e}_\alpha, t + \Delta t) = f_\alpha(\vec{x}, t), \quad (22) \qquad g_\alpha(\vec{x} + \vec{e}_\alpha, t + \Delta t) = g_\alpha(\vec{x}, t), \quad (23)$$

and a collision step:

$$f_\alpha(\vec{x}+\vec{e}_\alpha, t+\Delta t) = f_\alpha(\vec{x}, t)+\frac{1}{\tau_f}(f_\alpha^{eq}(\vec{x}, t)-f_\alpha(\vec{x}, t))+F_\alpha \quad (24) \qquad g_\alpha(\vec{x}+\vec{e}_\alpha, t+\Delta t) = g_\alpha(\vec{x}, t)+\frac{1}{\tau_g}(g_\alpha^{eq}(\vec{x}, t)-g_\alpha(\vec{x}, t))+G_\alpha. \quad (25)$$

Here F_α and G_α are the forcing terms terms and f_α^{eq} and g_α^{eq} are equilibrium distributions. The relaxation times τ_f and τ_g are related to the macroscopic thermal diffusivity (κ) and kinematic viscosity ν by [12]:

$$\tau_g = \frac{3\kappa}{S^2\Delta t} + \frac{1}{2}, \quad (26) \qquad \tau_f = \frac{3\nu}{S^2\Delta t} + \frac{1}{2}. \quad (27)$$

The equilibrium distributions are given by the BGK approximation [14, 15]:

$$f_\alpha^{eq}(\vec{x}, t) = \rho w_\alpha(1 + 3\frac{\vec{e}_\alpha \cdot \vec{u}}{s^2} + \frac{9}{2}\frac{(\vec{e}_\alpha \cdot \vec{u})^2}{c^4} - \frac{3}{2}\frac{\vec{u} \cdot \vec{u}}{c^2}), \quad (28) \qquad g_\alpha^{eq}(\vec{x}, t) = \epsilon\rho w_\alpha(1 + 3\frac{\vec{e}_\alpha \cdot \vec{u}}{s^2} + \frac{9}{2}\frac{(\vec{e}_\alpha \cdot \vec{u})^2}{c^4} - \frac{3}{2}\frac{\vec{u} \cdot \vec{u}}{c^2}) \quad (29)$$

Here ρ, ϵ and $\vec{u}$ are the macroscopic density, internal energy and velocity given by [12]:

$$\rho = \sum_{i=0}^{8} f_i, \quad (30) \qquad \rho\vec{u} = \sum_{i=0}^{i=8} f_i\vec{e}_i \quad (31) \qquad \rho\epsilon = \sum_{i=0}^{i=8} g_i. \quad (32)$$

The forcing terms F_i and G_i are problem dependent. For thermal convection $G_i = 0$ and F_i is a gravitational term Δf_α [12]:

$$\Delta f_\alpha = -\frac{w_\alpha \rho \alpha \epsilon}{\tau_{grav}} \frac{\vec{e}_\alpha}{|\vec{e}_\alpha|} \cdot \vec{g}, \tag{33}$$

where $|\cdot|$ is the vector norm and $\vec{g}$ is the gravitational acceleration. Here τ_{grav} is the relaxation time for the gravitational field. We take $\tau_g = 0.6$ here, following [12]. The algorithm used to evolve the above TLBM equations is given in algorithm 1 and closely follows [12].

To simulate our model using the Lattice Boltzmann method, we define a Cartesian domain in which we place a circular boundary. We view this internal circular region as our model, and simulate gravity, internal heating/cooling and our fluid within it. This is seen in figure 3. This is possible only due to the ease of setting boundary conditions in the TLBM and can not easily be done in the streamfunction-vorticity formulation. Critically, this simulation respects the polar geometry of our model while simulating its central region, which neither the polar or Cartesian streamfunction-vorticity simulations do simultaneously.

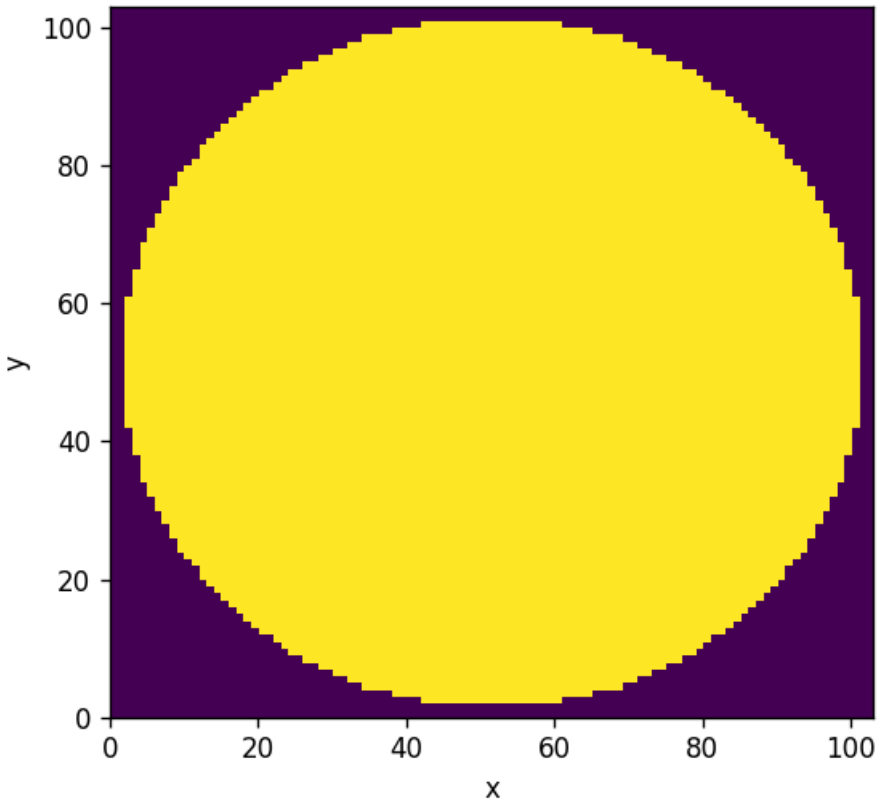


Figure 3: Lattice Boltzmann code simulated on a Cartesian domain with circular boundary. Inside the circular region (yellow) the fluid is simulated while the exterior (purple) contains nothing and is not relevant to the simulation.

Boundary Conditions

In both streamfunction-vorticity and Thermal Lattice Boltzmann code, the boundaries are fluid-impermeable, non-slip and thermally insulating. In the streamfunction-vorticity code this is done by directly imposing that the streamfunction (ψ) is constant on the boundaries and manually setting tangential velocities to zero. From this ω boundary conditions arise. In the Thermal Lattice Boltzmann code, these conditions are imposed by "bounce back" boundary conditions [11] where during the streaming step (equation 22) distributions f and g are reflected off the boundaries. These boundary conditions were selected as they are easily implemented in both codes, particularly the Lattice Boltzmann codes. To generate a cooling boundary condition, the heating field near the boundary was set negative. This method is crude and necessary to avoid greater complexity in the Lattice Boltzmann case, but should in future be replaced with the traditional method of fixing heat flux across the boundary in the streamfunction case.

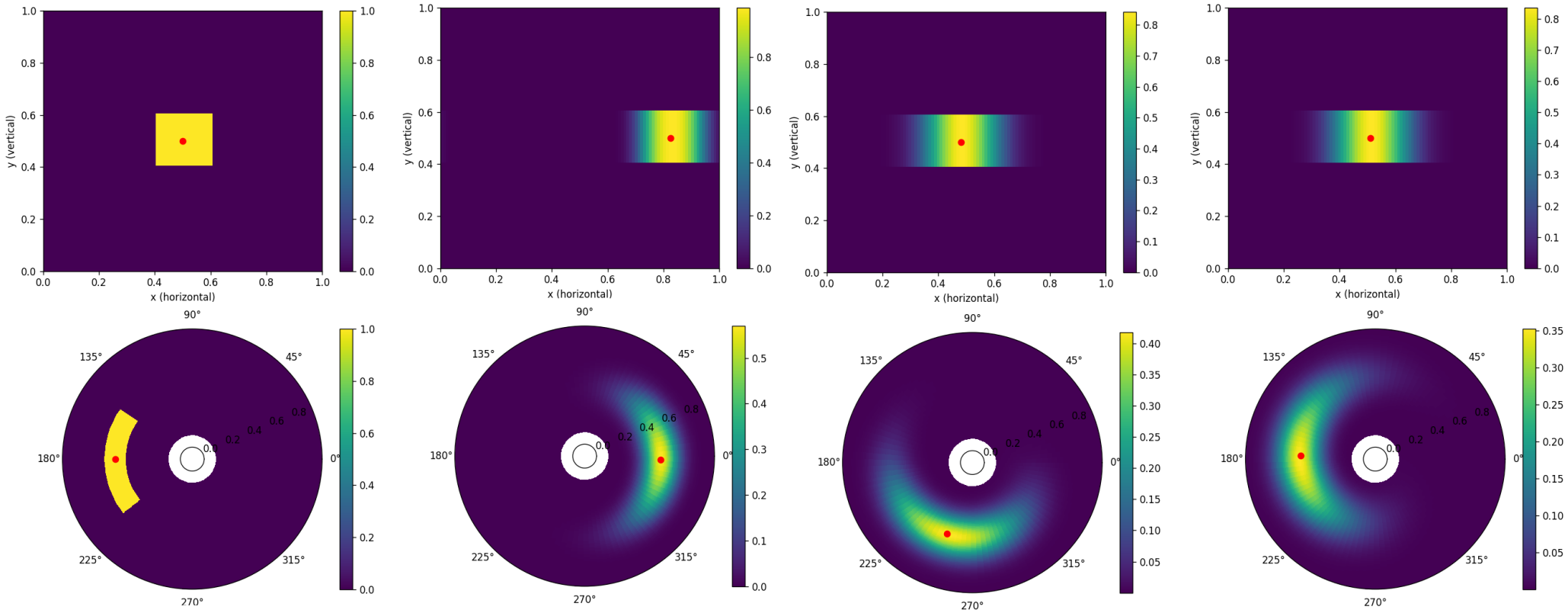


Figure 4: Streamfunction-vorticity Cartesian (top) and polar (bottom) advection tests for horizontal (top) and azimuthal (bottom) advection. Advection fields move left to right (top) and clockwise (bottom). Red dot is centroid location, temperature indicated by colour. First column is the initial state, second column is crossing of periodic boundary, third column is time when an exact scheme would return to the initial state, fourth column is advected field after returning to its initial location.

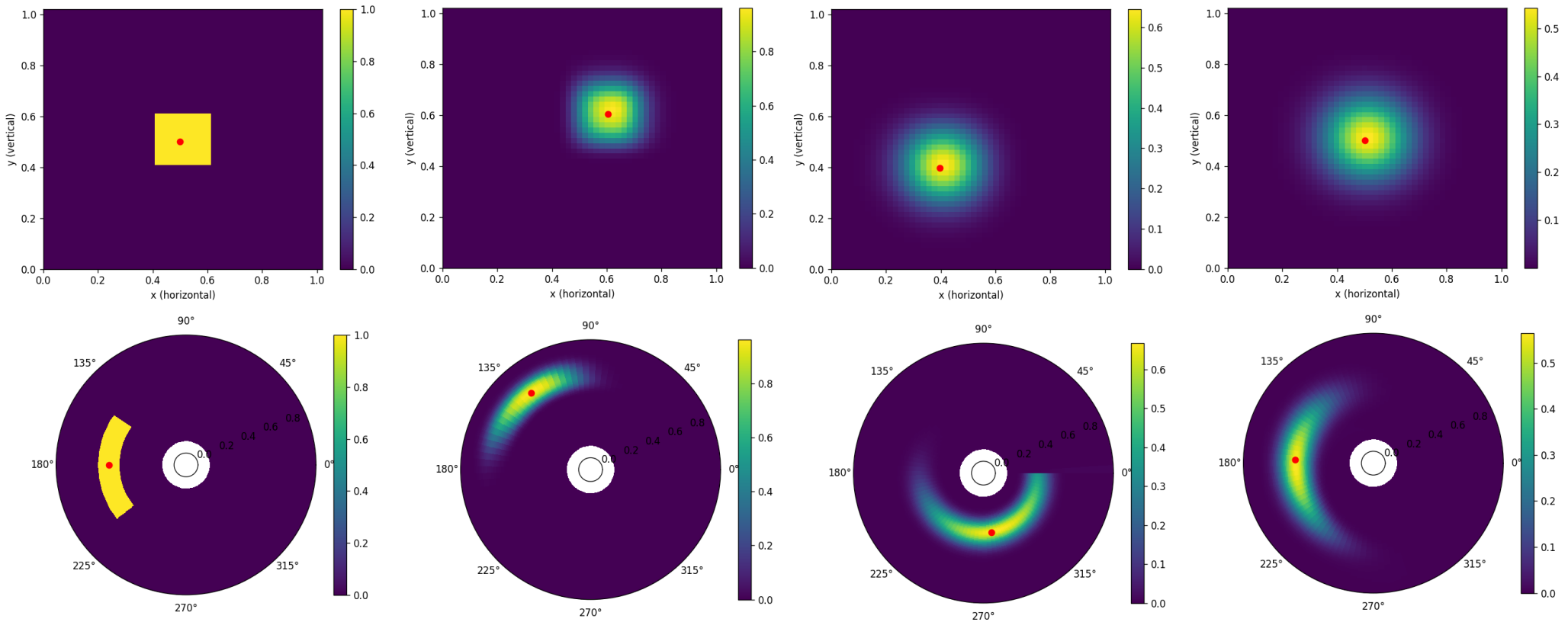


Figure 5: Streamfunction-vorticity Cartesian (top) and polar (bottom) advection tests for advection back and forth along the secondary diagonal. Red dot is the centroid location, temperature indicated by colour. First column is the initial state, second column is one extreme of the diagonal, the third column is the other extreme and the final column is the state once the centroid has returned to its initial position.

Algorithm 1 Thermal Lattice Boltzmann Algorithm

$f_\alpha = \rho_0 w_\alpha$ ▷ Set particle distribution in domain
$g_\alpha = 0$ ▷ Set internal energy zero in domain
$\epsilon = 0$ ▷ Allocate memory for internal energy
ϵ_b ▷ Set heat field in the domain
$\rho = 0$ ▷ Allocate memory for density
$\vec{u} = 0$ ▷ Allocate memory for velocities
while Simulation Running **do** ▷ Begin main simulation loop
  $g_\alpha \mathrel{+}= \rho\epsilon_b w_\alpha$ ▷ At each timestep, add the affect of the heating field
  if $x - \vec{e}_\alpha$ is solid **then** ▷ The following 3 lines enforce bounce-back boundary conditions
    $f_\alpha(x) = f_{\alpha'}(x)$
    $g_\alpha(x) = g_{\alpha'}(x)$
  else
    $f_\alpha(x) = f_\alpha(x - \vec{e}_\alpha)$ ▷ Streaming step
    $g_\alpha(x) = g_\alpha(x - \vec{e}_\alpha)$
  end if
  $\rho = \sum_{i=0}^{i=8} f_i$ ▷ Compute and set macroscopic density
  $\vec{u} = \frac{(\sum_{i=0}^{i=8} f_i \vec{e}_i)}{\rho}$ ▷ Compute and set macroscopic velocity
  $\epsilon = \frac{\sum_{i=0}^{i=8} g_i}{\rho}$ ▷ Compute and set internal energy
  $f_\alpha^{eq} = \rho w_\alpha (1 + 3\frac{\vec{e}_\alpha \cdot \vec{u}}{s^2} + \frac{9}{2}\frac{(\vec{e}_\alpha \cdot \vec{u})^2}{c^4} - \frac{3}{2}\frac{\vec{u}\cdot\vec{u}}{c^2})$ ▷ Get equilibrium distribution f
  $g_\alpha^{eq}(\vec{x}, t) = \epsilon\rho w_\alpha (1 + 3\frac{\vec{e}_\alpha \cdot \vec{u}}{s^2} + \frac{9}{2}\frac{(\vec{e}_\alpha \cdot \vec{u})^2}{c^4} - \frac{3}{2}\frac{\vec{u}\cdot\vec{u}}{c^2})$ ▷ Get equilibrium distribution g
  $\Delta f_\alpha = -w_\alpha \rho\alpha\epsilon \frac{\vec{e}_\alpha}{|\vec{e}_\alpha|} \cdot \vec{g}$ ▷ Get gravitational term
  $\tau_g = \frac{3\kappa}{S^2\Delta t} + \frac{1}{2}$ ▷ Compute relaxation times
  $\tau_f = \frac{3\nu}{S^2\Delta t} + \frac{1}{2}$
  $f_\alpha = f_\alpha + \frac{f_\alpha^{eq} - f_\alpha}{\tau_f} + \Delta f_\alpha$ ▷ Collision step
  $g_\alpha = g_\alpha + \frac{g_\alpha^{eq} - g_\alpha}{\tau_g}$
end while

Advection Tests

A particularly difficult part of the streamfunction-vorticity simulation is correctly modelling advection. In this section, I perform some tests to show that my advection scheme works.

Thermal diffusivity (κ) and thermal expansion (α) were set to zero to avoid convection. The fluid was set to temperature 0 (arb. units) except a small segment which was set to 1 shown in yellow top left of figures 4 and 5. A streamfunction ψ_c was enforced to produce diagonal back and forth motion or a continuous horizontal or azimuthal motion to cross a periodic boundary.
The advection results for both codes are similar. For the periodic advection test, top row figure 4 and 5 the temperature 1 region is advected across the periodic boundary conditions without distortion. The centroid (red dot) lagged the advection field by $\approx 1\%$ and $\approx 10\%$ in the horizontal and azimuthal advection case (figure 4) respectively. It is unknown exactly why these differ substantially, however a probable cause is the smaller timestep use to produce the polar tests. In all cases, the blurring shows numerical diffusion along the direction of travel which is consistent with the modified equivalent partial differential equation for equation 2 using our numerical scheme. The discontinuity in temperature in the bottom row third column of figure 5 near the $0, 360$ azimuthal angle boundary is a direct result of manually setting the streamfunction.

Thermal Lattice Boltzmann Tests

To test the validity of my thermal lattice Boltzmann code, results from [12] are replicated in in figure 6. I used a "bounceback" boundary condition on all walls of the domain (figure 6 top four panes) which was compared against a simulation (figure 6 bottom) with the vertical walls having periodic boundary conditions. In addition, I used random fluctuations in boundary temperature of 10% while [12] used random fluctuations of 25%. The results are similar in convective plume size, quantity and definition. The convection begins slower in the simulation written here, but this can be accounted for by the smaller random noise used.

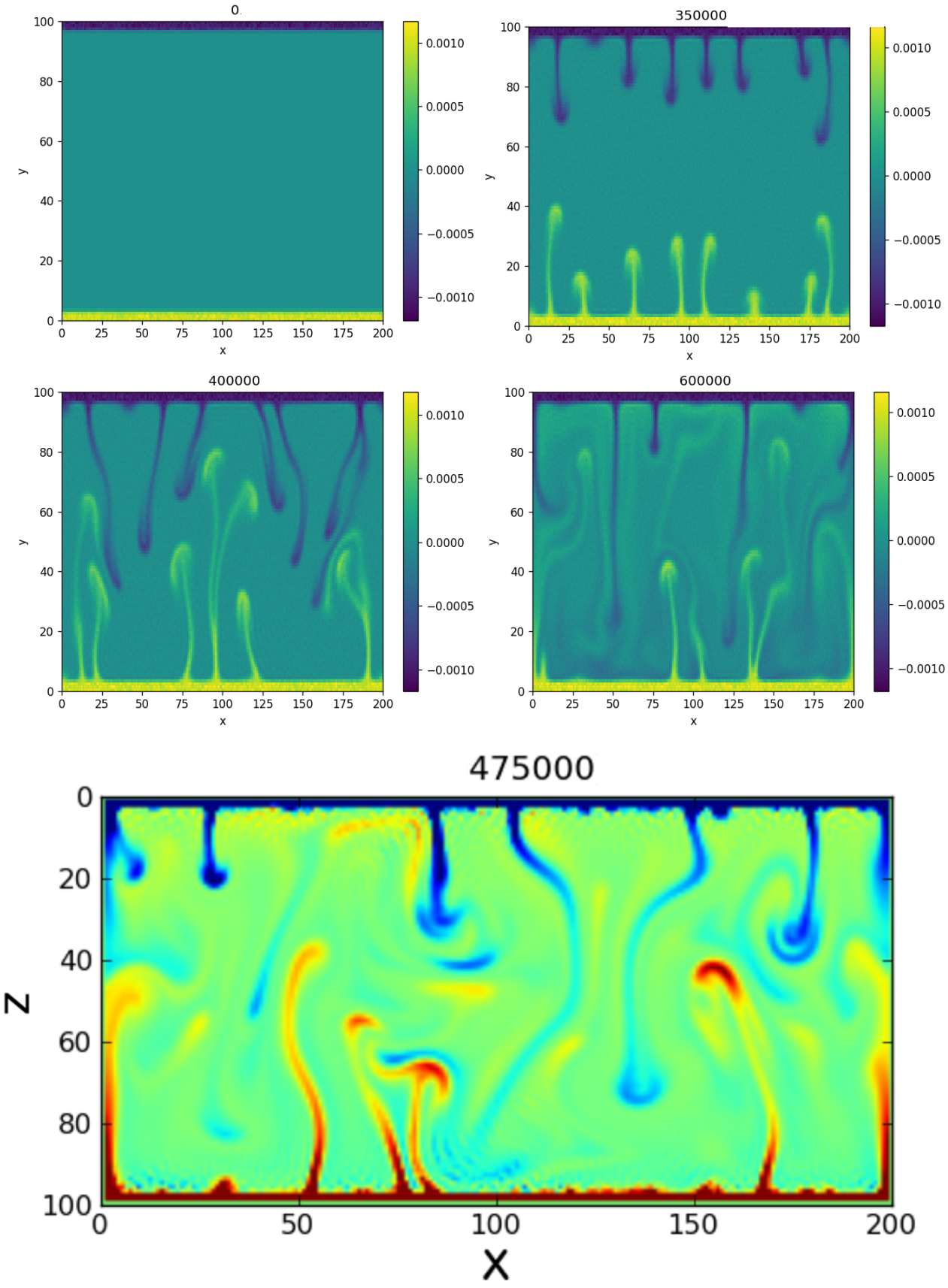


Figure 6: Comparison between Thermal Lattice Boltzmann coded here (top four figures) and Published Thermal Lattice Boltzmann results (bottom figure) [12]. Headings indicate timestep, red/yellow show hot regions while blue areas are cold. Top and bottom regions are held at constant temperature (no internal heating). All program settings used are similar. $Pr = 5000, Ra = 10^8$

Simulations

Geometry and the Central region of a self-gravitating fluid

In this section, I present results from my streamfunction-vorticity codes in Cartesian and polar coordinates. I show through comparison between simulations that taking into account the geometry and center of a 2-dimensional self gravitating fluid is important and significantly changes both the dynamics and long term convective patterns.

A gravitational field linearly proportional to height and radius for the Cartesian and polar codes respectively was set. A constant internal heating field H was set for all but the upper 10% of both domains. A compensating heating field was set at the remaining top of the domain to balance the total heat generation. Note this heat field is different in the Cartesian and polar cases. In all heating fields time-independent random fluctuations of $\approx 1\%$ were set to facilitate instabilities.
As seen in figure 8, the two geometries give vastly different convective behaviour. The Cartesian geometry (left) begins forming convective plumes at time ≈ 143000, long after the time 25000 when the polar (middle) model begins to convect. The convection patterns formed in the Cartesian model are consistent in time, forming the typical Rayleigh-Bernard convection cells [18], while the polar cases are much more erratic. The larger inner radius polar simulation (right) shows similar initial behaviour to the smaller internal radius polar simulation (middle), however, it initially forms higher frequency convective structures. Importantly, over large timescales, the typical wavelength for features in the Cartesian geometry is $3-5$ times longer. Therefore, in simulating a self-gravitating internally heated fluid such as the inner core accurately, the geometry and central region must be simulated. For traditional approaches which discretize a set of equations, this requires either complex spatial meshings or boundary conditions.

Thermal Lattice Boltzmann Simulations

In this section, I then present and discuss the results of my Thermal Lattice Boltzmann simulations. A video is available here or via the url `https://www.youtube.com/watch?v=OM1oqFluO7A` *to give the reader a better idea of the dynamics.*

Figures 9 and 10 show TLBM simulations for internally heated Rayleigh numbers between 10^8 and 10^4 for Prandtl numbers of $Pr = 5000$. A high Prandtl number was selected to show the applicability to geologic flows which operate in the infinte Prandtl number limit. The heating field H for these is constant below a radius of 90% and the remaining 10% cooled so that the internal energy is constant. To stimulate convection, random fluctuations of 10% in the heating field were set as well as random fluctuations to the heating/cooling boundary. The initial state common to simulations in figures 9 and 10 is shown in figure 7.
Figures 9 and 10 show that internally heated convection in our model begins at a thermal Rayleigh number below $Ra_H = 10^5$. This is consistent with analytical work that found a similar 3-dimensional system would have a critical Rayleigh number of ≈ 8000 [2]. it is unknown if the lack of convection in small thermal Rayleigh numbers like $Ra = 10^4$ in figure 9 is due to physical constraints, or the fact that we only simulated a finite amount of time. In future, the TLBM program could be parallelised (which is easy with goLang) and run for longer to simulate more steps and better bound the critical Ra_H from above. The thickness of the cooling boundary layer used here is also not analysed, in future this could be done away with by using more complex thermal boundary conditions.
he characteristic wavelength of the convective patterns in figures 9 and 10 decreases with increasing thermal Rayleigh number (Ra_H). For small convecting thermal Rayleigh numbers, $Ra_H = 10^6, Ra_H = 10^5$ the convection wavelength is fairly consistent through time, while for higher values $Ra_H = 10^8, Ra_H = 10^7$ the wavelength is more dynamic. In general, the characteristic convection cells seen in Cartesian convection such as that in figure 8 (left bottom) is not seen. Instead, we see periodic plumes with skinny stalks and wide heads. These patterns are not static, best seen by the dumbbell-shaped oscillations in figure 10 (left) indicating that these systems have not reached equilibrium.

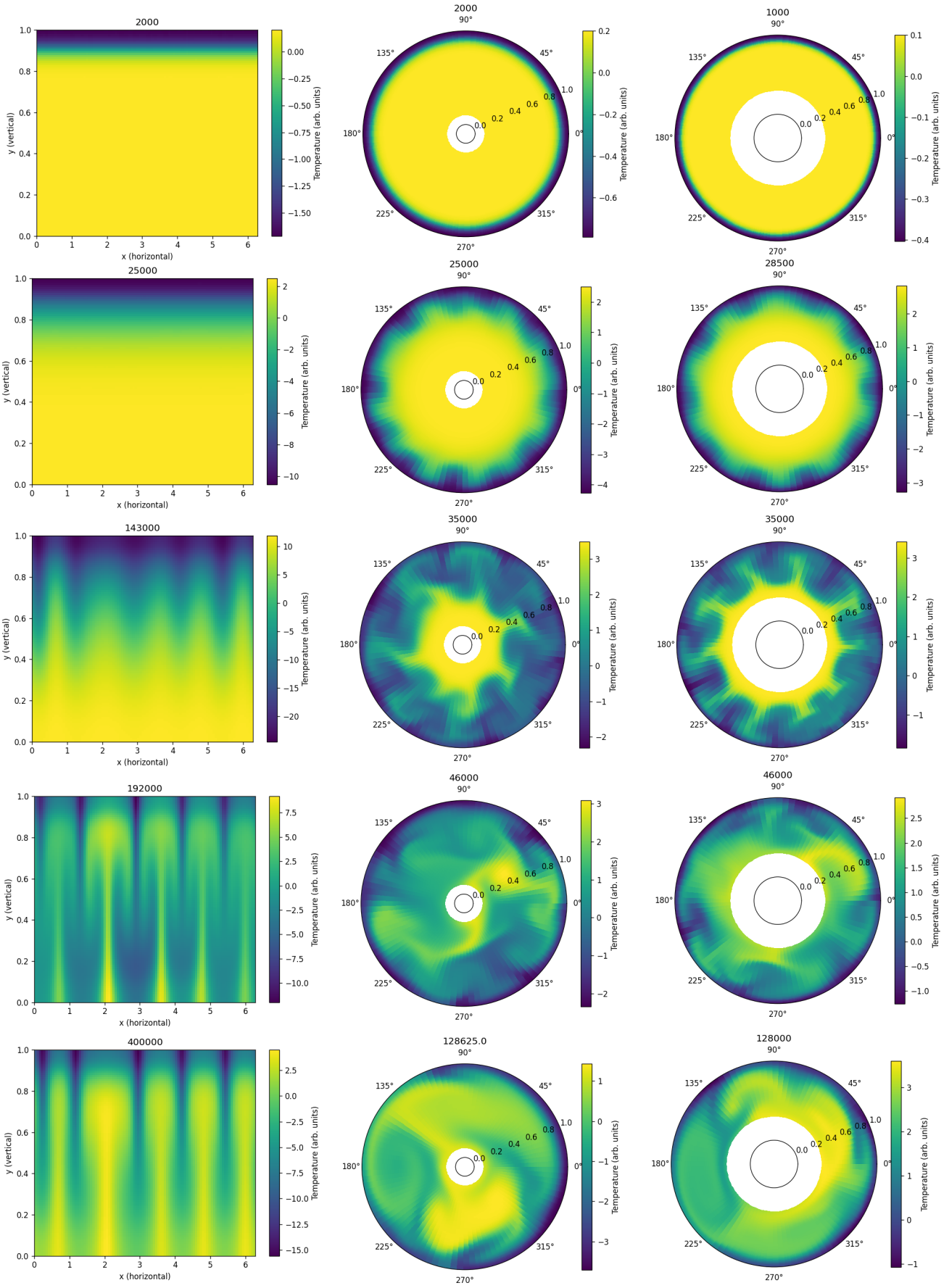


Figure 8: Simulations of internally heated, self-gravitating convection in Cartesian (left), polar with inner radius 0.1 (middle) and polar with inner radius 0.2 (right) coordinates. Times indicated by headings. Apart from inner radius, simulation settings are common. $Ra_H = 10^8, Pr = 5000$.

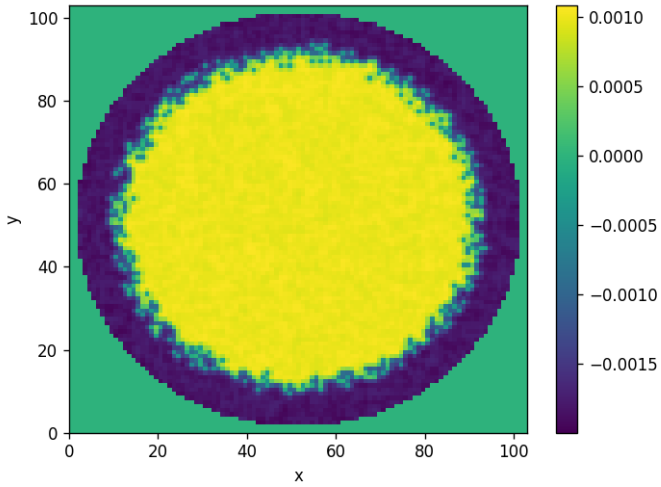


Figure 7: Example Initial state showing random fluctuations of the heating field of 10% throughout with random fluctuations across the hot/cold boundary.

Difficulties with the Thermal Lattice Boltzmann method

The TLBM coded here uses a fixed unit timestep and lattice spacing making the lattice speed $C = 1$. It was found that once velocities became comparable to the lattice speed the TLBM became unstable and distribution functions would become large and crash the program. For internally heated convective problems, the time to heat/cool the fluid before convective plumes can develop can be significant compared to the timescale for plume motion, particularly at smaller or near critical Rayleigh numbers. Because of this, much simulation time is spent computing an uninteresting part of the solution. This could be fixed by setting the initial condition according to simple analytical thermal diffusion models, and simulating from this point onwards. Extreme local heating or cooling was also an issue that would occur at high Rayleigh numbers $Ra > 10^9$ and induce velocities comparable to the lattice speed, causing instabilities to grow to infinity and crash the program.

Conclusion

A 2-dimensional self-gravitating internally heated model of the inner core was presented. Codes were developed from scratch in python3 and goLang to simulate convection in the model using streamfunction-vorticity and Lattice Boltzmann approaches. The streamfunction codes based in Cartesian and polar geometries were compared showing that the geometry and central region of the inner core were important to accurately simulating internally heated convection, however neither code could account for both. Finally, results from the TLBM code which accounted for both geometry and the central region were presented for high Prandtl number ($Pr = 5000$) giving an upper bound on the critical internally heated Rayleigh number of $Ra_H < 10^5$ which is consistant with other work [2]. In future, the same Lattice Boltzmann code could be used to model more realistic models of the inner core, which involve other heat sources such as heating from friction or heat left over from Earth's formation. It can also be applied to other problems with otherwise difficult non-slip boundary conditions.

Acknowledgements

I would like to thank Professor Louis Moresi for suggesting and supervising this project.

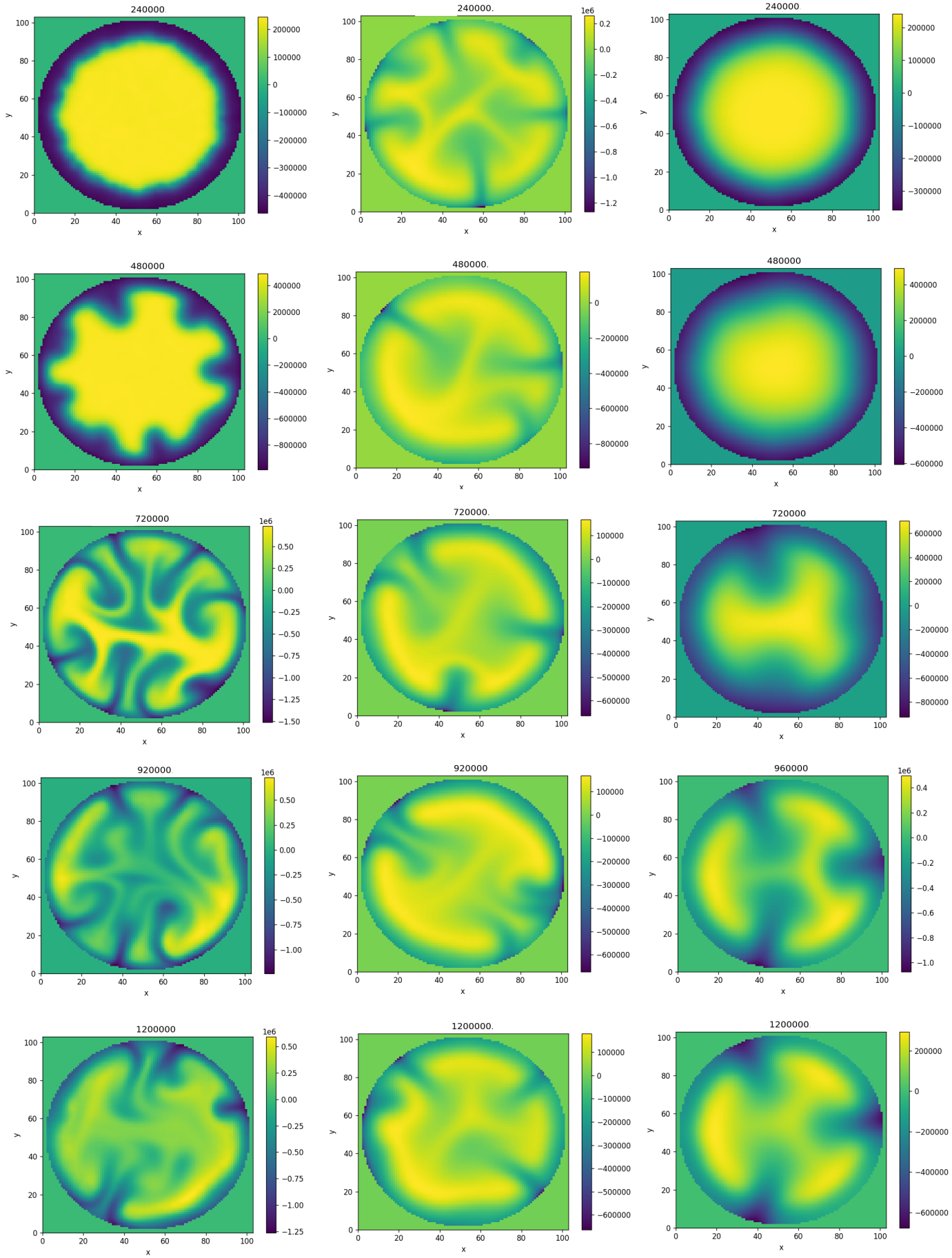


Figure 9: High Rayleigh number $Ra_H = 10^8$ (left column) $Ra_H = 10^7$ (middle column) $Ra_H = 10^6$ (right column) from Thermal Lattice Boltzmann simulations for an internally heated self-gravitating fluid with $Pr = 5000$. Headings are simulation timestep

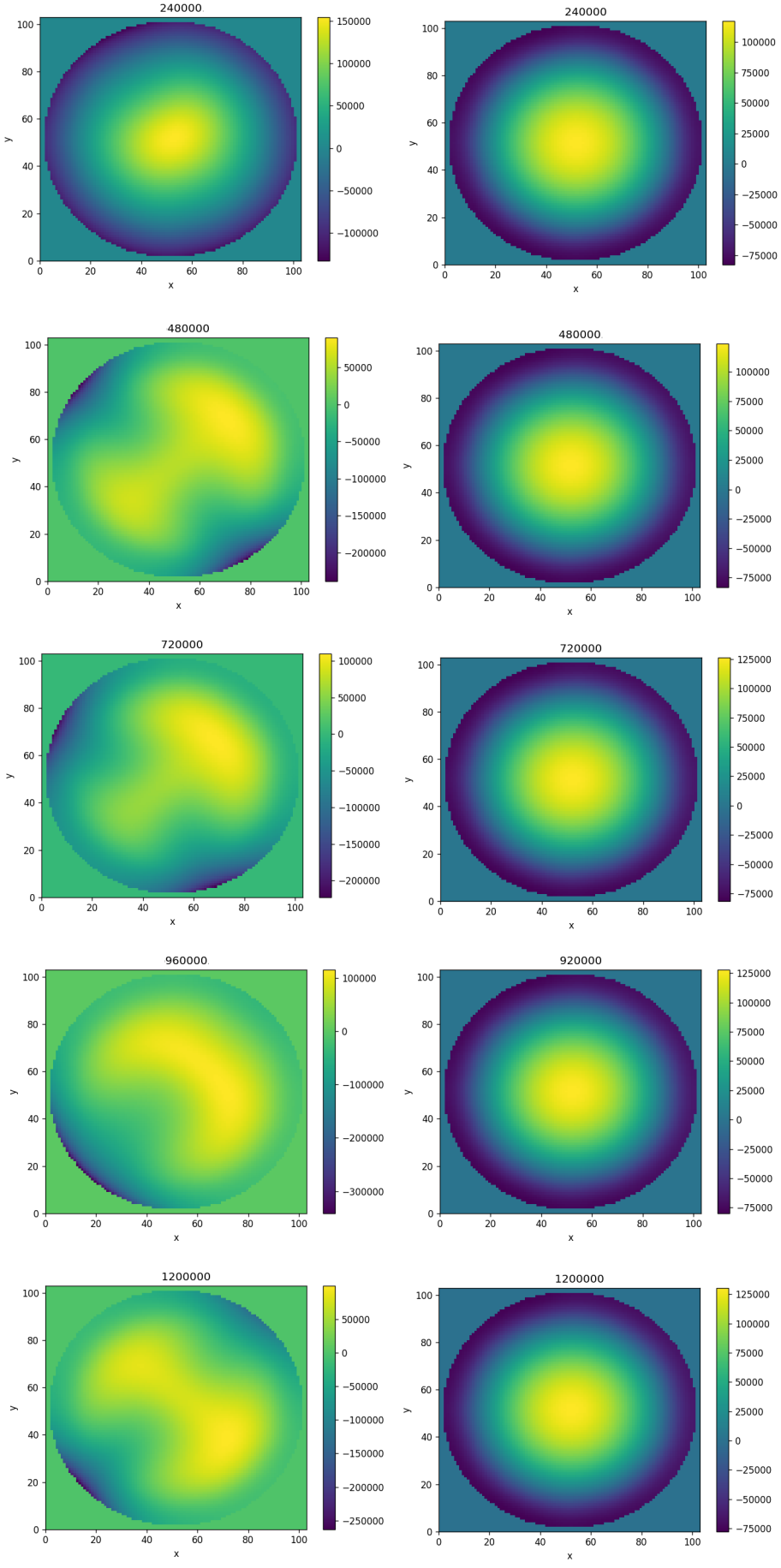


Figure 10: High Rayleigh number $Ra_H = 10^5$ (left column) $Ra_H = 10^4$ (right column) Thermal Lattice Boltzmann simulations for an internally heated self-gravitating fluid with $Pr = 5000$. Headings are simulation timestep

Appendix

The Jacobi Method

Here the Jacobi Method for solving the Poisson equation, like (9) is introduced. This is the method used for solving 9 used in my streamfunction-vorticity codes.

We cannot solve for ψ explicitly in equation 9. Instead we use an iterative Jacobi method. In Cartesian coordinates, equation 9 is:

$$\omega_{i,j} = \frac{\psi_{i+1,j} - 2\psi_{i,j} + \psi_{i-1,j}}{\Delta x^2} + \frac{\psi_{i,j+1} - 2\psi_{i,j} + \psi_{i,j-1}}{\Delta y^2} \tag{34}$$

Rearranging for ψ

$$\psi_{i,j} = \frac{\Delta x^2 \Delta y^2}{2(\Delta x^2 + \Delta y^2)}\left(\frac{\psi_{i+1,j} + \psi_{i-1,j}}{\Delta x^2} + \frac{\psi_{i,j+1} + \psi_{i,j-1}}{\Delta y^2} + \omega_{i,j}\right). \tag{35}$$

We then use the result of $\psi_{i,j}$ back into equation 35 to solve for $\psi_{i,j}$ iterately as shown in equation 36 [4, 1]:

$$\psi_{i,j}^{(k+1)} = \frac{\Delta x^2 \Delta y^2}{2(\Delta x^2 + \Delta y^2)}\left(\frac{\psi_{i+1,j}^{(k)} + \psi_{i-1,j}^{(k)}}{\Delta x^2} + \frac{\psi_{i,j+1}^{(k)} + \psi_{i,j-1}^{(k)}}{\Delta y^2} + \omega_{i,j}\right). \tag{36}$$

Where the superscript (k) means the result of the k^{th} iteration of the above equation and this operation us applied to all points in $\mathcal{D}'$. We terminate this iterative method once the error $\sum_{(i,j)\in\mathcal{D}'} \mid \psi_{i,j}^{(k+1)} - \psi_{i,j}^{(k)} \mid$ gets sufficiently small. A similar method is applied in polar coordinates.

References

[1] D. Adair and M. Jaeger. Developing an understanding of the steps involved in solving navier–stokes equations. 2015.

[2] V. Babskii and I. Sklovskaia. On convection onset in a self-gravitating fluid sphere with internal heating: Pmm vol. 35, n= 6, pp. 1000–1014, 1971. *Journal of Applied Mathematics and Mechanics*, 35(6):951–965, 1971.

[3] B. A. Bolt. Pdp and pkikp waves and diffracted pcp waves. *Geophysical Journal International*, 20(4):367–382, 1970.

[4] J. Burkardt. Jacobi iterative solution of poisson's equation in 1d, 2011.

[5] L. C. Earn, T. W. Yen, and T. L. Ken. The investigation on simple and simpler algorithm through lid driven cavity. *Journal of Advanced Research in Fluid Mechanics and Thermal Sciences*, 29(1):10–22, 2017.

[6] C. M. R. Fowler, C. M. R. Fowler, and M. Fowler. *The solid earth: an introduction to global geophysics*. Cambridge University Press, 1990.

[7] S. K. Godunov. A difference scheme for numerical solution of discontinuous solution of hydrodynamic equations. *Math. Sbornik*, 47:271–306, 1959.

[8] D. Goluskin. *Internally heated convection and Rayleigh-Bénard convection*. Springer, 2016.

[9] V. Guinot. *Godunov-type schemes: an introduction for engineers*. Elsevier, 2003.

[10] R. Khazaeli, M. Ashrafizaadeh, and S. Mortazavi. A ghost fluid approach for thermal lattice boltzmann method in dealing with heat flux boundary condition in thermal problems with complex geometries. 2015.

[11] P. Mora, G. Morra, and D. A. Yuen. A concise python implementation of the lattice boltzmann method on hpc for geo-fluid flow. *Geophysical Journal International*, 220(1):682–702, 2020.

[12] P. Mora and D. A. Yuen. Simulation of plume dynamics by the lattice boltzmann method. *Geophysical Journal International*, 210(3):1932–1937, 2017.

[13] W. H. Press, W. T. Vetterling, S. A. Teukolsky, and B. P. Flannery. *Numerical recipes*, volume 818. Cambridge university press Cambridge, 1986.

[14] Y.-H. Qian, D. d'Humières, and P. Lallemand. Lattice bgk models for navier-stokes equation. *EPL (Europhysics Letters)*, 17(6):479, 1992.

[15] D. H. Rothman and S. Zaleski. *Lattice-gas cellular automata*. 2004.

[16] M. Spiegelman and R. F. Katz. A semi-lagrangian crank-nicolson algorithm for the numerical solution of advection-diffusion problems. *Geochemistry, Geophysics, Geosystems*, 7(4), 2006.

[17] H. Tkalcic and T.-S. Pham. Shear properties of earth's inner core constrained by a detection of j waves in global correlation wavefield. *Science*, 362(6412):329–332, 2018.

[18] D. J. Tritton. *Physical fluid dynamics*. Springer Science & Business Media, 2012.

[19] A. J. Wagner. A practical introduction to the lattice boltzmann method. *Adt. notes for Statistical Mechanics*, 463:663, 2008.