

A DISSERTATION ON
Development of a Computational Model for
Simulating 2D Rayleigh–Bénard Convection

*in Partial Fulfilment of the Requirements
for the Degree of*

Masters in Physics
by
Adarsh Ramtel
(23615001)

under supervision of
Prof. Ambrish Pandey



**DEPARTMENT OF PHYSICS
INDIAN INSTITUTE OF TECHNOLOGY
ROORKEE**

2023–25

Acknowledgment

I would like to express my heartfelt gratitude to my professor, **Prof. Ambrish Pandey**, for his invaluable guidance, insightful lectures, and patient support throughout this project. His passion for physics and deep understanding of fluid dynamics were a constant source of inspiration and motivation.

I am also thankful to my teaching assistant, **Shreshthi**, whose detailed feedback, timely clarifications, and encouragement played a crucial role in shaping the direction and clarity of my work.

This project has been a meaningful learning experience, and I feel truly fortunate to have learned under such dedicated mentors.

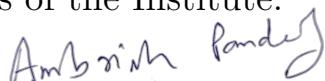
— *Adarsh*

CERTIFICATE

This is to certify that the work contained in the thesis entitled '**Development of a Computational Model for Simulating 2D Rayleigh–Bénard Convection**', submitted by **ADARSH RAMTEL(Enrolment No. 23615001**, in partial fulfillment of the requirements for the award of the Degree of Master of Science in Physics to the **INDIAN INSTITUTE OF TECHNOLOGY, ROORKEE**, is a record of bonafide research work carried out by him under my direct supervision and guidance.

I consider that the thesis has reached the standards and fulfilled the requirements of the rules and regulations relating to the nature of the degree. The contents embodied in the thesis have not been submitted for the award of any other degree or diploma in this or any other university.

I also certify that he has complied with the Plagiarism Guidelines of the Institute.



Dr. Ambrish Pandey
Professor, Supervisor
Department of Physics
Indian Institute of Technology, Roorkee

DECLARATION BY THE CANDIDATE

I, the undersigned, hereby solemnly declare that the project report entitled **Development of a Computational Model for Simulating 2D Rayleigh–Bénard Convection** represents my work conducted during my course of study under the supervision of **Dr. Ambrish Pandey**. I affirm that the statements and conclusions presented in this report are derived from my research efforts. Additionally, I certify the following:

- The work in the thesis is original and has been done by myself under the supervision of my supervisors.
- The work has not been submitted to any other Institute for any degree or diploma.
- I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
- Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.
- Whenever I have quoted written materials from other sources credit is given to the sources by citing them.



Adarsh Ramtel
M.Sc. Physics
Department of Physics
Indian Institute of Technology, Roorkee

*To my parents, my brother, and my grandparents,
whose unwavering respect for my goals
and tireless support have made them feel like shared dreams.
Your belief in me has been the greatest blessing.*

Contents

Contents	7
Abstract	8
1 Introduction	9
1.1 Background and Motivation	9
1.2 Rayleigh Bénard Convection	10
1.2.1 Convection in the Physical world	10
1.2.2 The controlled system of RB convection	10
1.3 Driving Equations of Fluid Mechanics	12
1.3.1 Continuity and Energy Conservation Equations	12
1.3.2 Momentum Conservation and Navier–Stokes Equation	13
1.3.3 The Boussinesq Approximation	14
1.3.4 Non-Dimensionalization	16
1.4 Computational Methods	17
1.4.1 Solving Differential Equations on a Computer - Finite Differences and Spectral Method	17
1.4.2 Advancing Solutions with Time	19
1.5 Structure of this Thesis and Layout	21
2 Modeling and Discretization of the Convection Problem	22
2.1 Vorticity-Streamfunction Formulation	22
2.2 Description of Simulation Domain	23
2.2.1 Spectral Decomposition along the Horizontal	23
2.2.2 Finite Differencing along the Vertical	25
2.2.3 Physical Significance behind Domain Structure	25
2.2.4 Boundary Conditions	27
2.3 Algorithm for the Solver	28
2.4 Flow Visualization Technique	29
2.5 Code availability	31
3 Implementation of the Numerical Model	32
3.1 Linearized Model	32

3.2	Adding Non-Linear Terms	35
3.2.1	Initial Temperature for simulation	36
4	Results and Analysis	38
4.1	Linear Stability Validation	38
4.2	Non-linear regime : Emergent Flow structures and Patterns .	43
4.3	Turbulent regimes : High Ra values	46
4.4	Sensitivity to Resolution and Time Step	48
5	Conclusion	50
5.1	Limitations and Future Work	50
5.2	Closing Remarks	51
A	Non-Dimensionalization	53
B	Thomas Algorithm (Tridiagonal matrix algorithm)	56
C	Additional Figures	58
	Bibliography	60

Abstract

Convection is a fundamental process observed across an astonishing range of physical systems—from boiling fluids to stellar interiors. Among the many models developed to understand it, **Rayleigh–Bénard convection** stands out as a classic and accessible framework for studying the onset and development of buoyancy-driven flow. This thesis presents the development of a computational solver for simulating two-dimensional Rayleigh–Bénard convection using a **hybrid numerical scheme: spectral decomposition** in the horizontal direction and **finite differences** in the vertical. Starting from the governing Boussinesq equations, we derive the **vorticity– streamfunction formulation**, implement linear and nonlinear dynamics, and evolve the system using an Adams–Bashforth time-stepping scheme. The solver is validated through linear stability analysis and captures the transition from conduction to convection at critical Rayleigh numbers. In the nonlinear regime, the model successfully reproduces the emergence of steady convection rolls and pattern formation. At higher Rayleigh numbers, the system exhibits features of turbulent convection, and we briefly explore the influence of the Prandtl number in shaping the flow structure. The work emphasizes clarity, modular design, and interpretability, providing a foundation for future exploration of more complex or three-dimensional convective systems.

Chapter 1

Introduction

1.1 Background and Motivation

Convection is a very important phenomenon in the universe, from influencing the atmosphere of planets to playing a significant role in the transport of energy in massive stars, the topic goes in depth much beyond the usual notion of hot air rising and cold air sinking[1, 2]. However, the last statement is indeed the defining characteristic of convection, and we build towards the much more involved process from this starting point.

The system of Rayleigh Bénard convection is a controlled setup that can be monitored efficiently and is incredibly useful for understanding onset of convection[3] in regard to the information it can provide for estimating larger models and more complex systems. This controlled system is very beneficial in studies of transitions from a purely conductive state to a convection dominated one, and eventually into a turbulent flow. It is characterized by a non-dimensional number known as the Rayleigh number, which governs the nature and onset of convection[3].

Such studies aim to answer fundamental questions : What difference in temperature between fluid layers is needed for the convection to begin?, What is the time required for convection to develop? When do turbulent properties emerge ? In this work, we attempt to address some of these questions by simulating the convection process via numerical methods.

In the modern era, with access to computational power, large-scale complex simulations can be run to model real-world convection more realistically. This enables us to simulate the phenomenon of convection on a computer and infer useful information, validate theoretical predictions, and draw physically

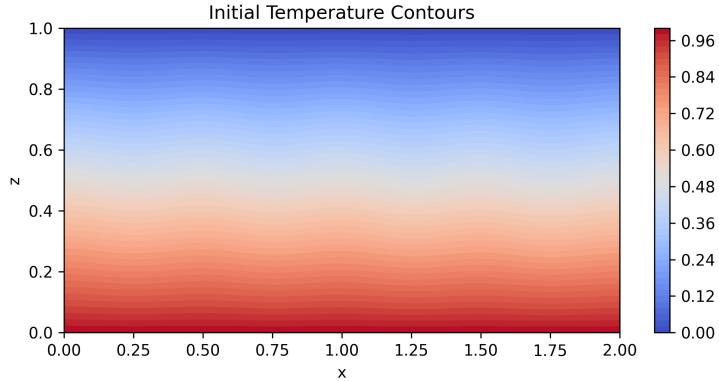


Figure 1.1: Standard Rayleigh-Bénard Setup with Hot bottom layer and cold upper layer

meaningful interpretations[4, 5].

1.2 Rayleigh Bénard Convection

1.2.1 Convection in the Physical world

Convection is one the earliest concepts a student studying science comes across[6] . It is a fundamental process of heat and mass transfer occurring in fluids under differences in temperature and density. When a fluid is heated, the warmer, less dense parts rise up the domain, while cooler denser regions sink, and set up circulating currents[7].

This mechanism governs a wide range of physical systems, of varying scales in the universe - from patterns in a cup of soup to weather patterns and ocean currents, and upto mantle dynamics beneath the crust and transport of energy in stars [2].

1.2.2 The controlled system of RB convection

Rayleigh Bénard Convection is a system that can be easily suited to one's needs in the simulation. It's origins lie in the 19th century when Henri Bénard conducted experiments[8] where a thin layer of oil was heated from below, observing hexagonal convection cells(named Bénard cells after him) forming spontaneously.

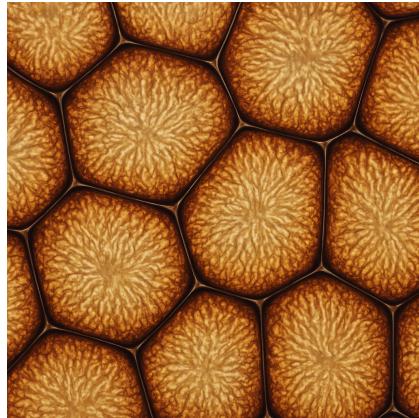


Figure 1.2: Artistic Representation of benard cells forming in a fluid system

Then Lord Rayleigh provided initial theoretical explanations for thermal convection instabilities in a horizontal fluid layer heated from below and cooled from above. Utilizing linear stability analysis, he showed that convection occurs when the buoyant forces overcome viscous damping, and predicted the onset of convection from a non-dimensional critical number, known as Rayleigh Number[3] :

$$Ra = \frac{g\alpha\Delta T d^3}{\nu\kappa}, \quad (1.1)$$

where,

- g : gravity
- α : thermal expansion coefficient
- ΔT : temperature difference
- d : layer thickness
- ν : kinematic viscosity
- κ : thermal diffusivity

Thermal expansion coefficient determines the extent of the expansion of the fluid upon heating, due to a decrease in density, kinematic viscosity determines the fluid's resistance to shear under gravity or internal flow. Thermal diffusivity describes how quickly heat is conducted through the fluid

relative to it's ability to store thermal energy.

The reason Rayleigh Bénard convection is widely studied is because it is simple but with deep physical relevance. It represents a well defined configuration : a horizontal fluid layer heated from below and cooled from above. This captures the essential physical mechanisms of buoyancy forces, diffusion, and viscous damping, while being easy to monitor and verifiable theoretically via linear stability analysis. This establishes Rayleigh number as a critical parameter that governs the onset and nature of convection[3].

1.3 Driving Equations of Fluid Mechanics

There are three equations at the core of any fluid mechanical system: The continuity equation, the energy conservation equation and of course the Navier Stokes equation or momentum conservation equation[9]. Our formulation of convective system will primarily comprise these equations.

1.3.1 Continuity and Energy Conservation Equations

Mass conservation is a fundamental requirement and a universal law. In fluids, it is given by the continuity equation.

$$\frac{\partial \rho}{\partial t} + \vec{u} \cdot \nabla \rho + \rho \nabla \cdot \vec{u} = 0, \quad (1.2)$$

where \vec{u} is the velocity field of the fluid, (\cdot) represents the dot product, ρ is the density, and ∇ represents the gradient. This means, from vector analysis, that the fluid's volume remains constant as it moves, and no net mass is accumulated nor depleted at any point in the domain [9].

Another important conservation equation, Energy conservation is described in fluids by the thermal energy equation, which describes how heat is transported due to both conduction and advection. The first law of thermodynamics states that energy is conserved in a fluid system. Specifically, the change in internal energy of a fluid element is equal to the heat it absorbs minus the work it does on its surroundings.

This can be written in terms of physical quantities as:

$$\rho \frac{de}{dt} + p \nabla \cdot \vec{v} = \rho T \frac{dS}{dt} = \nabla \cdot (k \nabla T) + Q, \quad (1.3)$$

Here, e is the internal energy per unit mass, \vec{v} is the velocity field, T is temperature, S is entropy per unit mass, and ρ is the fluid density. The term $p\nabla \cdot \vec{v}$ represents the mechanical work done due to expansion or compression of the fluid. The term $\nabla \cdot (k\nabla T)$ accounts for heat conduction through the fluid, where k is thermal conductivity. Q includes any other heat sources such as friction, electrical currents, or even nuclear reactions.

In essence, this equation tracks how heat, work, and internal energy interact as a fluid flows and evolves in time.[7, 9].

In the context of Rayleigh–Bénard convection, the balance between advection and diffusion in the temperature equation plays a central role in pattern formation, heat transfer efficiency (as measured by the Nusselt number), and the transition to turbulence. These equations, in conjunction with the Navier–Stokes equation for momentum conservation, form the basis of most theoretical and numerical models of convection [3, 9].

1.3.2 Momentum Conservation and Navier–Stokes Equation

One of the most fundamental equations in fluid dynamics is the Navier Stokes equation which elegantly describes the motion of fluids. The Navier Stokes equation encapsulates the conservation of momentum in a fluid and represents Newton’s second law applied to a continuous medium. It balances the rate of change of momentum with the forces acting on the fluid. These include pressure gradients, viscous stresses, and external body forces like gravity. The equation is given by:

$$\rho \frac{d\vec{v}}{dt} = -\nabla p + \nabla \cdot \boldsymbol{\sigma} + \rho \vec{g} \quad (1.4)$$

where ρ is fluid density, \vec{v} is the velocity vector, p is pressure, $\boldsymbol{\sigma}$ is the viscous stress tensor, and \vec{g} is the gravitational acceleration vector[10].

For a Newtonian fluid, the viscous stress tensor is directly proportional to the rate at which the fluid is deforming. It is given by:

$$\sigma_{ij} = 2\rho\nu \left(e_{ij} - \frac{1}{3}e_{kk}\delta_{ij} \right) \quad (1.5)$$

where ν is the kinematic viscosity and e_{ij} is the rate-of-strain tensor, defined as:

$$e_{ij} \equiv \frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \quad (1.6)$$

The term e_{kk} is the trace of the strain tensor (i.e., the sum of diagonal components), and δ_{ij} is the Kronecker delta. These expressions characterize how internal friction in the fluid resists deformation, an essential part of describing real fluid motion.[9]

1.3.3 The Boussinesq Approximation

The Boussinesq approximation is a popular simplification in fluid dynamics, especially for buoyancy-driven flows like Rayleigh Bénard convection. Proposed originally by Joseph Boussinesq in 1903[11], this approximation assumes that density variations in a fluid can be safely neglected except when they appear in the buoyancy (gravity) term of the momentum equation. That is, the fluid is treated as incompressible everywhere else, which simplifies the governing equations considerably [12].

Mathematically, this means we replace the full continuity equation, Eq. (1.2), with the incompressible equation,

$$\nabla \cdot \vec{u} = 0. \quad (1.7)$$

This does not imply that the density is strictly constant along fluid trajectories; rather, it implies that the relative change in density due to flow-induced compression or expansion is small compared to the effects of velocity gradients.

[9]The local influence of pressure on density is considered negligible, so density variations ρ' arise primarily from temperature differences T . This relationship is expressed by the linearized equation of state:

$$\rho' = -\rho_0\alpha(T - T_0), \quad (1.8)$$

where α is the thermal expansion coefficient and ρ_0 is the reference density.

These density variations generate buoyancy forces that drive fluid motion. Within the Boussinesq approximation, the momentum equation simplifies to:

$$\frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla) \vec{v} = -\frac{1}{\rho_0} \nabla p' + \alpha g_0 T \hat{z} + \nu \nabla^2 \vec{v}, \quad (1.9)$$

where p' is the pressure perturbation, ν is the kinematic viscosity, and \hat{z} is the vertical unit vector. which removes the static pressure gradient and density terms. In the remaining terms, density is approximated by the constant ρ_0 .

The viscous term represents molecular momentum diffusion and acts to smooth velocity gradients, effectively transporting momentum from faster to slower regions in the fluid.

The internal energy conservation, when simplified under the Boussinesq approximation, leads to an equation of the form[9]:

$$\frac{\partial T}{\partial t} + (\vec{v} \cdot \nabla)T = \kappa \nabla^2 T, \quad (1.10)$$

where κ is the thermal diffusivity.

This equation reflects the balance between advective transport of heat by the fluid motion and thermal diffusion. Only the diffusive contribution to heating is retained; other effects, such as viscous dissipation, are typically ignored. This is because in most practical Boussinesq flows, viscous heating is assumed to be negligible compared to conductive and convective heat transport.

These approximations are valid when temperature differences in the fluid are small enough that density variations are modest (e.g., a few percent), and when the vertical scale of the system is much smaller than the atmospheric scale height (for gases) or acoustic scale (for liquids). The Mach number, defined as the ratio of the flow speed to the speed of sound, should be less than about 0.3 to neglect compressibility effects, which is usually obeyed in thermally driven slow flows. [10, 12]

In summary, the Boussinesq approximation is an effective way to retain the essential physics of buoyancy while significantly simplifying the governing equations. It enables manageable modeling of convection and is foundational to the analysis of Rayleigh–Bénard systems.

1.3.4 Non-Dimensionalization

Nondimensionalization plays a crucial role in simplifying the governing equations of fluid flow by reducing the number of independent parameters. This process reveals key dimensionless numbers—such as the Rayleigh and Prandtl numbers—that encapsulate the relative influence of physical effects like buoyancy, viscosity, and thermal diffusion. It also enables comparison across systems with different scales and facilitates numerical stability in computational modeling.

We nondimensionalize the system using standard scales [9]. Length is scaled by the box depth D , time by the thermal diffusion time D^2/κ , and temperature by the imposed temperature difference ΔT . Pressure is scaled by $\rho_0 \kappa^2 / D^2$ for convenience.

Applying these scales to the governing equations (see Appendix A for full derivation), and multiplying Eq. (1.7) by D^2/κ , Eq. (1.9) by D^3/κ^2 , and Eq. (1.10) by $D^2/(\kappa\Delta T)$, we arrive at the nondimensional forms:

$$\nabla \cdot \vec{v} = 0, \quad (1.11)$$

$$\frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla) \vec{v} = -\nabla p + \text{Ra} \text{Pr} T \hat{z} + \text{Pr} \nabla^2 \vec{v}, \quad (1.12)$$

$$\frac{\partial T}{\partial t} + (\vec{v} \cdot \nabla) T = \nabla^2 T. \quad (1.13)$$

Here, all variables are nondimensional. The vertical domain spans $0 \leq z \leq 1$, and the horizontal extent is $0 \leq x \leq a$, where $a = L/D$ is the aspect ratio of the box. Two key nondimensional numbers arise from this scaling:

$$\text{Ra} = \frac{g_0 \alpha \Delta T D^3}{\nu \kappa}, \quad (1.14)$$

$$\text{Pr} = \frac{\nu}{\kappa}. \quad (1.15)$$

The Rayleigh number Ra quantifies the strength of buoyancy-driven convection: the numerator drives convection, while the denominator resists it through diffusion. The Prandtl number Pr is the ratio of momentum to thermal diffusivity [3]. These two equations are of paramount importance and we will discuss them in more detail in subsequent sections.

1.4 Computational Methods

In this section, we introduce some of the key techniques used as a base for developing a fully functional solver for Rayleigh-Bénard convection. The focus is on introduction of concepts of discretization and time integration required for solving the governing equations. Implementation details of the code are discussed further in Chapter 2.

1.4.1 Solving Differential Equations on a Computer - Finite Differences and Spectral Method

As it is easy to notice, the equations so far discussed Eq. (1.11), (1.12), (1.13) are a set of partial differential equations. While we can write them down on paper, solving them by hand is usually impossible—especially for realistic setups with complex boundaries or nonlinear behavior. Analytical solutions are solved in a continuous domain, however, in numerical simulations, we can't work with a truly continuous domain—so the first step is to break it up into smaller pieces.

This is where numerical methods come in. The core idea is to approximate continuous equations on a grid: a discrete collection of points in space (and time). On this grid, derivatives are replaced by differences (like finite differences), or sometimes by sums of basis functions (like spectral methods), making the equations suitable for computation [13]. For our purposes we make use of both of these techniques.

The first step in applying a finite-difference method to solve a partial differential equation (PDE) is to replace the continuous domain with a grid of discrete points. For example, consider a function $u(x, y)$ defined over a rectangular region $0 \leq x \leq 1$, $0 \leq y \leq L$. Instead of working with all possible values of x and y , we choose evenly spaced points using step sizes Δx and Δy , and evaluate the function only at those locations.

We can then represent the function at a grid point using indices i and j , so that

$$u(x, y) \rightarrow u_{i,j} = u(i\Delta x, j\Delta y).$$

This notation allows us to express finite-difference approximations in terms of a central grid point (i, j) and its neighboring points, such as $(i + 1, j)$, $(i - 1, j)$, $(i, j + 1)$, and $(i, j - 1)$. These neighbors are used to approximate derivatives and build algebraic equations that can be solved on a computer.

The first derivatives are then approximated as :

$$\left. \frac{\partial u}{\partial x} \right|_i \approx \frac{u_{i+1} - u_{i-1}}{2\Delta x} \quad (1.16)$$

in what is known as the central difference approximation, which gives a central difference approximation of the first derivative of u with respect to x at point i , using the values at neighbouring points $i + 1$ and $i - 1$ [13]. A more robust representation also adds a truncation error term. The difference between the partial derivative and its finite-difference representation is the truncation error (T.E.).

Similarly, for the second derivative, the central difference approximation gives:

$$\left. \frac{\partial^2 u}{\partial x^2} \right|_i \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} \quad (1.17)$$

Using these approximations for derivatives in an equation, leads to a linear system that can be written compactly as a *tridiagonal matrix equation*, as an example:

$$\begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ 0 & -1 & 2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-1} \end{bmatrix} = \Delta x^2 \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_{N-1} \end{bmatrix} + \begin{bmatrix} u_0 \\ 0 \\ 0 \\ \vdots \\ u_N \end{bmatrix} \quad (1.18)$$

This system arises from discretizing a 1D Poisson or diffusion equation with Dirichlet boundary conditions. Hence our problem is reduced from being one in calculus, to one in Linear algebra. Powerful algorithms, such as the Thomas algorithm can be leveraged to solve such systems.(appendix B)

Now let us introduce Spectral methods. They are a class of numerical techniques for solving differential equations by expanding the solution in terms of defined basis functions, such as sines, cosines, or orthogonal polynomials. Unlike finite difference methods that approximate derivatives using values at neighboring points, spectral methods approximate the entire solution as a weighted sum of known functions[9].

$$u(x, z, t) = \sum_{n=0}^N u_n^{(c)}(z, t) \cos\left(\frac{n\pi x}{a}\right) + \sum_{n=1}^N u_n^{(s)}(z, t) \sin\left(\frac{n\pi x}{a}\right), \quad (1.19)$$

where:

- $u_n^{(c)}(z, t)$ are the cosine mode amplitudes (even functions in x),
- $u_n^{(s)}(z, t)$ are the sine mode amplitudes (odd functions in x),
- The sine sum starts from $n = 1$ because $\sin(0) = 0$,
- The factor $\frac{n\pi}{a}$ sets the spatial frequency in the x -direction.

This real-valued expansion is particularly useful when the physical domain or boundary conditions imply certain symmetries (e.g., free-slip, no-flux, or periodic boundaries in x). The z -dependence is typically handled using a separate numerical method, such as finite differences or Chebyshev polynomials.

These methods are especially powerful when the solution is smooth and the geometry is simple (like periodic domains or rectangular boxes). The accuracy of spectral methods increases exponentially with the number of modes used.

However, they are not ideal for domains with complicated geometries or sharp discontinuities, where local methods like finite difference or finite element approaches may work better.

1.4.2 Advancing Solutions with Time

Once the spatial discretization is handled—either by finite difference, spectral, or other methods—the next step in solving a time-dependent partial differential equation is to advance the solution forward in time. This means approximating the time derivative $\partial u / \partial t$ using discrete steps, so that the solution evolves in small time increments from a known initial state.

A general form of a time-dependent PDE looks like:

$$\frac{\partial u}{\partial t} = \mathcal{L}(u), \quad (1.20)$$

where $\mathcal{L}(u)$ is some spatial operator (e.g., containing derivatives in space, diffusion, or advection terms).

To solve this numerically, we approximate $u(x, t)$ at discrete time levels $t^n = n\Delta t$. The most basic approach is the forward Euler method:

$$u^{n+1} = u^n + \Delta t \mathcal{L}(u^n), \quad (1.21)$$

which is first-order accurate in time and easy to implement, but can become unstable for stiff problems or small spatial steps. Higher-order methods like Runge–Kutta schemes are commonly used to improve stability and accuracy.

There are two types of time stepping methods : explicit and implicit. Explicit time-stepping methods compute the solution at the next time step directly from known quantities at the current step, while implicit methods involve solving equations that depend on the unknown future state (therefore more computationally expensive). For simplicity and ease of implementation, we will use explicit methods in this work, which are suitable for problems where the time step can be kept sufficiently small to maintain stability.

[13]Explicit methods are simple and efficient but often require small time steps due to stability limits like the CFL condition. A key idea here is that the time step Δt must be small enough to resolve the transport of information between fluid elements—typically, it should not exceed the time required for a fluid parcel to move across one grid cell. This constraint is closely tied to the Courant–Friedrichs–Lowy (CFL) condition [13], which governs the stability of explicit schemes.

In this work, we use the Adams–Bashforth method, a widely used explicit multistep technique that provides a good balance between accuracy and computational efficiency.

1.5 Structure of this Thesis and Layout

This thesis is organized to reflect both the logical development of the numerical model and the physical intuition behind Rayleigh–Bénard convection.

Chapter 1 begins with the physical background and motivation for studying thermal convection, leading to a discussion of the governing fluid equations and the Boussinesq approximation. Chapter 2 introduces the vorticity–streamfunction formulation, outlines the numerical discretization strategies used, and presents the solver algorithm in detail. Chapter 3 focuses on implementation, beginning with the linear model and gradually introducing nonlinear effects, including mode initialization strategies and how flow patterns emerge.

Chapter 4 presents the results—starting from validation against linear theory and progressing to nonlinear and turbulent behavior at higher Rayleigh numbers. The final chapter offers conclusions, key reflections, and suggestions for future improvement.

The appendices include technical derivations (such as non-dimensionalization), numerical algorithms like the Thomas method, and additional figures that support the main results but were not included in the core discussion.

The structure is intended to guide the reader from foundational concepts to advanced simulation results in a coherent and accessible manner.

Chapter 2

Modeling and Discretization of the Convection Problem

2.1 Vorticity-Streamfunction Formulation

To simplify the numerical treatment of the incompressible Navier–Stokes equations for 2D Rayleigh–Bénard convection, we work with a vorticity–streamfunction formulation[9]. This approach removes pressure from the equations and naturally satisfies the incompressibility condition.

We begin by taking the curl of the momentum equation to eliminate the pressure term. In 2D, where the velocity field \vec{v} has components only in the x and z directions (i.e., $v_y = 0$ and $\partial/\partial y = 0$), the vorticity ω has only a y -component and is given by:

$$\omega = (\nabla \times \vec{v})_y = \frac{\partial v_x}{\partial z} - \frac{\partial v_z}{\partial x}. \quad (2.1)$$

Taking the curl of the momentum equation, and applying standard vector identities, we arrive at an evolution equation for the vorticity:

$$\frac{\partial \omega}{\partial t} = -(\vec{v} \cdot \nabla)\omega - Ra \cdot Pr \frac{\partial T}{\partial x} + Pr \nabla^2 \omega. \quad (2.2)$$

This equation shows that vorticity changes due to advection, thermal buoyancy, and viscous diffusion. The term $-(\vec{v} \cdot \nabla)\omega$ transports vorticity along the flow; the $-RaPr\partial T/\partial x$ term represents the generation of vorticity due to temperature gradients, and $Pr\nabla^2\omega$ accounts for viscous smoothing of vorticity variations.

To recover the velocity field from the vorticity, we introduce a streamfunction $\psi(x, z, t)$ such that:

$$v_x = -\frac{\partial \psi}{\partial z}, \quad v_z = \frac{\partial \psi}{\partial x}. \quad (2.3)$$

This definition ensures that the continuity equation $\nabla \cdot \vec{v} = 0$ is automatically satisfied. Substituting these expressions into the definition of vorticity leads to a Poisson equation for the streamfunction:

$$\omega = -\nabla^2 \psi. \quad (2.4)$$

Solving this equation for ψ using the updated vorticity allows us to reconstruct the velocity field at each time step. Contours of the streamfunction ψ represent streamlines of the flow, with their spacing indicating the local speed of the fluid. This particular poisson equation is solved via a tridiagonal matrix, as given in Appendix B.

This vorticity–streamfunction formulation is computationally convenient because it avoids the explicit calculation of pressure and ensures incompressibility by construction.

2.2 Description of Simulation Domain

To solve the governing equations numerically, we approximate the physical Rayleigh–Bénard convection system using a discretized computational domain. This involves defining a grid of points where the fluid properties—such as velocity, temperature, and vorticity—are evaluated and evolved in time, thereby mimicking the behavior of flow, heat transport, and diffusion observed in the real system.

In our approach, we employ a hybrid numerical scheme: finite differences are used to discretize the vertical direction, while a spectral decomposition using trigonometric basis functions is applied along the horizontal direction. This allows us to efficiently capture vertical gradients with flexibility while leveraging the accuracy of spectral methods in the periodic horizontal direction.

2.2.1 Spectral Decomposition along the Horizontal

In the horizontal direction (x), we are decomposing the temperature field $T(x, z, t)$ using a Fourier series with only the cosine component, which is a natural expansion incorporating the boundary conditions:

$$T(x, z, t) = \sum_{m=0}^{N_n} \hat{T}_m(z, t) \cos\left(\frac{m\pi x}{a}\right) \quad (2.5)$$

Where:

- x is the horizontal coordinate, $0 \leq x \leq a$.
- z is the vertical coordinate, discretized using finite differences. t is time.
- m is the horizontal wavenumber. $m = 0$ is the mean flow. $m > 0$ represent horizontal variations.
- $\hat{T}_m(z, t)$ and $\tilde{T}_m(z, t)$ are the spectral coefficients (functions of z and t).
- N_n is the number of horizontal modes.

Physical Interpretation of the modes :

- $n = 0$ mode :

The term becomes $T_0(z, t)\cos(0) = T_0(z, t)$. This mode represents the component of the temperature field that is uniform in the horizontal direction. It only varies with the vertical coordinate z and time t . $T_0(z, t)$ is the horizontal average of the temperature at each vertical level.

- $n = 1$ mode :

the term is $T_1(z, t)\cos(\frac{\pi x}{a})$. The cosine function $\cos(\frac{\pi x}{a})$ completes half of its wavelength within the domain $0 \leq x \leq a$. It goes from $\cos(0) = 1$ at $x = 0$ to $\cos(\pi) = -1$ at $x = a$. The full wavelength of this mode is $\lambda_1 = \frac{2\pi}{(\pi/a)} = 2a$.

The Vorticity field, $\omega(x, z, t)$ is expanded in sines, again pertaining to the boundary conditions. Revisiting the equations that need to be solved in their final form, after all the approximations have been made, and all of the numerical schemes discussed yet have been implemented [9] :

The temperature equation -

$$\frac{\partial T_n}{\partial t} = -[(\mathbf{v} \cdot \nabla)T]_n + \left(\frac{\partial^2 T_n}{\partial z^2} - \left(\frac{n\pi}{a} \right)^2 T_n \right) \quad (2.6)$$

The vorticity equation -

$$\frac{\partial \omega_n}{\partial t} = -[(\mathbf{v} \cdot \nabla)\omega]_n + Ra \cdot Pr \cdot \frac{n\pi}{a} T_n + Pr \left(\frac{\partial^2 \omega_n}{\partial z^2} - \left(\frac{n\pi}{a} \right)^2 \omega_n \right) \quad (2.7)$$

and streamfunction update -

$$\omega_n = - \left(\frac{\partial^2 \psi_n}{\partial z^2} + \left(\frac{n\pi}{a} \right)^2 \psi_n \right) \quad (2.8)$$

2.2.2 Finite Differencing along the Vertical

For the main simulation, along the vertical, we will use finite differencing as mentioned in section 1.4.1. Approximating the above equations, we get, the linear terms as :

$$\begin{aligned} \text{dtemdt}[k, n, 1] &= \text{oodz2} \cdot (\text{tem}[k+1, n] - 2 \cdot \text{tem}[k, n] + \text{tem}[k-1, n]) \\ &\quad - (n \cdot c)^2 \cdot \text{tem}[k, n], \\ \text{domgdt}[k, n, 1] &= \text{Ra} \cdot \text{Pr} \cdot (n \cdot c) \cdot \text{tem}[k, n] \\ &\quad + \text{Pr} \cdot (\text{oodz2} \cdot (\text{omg}[k+1, n] - 2 \cdot \text{omg}[k, n] + \text{omg}[k-1, n]) \\ &\quad - (n \cdot c)^2 \cdot \text{omg}[k, n]) \end{aligned} \quad (2.9)$$

Here, oodz2 is $1/(dz^2)$, the numerical form for $1/\Delta^2$, and $\text{domgdt}[k, n, 1]$, $\text{tem}[k-1, n]$ etc are the arrays used for storing the values of the quantities.

The above equations do not include the non-linear quantities as presented within square brackets in Eqn. (2.7),(2.6), as they are more involved and require a special mathematical treatment, given in section ,

2.2.3 Physical Significance behind Domain Structure

The choice of domain structure in our numerical model of Rayleigh–Bénard convection is closely tied to the underlying physics and the boundary conditions of the system. The fluid is confined between two horizontal plates and extends periodically in the horizontal (x) direction. This geometry naturally motivates a hybrid numerical scheme: a spectral decomposition in the horizontal direction and a finite difference approach in the vertical direction.

Spectral Decomposition in the Horizontal

The horizontal direction is treated using a cosine Fourier decomposition. This is physically appropriate because the lateral boundary conditions are periodic or symmetric in typical Rayleigh–Bénard setups. Expanding the temperature field as

$$T(x, z, t) = \sum_{n=0}^{N_n} \hat{T}_n(z, t) \cos\left(\frac{n\pi x}{a}\right), \quad (2.10)$$

allows us to naturally incorporate Neumann or periodic boundary conditions. The spectral method offers high accuracy for smooth fields and enables us to work efficiently in Fourier space for horizontal derivatives, reducing computational cost while preserving precision.

Finite Differencing in the Vertical

In contrast, the vertical (z) direction has rigid, no-slip boundaries at the top and bottom plates, and thermal boundary conditions (e.g., fixed temperatures or heat fluxes). These require a more localized treatment, and finite differencing is particularly well-suited here.

For instance, the second derivative in z of the temperature mode \hat{T}_n is approximated using a central difference scheme:

$$\frac{\partial^2 \hat{T}_n}{\partial z^2} \approx \frac{\hat{T}_n[k+1] - 2\hat{T}_n[k] + \hat{T}_n[k-1]}{\Delta z^2}. \quad (2.11)$$

This spatial discretization transforms the continuous partial differential equations into a solvable system of ODEs in time.

Advantages of the Hybrid Structure

The hybrid structure reflects the anisotropic nature of the problem. Horizontally, the flow tends to be smooth and often large-scale, benefitting from the spectral method's global accuracy. Vertically, the influence of sharp gradients near the boundaries (thermal and velocity boundary layers) makes local finite differences more suitable.

Furthermore, this separation simplifies the treatment of the governing equations. For instance, the temperature and vorticity equations for each mode n take the form as mentioned in Eqn. (2.7),(2.6)

By projecting the equations onto individual Fourier modes, we decouple the horizontal modes and reduce the problem to solving a set of 1D equations in the vertical direction. This significantly reduces complexity and computational load.

Thus, the hybrid domain structure is not only computationally advantageous but also grounded in the physical characteristics of the Rayleigh–Bénard system. It captures both the periodic or symmetric behavior in the horizontal and the rigid boundary layers in the vertical, allowing us to simulate convective dynamics with high fidelity and efficiency.

2.2.4 Boundary Conditions

In any numerical solution of partial differential equations, the choice of boundary conditions is critical to both physical fidelity and numerical stability. In Rayleigh–Bénard (RB) convection, the boundary conditions reflect the physical configuration: a fluid layer confined between two horizontal plates that are typically rigid, impermeable, and held at fixed temperatures.

Thermal Boundary Conditions

In our model, we impose Dirichlet boundary conditions on the temperature field for all non-zero horizontal Fourier modes, represented by array with rows corresponding to vertical z-levels and columns storing information on the modes.

$$\text{tem}[0, 1 :] = 0, \quad \text{tem}[-1, 1 :] = 0. \quad (2.12)$$

This corresponds to fixing the temperature perturbations at the top and bottom plates to zero. Physically, this means the plates maintain constant temperature, which is a common assumption in laboratory RB convection experiments where the top plate is cooler than the bottom.

Kinematic Boundary Conditions (Free-slip)

We implement free-slip velocity boundary conditions, which permit tangential motion at the plates but prohibit normal flow and shear stress. These are mathematically implemented as:

$$\psi[0, :] = 0, \quad \psi[-1, :] = 0, \quad (2.13)$$

$$\omega[0, :] = 0, \quad \omega[-1, :] = 0. \quad (2.14)$$

Here, ψ is the streamfunction, and ω is the vorticity. Setting $\psi = 0$ at the boundaries ensures no normal flow (i.e., $v_z = 0$), while setting $\omega = 0$ corresponds to a vanishing shear stress, enforcing a free-slip condition.

Physical Interpretation

Free-slip conditions are often used in theoretical and numerical studies to simplify the problem while still capturing the essential physics of thermal convection. Although real plates are typically no-slip (i.e., zero velocity at the boundary), the free-slip approximation still provides qualitatively accurate insights into pattern formation, heat transport, and stability behavior—especially in the absence of strong boundary layer effects.

This approach also eliminates the need to resolve thin viscous boundary layers near the walls, allowing for coarser vertical resolution and more efficient simulation. Nevertheless, for studies that emphasize near-wall dynamics or detailed heat transport, no-slip conditions may be more appropriate (see e.g., [14]).

The boundary conditions adopted here strike a balance between physical realism and numerical simplicity. They reflect an idealized setup commonly employed in RB studies and enable a clear investigation of convective dynamics without being dominated by near-wall viscous effects.

2.3 Algorithm for the Solver

The following algorithm outlines the core computational loop used in our simulation of two-dimensional Rayleigh–Bénard convection under the Boussinesq approximation. The numerical method combines a spectral decomposition in the horizontal direction with a second-order finite difference discretization in the vertical direction, capturing both the efficiency of spectral methods and the flexibility of finite difference schemes near boundaries. This hybrid approach allows for efficient computation of spatial derivatives and accurate enforcement of boundary conditions.

Time integration is carried out using a second-order Adams–Bashforth scheme, and the nonlinear terms are evaluated using spectral convolution in Fourier space. The streamfunction–vorticity formulation is employed to reduce the system to scalar evolution equations, and the Poisson equation for the streamfunction is solved at each timestep using a tridiagonal matrix solver.

This implementation closely follows the scheme described in Glatzmaier’s work [9], and we do not claim originality in the structure of the algorithm itself. Instead, our focus is on verifying and adapting the method to the current simulation context.

The algorithm below summarizes the full procedure for evolving the temperature, vorticity, and streamfunction fields over time.

Algorithm 1: Spectral–Finite Difference Solver for 2D Rayleigh–Bénard Convection

Input: Parameters: Ra , Pr , Nz , N_n , a , dt , n_{steps}
Output: Fields: ψ , ω , T

- 1 Initialize arrays: ψ , ω , T , $\frac{dT}{dt}$, $\frac{d\omega}{dt} \leftarrow 0$;
- 2 Set initial temperature: linear profile in $n = 0$, small perturbation in $n = 1$;
- 3 **for** time step $t = 1$ **to** n_{steps} **do**
- 4 **for** grid point $k = 1$ **to** $Nz - 2$ **do**
- 5 **for** mode $n = 0$ **to** N_n **do**
- 6 Compute linear terms of $\frac{dT}{dt}$ and $\frac{d\omega}{dt}$;
- 7 **if** $n > 0$ **then**
- 8 Add buoyancy and diffusion terms to $\frac{d\omega}{dt}$;
- 9 Compute nonlinear advection terms via spectral convolution in x ;
- 10 Accumulate nonlinear contributions into $\frac{dT}{dt}$ and $\frac{d\omega}{dt}$ for all modes;
- 11 Update T and ω using Adams–Bashforth time integration;
- 12 **for** mode $n = 0$ **to** N_n **do**
- 13 Solve Poisson equation for $\psi_n(z)$ via tridiagonal solver;
- 14 Enforce boundary conditions for all fields;
- 15 Shift time derivatives for next step;
- 16 **return** ψ , ω , T ;

2.4 Flow Visualization Technique

To analyze the behavior of the convective flow over time, we visualize the temperature and streamfunction fields at selected time intervals during the simulation. These two quantities provide complementary insights into the system: the temperature field highlights thermal structures such as rising plumes and boundary layers, while the streamfunction gives a direct picture of the flow circulation through its associated velocity field.

In our implementation, the fields are evolved in spectral space along the horizontal direction and in physical space along the vertical. At each output step, we reconstruct the full two-dimensional fields by performing an inverse Fourier transform over the horizontal modes. This yields temperature $T(x, z)$ and streamfunction $\psi(x, z)$ on a uniform spatial grid.

The resulting fields are plotted using standard colormaps to indicate

scalar values across the domain. For temperature, we typically use a diverging or sequential colormap to distinguish hot and cold regions clearly. The streamfunction is visualized using contour lines, which trace out the streamlines of the flow. Because the velocity field is given by $\mathbf{u} = (-\partial\psi/\partial z, \partial\psi/\partial x)$, these contours effectively represent the flow direction and the structure of convective rolls.

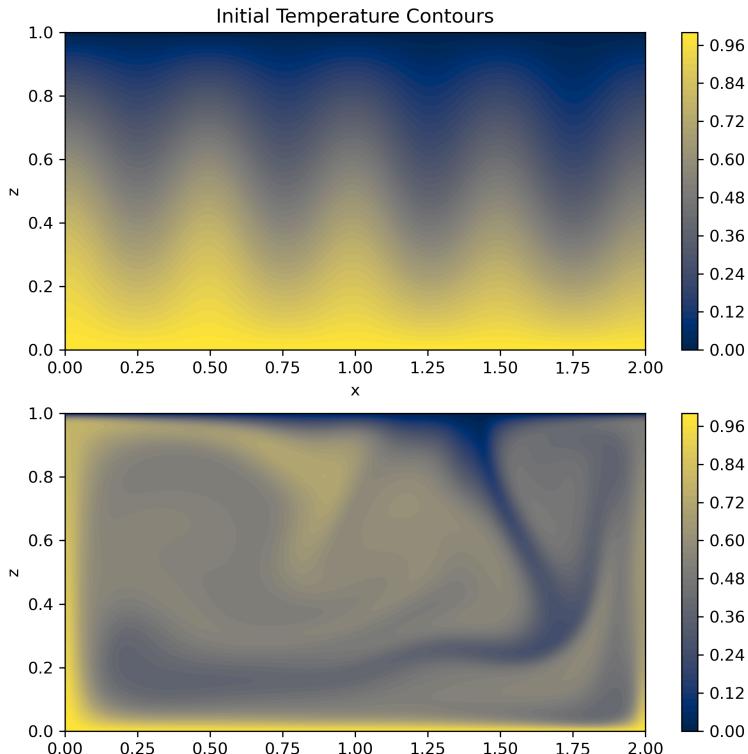


Figure 2.1: Evolution of Flow Structures

All plots are generated using Python's `matplotlib` library. These visualizations help track the development of coherent structures, transitions to unsteady or chaotic flow, and the influence of parameters such as the Rayleigh number on flow morphology.

Here is an example 2.1 for the contour plot, with the uppermost denoting initial state and the subsequent showing the evolution after time has elapsed. A detailed result and analysis section dives deeper into the structures and patterns.

2.5 Code availability

The Python code used for simulating two-dimensional Rayleigh–Bénard convection is available on GitHub. It includes all source files, configuration settings, and instructions to run the solver.

You can access the repository at the following link:

[https://github.com/Adarshramtel28/
PythonBased-Rayleigh-Benard-2D-convection-Solver.git](https://github.com/Adarshramtel28/PythonBased-Rayleigh-Benard-2D-convection-Solver.git)

Chapter 3

Implementation of the Numerical Model

3.1 Linearized Model

[9] Our methodology will involve first creating a linear version of the simulation, by ignoring the non-linear terms within square brackets in Eqn. (2.7), (2.6)

To understand the onset of convection, it's useful to first consider the linearized version of the governing equations. In this simplified setup, the temperature evolution becomes independent of the flow—it just diffuses and eventually decays. This would also cause the flow to vanish, leaving no meaningful dynamics.

To prevent this, we prescribe a simple, steady background temperature profile that varies only with height:

$$T_0(z) = 1 - z. \quad (3.1)$$

This conductive state introduces a fixed vertical temperature gradient, $\partial T_0 / \partial z = -1$, which drives buoyancy forces in the flow.

By keeping this background gradient and considering only small perturbations, we can linearize the temperature advection term and couple it to the flow. This lets us study how small disturbances grow or decay before nonlinear effects kick in. The vertical velocity needed for this coupling is obtained from the streamfunction via:

$$(v_z)_n = \frac{n\pi}{a} \psi_n. \quad (3.2)$$

This forms the basis for analyzing early-stage convection and since concrete linear stability calculations reveal precise theoretical predictions, our code can be verified.

The temperature equation -

$$\frac{\partial T_n}{\partial t} = \frac{n\pi}{a} \psi_n + \left(\frac{\partial^2 T_n}{\partial z^2} - \left(\frac{n\pi}{a} \right)^2 T_n \right) \quad (3.3)$$

Vorticity equation -

$$\frac{\partial \omega_n}{\partial t} = Ra \cdot Pr \cdot \frac{n\pi}{a} T_n + Pr \left(\frac{\partial^2 \omega_n}{\partial z^2} - \left(\frac{n\pi}{a} \right)^2 \omega_n \right) \quad (3.4)$$

Streamfunction update -

$$\omega_n = - \left(\frac{\partial^2 \psi_n}{\partial z^2} + \left(\frac{n\pi}{a} \right)^2 \psi_n \right) \quad (3.5)$$

Critical Mode and Number

A key question in the study of Rayleigh–Bénard convection is: under what conditions does a static, conductive fluid layer become unstable and begin to convect? The Rayleigh number, Ra , serves as a dimensionless measure of the balance between buoyant driving forces and viscous and thermal damping. When Ra is small, diffusion dominates and any perturbations decay. But beyond a certain threshold—called the critical Rayleigh number, Ra_{crit} —small perturbations begin to grow, marking the onset of convection[3, 12].

To determine this threshold, we analyze the linearized equations under the assumption of normal modes, where each dependent variable varies sinusoidally in both space and time. We further set time derivatives to zero, looking for the conditions under which steady (marginally stable) solutions first appear. Thanks to the orthogonality of the sine basis in z , each horizontal mode number n and vertical mode number m decouples, reducing the system to three algebraic equations for the mode amplitudes $T_{nm}, \omega_{nm}, \psi_{nm}$.

Solving this system yields the critical Rayleigh number for each mode:

$$Ra_{\text{crit}}(n, m) = \left(\frac{\pi}{a} \right)^4 \frac{(n^2 + (am)^2)^3}{n^2}, \quad (3.6)$$

where a is the aspect ratio of the domain. The most unstable mode—i.e., the one that triggers convection first—is the one that minimizes Ra_{crit} . The corresponding horizontal wavenumber is given by:

$$n_{\text{crit}} = \text{nearest integer to } \left(\frac{a}{\sqrt{2}} \right). \quad (3.7)$$

This result identifies the preferred spatial pattern at the onset of convection: typically a series of convection rolls with horizontal wavelength tuned to the geometry of the domain.

Given below is in tabular form, different values of Ra_{crit} for different mode numbers. We shall verify these in the results section. As a benchmark case, we will present the trend at aspect ratio , $a = \sqrt{2}$ for which, the critical mode number, $n = 1$ (Eq (3.7) and Critical Rayleigh number = 657.

Aspect Ratio	Critical Ra	Critical n
0.5	3044.034	0
1.0	779.273	1
$\sqrt{2}$	657.5	1
1.5	660.518	1
2.0	761.009	1
2.5	950.284	2
3.0	1202.581	2
3.5	1509.993	2
4.0	1869.417	3
4.5	2279.430	3
5.0	2739.299	4
5.5	3248.619	4
6.0	3807.147	4
6.5	4414.733	5
7.0	5071.277	5
7.5	5776.715	5
8.0	6530.999	6
8.5	7334.097	6
9.0	8185.986	6
9.5	9086.648	7
10.0	10036.068	7
10.5	11034.238	7
11.0	12081.149	8
11.5	13176.795	8

Table 3.1: Critical Rayleigh number and corresponding critical mode number n_{crit} as a function of aspect ratio, in steps of 0.5.

3.2 Adding Non-Linear Terms

The linear solution describes only the very early stage of convection, where small temperature perturbations grow exponentially due to instability. However, this growth cannot continue indefinitely. As the amplitudes of velocity and temperature increase, the non-linear terms in the governing equations—especially those involving products like $(\mathbf{v} \cdot \nabla)\omega$ and $(\mathbf{v} \cdot \nabla)T$ —start to play a dominant role. These terms eventually halt the exponential growth, leading the system into what is called a *finite amplitude regime*. In this regime, the magnitudes of variables are constrained by the full non-linear dynamics and cannot be freely scaled. For instance, doubling all amplitudes would double linear terms, but would quadruple the non-linear terms, violating the balance required by the equations.

To capture this behavior, we must include the complete non-linear advection terms in the vorticity and temperature equations:

$$-(\mathbf{v} \cdot \nabla)\omega = - \left(v_x \frac{\partial \omega}{\partial x} + v_z \frac{\partial \omega}{\partial z} \right), \quad (3.8)$$

$$-(\mathbf{v} \cdot \nabla)T = - \left(v_x \frac{\partial T}{\partial x} + v_z \frac{\partial T}{\partial z} \right). \quad (3.9)$$

The velocity components themselves are given in terms of the streamfunction ψ via:

$$v_x = -\frac{\partial \psi}{\partial z}, \quad v_z = \frac{\partial \psi}{\partial x}. \quad (3.10)$$

Unlike in the linear case, the horizontally averaged temperature component $T_0(z, t)$ is no longer fixed. It now evolves as part of the solution, though it must still satisfy the boundary conditions $T_0(0, t) = 1$ and $T_0(1, t) = 0$. This adds a new layer of realism, as the background temperature profile can now be altered by convective motion.

The non-linear terms involve products of fields that are expanded in Fourier series in the horizontal direction. To evaluate these terms consistently, we apply trigonometric identities:

$$\sin a \cos b = \frac{1}{2}[\sin(a + b) + \sin(a - b)], \quad (3.11)$$

$$\cos a \cos b = \frac{1}{2}[\cos(a + b) + \cos(a - b)], \quad (3.12)$$

and so on, which allows us to express products of modes as sums over single-mode trigonometric functions. This lets us isolate the contribution to each Fourier mode from interactions of all other pairs of modes.

We compute the resulting mode contributions using a *Galerkin method*, where all calculations remain in spectral space for the x direction. For example, the non-linear advection of vorticity becomes a double summation over mode indices n' and n'' , and the contribution to a specific mode n is given by:

$$[(\mathbf{v} \cdot \nabla) \omega]_n = \frac{\pi}{2a} \sum_{n'} \sum_{n''} \left[\left(-n' \frac{\partial \psi_{n''}}{\partial z} \omega_{n'} + n'' \psi_{n''} \frac{\partial \omega_{n'}}{\partial z} \right) \delta_{n''+n',n} \right. \quad (3.13)$$

$$\left. - \left(n' \frac{\partial \psi_{n''}}{\partial z} \omega_{n'} + n'' \psi_{n''} \frac{\partial \omega_{n'}}{\partial z} \right) (\delta_{n''-n',n} - \delta_{n'-n'',n}) \right], \quad (3.14)$$

where $\delta_{i,j}$ is the Kronecker delta function.

The same approach is used for the temperature advection term, keeping in mind that temperature is expanded in cosines and includes the $n = 0$ mode, which must be treated separately.

This Galerkin projection method allows us to remain entirely in spectral space when computing the non-linear terms, avoiding the need to transform to physical space in the x direction. Although less efficient than spectral-transform techniques, it provides clear insight into how interactions between different horizontal modes feed energy into new modes, leading to the rich dynamics of time-dependent convection observed at higher Rayleigh numbers.

3.2.1 Initial Temperature for simulation

$n = 0$ Mode:

$T_0(z, t = 0) = 1 - z$. This represents a linear temperature profile in the vertical, which is uniform across the horizontal domain.

$n = 1$ Mode:

$T_1(z, t = 0) = 0.01 \sin(\pi z)$. This introduces a horizontal variation with half a wavelength across the domain. The amplitude of this variation changes sinusoidally with height, being zero at the top and bottom boundaries and maximum at the mid-height.

$n > 1$ Modes:

$T_n(z, t = 0) = 0$ for $n > 1$. These modes are initially zero, meaning there are no initial horizontal temperature variations with shorter wavelengths.

Physical Interpretation of the Initial State:

Our initial state describes a fluid that:

- Has a base temperature profile that decreases linearly with height and is uniform horizontally (due to the $n = 0$ mode).
- Has a small superimposed horizontal temperature perturbation (due to the $n = 1$ mode). This perturbation has a horizontal structure that varies as $\cos\left(\frac{\pi x}{a}\right)$, meaning it has one region of higher temperature and one region of lower temperature across the domain. The strength of this horizontal variation is very weak (1% of the background temperature scale) and its vertical structure is such that it's strongest in the middle of the domain and vanishes at the top and bottom boundaries.

This initial condition is again designed to study the stability of the linearly stratified fluid. The small $n = 1$ perturbation acts as a trigger. If the Rayleigh number is high enough, this perturbation (or other modes that might become unstable) will grow, leading to convective patterns in the flow. The choice of the $n = 1$ mode often corresponds to the fastest growing mode in Rayleigh-Bénard convection with these types of boundary conditions.

Chapter 4

Results and Analysis

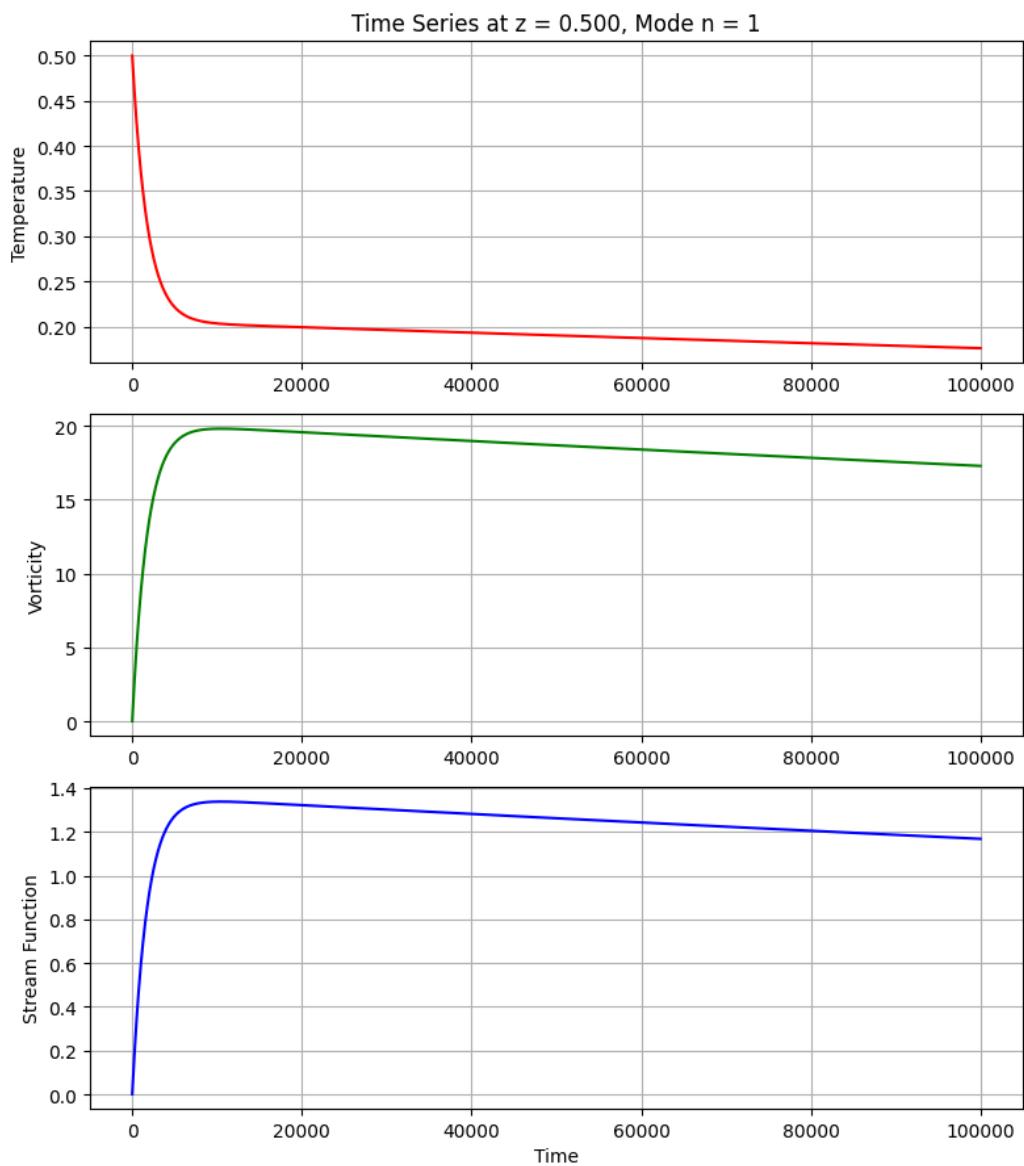
4.1 Linear Stability Validation

Each horizontal mode number n is associated with a specific threshold Rayleigh number, called the *critical Rayleigh number*, denoted $\text{Ra}_{\text{crit}}(n)$. When the Rayleigh number Ra is greater than this threshold, i.e., $\text{Ra} > \text{Ra}_{\text{crit}}(n)$, the corresponding mode becomes *unstable* and its amplitudes— $T_n(z, t)$, $\omega_n(z, t)$, and $\psi_n(z, t)$ —grow exponentially with time. This is referred to as a *supercritical* solution. Conversely, if $\text{Ra} < \text{Ra}_{\text{crit}}(n)$, the mode is *subcritical*, and its amplitudes decay exponentially.

Different modes have different values of $\text{Ra}_{\text{crit}}(n)$, and the mode that becomes unstable first—i.e., the one with the smallest $\text{Ra}_{\text{crit}}(n)$ —is called the *critical mode number*, denoted n_{crit} . The corresponding Rayleigh number $\text{Ra}_{\text{crit}}(n_{\text{crit}})$ is the overall critical Rayleigh number for the system.

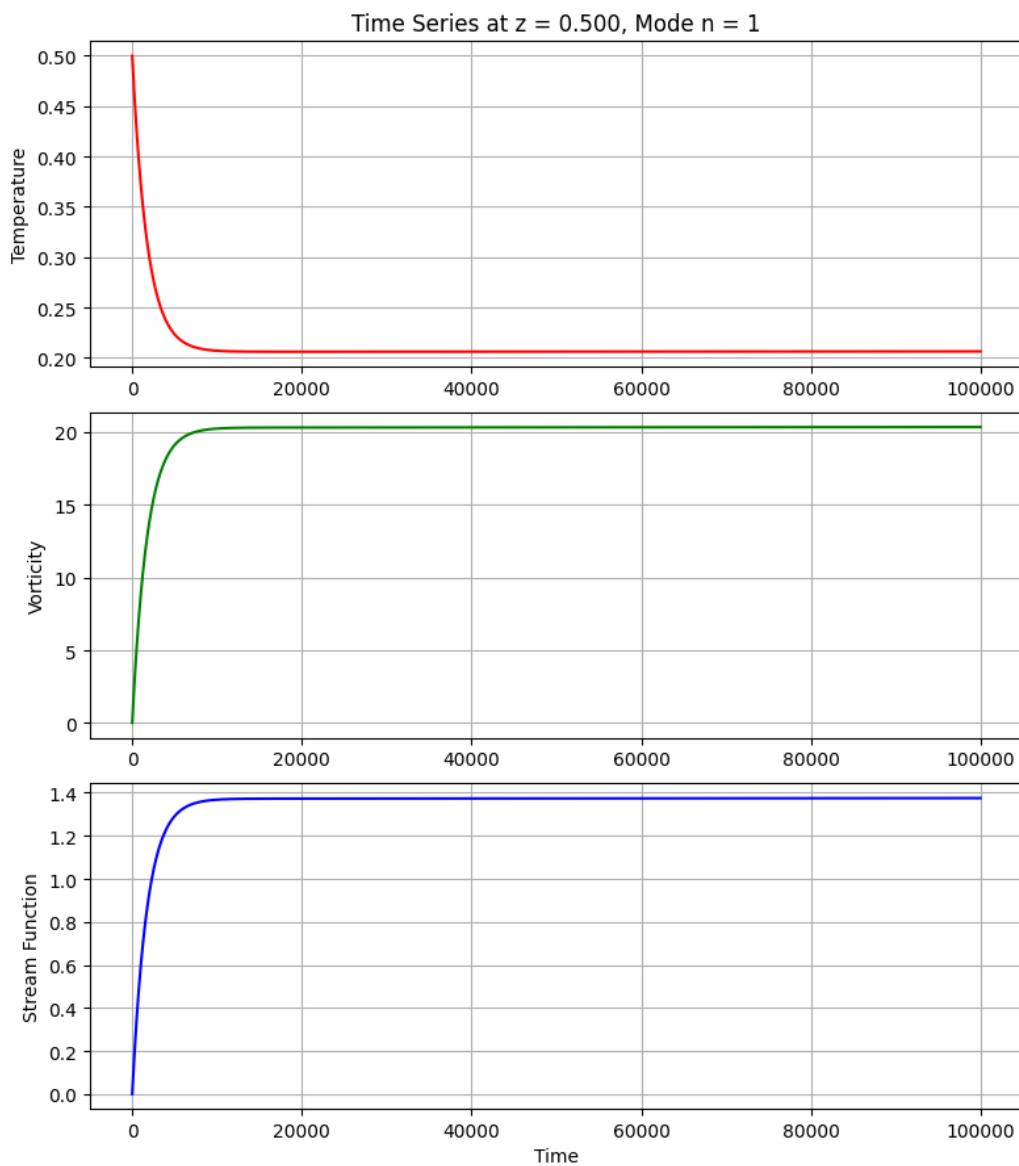
We shall display below the benchmark case of aspect ratio $a = \sqrt{2}$, where critical Rayleigh number is 657, and show how our code effectively captures (with some amount of numerical artefact errors), the transition from subcritical to critical and then supercritical growth in parameter amplitudes.

Here, temperature, vorticity and streamfunction values are monitored over timesteps, and three simulations are displayed out of a number of simulations across different values of Rayleigh numbers. The Prandtl number is kept at 0.5 and the aspect ratio fixed at squared root of 2. In Appendix C, figures for aspect ratio 3, critical number about 1200 is given.



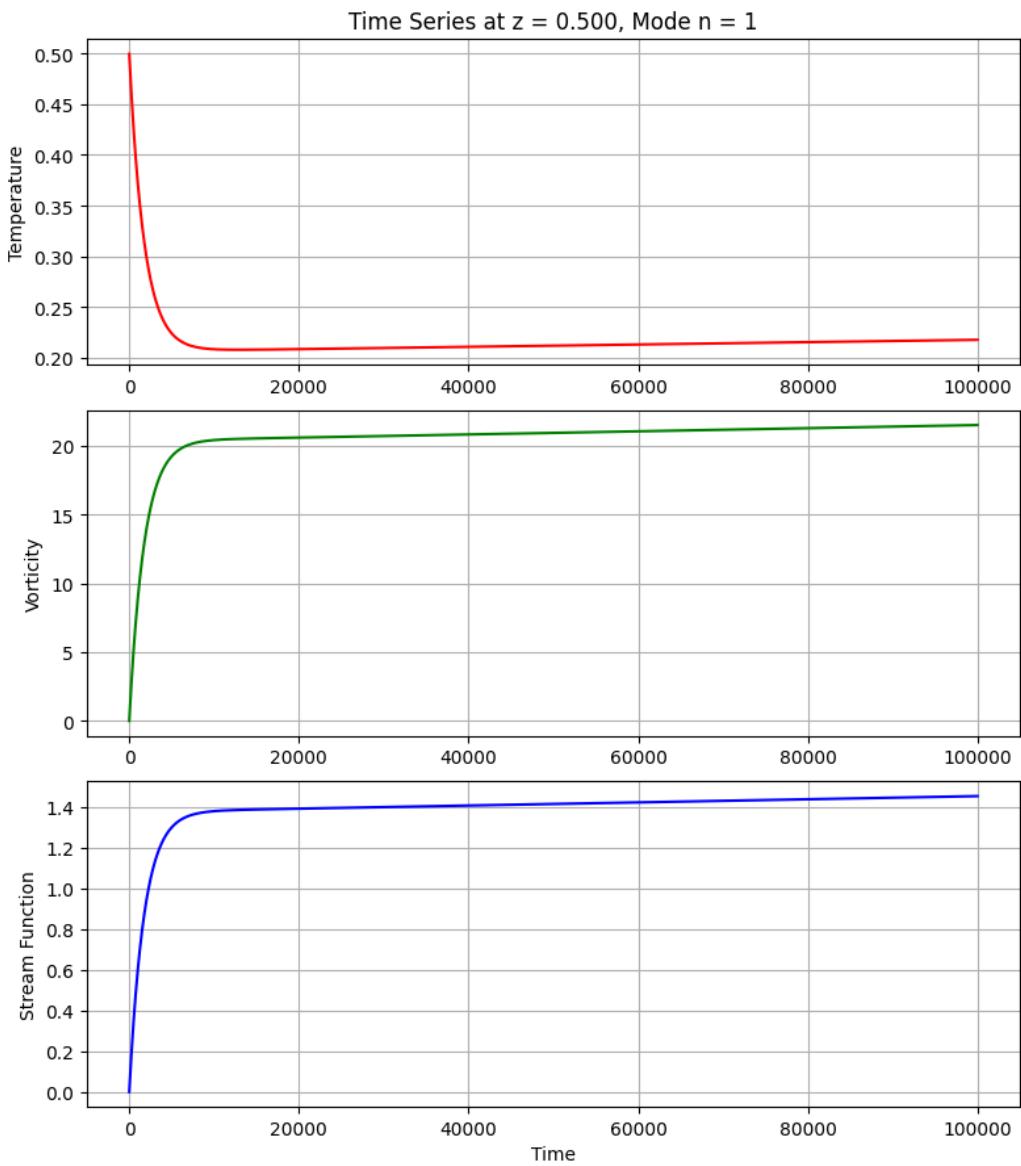
$\text{Ra} = 650$, $\text{Pr} = 0.7$, Steps = 100000, $\text{dt} = 0.000023$

Figure 4.1: $\text{Ra} = 650$ subcritical solution



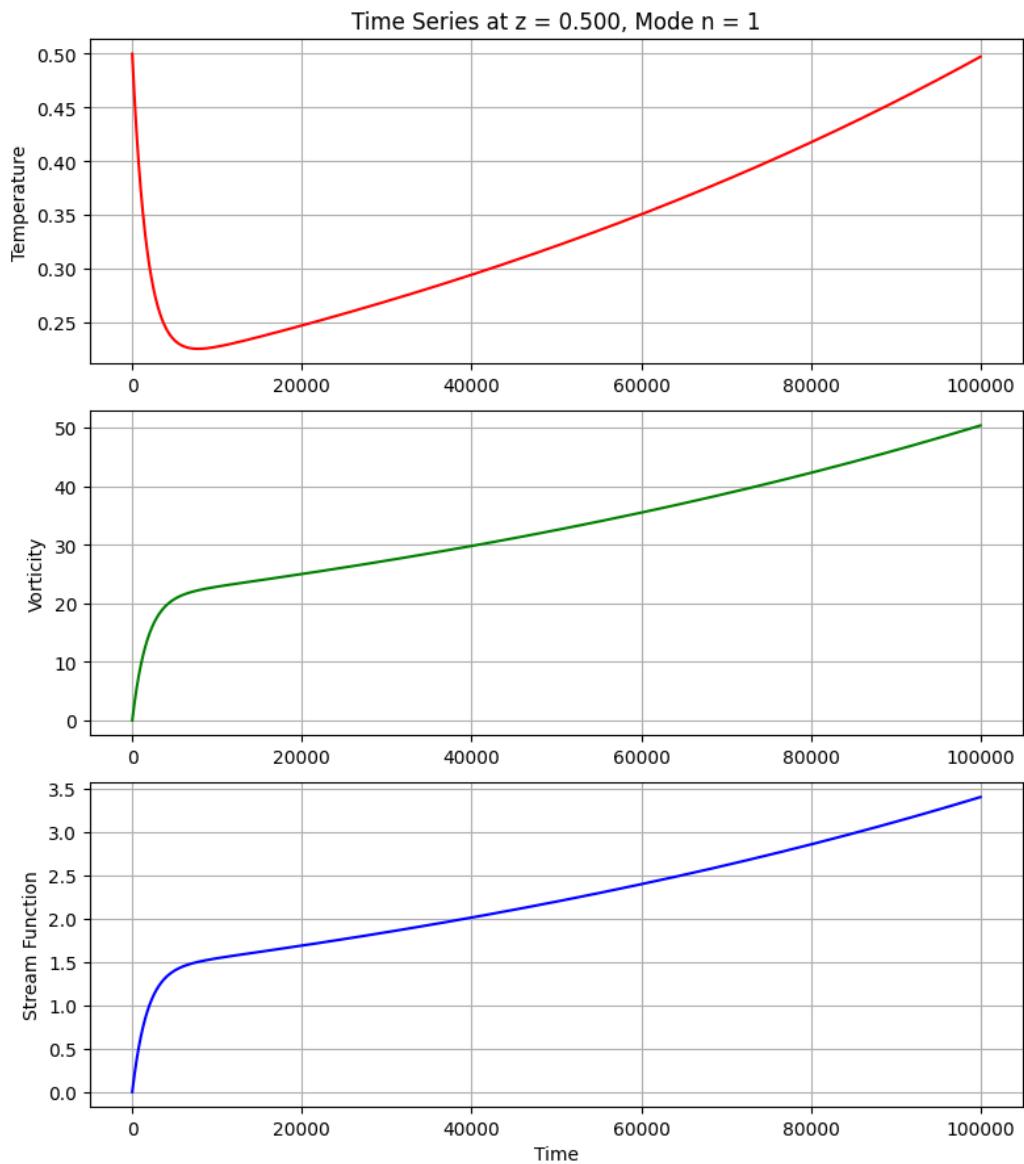
Ra = 657.5, Pr = 0.7, Steps = 100000, dt = 0.000023

Figure 4.2: Ra = 657 critical solution



$\text{Ra} = 660$, $\text{Pr} = 0.7$, Steps = 100000, $\text{dt} = 0.000023$

Figure 4.3: $\text{Ra} = 660$ supercritical solution

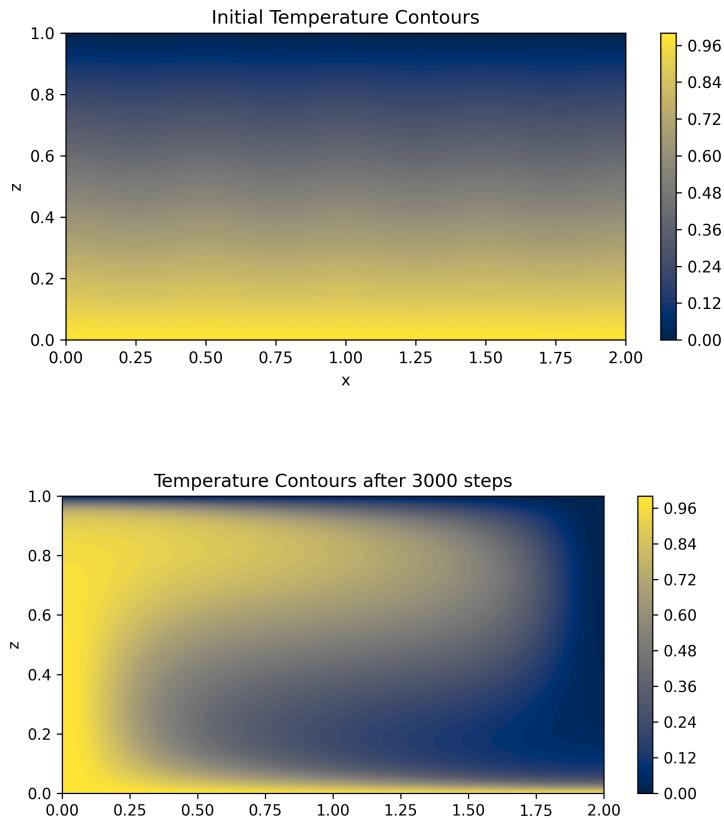


$\text{Ra} = 700$, $\text{Pr} = 0.7$, Steps = 100000, $\text{dt} = 0.000023$

Figure 4.4: $\text{Ra} = 700$ supercritical solution

4.2 Non-linear regime : Emergent Flow structures and Patterns

We display two initializations of temperature as mentioned in section 3.2.1. We start with initializing only 1 mode, that is $n = 1$, other than our background linearly decreasing temperature gradient, $1 - z$, with $0.01 * 0.01 * np.sin(np.pi * zvals)$. This effectively is supposed to produce a single convection roll in the system. The result is shown with intermediate transitions :



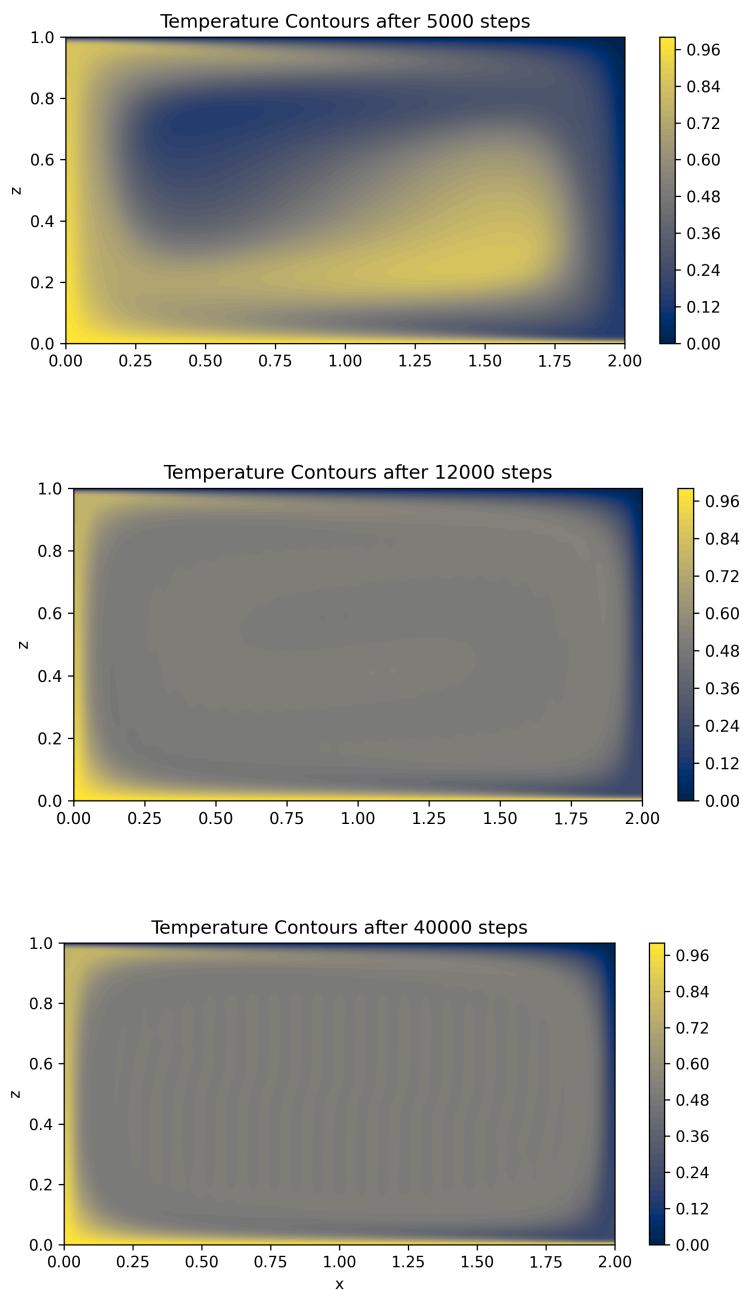
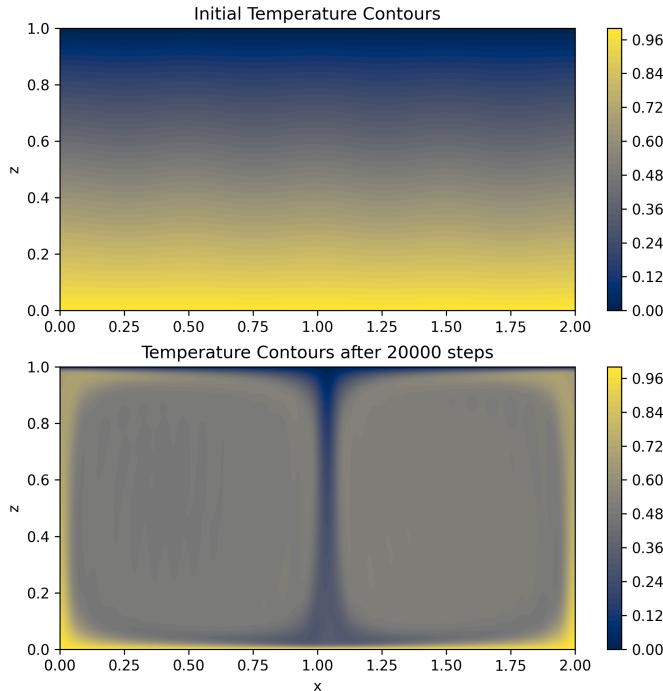


Figure 4.5: Time evolution of temperature contours, with system attaining stability with one convection roll. $Ra = 1e6$, $Pr = 0.5$, $aspect = 3$

Next, we initialize one more mode, that is mode 8 with similar amplitude as mode 1 : with $0.01 * np.sin(np.pi * zvals)$. When we initialize mode 1 and mode 8 in the horizontal direction, we're essentially seeding the fluid with both large-scale and finer-scale temperature disturbances. Mode 1 tries to form one big roll across the domain, while mode 8 introduces small ripples.

As the system evolves, nonlinear interactions often amplify the most energetically favorable pattern — and for our aspect ratio of 3, this can leads to the fluid settling into a structure with two convection rolls. So even though we started with more than one mode, the dynamics naturally select a flow pattern where two rolls fit best, influenced by the initial mix of long and short wavelengths.



4.3 Turbulent regimes : High Ra values

Brief Note on Turbulent Regimes in Rayleigh–Bénard Convection

While the focus of this thesis is on the onset and early nonlinear evolution of Rayleigh–Bénard convection, it is worth briefly noting the rich and complex behavior that emerges in the turbulent regime at very high Rayleigh numbers ($Ra \gg 10^7$). In this limit, convection becomes highly vigorous, with a cascade of eddies and plumes that span a wide range of spatial and temporal scales.

In turbulent Rayleigh–Bénard convection[3], the flow structure is dominated by strong thermal plumes and a large-scale circulation that constantly reorganizes. Heat transport in this regime deviates significantly from laminar predictions and is often characterized by empirical scaling laws Reynolds number Re as functions of Ra and the Prandtl number Pr . In this regime, the role of Prandtl starts to become important in determining the nature of the turbulence.

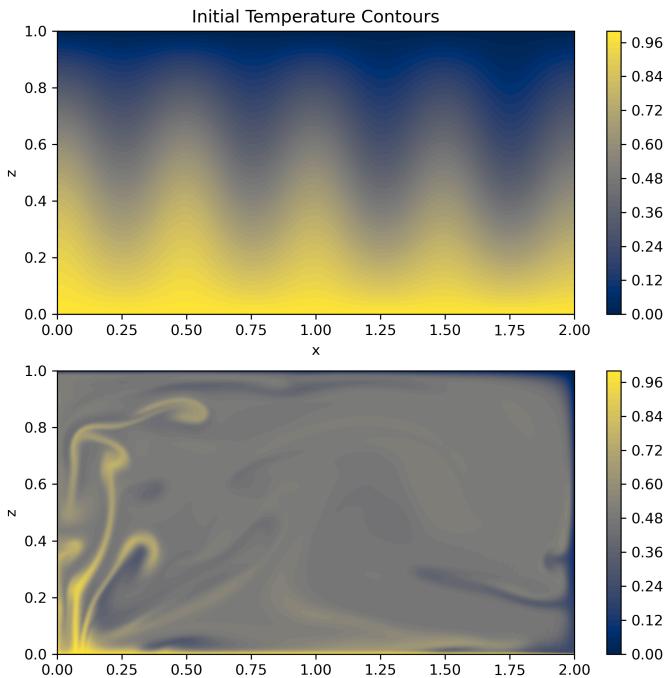


Figure 4.6: $Ra = 10^8$, $Pr = 1.0$, after $2e5$ timesteps

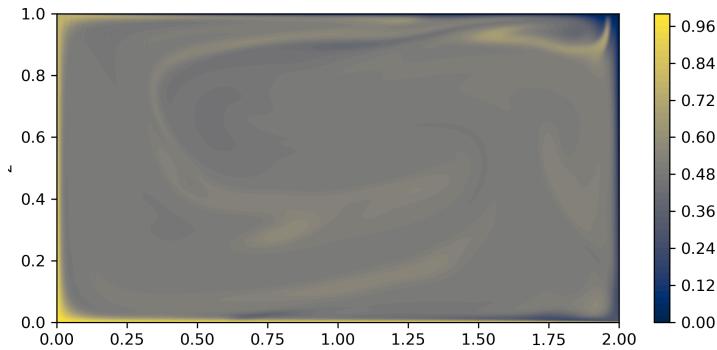


Figure 4.7: $Ra = 10^8$, $Pr = 2.0$, after $2e5$ timesteps

We can see higher Pr leads to faster developing flows.

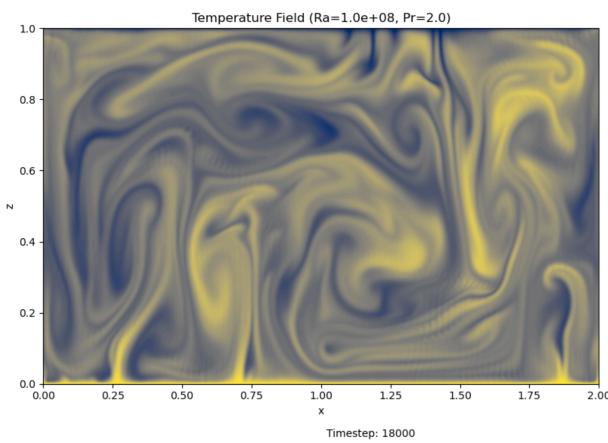
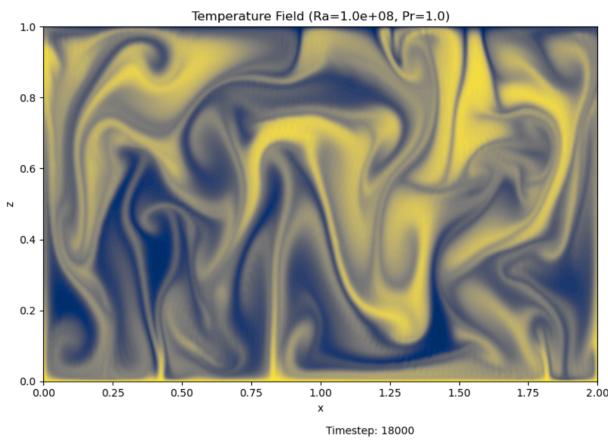


Figure 4.8: $Ra = 10^8$, $Pr = 1$ and $Pr = 2.0$, after $1.8e4$ timesteps

Physical Interpretation

Theoretical Support: The Prandtl number (Pr) is a ratio of momentum diffusivity (viscosity) to thermal diffusivity. A higher Pr means heat diffuses relatively slower compared to momentum. In thermal convection, this often leads to sharper thermal gradients and stronger coupling between the velocity and temperature fields, which can drive instabilities and turbulence to develop more quickly.

The last two images above show simulations of how temperature is distributed in a fluid over time (the "Temperature Field") under specific conditions (defined by the Rayleigh number, Ra). The top image ($\text{Pr} = 1$) shows some swirling patterns, while the bottom image ($\text{Pr} = 2$) at the same time step shows much more intricate, complex, and finer-scale structures, indicating the flow has become more chaotic and developed faster.

4.4 Sensitivity to Resolution and Time Step

An important aspect of numerical simulations, especially in fluid dynamics, is ensuring that the spatial and temporal resolution are fine enough to accurately capture the essential features of the flow. In the present study, we examined how the choice of vertical grid points (Nz), number of horizontal modes (Nn), and time step (dt) influence the stability and accuracy of the solution across varying Rayleigh numbers.

For moderate Rayleigh numbers, such as $\text{Ra} = 10^6$, we observed that the system could be evolved stably with relatively modest resolution: $\text{Nz} = 101$, $\text{Nn} = 50$, and $\text{dt} = 1 \times 10^{-6}$. However, as we increased the Rayleigh number to 10^8 and 10^{10} , we found that significantly finer resolution and smaller time steps were required to maintain numerical stability and to resolve the increasingly sharp thermal and velocity gradients.

Specifically, for $\text{Ra} = 10^8$, we needed to double the number of vertical points and modes to $\text{Nz} = 201$ and $\text{Nn} = 100$, while halving the time step to $\text{dt} = 0.5 \times 10^{-6}$. At the extreme end, for $\text{Ra} = 10^{10}$, we required $\text{Nz} = 401$, $\text{Nn} = 200$, and a time step as small as $\text{dt} = 0.1 \times 10^{-6}$.

This sensitivity is expected and reflects the nature of high-Rayleigh number convection. As the Rayleigh number increases, the flow becomes more vigorous and the thermal boundary layers near the top and bottom boundaries become thinner. Capturing these layers accurately demands a finer spatial resolution. Likewise, the velocity field develops sharper structures and faster dynamics, which require smaller time steps to satisfy stability criteria, especially for explicit or semi-implicit time integration schemes.

If these adjustments are not made, the simulation either fails to converge or yields physically incorrect results due to numerical instabilities or under-resolution of key features. Thus, adapting the resolution and time step to the Rayleigh number is a necessary part of running robust and meaningful simulations in this context.

Chapter 5

Conclusion

5.1 Limitations and Future Work

Like any simulation-based project, this work involves a number of simplifying assumptions and trade-offs made for the sake of clarity, computational feasibility, and interpretability. The model is restricted to two spatial dimensions and uses the Boussinesq approximation, which assumes small temperature variations and neglects compressibility effects. While this is perfectly valid for capturing the onset and early stages of convection, it limits the realism of the turbulent regime, especially when extending the model toward astrophysical or geophysical contexts where compressibility becomes important.

Another simplification lies in the boundary conditions. We employed free-slip conditions on the top and bottom plates, which are mathematically convenient and reduce numerical stiffness, but do not fully capture the boundary-layer dynamics present in real-world convection experiments. Implementing no-slip conditions and examining their impact on flow structures—particularly at higher Rayleigh numbers—would be a natural next step.

The hybrid numerical method used here strikes a balance between flexibility and performance, but it can be improved. For example, adaptive mesh refinement (AMR) in the vertical direction could allow for better resolution of thermal and velocity boundary layers without a significant increase in computational cost. Additionally, the current spectral method in the horizontal direction could be extended to deal with more general boundary geometries using Chebyshev or Legendre polynomials.

In terms of physics, expanding the model to include three-dimensional effects would dramatically enhance its applicability, particularly for studying secondary instabilities and turbulent cascades. Incorporating rotation, magnetic fields, or compositional gradients could open pathways toward simulating planetary interiors or stellar convection zones with greater fidelity.

While the code developed for this thesis is functional and modular, it can benefit from further optimization and parallelization methods such as pencil decomposition[15]. As Rayleigh numbers increase and finer resolutions are required, performance bottlenecks become increasingly limiting. Leveraging GPU acceleration or domain decomposition methods would make the solver more scalable for longer or higher-resolution runs.

The results section denote a qualitative description and validation of the work, particularly for the non-linear regime, but next, a quantitative diagnosis will be a focus. Analysis of important characteristic trends in Nusselt and Reynolds numbers[16, 17], Nu and Re, would be the next step.

In summary, this project lays a solid groundwork, but many extensions remain both possible and worthwhile. These future directions are not just technical upgrades—they represent opportunities to deepen the physical insights that numerical simulations can offer.

5.2 Closing Remarks

Summary and Reflections

This thesis set out to simulate two-dimensional Rayleigh–Bénard convection using a hybrid numerical approach combining spectral and finite-difference methods. The project began with the linearized version of the governing Boussinesq equations, which provided a solid theoretical foundation for understanding the onset of convection. We validated the solver against known critical Rayleigh numbers and mode structures, verifying that the numerical model reproduces expected results within acceptable error margins.

Building on this, we incorporated non-linear terms to allow for the emergence of complex flow patterns beyond the linear regime. The solver was then able to capture a range of behaviors—from the smooth formation of convection rolls in the near-critical regime to increasingly intricate dynamics as Rayleigh numbers were increased. Simulations in the turbulent regime

offered a glimpse into the fascinating complexity of high-Rayleigh-number convection, including the role of the Prandtl number in controlling flow morphology and the rate at which turbulence emerges.

From a computational perspective, this project served as an excellent platform to engage with core numerical techniques such as finite differencing, spectral methods, time integration schemes like Adams–Bashforth, and efficient solvers like the Thomas algorithm. Equally important was learning to balance numerical stability with computational efficiency—a practical skill that can only be internalized through trial, error, and iteration.

Personal Takeaway

A year in making, I started this thesis by familiarizing myself with the 1D heat equation, which proved difficult enough then. Now, after simulating these complex calculations and moving much closer to realistic physical world, I am indebted to this work for my gain in experience and knowledge. More than just a technical endeavor, this project taught me how subtle physics becomes tangible through the lens of computation. Watching fluid patterns evolve from a few lines of code felt almost like discovering them first-hand—each roll, plume, and instability a direct outcome of well-understood equations. Along the way, I learned to think critically about model assumptions, boundary conditions, and the limitations that come with numerical simulations.

While the model developed here is intentionally idealized—limited to two dimensions, free-slip boundaries, and moderate Prandtl numbers—it serves as a stepping stone to richer and more realistic simulations. Future extensions could include three-dimensional modeling, no-slip boundaries, adaptive meshing, or coupling with external fields. These remain open doors that I now feel more equipped to walk through.

In closing, this work has been both a technical and intellectual journey. It deepened my appreciation for fluid dynamics and numerical modeling, and above all, reaffirmed the excitement of exploring physics through computation.

Appendix A

Non-Dimensionalization

We non-dimensionalize the Boussinesq equations to reduce the number of parameters and highlight the essential physics governing the Rayleigh–Bénard convection problem.

A.1 Choice of Scales

[9] We adopt a traditional scaling based on the depth of the convection cell D , the temperature difference across the fluid layer ΔT , and the thermal diffusivity κ . The characteristic scales are:

- Length scale: D
- Time scale: $\frac{D^2}{\kappa}$ (thermal diffusion time)
- Temperature scale: ΔT
- Pressure scale: $\frac{\rho_0 \kappa^2}{D^2}$
- Velocity scale: $\frac{\kappa}{D}$

We define nondimensional variables (denoted with a tilde) as:

$$\begin{aligned} x &= D\tilde{x}, & z &= D\tilde{z}, & t &= \frac{D^2}{\kappa}\tilde{t}, \\ T &= T_0 + \Delta T \tilde{T}, & p &= \frac{\rho_0 \kappa^2}{D^2} \tilde{p}, \\ \mathbf{v} &= \frac{\kappa}{D} \tilde{\mathbf{v}}. \end{aligned}$$

Dropping the tildes henceforth for clarity, all variables will be treated as nondimensional.

A.2 Original Dimensional Equations

The dimensional Boussinesq equations for an incompressible fluid are:

$$\nabla \cdot \mathbf{v} = 0, \quad (\text{A.1})$$

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = -\frac{1}{\rho_0} \nabla p + \nu \nabla^2 \mathbf{v} + g\alpha T \hat{\mathbf{z}}, \quad (\text{A.2})$$

$$\frac{\partial T}{\partial t} + (\mathbf{v} \cdot \nabla) T = \kappa \nabla^2 T. \quad (\text{A.3})$$

A.3 Substituting Scales into the Equations

We now substitute the scaled variables into each equation.

Continuity:

$$\nabla \cdot \mathbf{v} = \frac{\kappa}{D} \nabla' \cdot \tilde{\mathbf{v}} = 0 \Rightarrow \nabla \cdot \mathbf{v} = 0 \quad (\text{A.4})$$

Momentum: Start from:

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = -\frac{1}{\rho_0} \nabla p + \nu \nabla^2 \mathbf{v} + g\alpha T \hat{\mathbf{z}}$$

Substitute the nondimensional variables:

$$\frac{\kappa}{D} \cdot \frac{\partial \tilde{\mathbf{v}}}{\partial (\frac{D^2}{\kappa} \tilde{t})} + \left(\frac{\kappa}{D} \tilde{\mathbf{v}} \cdot \nabla \right) \left(\frac{\kappa}{D} \tilde{\mathbf{v}} \right) = -\frac{1}{\rho_0} \cdot \frac{1}{D} \cdot \frac{\rho_0 \kappa^2}{D^2} \nabla \tilde{p} + \nu \cdot \frac{1}{D^2} \cdot \left(\frac{\kappa}{D} \tilde{\mathbf{v}} \right) + g\alpha \Delta T \tilde{T} \hat{\mathbf{z}}$$

Simplifying:

$$\frac{\kappa^2}{D^3} \left(\frac{\partial \tilde{\mathbf{v}}}{\partial \tilde{t}} + (\tilde{\mathbf{v}} \cdot \nabla) \tilde{\mathbf{v}} \right) = -\frac{\kappa^2}{D^3} \nabla \tilde{p} + \nu \cdot \frac{\kappa}{D^3} \nabla^2 \tilde{\mathbf{v}} + g\alpha \Delta T \hat{\mathbf{z}} \tilde{T}$$

Divide both sides by $\frac{\kappa^2}{D^3}$, then define:

$$\text{Pr} = \frac{\nu}{\kappa}, \quad \text{Ra} = \frac{g\alpha \Delta T D^3}{\nu \kappa}$$

Final nondimensional momentum equation:

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = -\nabla p + \text{Pr} \nabla^2 \mathbf{v} + \text{Ra} \text{Pr} T \hat{\mathbf{z}} \quad (\text{A.5})$$

Temperature:

$$\frac{\partial T}{\partial t} + (\mathbf{v} \cdot \nabla) T = \nabla^2 T \quad (\text{A.6})$$

A.4 Final Nondimensional Equations

Thus, the fully nondimensional system becomes:

$$\nabla \cdot \mathbf{v} = 0, \quad (\text{A.7})$$

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = -\nabla p + \text{Pr} \nabla^2 \mathbf{v} + \text{Ra} \text{Pr} T \hat{\mathbf{z}}, \quad (\text{A.8})$$

$$\frac{\partial T}{\partial t} + (\mathbf{v} \cdot \nabla) T = \nabla^2 T. \quad (\text{A.9})$$

A.5 Nondimensional Numbers

- **Rayleigh number** Ra measures the ratio of buoyancy to thermal and viscous damping[9]:

$$\text{Ra} = \frac{g\alpha\Delta TD^3}{\nu\kappa}$$

- **Prandtl number** Pr compares viscous diffusion to thermal diffusion:

$$\text{Pr} = \frac{\nu}{\kappa}$$

These two nondimensional numbers fully characterize the behavior of the system for given boundary conditions and geometry.

Appendix B

Thomas Algorithm (Tridiagonal matrix algorithm)

Appendix B: Thomas Algorithm for Tridiagonal Systems

Background

The Thomas algorithm is a straightforward and efficient method to solve systems of linear equations when the coefficient matrix is *tridiagonal*[18]. This means that the matrix has non-zero elements only on the main diagonal, the diagonal just above it, and the diagonal just below it; all other entries are zero. Such matrices commonly arise when discretizing differential equations using finite difference methods.

A tridiagonal system can be written as:

$$a_i x_{i-1} + b_i x_i + c_i x_{i+1} = d_i, \quad i = 1, 2, \dots, n,$$

where the coefficients a_i , b_i , c_i and the right-hand side d_i are known, and x_i are the unknowns we want to find. Here, it is important to note that $a_1 = 0$ because there is no x_0 , and $c_n = 0$ since there is no x_{n+1} .

Matrix form

This system corresponds to the linear system

$$\begin{bmatrix} b_1 & c_1 & 0 & \cdots & 0 \\ a_2 & b_2 & c_2 & \cdots & 0 \\ 0 & a_3 & b_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & c_{n-1} \\ 0 & \cdots & 0 & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_n \end{bmatrix}.$$

Because this matrix is tridiagonal, the Thomas algorithm can solve it in only $\mathcal{O}(n)$ operations, which is much faster than general Gaussian elimination.

Algorithm 2: Thomas Algorithm for Solving a Tridiagonal System

Input: Vectors a, b, c, d of length n , where

[noitemsep,topsep=0pt] $a = (a_1, a_2, \dots, a_n)$, with $a_1 = 0$

$b = (b_1, b_2, \dots, b_n)$ $c = (c_1, c_2, \dots, c_n)$, with $c_n = 0$ $d = (d_1, d_2, \dots, d_n)$

Output: Solution vector $x = (x_1, x_2, \dots, x_n)$ of the system $Ax = d$

```

1 Initialize modified coefficients:  $c'_1 \leftarrow \frac{c_1}{b_1}$ ,  $d'_1 \leftarrow \frac{d_1}{b_1}$ ;
2 for  $i \leftarrow 2$  to  $n$  do
3   Compute denominator:  $m_i \leftarrow b_i - a_i c'_{i-1}$ ;
4   Update coefficients:  $c'_i \leftarrow \frac{c_i}{m_i}$ ,  $d'_i \leftarrow \frac{d_i - a_i d'_{i-1}}{m_i}$ ;
5 Back substitution:;
6 Set  $x_n \leftarrow d'_n$ ;
7 for  $i \leftarrow n - 1$  to 1 do
8   Compute  $x_i \leftarrow d'_i - c'_i x_{i+1}$ ;
9 return  $x$ ;
```

Remarks

- The Thomas algorithm assumes that none of the denominators m_i become zero during the process, which is guaranteed if the coefficient matrix is diagonally dominant or positive definite.
- Compared to standard Gaussian elimination, which takes $\mathcal{O}(n^3)$ operations for a general matrix, the Thomas algorithm is very efficient and takes only $\mathcal{O}(n)$.
- The vectors a, b, c represent the sub-diagonal, main diagonal, and super-diagonal of the matrix, respectively.

Appendix C

Additional Figures

We present here the results for the linear code and stability analysis, when aspect ratio is 3 :

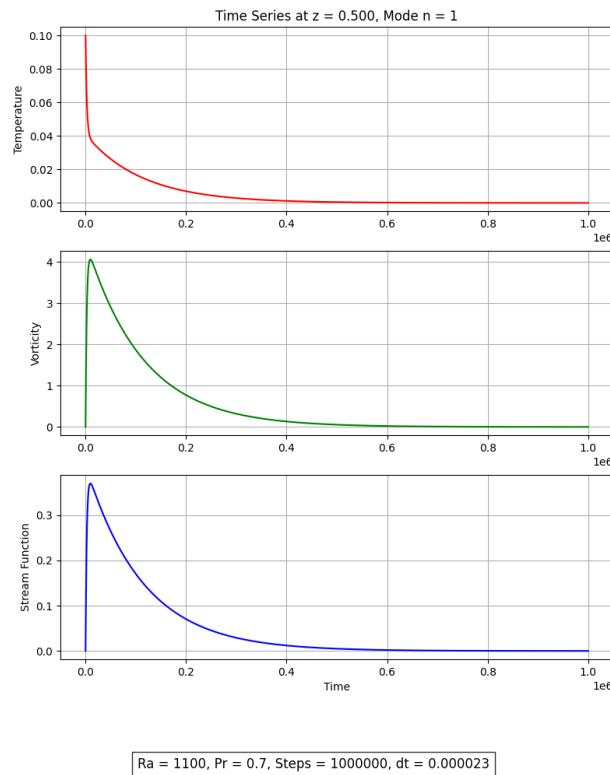


Figure C.1: $\text{Ra} = 1100$, subcritical solution

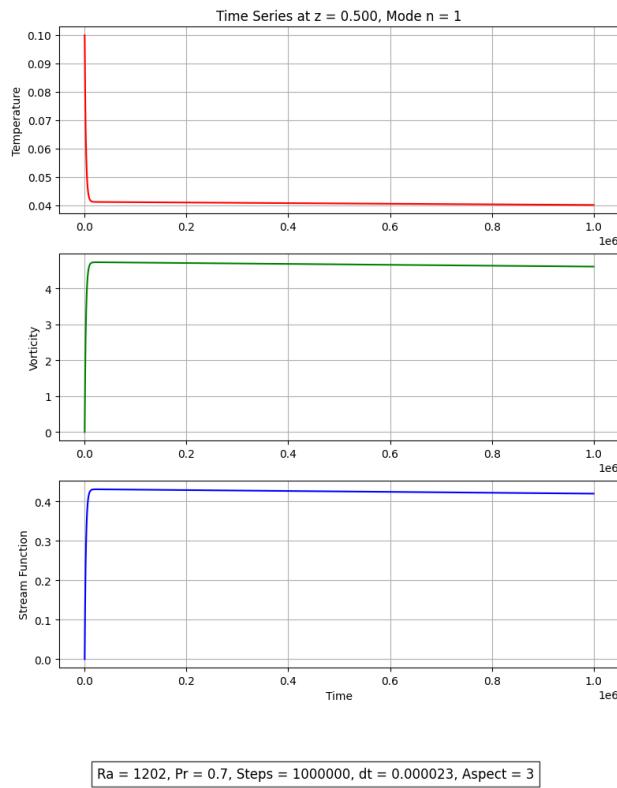


Figure C.2: $Ra = 1202$ (near-critical solution)

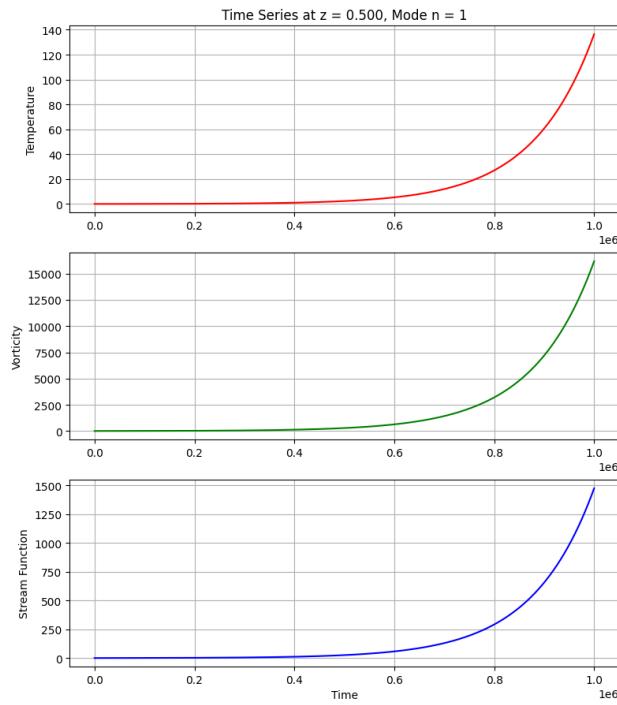


Figure C.3: $Ra = 1300$ (supercritical solution)

Bibliography

- [1] E. A. Spiegel. Convection in stars i. basic boussinesq convection. *Annual Review of Astronomy and Astrophysics*, 9:323–352, 1971.
- [2] C. J. Hansen, S. D. Kawaler, and V. Trimble. *Stellar Interiors: Physical Principles, Structure, and Evolution*. Springer, 2nd edition, 2004.
- [3] S. Chandrasekhar. *Hydrodynamic and Hydromagnetic Stability*. Oxford University Press, 1961.
- [4] A. V. Getling. *Rayleigh–Bénard Convection: Structures and Dynamics*. World Scientific, 1998.
- [5] R. Verzicco and R. Camussi. Numerical experiments on strongly turbulent thermal convection in a slender cylindrical cell. *Journal of Fluid Mechanics*, 477:19–49, 2003.
- [6] National Council of Educational Research and Training (NCERT). *Science - Class 7*. NCERT, 2020. Chapter 4: Heat.
- [7] Frank P. Incropera and David P. DeWitt. *Fundamentals of Heat and Mass Transfer*. John Wiley & Sons, 7th edition, 2011.
- [8] Henri Bénard. Les tourbillons cellulaires dans une nappe liquide transportant de la chaleur par convection en régime permanent. *Annales de Chimie et de Physique*, 23:62–144, 1901.
- [9] Gary A. Glatzmaier. *Introduction to Modeling Convection in Planets and Stars: Magnetic Field, Density Stratification, Rotation*. Princeton University Press, 2014.
- [10] L.D. Landau and E.M. Lifshitz. *Fluid Mechanics*, volume 6 of *Course of Theoretical Physics*. Pergamon Press, 1987.
- [11] Joseph Boussinesq. *Théorie analytique de la chaleur, Vol. 2*. Gauthier-Villars, Paris, 1903.

- [12] Pijush K. Kundu, Ira M. Cohen, and David R. Dowling. *Fluid Mechanics*. Academic Press, 6 edition, 2015.
- [13] Dale A. Anderson, John C. Tannehill, Richard H. Pletcher, Ramakanth Munipalli, and Vijaya Shankar. *Computational Fluid Mechanics and Heat Transfer*. CRC Press, 4 edition, 2020.
- [14] Andreas Tilgner and Friedrich H Busse. Convection in a box: Boundary conditions and pattern selection. *Physical Review E*, 47(6):3853–3860, 1993.
- [15] Erwin P Van Der Poel, Rodolfo Ostilla-Mónico, John Donners, and Roberto Verzicco. A pencil distributed finite difference code for strongly turbulent wall-bounded flows. *Computers & Fluids*, 116:10–16, 2015.
- [16] David Goluskin, Hans Johnston, Glenn R Flierl, and Edward A Spiegel. Convectively driven shear and decreased heat flux. *Journal of Fluid Mechanics*, 759:360–385, 2014.
- [17] Ambrish Pandey, Mahendra K Verma, Anando G Chatterjee, and Biplob Dutta. Similarities between 2d and 3d convection for large prandtl number. *Pramana*, 87:1–10, 2016.
- [18] William H Press, Saul A Teukolsky, William T Vetterling, and Brian P Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, Cambridge, UK, 3rd edition, 2007.

**Mostly Human 10%**

Almost all of the text is written by humans with some hints of AI assistance.

AI weightage	Content weightage	Sentences
H Highly AI written	3% Content	12
M Moderately AI written	5% Content	20
L Lowly AI written	2% Content	6



Adarsh Ramtel

Final

 Thesis

 2025

 Indian Institute of Technology, Roorkee



Document Details

Submission ID

trn:oid:::1:3254278010

55 Pages

Submission Date

May 19, 2025, 2:39 PM GMT+5:30

9,977 Words

Download Date

May 19, 2025, 3:23 PM GMT+5:30

51,484 Characters

File Name

ThesisMSc-Final.pdf

File Size

5.2 MB

16% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Ambreen Pandey
B.Tech. (Hons.)

Filtered from the Report

- ▶ Bibliography
- ▶ Quoted Text

Match Groups

- █ 119 Not Cited or Quoted 13%
Matches with neither in-text citation nor quotation marks
- █ 29 Missing Quotations 3%
Matches that are still very similar to source material
- █ 0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation
- █ 0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

- 12% █ Internet sources
- 11% █ Publications
- 4% █ Submitted works (Student Papers)

Adarsh Ramtel