Spectral Simulation Methods : Rayleigh-Benard Convection with Python

Final Report

Dwaipayan Roy Chowdhury

5360085

dwaipayan0502@gmail.com

March 30, 2023

Contents

1

Introduction

Pseudospectral methods are a powerful class of numerical techniques used to solve differential equations in various fields of science and engineering. These methods are based on approximating the solution to a differential equation using a weighted sum of basis functions, which can accurately capture the behavior of the solution.

One important advantage of pseudospectral methods is that they can achieve very high accuracy with relatively few basis functions. This is because the basis functions used in pseudospectral methods are often chosen to be orthogonal, which allows for efficient and accurate approximation of the solution. There are several different types of pseudospectral methods, including Galerkin methods, collocation methods, and spectral methods. In Galerkin methods, the differential equations are projected onto a finite-dimensional subspace of functions spanned by the chosen basis functions. This results in a set of algebraic equations that can be solved for the coefficients of the basis functions. Collocation methods, on the other hand, involve evaluating the differential equations at a finite number of collocation points, and then using these equations to solve for the coefficients of the basis functions. Spectral methods are a combination of Galerkin and collocation methods, where the basis functions are chosen to be orthogonal and satisfy certain boundary conditions.[1]

Pseudospectral methods have many applications in various fields, including fluid dynamics, acoustics, electromagnetic theory, and quantum mechanics. They have been used to solve a wide range of problems, from the simulation of turbulent flows to the calculation of the energy levels of quantum systems. In recent years, pseudospectral methods have become increasingly popular due to their high accuracy, spectral convergence properties, and computational efficiency.

In this report, we will explore the use of pseudospectral methods for solving thermal diffusion and convection problems, using both Galerkin and FFT methods. The Galerkin method involves projecting the differential equations onto a finite-dimensional subspace of functions, while the FFT method involves using the fast Fourier transform to compute the spectral coefficients of the solution. We will discuss the mathematical basis of these methods, including the spectral approximation of the solution, the numerical discretization of the differential equations, and the implementation of the methods in practical applications. We will also compare the performance of the Galerkin and FFT methods in terms of accuracy, efficiency, and computational cost.[1] [2]

Overall, this report aims to provide a comprehensive overview of pseudospectral methods for solving thermal diffusion and convection problems, and to demonstrate the effectiveness of these methods in various applications.

2

Spectral Concepts

Heat transport is an important physical process that occurs in many fields, including engineering, physics, and materials science. To accurately model and analyze heat transport, it is necessary to solve the heat transport equation, which describes the evolution of temperature in a system over time.

There are several numerical techniques that can be used to solve the heat transport equation, including finite difference, Fourier interpolation, and Fourier spectral methods. Finite difference methods involve discretizing the spatial domain into a grid of points and approximating the derivatives in the heat transport equation using finite differences. This results in a set of algebraic equations that can be solved for the temperature at each grid point

Fourier interpolation is a technique that involves approximating the temperature field using a sum of sinusoidal functions with unknown coefficients. These coefficients can be determined by matching the temperature values at a finite number of interpolation points. Once the coefficients are determined, the temperature field can be reconstructed using the Fourier series.

Fourier spectral methods are a class of numerical techniques that use the Fourier transform to solve the heat transport equation in the frequency domain. In Fourier spectral methods, the temperature field is represented as a sum of sinusoidal functions with unknown coefficients. The Fourier transform is then applied to the heat transport equation, resulting in a set of algebraic equations that can be solved for the coefficients of the sinusoidal functions. The temperature field can then be reconstructed in the spatial domain using the inverse Fourier transform.

In this report, we will explore these three numerical techniques in detail and compare their performance in solving the 1D heat transport equation. We will discuss the mathematical basis of each technique, including the discretization and approximation methods used. We will also demonstrate how to implement each technique in practical applications and compare their accuracy and computational efficiency.

2.1 Explicit Heat Diffusion (FDE)

The Fourier heat transport equation is mathematically identical to the diffusion equation discussed in class. A simple underlying picture would be the diffusion of heat packets or the diffusion of particles carrying a certain amount of heat. Heat is energy, but it is commonly measured in terms of temperature T. In order to convert between these two, we need a material property, the heat capacity c, then $q = \rho c_p T$ with density ρ. q is the energy density, i.e., energy per unit volume. Note that in general, $q(x, y, z, t)$, $T(x, y, z, t)$, $\rho(x, y, z)$, and $c_p(x, y, z)$ are fields that vary with position (x, y, z) and potentially time t.

Since energy is a conserved quantity, the continuity equation holds for q. (Remember that we derived the continuity equation in class for particle numbers, which are obviously conserved. Energy is similarly conserved in any physical system, and hence the same expression holds.)[2]

2.1.1 Derivation

In mathematical terms:

$$\frac{\partial q}{\partial t} = \rho c_p \frac{\partial T}{\partial t} = -\nabla \cdot \vec{j}_q + \dot{q}_S \tag{2.1}$$

where $\vec{j}_q$ is the heat current, and $\dot{q}_S$ are heat sources or sinks. (A lighted candle could, for example, be a source of heat. $\dot{q}_S$ describes the energy generated per unit volume and time.) Note that we have assumed that ρ and c_p do not vary with time t, so that we can pull it out of the derivative. [2]

The constitutive expression for the heat current goes back to Fourier and states that the current is proportional to the temperature gradient,

$$\vec{j}_q = -k\nabla T \tag{2.2}$$

where k is called the thermal conductivity. Putting these two equations together, we get:

$$\rho c_p \frac{\partial T}{\partial t} = k\nabla^2 T + \dot{q}_S \tag{2.3}$$

if we assume that k also does not depend on position. Note that we can rewrite this equation as:

$$\frac{\partial T}{\partial t} = \alpha \nabla^2 T + \dot{q}_S = \alpha \frac{\partial^2 T}{\partial x^2} + \dot{q}_S \tag{2.4}$$

with $\alpha = k/\rho c_p$. The equation hence only depends on a single parameter α that contains the density ρ, the heat capacity c_p, and the thermal conductivity k of the sample. In the steady-state (that is typically achieved for long times t), all derivatives with respect to t vanish, and we obtain:

$$\alpha \nabla^2 T + \dot{q}_S = 0 \tag{2.5}$$

We will in this exercise only discuss the 1D variant of these equations. The dynamic equation then becomes:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} + \dot{q}_S \tag{2.6}$$

This equation represents the one-dimensional heat transport equation, where T is the temperature distribution as a function of position x and time t, α is the thermal diffusivity of the material, and $\dot{q}_S$ is the heat source/sink term.[2]

There are several numerical techniques for solving this equation. One approach is to use finite difference methods, where the differential equation is discretized using a finite set of points in space and time. This method involves approximating the derivatives of T with respect to x and t using finite difference approximations, and then solving the resulting set of algebraic equations.

2.1.2 Finite-Difference scheme

To obtain a discrete representation of the derivative operator, we can use Taylor series expansion of some function $f(x)$. Let us evaluate the value of f at $x + \Delta x$. Then

$$f(x+\Delta x) = f(x) + \Delta x f'(x) + \frac{1}{2}\Delta x^2 f''(x) + \frac{1}{6}\Delta x^3 f'''(x) + O(\Delta x^4), \tag{2.7}$$

where primes indicate derivatives at the function value, $f'(x) = df/dx$, $f''(x) = d^2f/dx^2$, and so on. Similarly, we can evaluate the function value at $x - \Delta x$. This yields

$$f(x-\Delta x) = f(x) - \Delta x f'(x) + \frac{1}{2}\Delta x^2 f''(x) - \frac{1}{6}\Delta x^3 f'''(x) + O(\Delta x^4). \tag{2.8}$$

From a suitable linear combination of these two equations and termination at different order in Δx^4, we obtain discrete representations for the derivative operator.

2.1.3 Forward differences

The forward difference method is used to approximate the derivative of a function $f(x)$ at point x with a finite difference Δx. We evaluate the function at $x + \Delta x$ and x, and take the difference divided by Δx to obtain an approximation of the derivative at point x. This can be expressed mathematically as:

$$f'(x) = \frac{f(x+\Delta x) - f(x)}{\Delta x} + O(\Delta x). \tag{2.9}$$

The $O(\Delta x)$ term represents the truncation error and indicates that the approximation is accurate up to first order in Δx. The error decreases as we decrease the grid spacing Δx.

This first order forward difference method can be used to propagate the temperature field $T(x, t_0)$ in an explicit solution scheme for the heat transport equation. We can approximate the derivative of temperature with respect to time $\partial T/\partial t$ using the first order forward difference method and use it to update the temperature field at a later time $t + \Delta t$. This can be expressed as:

$$T(t+\Delta t) = T(t) + \frac{\partial T}{\partial t}\Delta t = T(t) + (\alpha \frac{\partial^2 T}{\partial x^2} + \dot{q}_S)\Delta t. \tag{2.10}$$

where α is the thermal diffusivity and $\dot{q}_S$ is the rate of heat generation per unit volume.

This type of explicit time stepping is called Euler integration. Although there are more accurate methods for solving initial value problems, such as Runge-Kutta integration, Euler integration is simpler and easier to implement. Note that a very small time step Δt is required to ensure accuracy.[2]

2.1.4 Central differences

The central differences approximation is a technique for approximating derivatives numerically using a discrete representation of a function. One way to obtain such a representation is by Taylor series expansion of the function. Let's consider a function $f(x)$ evaluated at two points $x + \Delta x$ and $x - \Delta x$. From the difference of these two evaluations, we can obtain an approximation for the first derivative of $f(x)$ at x.[2]

$$f(x+\Delta x) - f(x-\Delta x) = 2\Delta x f'(x) + O(\Delta x^3) \tag{2.11}$$

Dividing both sides by $2\Delta x$ and solving for $f'(x)$, we get:

$$f'(x) = \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} + O(\Delta x^2) \tag{2.12}$$

This is the second order central difference approximation to the derivative. The error term is of order Δx^2, which means that the approximation is more accurate than the first order forward difference approximation.

Similarly, we can obtain an approximation for the second derivative of $f(x)$ by summing the evaluations of $f(x + \Delta x)$ and $f(x - \Delta x)$, and solving for $f''(x)$.

$$f(x + \Delta x) + f(x - \Delta x) = 2f(x) + \Delta x^2 f''(x) + O(\Delta x^4) \tag{2.13}$$

Solving for $f''(x)$, we get:

$$f''(x) = \frac{f(x + \Delta x) - 2f(x) + f(x - \Delta x)}{\Delta x^2} + O(\Delta x^2) \tag{2.14}$$

This is a second order approximation for the second derivative, and we can use it to discretize the second derivative in the heat transport equation.[2]

2.2 Implicit Solution

The implicit solution is the steady-state solution where the temperature field no longer changes, i.e., $\frac{\partial T}{\partial t} = 0$. To obtain this solution, we first need to discretize the heat transport equation.[2]

The implicit equation that we need to solve is given by:

$$\alpha \frac{\partial^2 T}{\partial x^2} + \dot{q}_S = 0. \tag{2.15}$$

We can discretize this equation using the fourth-order finite differences approximation to the second derivative. Thus, we obtain:

$$\frac{T_{i-1} - 2T_i + T_{i+1}}{\Delta x^2} + \frac{\dot{q}_{S,i}}{\alpha} = 0, \tag{2.16}$$

where we have shifted the transport coefficient α to the source term $\dot{q}_{S,i}$.

Earlier, we discretized the temperature field in terms of equally spaced grid points i located at positions $i\Delta x$ for $i = 1, \ldots, N$, where N is the total number of grid points. This gives the discrete temperature field T_i. Note that this is actually a set of N equations for each i. For $N = 5$, the set of equations is [2]:

$$T_5 - 2T_1 + T_2 \Delta x^2 = -\frac{\dot{q_{S,1}}}{\alpha}, \tag{2.17}$$

$$T_1 - 2T_2 + T_3 \Delta x^2 = -\frac{\dot{q_{S,2}}}{\alpha}, \tag{2.18}$$

$$T_2 - 2T_3 + T_4 \Delta x^2 = -\frac{\dot{q_{S,3}}}{\alpha}, \tag{2.19}$$

$$T_3 - 2T_4 + T_5 \Delta x^2 = -\frac{\dot{q_{S,4}}}{\alpha}, \tag{2.20}$$

$$T_4 - 2T_5 + T_1 \Delta x^2 = -\frac{\dot{q_{S,5}}}{\alpha}. \tag{2.21}$$

If we interpret the discrete field T_i as a vector $\vec{T}$, we can rewrite this set of equations as:

$$\mathbf{A}\vec{T} = -\Delta x^2 \frac{\vec{\dot{q}}_S}{\alpha}, \tag{2.22}$$

where $\mathbf{A}$ is the system matrix given by:

$$\mathbf{A} = \begin{pmatrix} -2 & 1 & 0 & 0 & 1 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \\ 1 & 0 & 0 & 1 & -2 \end{pmatrix}. \tag{2.23}$$

And, thus we solve the linear matrix operation $\vec{T} = A^{-1}B$

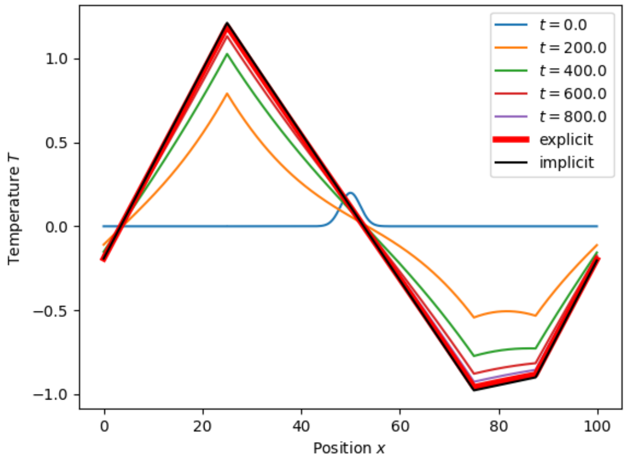


Figure 2.1: This figure represents the explicit and implicit heat diffusion using Finite difference [2]

2.3 Fourier Interpolation

The Discrete Fourier Transform (DFT) is a mathematical tool used to convert a finite sequence of equally-spaced samples of a function into a sequence of complex numbers, which can be analyzed in the frequency domain. It is widely used in signal processing, image processing, audio analysis, and many other fields. The DFT is based on the Fourier series, which states that any periodic function can be expressed as a sum of sinusoids with different frequencies.

The fourier series is written by,

$$f(x) = \frac{1}{L} \sum_{n=-\infty}^{\infty} (c_k \Delta x) e^{iq_k x} \tag{2.24}$$

where L is the total length of the sequence, and k is an integer index ranging from $-\infty$ to ∞. The coefficients c_k in the Fourier series expansion of f(x) are related to the DFT coefficients by

$$c_k \Delta x = \int_0^L dx f(x) e^{-iq_k x}$$

Given a sequence of N equally-spaced samples of a function $f(x)$ with spacing Δx, the DFT of the sequence is given by:

$$f(x_l) = \frac{1}{N} \sum_{n=0}^{N-1} c_k e^{iq_k x_l} \tag{2.25}$$

where $x_l = l\Delta x$ is discretized with Δx space and $q_k = k\Delta q$ and the Fourier series is defined on $\Delta q = \frac{2\pi}{L}$ and $L = \Delta x N$. Thus, the coefficient can be written as,

$$c_k = \sum_{n=0}^{N-1} f(x_l) e^{-iq_k x_l}$$

And we substitute some variables like $q_k x_l = kl\Delta x\Delta q = \frac{2\pi kl}{N}$ and the formulation becomes,

$$f(x_l) = \frac{1}{N} \sum_{n=0}^{N-1} c_k e^{\frac{i2\pi kl}{N}}$$

and,

$$c_k = \sum_{n=0}^{N-1} f(x_l) e^{\frac{i2\pi kl}{N}}$$

We use the DFT concept to perform Fourier interpolation. Fourier interpolation is a method used to estimate the value of a function at a point where it is not explicitly defined, but can be estimated using the known values of the function at other points. This is done by first taking the Discrete Fourier Transform (DFT) of the known data, and then using the DFT to estimate the values of the function at intermediate frequencies. This can be achieved through a process called zero-padding.

The process of zero-padding involves extending the length of the original data sequence by appending zeros to the end of the sequence. This effectively increases the resolution of the DFT, allowing us to estimate the values of the function at intermediate frequencies. The amount of zero-padding required depends on the level of interpolation required. Generally, the amount of zero-padding required is proportional to the ratio of the desired output resolution to the input resolution. [2]

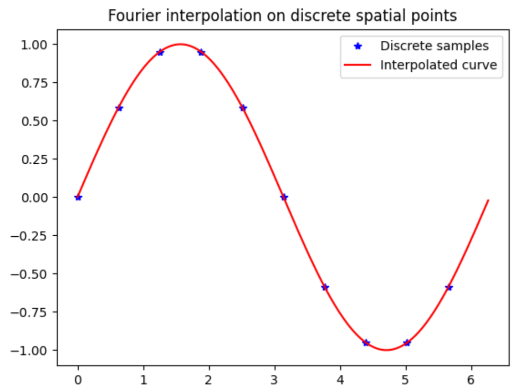


Figure 2.2: Interpolation of sine curve using Fourier method

2.3.1 Steps

1. Compute the discrete Fourier transform (DFT) of the input signal using the Fast Fourier Transform (FFT) algorithm:
   $\texttt{fft\_transform} = \texttt{np.fft.fft}(\texttt{func}(\texttt{np.arange}(N\_samples)/N\_samples))$

   where `func` is the input function, `N_samples` is the number of samples of the input function, and `np.arange` generates an array of evenly spaced numbers from 0 to `N_samples-1`.

2. Find the midpoint index of the input signal:

   $\texttt{mid\_index} = N\_samples//2$

   where "//" is the integer division operator.

3. Create a zero-padded DFT array of length `N_pad`:

   $\texttt{fft\_pad} = \texttt{np.zeros}(N\_pad, \texttt{dtype} = \texttt{complex})$

4. Copy the positive frequency components of the DFT up to the midpoint index into the first half of the zero-padded DFT:

   $\texttt{fft\_pad}[0 : \texttt{mid\_index}] = \texttt{fft\_transform}[0 : \texttt{mid\_index}]$

5. Copy the negative frequency components of the DFT from the midpoint index to the end into the second half of the zero-padded DFT:

   $\texttt{fft\_pad}[N\_pad - \texttt{mid\_index} :] = \texttt{fft\_transform}[N\_samples - \texttt{mid\_index} :]$

6. Compute the inverse DFT of the zero-padded DFT using the IFFT algorithm and scale by the ratio of the padded and original number of samples:

   $\texttt{fft\_invs} = \texttt{np.fft.ifft}(\texttt{fft\_pad}) * N\_pad/N\_samples$

   The resulting array `fft_invs` contains the interpolated values of the input function.

The algorithm performs Fourier interpolation by increasing the number of samples in the frequency domain (i.e. the DFT) through zero-padding. The input signal is first transformed into the frequency domain using the FFT algorithm, and then zero-padded to increase the number of frequency components. The positive and negative frequency components are then rearranged into the first and second half of the zero-padded DFT, respectively, and the inverse DFT is computed to obtain the interpolated values of the input signal. Finally, the interpolated values are scaled by the ratio of the padded and original number of samples to obtain the correct amplitude values. [3][2]

2.4 Fourier Derivative

One of the biggest advantage of using spectral method is to use simple fourier derivatives with higher accuracy and the formulation of higher order derivative is just taking the derivatives on exponential terms.

$$f'(x) = \frac{1}{N} \sum_{n=0}^{N-1} iq_k c_k e^{iq_k x_l}$$

Thus, the Fourier transform of the derivative of the interpolated function is just a multiplication with a factor $iq_k c_k$.

2.5 FD in Fourier representation

We here try to formulate an equation that uses finite difference as the first order derivative formulation, but the next step is calculated using the Fourier method described in the last section. We write the forward difference equation as,

$$f'(x_l) = \frac{1}{N} \sum_{n=0}^{N-1} \frac{e^{iq_k x_{l+1}} - e^{iq_k x_l}}{\Delta x} c_k = \frac{1}{N} \sum_{n=0}^{N-1} \frac{e^{iq_k \Delta x} - 1}{\Delta x} c_k e^{iq_k x_l}$$

This formula uses the Fourier representation within the FD and for the fourier derivative, we can write $D_k = iq_k$ and the FD scheme becomes,

$$D_k = \frac{e^{iq_k \Delta x} - 1}{\Delta x}$$

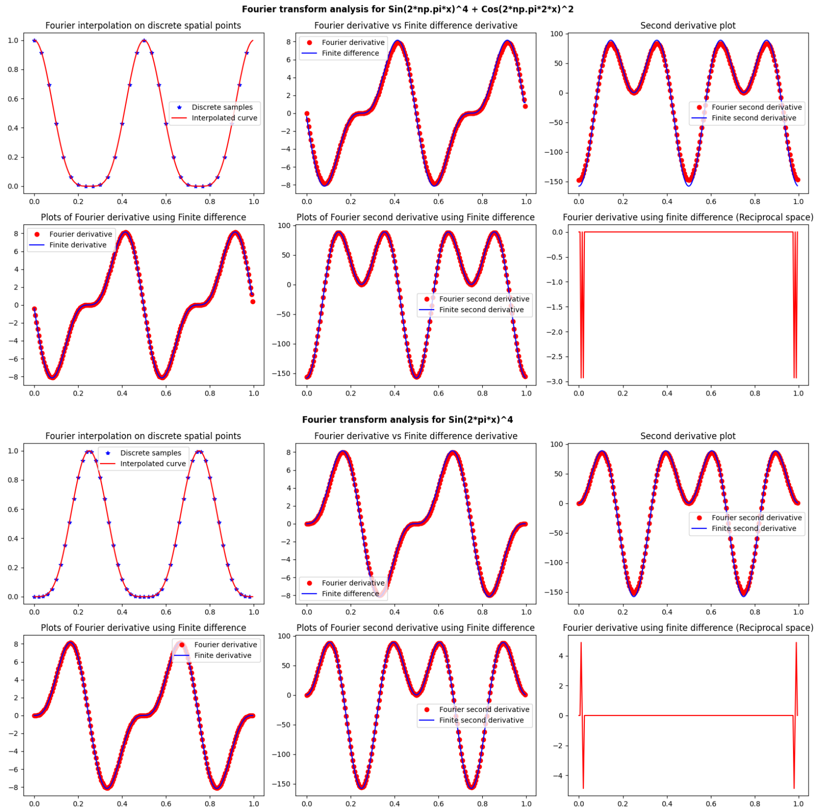


Figure 2.3: In both the plots, the first subplot refers to the interpolation of the two functions. The second subplot represents the derivatives of the equations using finite difference and fourier derivative. The third plot calculates the second derivative using FD and spectral method. Next, we use the fourier derivative in finite difference equation to compare with the finite difference. The fifth plot refers to the same for second derivative. The last plot refers to the same in reciprocal space. The functions used are $f(x) = sin(2\pi x)^4$ and $f(x) = sin(2\pi x^2)^4 + cos(2\pi x^2)^2$

2.6 Heat diffusion using Spectral

Here, we discuss about the solution of explicit and implicit steady state heat diffusion using Galerkin and spectral method. Let us begin with the heat transport equation for steady state given by,

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} + \dot{q}_S$$
$$R \equiv \alpha \frac{\partial^2 T}{\partial x^2} + \dot{q}_S \equiv 0$$

Now, we introduce the concept of Galerkin formulation to solve the PDE assuming the solution to a sum of weighted residuals. Also, the targeted temperature here can be approximated as,

$$T_N(x) = \sum_{k=0}^{N-1} \tilde{T}_k v_k(x)$$

where $v_k(x)$ is the set of basis functions.

2.6.1 Galerkin Method

Here, we try to solve the Diffusion Equation using Galerkin Method. To solve this equation, we will use the Galerkin method, which involves assuming that the solution can be represented as a linear combination of basis functions. We will choose our basis functions as $\psi_j(x) = x^i(x-L)$, where L is the length of the domain and i is an integer.[4]

We will then use the residual of the equation, $R(x; c_1, c_2, .., c_n) = u'' + f(x)$, and impose the condition that it is orthogonal to all the basis functions. This leads to a system of linear equations which can be solved to find the coefficients of the trial solution.

We will implement this method by first taking the second derivative of the trial solution and substituting it and the basis functions into the residual equation. Then, we will multiply both sides of the equation by each basis function, take their inner product, and use the orthogonality condition to obtain a system of linear equations. Finally, we will solve the system of linear equations to find the coefficients of the trial solution.[5]

$$
\begin{aligned}
\textbf{Given:} \quad & u'' + f(x) = 0 \\
\text{Let's assume the solution can be represented as} \quad & u_h = \sum_{j=1}^{N} c_j \psi_j(x) \\
\text{where} \quad & \psi_j(x) = x^i(x-L) \text{ is a basis function} \\
\text{Taking the second derivative of} \quad & u_h \\
u_h'' &= \sum_{j=1}^{N} c_j \frac{d^2}{dx^2}\psi_j(x) \\
&= \sum_{j=1}^{N} c_j \frac{d^2}{dx^2} x^i(x-L) \\
&= \sum_{j=1}^{N} c_j \frac{i(i-1)}{(x-L)^2} x^i(x-L) \\
\text{Substituting} \quad & u_h'' \text{ and } \psi_j(x) \text{ into the residual equation} \\
R(x; c_1, c_2, ..., c_N) &= u'' + f(x) \\
&= \sum_{j=1}^{N} c_j \frac{i(i-1)}{(x-L)^2} x^i(x-L) + f(x)
\end{aligned}
$$

$$
\begin{aligned}
\text{Multiplying both sides by} \quad & \psi_k(x) \\
(R(x; c_1, c_2, ..., c_N), \psi_k(x)) &= \left(\sum_{j=1}^{N} c_j \frac{i(i-1)}{(x-L)^2} x^i(x-L) + f(x), x^k(x-L) \right) \\
\text{We get a system of linear equations} \quad & \mathbf{Ac} = \mathbf{b} \text{ where} \\
A_{ij} = \left(\psi_i(x), \frac{d^2}{dx^2}\psi_j(x) \right) &= \frac{j(j-1)}{L^2} \int_0^L x^{i+j-2}(x-L)dx \\
b_i = -\left(f(x), \psi_i(x)\right) &= -\int_0^L f(x) x^i (x-L) dx \\
\text{Finally, we solve the linear system to get the coefficients} \quad & \mathbf{c} = \mathbf{A}^{-1}\mathbf{b}
\end{aligned}
$$

Algorithm 1 Solving the Diffusion Equation

Length of the domain L, number of nodes N, number of divisions $divs$, and diffusion coefficient α Temperature distribution U **Initialize:** $x \leftarrow$ evenly spaced points from 0 to L with $divs$ divisions, $h \leftarrow$ grid spacing, $A \leftarrow$ zero matrix of size $N \times N$, $B \leftarrow$ zero vector of size $N \times 1$; **Define:**

- $f(x) \leftarrow$ source term at position x
- $\psi(x, i_0) \leftarrow$ spectral basis function of order i_0 at position x
- $\mathrm{psidd}(x, i_0) \leftarrow$ second derivative of $\psi(x, i_0)$ at position x
- $\mathrm{As}(x, i_0, j_0) \leftarrow$ inner product of $\psi(x, i_0)$ and $\mathrm{psidd}(x, j_0)$
- $\mathrm{bs}(x, i_0) \leftarrow$ inner product of $\psi(x, i_0)$ and $-f(x)$

Calculate:

- Assemble matrix A and vector B using $\mathrm{As}(x, i_0, j_0)$ and $\mathrm{bs}(x, i_0)$
- Solve for C using A and B
- Calculate the spectral basis functions $\mathrm{ps}(x, i_2)$ and their second derivatives $\mathrm{psdd}(x, i_2)$
- Calculate the temperature distribution U using $\mathrm{ps}(x, i_2)$ and C

Solving Matrix

- Initialize *sources* vector with ones, assemble matrix A and vector b
- Solve for T_{FD} using A and b

Calculate:

- Analytical solution $T_{analytical}$ using *sources* and x
- Residual and error using $\mathrm{psdd}(x, i_2)$, C, $f(x)$, and $T_{analytical}$

Plot: $T_{analytical}$, T_{FD}, and U in one plot, and the residual and error in another plot;

Here, we write an algorithm to solve the Galerkin method in the context of diffusion equation. The code solves the 1D diffusion equation using three different methods: Galerkin spectral method, finite difference method, and analytical solution. The Galerkin spectral method uses a finite set of basis functions to approximate the solution, and the finite difference method discretizes the domain and uses a numerical scheme to approximate the solution. The analytical solution provides a reference solution for comparison. The code first sets up the problem parameters and defines the required functions for the Galerkin spectral method. It then constructs the matrices needed to solve the problem and solves it using the spsolve function. Next, it calculates the solution using the finite difference method by setting up a matrix and solving the resulting linear system. Finally, it calculates the analytical solution and plots the results for comparison. The residual and error are also calculated and plotted to evaluate the accuracy of the numerical solutions.

Now, we write a code that plots the explicit and implicit heat transfer using spectral method and galerkin formulation. The code first defines the necessary parameters for the problem, such as the length of the domain L, the number of nodes N, and the number of divisions divs in the domain. It then defines the function f(x) to represent the sources of the PDE.

The Galerkin spectral method solves the PDE by approximating the solution as a linear combination of known functions. In this code, the known functions are defined by the functions psi(x,i0) and psidd(x,i0). These functions are used to compute the matrix A and vector B by evaluating integrals over the domain. The A matrix and B vector are then used to solve the system of linear equations Ax=B using the spsolve function from the scipy.sparse.linalg module.

The code also computes the solution of the same PDE using the finite difference method and compares it with the Galerkin spectral method solution and the analytical solution. The finite difference solution is computed by solving the discretized form of the PDE using a sparse matrix and the spsolve function. The analytical solution is computed using the known analytical solution of the PDE. The code also computes and plots the residual and error between the numerical and analytical solutions. In the second plot, we initialized the

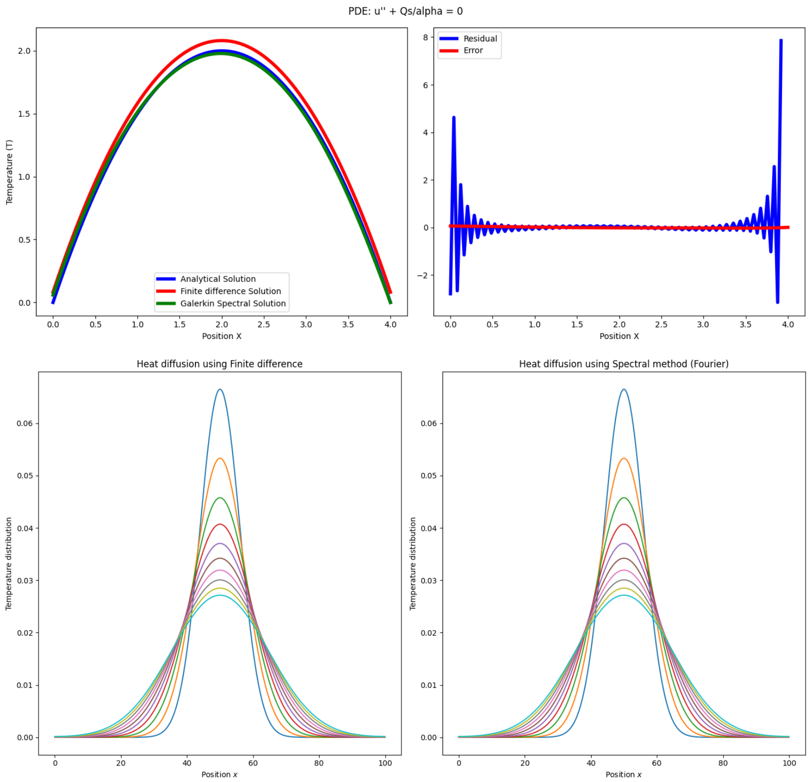


Figure 2.4: The first two figures plot the steady state Heat diffusion equation using galerkin method, FD and analytically. Also the residual with the error is plotted. The second plot explains the explicit flow of temperature with time using Finite difference and spectral.

temperature with a Gaussian distribution and we observe how with time the temperature flows. We use both Finite difference and Spectral method to verify if both the model works properly.[2]

Now, we plot a 3D heat diffusion equation using the galerkin method. Here we implement a numerical scheme which solves the coefficients of $\tilde{T}$ with a given source $\dot{q}_s$. We use an ansatz $\tilde{T} = T_n sin(n\pi x)sin(n\pi y)sin(n\pi z)$ and we take the eigen values as the derivatives using fourier method to set it as A. The b is created using the source terms multiplied with the ansatz to match the orthogonality. The ansatz is chosen to satisfy the boundary conditions of $T(x, y, z) = 0$ at $z = (0, 1), x = (0, 1), y = (0, 1)$.

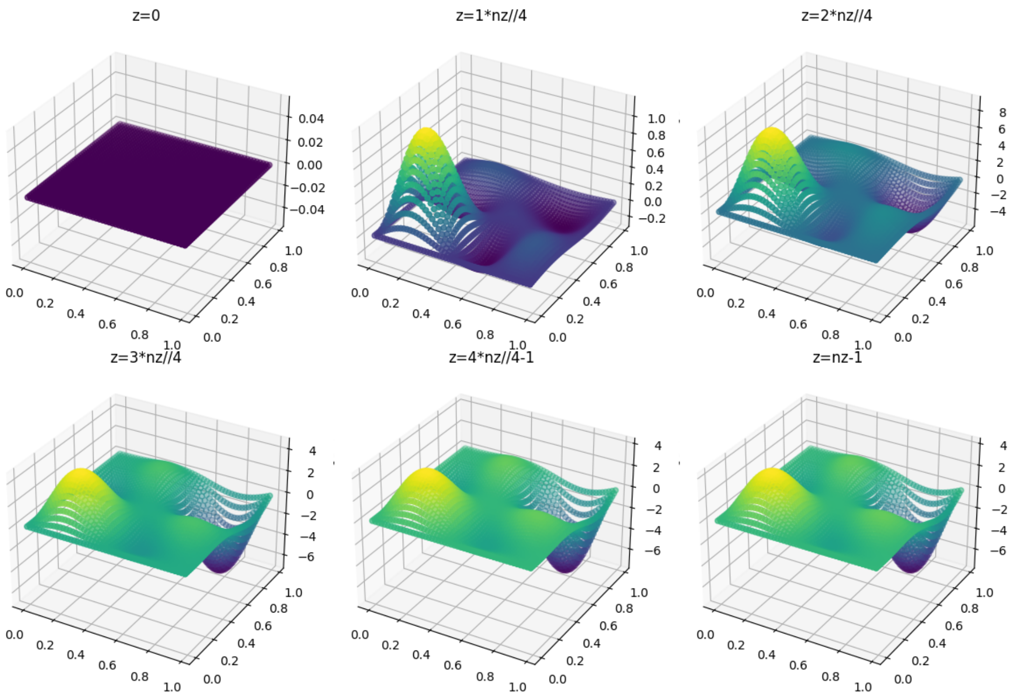


Figure 2.5: The figure represents the steady state of a 3D heat diffusion flow plotted along XY plane at $z = 0$,$z = \frac{Nz}{4}$,$z = \frac{2N_z}{4}$,$z = \frac{3N_z}{4}$,$z = \frac{4N_z}{4}$,$z = N_z - 1$ and $\dot{q}_s = 0.5$ at $\frac{N_x}{4}$, $\frac{N_y}{4}$, $\frac{N_z}{4}$ and $\dot{q}_s = -0.3$ at $\frac{3N_x}{4}$, $\frac{3N_y}{4}$, $\frac{3N_z}{4}$

Here, the code solves a 3D heat equation with variable sources and sinks. The solution is obtained by solving a linear system of equations using finite Fourier transforms and finite differences. First, we define the basis functions. Next, the spatial grid and the time increment are defined. Heat sources and sinks are also defined. The code then calculates the basis functions using a 3D sinusoidal function. The basis functions are then used to define a matrix of coefficients which are used to solve the heat equation using finite Fourier transforms. A linear system of equations is then defined and solved using finite Fourier transforms and finite differences. [3][1][2]

3

Rayleigh Bénard Convection

Thermal convection is a process that occurs when a fluid is heated from below and cooled from above, resulting in the formation of convective cells. The density difference between the hot and cold regions of the fluid causes buoyancy forces that drive fluid motion. One of the most well-known examples of thermal convection is the Rayleigh Benard convection, which has been extensively studied in the field of fluid mechanics.

The Rayleigh Benard convection is named after Lord Rayleigh and Henri Benard, who independently investigated this phenomenon in the late 19th and early 20th centuries. Rayleigh conducted experiments on a thin layer of fluid heated from below, while Benard studied the convection in a cylindrical container filled with fluid. Both researchers observed the formation of cellular flow patterns, which they attributed to the instability of the fluid interface between the hot and cold regions.

Rayleigh Benard convection is not only a fascinating phenomenon but also plays a crucial role in many natural and industrial processes. For instance, it occurs in the Earth's mantle and drives plate tectonics, which is responsible for the formation of mountains and volcanic activity. It also plays a crucial role in atmospheric circulation, ocean currents, and heat transfer in industrial processes such as cooling towers and boilers.

The study of Rayleigh Benard thermal convection has therefore attracted significant attention from physicists, mathematicians, and engineers over the past century. In recent years, advances in experimental techniques and computational methods have allowed researchers to gain a deeper understanding of the dynamics and flow patterns of Rayleigh Benard convection. These studies have revealed new insights into the mechanisms of heat transfer, mixing, and turbulence in fluid flows, with important implications for a wide range of applications.

In this report, we will discuss how we solve the convective instability and how to apply linear stability analysis, how we have identified critical Rayleigh number on the onset of convection, visualize static and time varying dynamic velocity and thermal profiles as well as the effects of parameters such as the fluid properties, geometry, and boundary conditions. [3]

3.1 Coupled Differential Equation

Let us derive the onset of the instability of Rayleigh Bénard convection. For the derivation purpose, we use the non-dimensionalization approach and the linear stability theory. We go step by step to solve the stability analysis.

3.1.1 Setting up the problem

We here demonstrate the a Rayleigh–Bénard convection setup. In the figure below, we have fluid layer bounded by two plates on the top and bottom. The bottom plate is uniformly heated up and thus $T_0 > T_h$. Due to difference in temperature, the density is no more same and a variation in density leads to the flow of movement of fluid particles in a pattern. This is the fundamental idea of Rayleigh-Bénard convection. We are under the assumption here the Navier stokes equation still stands valid even though NSE is derived on the principle of in-compressible fluid. The density deviation creates a buoyancy and the movement occurs. the fluid motion here is balanced by the viscosity of the fluid and the ration between these two forces explain the stability. Also, a small heating up might not be enough to observe the fluid motion as the buoyancy force may not be predominant then. To observe something interesting, we have to generate enough heating so that the viscosity becomes less effective than the buoyancy.[6]

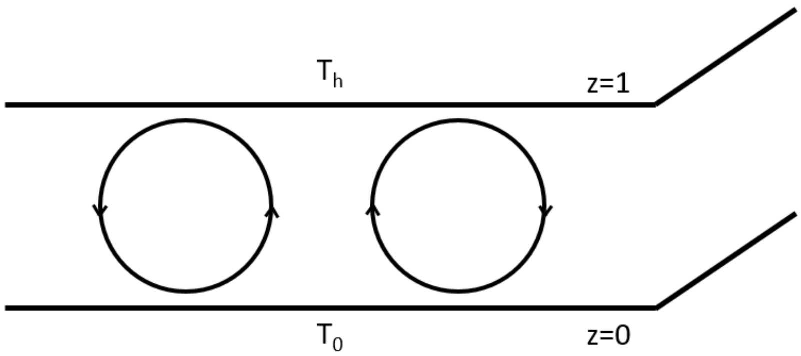


Figure 3.1: Experimental setup for the derivation

We are considering 2D case for our derivation (x, z) plane and we assume all operations in x and y plane are similar and that x and z plane has something interesting to observe.

3.1.2 Governing equations

In the previous subsection, we already explained that density is no more constant and thuis we cannot directly incorporate the NSE equations into our derivation. Recalling the equation of state, where density of fluid particles (ρ) depends on Temperature (T) and pressure (p) can be written with Boussinesq approximation as, [6]

$$\boxed{\rho = \rho_0(1 + \alpha(T - T_0))} \tag{3.1}$$

here, $\alpha < 0$ is the volume coefficient of thermal expansion. When we heat up a plate, the adjacent area becomes less dense. This Boussinesq approximation is important for the buoyancy term.

Now, the mass and the momentum conservation,

$$\boxed{\partial_t \rho + \nabla \cdot (\rho u) \Longrightarrow \nabla \cdot u = 0} \tag{3.2}$$

$$\boxed{\rho_0 \frac{Du}{Dt} = -\nabla p + \mu \nabla^2 u - \rho g \hat{(z)} \Longrightarrow \rho_0 \left(\frac{\partial u}{\partial t} + (u \cdot \nabla) u \right) = -\nabla p + \mu \nabla^2 u - \rho_0 (1 + \alpha(T - T_0)) g \hat{z}} \tag{3.3}$$

Now, we need one for equation as we have 4 variables to solve and we use advection-diffusion equation as Temperature T is a scalar field

$$\boxed{\partial_t T + (u \cdot \nabla)T = \kappa \nabla^2 T} \tag{3.4}$$

Thus, the system of equations that we try to solve are

$$\boxed{\begin{aligned}
\text{Momentum:} \quad & \rho_0 \left(\frac{\partial u}{\partial t} + (u \cdot \nabla)u \right) = -\nabla p + \mu \nabla^2 u - \rho_0(1 + \alpha(T - T_0))g\hat{z} \\
\text{Mass:} \quad & \nabla \cdot u = 0 \\
\text{Energy:} \quad & \frac{\partial T}{\partial t} + (u \cdot \nabla)T = \kappa \nabla^2 T \\
\text{Equation of state:} \quad & \rho = \rho_0(1 + \alpha(T - T_0))
\end{aligned}}$$

3.1.3 Non-dimensionalize the equations

We can solve the system of governing equations but the outcome will heavily be influenced by computational and physical units. To make a generic derivation, we start by deducing some dimensionless parameters to work with. Since, the distnce between two plates is 1, the length scale is automatically 1. We can use the following scales to normalize the physical parameters,[6]

$$\boxed{\begin{aligned}
\text{Length scale:} \quad & 1 \\
\text{Time scale:} \quad & \tau = \frac{1}{\kappa} \\
\text{Temperature scale:} \quad & \Delta T = T - T_0 \\
\text{Velocity scale:} \quad & U = \frac{1}{\tau} = \frac{\kappa}{1} \\
\text{Pressure scale:} \quad & P = \rho_0 U^2 \frac{\nu}{\kappa}
\end{aligned}}$$

To verify that our choice matches the units, we can write the variables like the following

$$[u] = \left[\frac{L}{T}\right], \quad [\rho_0] = \left[\frac{M}{L^3}\right], \quad [p] = \left[\frac{M}{LT^2}\right], \; [\nu] \quad = [\kappa] = \left[\frac{L^2}{T}\right], \quad [\nabla] = \left[\frac{1}{L}\right], \quad [\partial_t] = [u \cdot \nabla] = \left[\frac{1}{T}\right]$$

Decomposing the variables and operators into the dimensionless parameters,

$$u = Uu = \frac{\kappa}{h}u, \quad p = Pp = \frac{\rho_0}{\kappa^2}\frac{\nu}{h^2}p, \; T \quad = \Delta T T + T_0, \quad \nabla = \frac{1}{h}\nabla, \quad \partial_t = u \cdot \nabla = \frac{\kappa}{h^2}\partial_t$$

Simplifying the momentum equation using the following parameters, we refer to the solution [6] and can write the following as,

$$\boxed{\frac{1}{\mathrm{Pr}} \left(\frac{\partial u}{\partial t} + (u \cdot \nabla)u \right) = -\nabla p + \nabla^2 u - RT\hat{z}}$$

Here, we have two conventional dimensionless number,

$$\text{Prandtl number: } \mathrm{Pr} = \frac{\nu}{\kappa} = \frac{\text{momentum diffusivity}}{\text{thermal diffusivity}},$$

$$\text{Rayleigh number: } \mathrm{R} = \frac{\alpha \Delta T g h^3}{\nu\kappa} = \frac{\text{convection}}{\text{conduction}} = \frac{\alpha \Delta T g}{\nu\kappa/h^3} = \frac{\rho g h^3}{\nu U} = \frac{\text{buoyant force}}{\text{viscous force}}.$$

As discussed in the experimental setup subsection, the convection is driven by the buoyancy force and the ration of convection decides the buoyancy to viscous force, it can be observed here that it is nothing but the Rayleigh number.

In Rayleigh-Bénard convection, Rayleigh number is the most important dimensionless number. Thus, the modified equations of momentum, mass and energy ca be written as [6],

$$\boxed{\begin{aligned} \text{Momentum: } & \frac{1}{\mathrm{Pr}}\left(\frac{\partial u}{\partial t} + (u\cdot\nabla)u\right) = -\nabla p + \nabla^2 u - \mathrm{Ra}T\hat{z}, \\ \text{Mass: } & \nabla\cdot u = 0, \\ \text{Energy: } & \frac{\partial T}{\partial t} + (u\cdot\nabla)T = \nabla^2 T. \end{aligned}}$$

3.1.4 Steady State solutions

We first understand our boundary conditions,

$$\begin{aligned} \text{at } z = 0 : & u = 0 \text{ and } T = 0 \\ \text{at } z = 1 : & u = 0 \text{ and } T = 1 \end{aligned}$$

We now find the momentum along x and z direction,

$$\begin{cases} \text{momentum } \hat{x}: \ \frac{1}{Pr}\left(\frac{\partial u_S}{\partial t} + (u_S\cdot\nabla)u_S\right) = -\frac{\partial p_S}{\partial x} + \nabla^2 u_S \\ \text{momentum } \hat{z}: \ \frac{1}{Pr}\left(\frac{\partial v_S}{\partial t} + (u_S\cdot\nabla)v_S\right) = -\frac{\partial p_S}{\partial y} + \nabla^2 v_S - RT_S\hat{g}_z \end{cases}$$

$$\Rightarrow \begin{cases} \text{momentum } \hat{x}: \ \frac{\partial p_S}{\partial x} = 0 \\ \text{momentum } \hat{y}: \ \frac{\partial p_S}{\partial y} = -RT_S g \end{cases} \quad \Rightarrow \frac{\partial p_S}{\partial y}(y) = -RT_S(z)g$$

And the energy equation can be simplified as

$$\begin{aligned} \frac{\partial T_S}{\partial t} + (u_S\cdot\nabla)T_S = \kappa\left(\frac{\partial^2 T_S}{\partial x^2} + \frac{\partial^2 T_S}{\partial y^2}\right) \\ \Rightarrow \frac{\partial^2 T_S}{\partial z^2} = 0 \Rightarrow T_S(z) = z \end{aligned}$$

The final linearized governing equations would be[6]

$$\boxed{\begin{aligned} \text{Momentum } \hat{x}: \quad & \frac{1}{\mathrm{Pr}}\frac{\partial \tilde{u}}{\partial t} = -\frac{\partial \tilde{p}}{\partial x} + \nabla^2\tilde{u} \\ \text{Momentum } \hat{z}: \quad & \frac{1}{\mathrm{Pr}}\frac{\partial \tilde{w}}{\partial t} = -\frac{\partial \tilde{p}}{\partial z} + \nabla^2\tilde{v} - \mathrm{R}\tilde{T}g \\ \text{Energy: } \quad & \frac{\partial \tilde{T}}{\partial t} + \tilde{w}\frac{\partial \tilde{T}}{\partial y} = \kappa\left(\frac{\partial^2\tilde{T}}{\partial x^2} + \frac{\partial^2\tilde{T}}{\partial y^2}\right) \\ \text{Mass: } \quad & \frac{\partial \tilde{w}}{\partial x} + \frac{\partial \tilde{w}}{\partial y} = 0 \end{aligned}}$$

3.1.5 Simplification to ODE

To convert the following 4 PDEs to 1 ODE, let us first try to eliminate pressure variable by,

$$\partial_z[\frac{1}{\mathrm{Pr}}\frac{\partial \tilde{u}}{\partial t}] - \partial_x[\frac{1}{\mathrm{Pr}}\frac{\partial \tilde{w}}{\partial t}]$$

This gives us,

$$\frac{1}{Pr}\frac{\partial}{\partial t}\left(\frac{\partial_z\tilde{u}}{\partial x} - \frac{\partial_x\tilde{w}}{\partial y}\right) = \frac{\partial^2}{\partial x^2}\left(\frac{\partial_z\tilde{u}}{\partial x} - \frac{\partial_x\tilde{w}}{\partial z}\right) + R\frac{\partial\tilde{T}}{\partial x} \tag{3.5}$$

Now, to eliminate $\tilde{u}$, we perform $\partial_x[3.5]$ and the mass equation and get,

$$-\frac{1}{Pr}\frac{\partial}{\partial t}\nabla^2\tilde{w} = -\nabla^4\tilde{w} + Ra\frac{\partial^2}{\partial x^2}\tilde{T} \tag{3.6}$$

Now, We try implementing the Galerkin formulation and we have two unknowns here $\tilde{w}$ and $\tilde{T}$. We choose our ansatz along x and t to meet the boundary and the ansatz follows,[6]

$$\begin{aligned}\tilde{w} &= w^* e^{\lambda t} sin(kx)\\ \tilde{T} &= T^* e^{\lambda t} sin(kx)\end{aligned} \tag{3.7}$$

where $k = (k_x, k_y)$ and we now derive the critical Rayleigh number $R(k^2)$. The critical rayleigh number is defined at $\lambda = 0$. Using these ansatz functions into 3.3 referring to [6], [2], we now solve the following two PDEs,

$$\left(\frac{d^2}{dz^2} - k^2\right)^2 w(z) - Rk^2\theta = 0 \tag{3.8}$$

$$\left(\frac{d^2}{dz^2} - k^2\right)\theta(z) + w(z) = 0 \tag{3.9}$$

By substituting $\theta(z)$ from the later equation to the first one, we can rewrite the ODE as,

$$(D^2 - k^2)^3 w(z) = -Rk^2 w(z)$$

3.1.6 Solving Eigen values

Now, referring to the B.Cs **??**, we assume the ansatz along z direction as,

$$\begin{aligned}w^* &= w(z) = w_n sin(n\pi z)\\ T^* &= \theta(z) = \theta_n sin(n\pi z)\end{aligned}$$

And, we formulate the ODEs and let, $D_k = -\frac{(D^2-k^2)^3}{k^2} = -\frac{(\partial_z^2-k^2)^3}{k^2}$ and $f = w(z)$, the ODE becomes

$$D_k f = Rf \tag{3.10}$$

Inserting the ansatz to the matrix, we can analytically solve the eigen value problem [2][6],

$$\begin{aligned}D_k f &= Rf\\ (-n^2\pi^2 - k^2)^3 w_n sin(n\pi z) &= -Rk^2 w_n sin(n\pi z)\end{aligned}$$

$$\boxed{R_c(k^2) = \frac{(-n^2\pi^2 - k^2)^3}{k^2}} \tag{3.11}$$

3.2 Non-linear Heat transport

In this section we aim to explain the dynamic heat flow in nonlinear system. Here, we understand the solution of coupled differential equation using spectral method and visualizing concept of the convective patterns at critical Rayleigh condition. In this section, we first introduce the coupled differential equations in a bit different form that we introduced in the derivation of the critical Rayleigh number where we only considered the equations to vary based on z axis. The coupled differential equation can be written as,

$$\boxed{\left(\nabla - \frac{\partial}{\partial t}\right)\Theta(r,t) + u(r,t)\cdot\nabla\Theta(r,t) + W(r,t) = 0} \tag{3.12}$$

In this coupled equation, $\Theta(r,t)$ implies to the temperature deviation with respect to $r = (x,y,z)$ and $W(r,t)$ is the z-component of the streaming velocity field, $u = (U,V,W)$. The non-linear part introduced here is the $\Theta(r,t) + u(r,t)$ because it comprises of the product of two fields. Within the limit of large Prandtl numbers, we apply twice the curl operator to this Navier stokes equation and get,

$$\nabla^2 W(r,t) + R\nabla_{(x,y)}\Theta(r,t) = 0 \tag{3.13}$$

where, $\nabla_{(x,y)} = \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial y^2}$. Now, to solve this nonlinear solution and to visualize the convective patter, we first derive the $R_c(k^2)$ again considering the non-linear terms and the boundary now has a higher derivative restriction. We now require that the derivative of the velocity W normal to the boundary vanishes, $\frac{d}{dz}w(z) = 0$ at $z = 0, 1$. This let's us redefine our ansatz to a some polynomials and is explained in the below section of derivation,

3.2.1 Derivation

To integrate the the complete equations, we first calculate some small integrals to make our computation easy. Let $u = z - 1$. Then $du/dz = 1$ and $dz = du$. We start by integrating $\int_0^1 z^2(z-1)^2, dz$

$$\begin{aligned}
\int_0^1 z^2(z-1)^2, dz &= \int_{-1}^0 (u+1)^2 u^2, du \\
&= \int_{-1}^0 u^4 + 2u^3 + u^2, du \\
&= \left[\frac{1}{5}u^5 + \frac{1}{2}u^4 + \frac{1}{3}u^3\right]_{-1}^0 \\
&= \frac{1}{30}
\end{aligned}$$

Now, we integrate the expression $\int_0^1 z^4(z-1)^4, dz$ using integration by substitution.

$$\begin{aligned}
\int_0^1 z^4(z-1)^4, dz &= \int_{-1}^0 (u+1)^4 u^4, du \\
&= \int_{-1}^0 u^8 + 4u^7 + 6u^6 + 4u^5 + u^4, du \\
&= \left[\frac{1}{9}u^9 + \frac{4}{8}u^8 + \frac{6}{7}u^7 + \frac{4}{6}u^6 + \frac{1}{5}u^5\right]_{-1}^0 \\
&= \frac{1}{630}
\end{aligned}$$

Next, we calculate the integrals of $\int_0^1 z^2(z-1)^2(6z^2 - 6z + 1), dz$ over the range $[0,1]$:

$$\begin{aligned}\int_0^1 z^2(z-1)^2(6z^2-6z+1), dz &= \int_0^1 (6z^6-18z^5+16z^4-6z^3+z^2), dz \\ &= \left[\frac{6}{7}z^7-\frac{9}{2}z^6+\frac{16}{3}z^5-\frac{3}{2}z^4+\frac{1}{3}z^3\right]_0^1 \\ &= -\frac{1}{105}\end{aligned}$$

Following this, we calculate the integral of $z \cdot (z-1)$ from 0 to 1 can be written,

$$\begin{aligned}\int_0^1 z(z-1), dz &= \int_0^1 (z^2-z), dz \\ &= \left[\frac{1}{3}z^3-\frac{1}{2}z^2\right]_0^1 \\ &= \frac{1}{6}\end{aligned}$$

Now, we move towards the comprehensive derivation by initializing the ansatz required for derivation. These ansatz are chosen keeping in mind that not only the velocity $w(z)=0$ and temperature $\theta(z)=0$, but also $\nabla\theta(z)=0$ at $z=0,1$ [2].

$$\begin{aligned}\theta(z) &= \theta_0 z(1-z) \\ w(z) &= w_0 z^2(1-z)^2 \\ \phi_1(z) &= z^2(1-z)^2 \\ \phi_2(z) &= z(1-z)\end{aligned}$$

Next, we define the residuals R_1 and R_2:

$$\begin{aligned}R_1 &= \left(\frac{d^2}{dz^2}-k^2\right)^2 w(z) - Rk^2\theta(z) = 0 \\ R_2 &= \left(\frac{d^2}{dz^2}-k^2-\lambda\right)\theta(z) + w(z) = 0\end{aligned}$$

To obtain the weak form of the problem, we multiply each residual by a test function ϕ_i and integrate over the domain $0 \le z \le 1$:

$$\begin{aligned}(\phi_1, R_1) &= \int_0^1 \phi_1 \left(\frac{d^2}{dz^2}-k^2\right)^2 w(z) - Rk^2\theta(z), dz = 0 \\ (\phi_2, R_2) &= \int_0^1 \phi_2 \left(\frac{d^2}{dz^2}-k^2-\lambda\right)\theta(z) + w(z), dz = 0\end{aligned}$$

Now, we simplify the first orthogonal operation of (ϕ_1, R_1),

$$
\begin{aligned}
(\phi_1, R_1) &= \int_0^1 \phi_1 \left(\frac{d^2}{dz^2} - k^2\right)^2 w(z) - Rk^2\theta(z), dz \\
&= \int_0^1 z^2(1-z)^2 \left(\left(\frac{d^2}{dz^2} - k^2\right)^2 w_0 z^2(1-z)^2\right) dz - \int_0^1 Rk^2 z^3(z-1)^3\theta_0 dz \\
&= \int_0^1 z^2(1-z)^2 \left(\left(\frac{d^2}{dz^2} - k^2\right)^2 w_0 z^2(1-z)^2\right) dz \\
&= \int_0^1 z^2(1-z)^2 \left[\left(\frac{d^4}{dz^4} - 2k^2\frac{d^2}{dz^2} + k^4\right) z^2(z^2-2z+1)w_0\right] dz \\
&= 24w_0 \int_0^1 z^2(1-z)^2 dz + k^4 w_0 \int_0^1 z^4(1-z)^4 dz - 2w_0k^2 \int_0^1 z^2(1-z)^2(12z^2-12z+2)dz \\
&= \frac{24w_0}{30} + \frac{k^4w_0}{630} - 4k^2w_0 \int_0^1 z^2(1-z)^2(6z^2-6z+1)dz \\
&= \frac{24w_0}{30} + \frac{k^4w_0}{630} + \frac{4k^2w_0}{105}
\end{aligned}
$$

The simplification uses the results of the equations $\int_0^1 z^4(z-1)^4, dz$, $\int_0^1 z^2(z-1)^2(6z^2-6z+1), dz$ and $\int_0^1 z^2(z-1)^2$ **??**,**??** and **??**

Next, let's deduce the later part of the first integral,

$$
\int_0^1 Rk^2 z^3(z-1)^3\theta_0 dz = Rk^2\theta_0 \int_0^1 z^3(z-1)^3 dz = \frac{-Rk^2\theta_0}{140}
$$

After simplifying and calculating the first integral, we have to calculate the integral for the second orthogonality of the Residual.

$$
\begin{aligned}
(\phi_2, R_2) &= \int_0^1 \phi_2 \left(\frac{d^2}{dz^2} - k^2 - \lambda\right) \theta(z) + w(z), dz \\
&= \int_0^1 z(1-z) \left(\frac{d^2}{dz^2} - k^2 - \lambda\right) \theta_0 z(1-z) + w_0 z^2(1-z)^2 \\
&= \int_0^1 z(z-1) \left[(z^2(z-1)^2 w_0 + \left(\frac{d^2}{dz^2} - k^2 - \lambda\right) z(z-1)\theta_0\right] dz \\
&= w_0 \int_0^1 z^3(z-1)^3 dz + \int_0^1 z(z-1) \left[\left(\frac{d^2}{dz^2} - k^2 - \lambda\right) z(z-1)\theta_0\right] dz \\
&= \frac{-w_0}{140} + \theta_0 \int_0^1 z(z-1) \Big(2 - k^2 z(z-1) - \lambda(z-1)\Big) dz \\
&= \frac{-w_0}{140} + 2\theta_0 \int_0^1 z(z-1)dz - k^2\theta_0 \int_0^1 z^2(z-1)^2 dz - \lambda\theta_0 \int_0^1 z^2(z-1)^2 dz \\
&= \frac{-w_0}{140} - \frac{\theta_0}{3} - \frac{k^2\theta_0}{30} - \frac{\lambda\theta_0}{30} = 0
\end{aligned}
$$

Again, we use the results of the equations $\int_0^1 z^4(z-1)^4, dz$, $\int_0^1 z^2(z-1)^2(6z^2-6z+1), dz$ and $\int_0^1 z^2(z-1)^2$ **??**,**??** and **??**.

Now, we have integral outcome of the two integration and we here calculate the λ and θ_0. The two results are written as,

$$\frac{24w_0}{30} + \frac{k^4 w_0}{630} + \frac{4k^2 w_0}{105} + \frac{Rk^2\theta_0}{140} = 0 \tag{3.14}$$

$$\frac{-w_0}{140} - \frac{\theta_0}{3} - \frac{k^2\theta_0}{30} - \frac{\lambda\theta_0}{30} = 0 \tag{3.15}$$

Solving λ from Eq.3.15,

$$\begin{aligned}
\frac{\lambda\theta_0}{30} &= \frac{-w_0}{140} - \frac{\theta_0}{3} - \frac{k^2\theta_0}{30} \\
\lambda\theta_0 &= \frac{-30w_0}{140} - 10w_0 - k^2\theta_0 \\
\lambda &= \frac{-30}{140}\frac{0}{\theta_0} - 10 - k^2
\end{aligned}$$

Solving θ_0 from Eq. 3.14

$$\begin{aligned}
\frac{Rk^2\theta_0}{140} &= \frac{24w_0}{30} + \frac{k^4 w_0}{630} + \frac{4k^2 w_0}{105} \\
Rk^2\theta_0 &= -\frac{140k^4 w_0}{630} - \frac{3360w_0}{30} - \frac{560k^2 w - 0}{105} \\
\theta_0 &= \frac{\left(-\frac{140k^4 w_0}{630} - \frac{3360w_0}{30} - \frac{560k^2 w - 0}{105}\right)}{Rk^2}
\end{aligned}$$

Our end goal is to calculate $\lambda(k^2)$ and thus we need to substitute the θ_0 from the earlier equation to the λ expression,

$$\begin{aligned}
\lambda(k^2) &= -\frac{30}{140}\frac{w_0}{\theta_0} - 10 - k^2 \\
&= -\frac{30}{140}\frac{Rk^2 w_0}{\left(-\frac{140k^4 w_0}{630} - \frac{3360w_0}{30} - \frac{560k^2 w - 0}{105}\right)} - 10 - k^2 \\
&= \frac{0.964Rk^2}{k^4 + 23.9k^2 + 503} - 10 - k^2
\end{aligned}$$

Thus, We arrive to the following expression solving the two Residual integrations and substituting the ansatz functions that satisfy our boundary conditions,

$$\boxed{\lambda(k^2) = \frac{0.964Rk^2}{k^4 + 23.9k^2 + 503} - 10 - k^2} \tag{3.16}$$

3.2.2 Velocity and heat calculations

From 3.1.4, we observe that we have four variables to solve (velocity in three directions $u(r,t), v(r,t), w(r,t)$ and temperature $u(r,t)$). Also, simplifying the mass, energy and momentum equations, we can rewrite the equations like the following[2],

$$\begin{aligned}
w(r,t) + u(r,t)\cdot\nabla\theta(r,t) + (\Delta - \frac{\partial}{\partial t})\theta(r,t) &= 0 \\
\Delta^2 u(r,t) + R\frac{\partial^2}{\partial x\partial z}\theta(r,t) &= 0 \\
\Delta^2 v(r,t) + R\frac{\partial^2}{\partial y\partial z}\theta(r,t) &= 0 \\
\Delta^2 w(r,t) + R(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})\theta(r,t) &= 0
\end{aligned} \tag{3.17}$$

We note the boundary conditions for having no slip condition and write the following ansatz in fourier space along x, y.

$$
\begin{aligned}
\theta(r,t) &= \sum_{k_x,k_y} \theta_{k_x,k_y}(t) z(z-1) e^{ik_x} \\
u(r,t) &= \sum_{k_x,k_y} u_{k_x,k_y}(t) z(z-1) e^{ik_x} \\
v(r,t) &= \sum_{k_x,k_y} v_{k_x,k_y}(t) z(z-1) e^{ik_x} \\
w(r,t) &= \sum_{k_x,k_y} w_{k_x,k_y}(t) z^2(z-1)^2 e^{ik_x}
\end{aligned}
\tag{3.18}
$$

After inserting these ansatz in the 4 PDEs, we perform the weighted average along z direction and project all the equations onto $e^{ik'x}$ and get system of coupled equations.

$$
\begin{aligned}
\left(\frac{k^4}{140} + \frac{2k^2}{15} + 4\right) w_{k_x,k_y}(t) + \frac{Rk^2}{30}\theta_{k_x,k_y}(t) &= 0 \\
\left(\frac{-k^4}{140} - 2k^2\right) u_{k_x,k_y}(t) + \frac{R \cdot ik_x}{6}\theta_{k_x,k_y}(t) &= 0 \\
\left(\frac{-k^4}{140} - 2k^2\right) v_{k_x,k_y}(t) + \frac{R \cdot ik_x}{6}\theta_{k_x,k_y}(t) &= 0 \\
\dot{\theta}_{k_x,k_y}(t) = -\left(\frac{28}{3} + k^2\right)\theta_{k_x,k_y}(t) - \frac{7}{63}w_{k_x,k_y}(t) - 140 \cdot u(r,t) \cdot \nabla\theta(r,t)
\end{aligned}
\tag{3.19}
$$

We use these equations in the code to plot the velocity streams and thermal contours.

3.3 Implementation and Results

This section is divided into 2 sections, Linear and Non-linear. At first, we ignore the non-linear to use simple formulation of wave vectors and then compute the steady flow of the heat convection.

3.3.1 Static Linear Simulation

We first start by plotting the Rayleigh vs wavevector plot. The derivation to this result has been stated in 3.11

Algorithm 2 Calculate critical Rayleigh number and wavenumbers

$k_v alues \leftarrow$ np.linspace(0.8, 7, 100)
$Rc_{values} \leftarrow$ np.zeroslike(k_{values})
for $i \leftarrow 0$ **to** (len(k_{values}) $- 1$) **do**
    $k \leftarrow k_{values}[i]$
    $Rc \leftarrow (\pi^2 + k^2)^3/k^2$
    $Rc_{values}[i] \leftarrow Rc$
end for
$k_c \leftarrow k_{values}$[argmin($Rc_{values}$)]
$R_c \leftarrow$ np.min(Rc_{values})

Now, we try to plot the convective pattern for the velocity field $W(x, y, z)$ and the temperature field $\Theta(x, y, z)$ at $z = \frac{1}{2}$. First, we identify the solution along z and to achieve this, we again consider the ansatz **??** and considering $\theta_n = 1$, we can solve this parametric equations,

$$\begin{bmatrix} (n^2\pi^2 + k^2)^2 & -Rk^2 \\ 1 & -(n^2\pi^2 + k^2) \end{bmatrix} \begin{bmatrix} w_n \\ \theta_n \end{bmatrix} = 0$$

From here, we get,

$$w_n = \frac{Rk^2\theta_n}{(n^2\pi^2 + k^2)^2}$$

We have the value of w_n and we have the modulus of the wavevector k_c. Now we find the available wavevectors from lateral periodic structure. Finally we choose a simulation domain of 100∗100 and plot the $Theta$ and W. Also, we use the wavevector formula as $k_{x(y)} = \frac{2\pi x(y)}{L}$.

Algorithm 3 Compute convective patterns

$k_x \leftarrow 2\pi \text{np.fft.fftshift}(\text{np.fft.fftfreq}(L, L/(L-1)))$
$k_y \leftarrow k_x.\text{copy}()$
$KX, KY \leftarrow \text{np.meshgrid}(k_x, k_y)$
$x \leftarrow \text{np.linspace}(0, L, L)$
$y \leftarrow \text{np.linspace}(0, L, L)$
$z \leftarrow \text{np.linspace}(0, 1, L)$
$X, Y \leftarrow \text{np.meshgrid}(x, y)$
$Z_X, Z \leftarrow \text{np.meshgrid}(x, z)$
$kx_{critical} \leftarrow 2\pi/L * k_m$
$kz_{critical} \leftarrow 2\pi/L * n$
$U \leftarrow -\text{np.real}(w(Z, n, R_m, kz_{critical}) * \text{np.sin}(kx_{critical} * X))$
$W \leftarrow \text{np.real}(w(Z, n, R_m, kz_{critical}) * \text{np.cos}(kx_{critical} * X))$
$\Theta \leftarrow \text{np.real}(\text{theta}(0.5, n) * \text{np.cos}(kx_{critical} * X))$

Now, we plot the critical Rayleigh number, wave vector following the algorithm 4

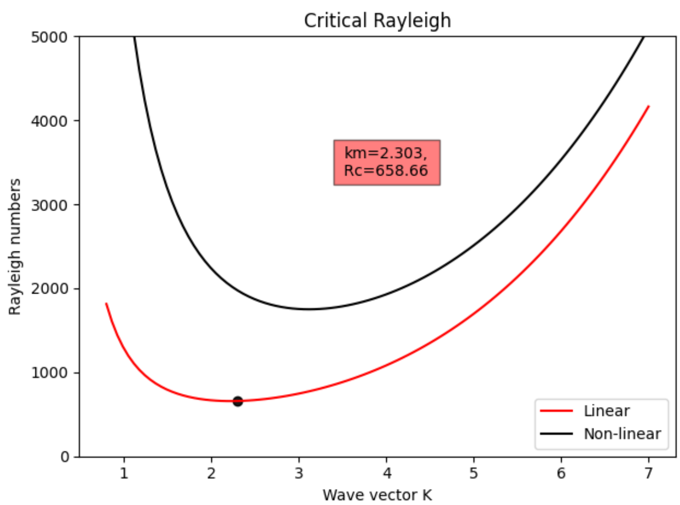


Figure 3.2: Rayleigh numbers vs wave vector

This is plotted at $\lambda = 0$ and we observe the critical Rayleigh number at $R_c = 659$ and $k_m = 2.3$. We plotted two equations from the derivations for linear and nonlinear PDEs.

In the next part, we now use the algorithm 0 in both XY plane at $Z = \frac{1}{2}$ and XZ plane. In both the plots we see that the fluid flow from cooler region to hot plane and hot region to cooler region. This spiral convective patterns are observed using the *sin* ansatz along z and *sin* and *cos* ansatz along x and y plane.

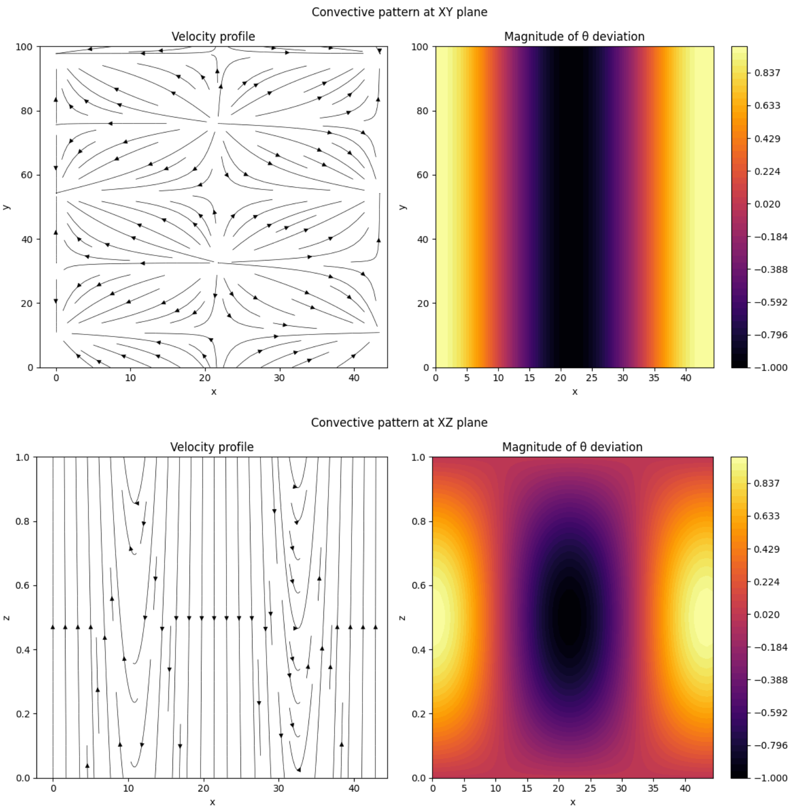


Figure 3.3: The first plot explains the convective pattern along XY plane at $Z = \frac{1}{2}$ and second plot explains the same on XY plane.

The above plot 3.3 is for static case explained as a stability analysis with $R > R_c$. The value of $R_c = 659$ and $k_c = 2.23$ is used which is almost similar to the analytical value calculated at $k_c = \frac{\pi}{\sqrt{2}}$ and $R_c = 6.75\pi^2$. The temperature gradient is set to observe with magnitude of 1 and the length scale zoomed from 100 to 40 in XY plane. This matches our experiment setup image 3.1 which means the the fluid flow from lower temperature to higher temperature as the density is less in higher temperature region. The second plot along XZ planes thereby helps us to recognize the *sin* ansatz along both X and Z direction. Resultant of two directional sinusoids is a ellipsoid by geometry. Hence, the temperature deviation looks the same. On the velocity profile, we observe the movement of velocity from the cooler region to the hot bed.

3.3.2 Non-linear simulation

In this simulation, we look back to our derivation 3.2.2 where one additional term was introduced in the coupled differential equation $u(r,t) \cdot \nabla\theta(r,t)$. We again redefine all our ansatz with more restrictions of slope being 0 at boundaries. We now write the algorithm for solving the problem of non linear Rayleigh convection following this 3.2.2 mathematical background. First we write the $\dot{\theta}_{xy}$ algorithm that propagates the temperature with respect to time.

Using this algorithm, **??** we calculate the $\dot{\theta}$ which helps to propagate the thermal magnitude with time. Now, we write the algorithm for the velocity propagation with time.

Algorithm 4 Calculate time marching of θ_{xy}

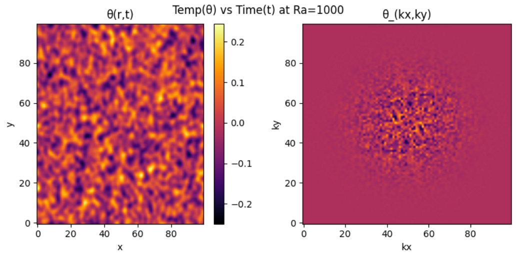
Initialize Variables to plot (Fourier space)
$\theta_{xy} \leftarrow$ np.zeros((Nx, Ny), dtype=np.complex128)
$u_{xy} \leftarrow$ np.copy(θ_{xy})
$v_{xy} \leftarrow$ np.copy(θ_{xy})
$w_{xy} \leftarrow$ np.copy(θ_{xy})
$\theta_{xy} \leftarrow$ np.fft.fftshift(np.fft.fft2(np.random.rand(Nx, Ny)))
Real space θ
$\theta_{rp} \leftarrow$ np.real(np.fft.ifftshift(θ_{xy}))
Calculate $\nabla\theta(r,t)$
$grad_{\theta x}, grad_{\theta y} \leftarrow$ np.gradient(θ_{rp})
Real space velocities
$u_{rp} \leftarrow$ np.real(np.fft.ifft2(u_{xy}))
$v_{rp} \leftarrow$ np.real(np.fft.ifft2(v_{xy}))
$w_{rp} \leftarrow$ np.real(np.fft.ifft2(w_{xy}))
Calculate $u(r,t)$
Calculate N in x and y directions
$N_{rp} \leftarrow u_{rp} \times grad_{\theta x} + v_{rp} \times grad_{\theta y}$
Calculate $N(x,y,z,t)$ in Fourier space
$N_{zrp} \leftarrow w_{rp} \times$ np.fft.ifft2(-1j * self.Kx *θ_{xy}).real
$N_{rp} += N_{zrp}$
$N_{xy} \leftarrow$ np.fft.fft2(N_{rp})
Calculate Time marching of θ_{xy} using $N_{xy} = u(r,t) \cdot \nabla\theta(r,t)$
$\theta_{xy} += dt \times (-(28/3 + self.K_2) \times \theta_{xy} - (7/63) \times w_{xy} - 1/140 \times N_{xy})$
return $\theta_{xy}, u_{xy}, v_{xy}, w_{xy}$

Algorithm 5 Calculate x,y,z component of velocity in real and Fourier space

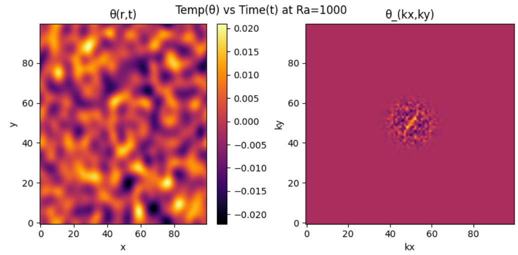
Masking to avoid zero division
$mask \leftarrow (K_2^2 > 0);;(K_2 < 1e3)$
Calculate x,y,z component of velocity here
$u_{xy}[mask] \leftarrow \frac{Rayleigh \cdot i \cdot Kx[mask] \cdot \theta_{xy}[mask]}{(\frac{K_2[mask]^2}{2} + 12 \cdot K_2[mask])}$
$v_{xy}[mask] \leftarrow \frac{Rayleigh \cdot i \cdot Ky[mask] \cdot \theta_{xy}[mask]}{(\frac{K_2[mask]^2}{2} + 12 \cdot K_2[mask])}$
$w_{xy}[mask] \leftarrow -\frac{(Rayleigh \cdot K_2[mask] \cdot \theta_{xy}[mask])/30}{(\frac{K_2[mask]^2}{140} + \frac{2 \cdot K_2[mask]}{15} + 4)}$
Velocity in real space
$u \leftarrow real(ifft2(u_{xy}))$
$v \leftarrow real(ifft2(v_{xy}))$
$w \leftarrow real(ifft2(w_{xy}))$
return $u, v, w, u_{xy}, v_{xy}, w_{xy}$

This algorithm calculates the x, y, z components of velocity using the Rayleigh number (Rayleigh), the wave vectors in the x and y direction (K_x, K_y), and the variables in Fourier space (θ_{xy} and K^2). Then it calculates the velocity in real space by taking the inverse Fourier transform of the variables u_{xy}, v_{xy}, w_{xy} using np.fft.ifft2() and np.real() functions, which returns the real part of the resulting complex numbers.

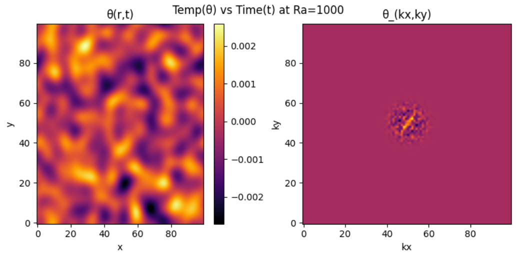
Figure 3.4 plots the temperature flow with time. We here initialize the θ to be random across the real space. And the physics behind this simulation over time states that we have random high and low temperature points defined in real space and the corresponding



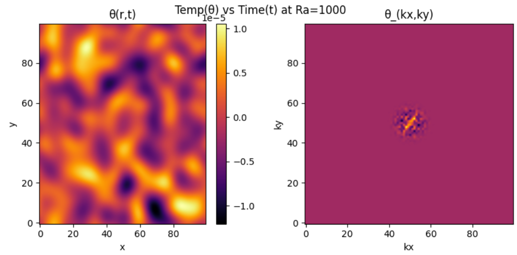
(a) at $t = 0.02$ secs



(b) at $t = 0.2$ secs



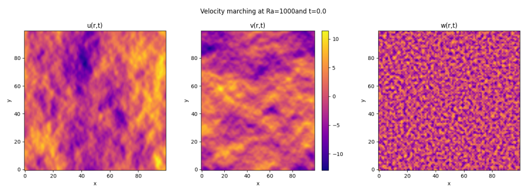
(c) at $t = 0.4$ secs



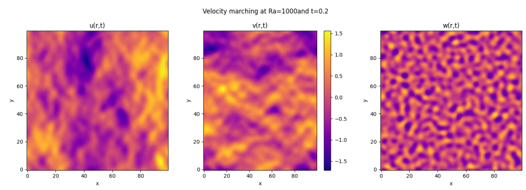
(d) at $t = 0.95$ secs

Figure 3.4: Temperature deviation plot θ. We choose to take Rayleigh number at $Ra = 1000$ which is $Ra > R_c$ and the timestep is chosen at $t = 0.001$ seconds.

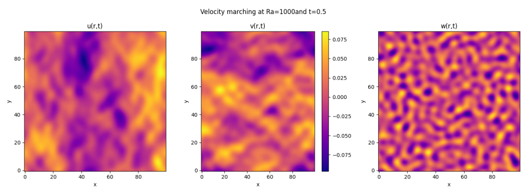
reciprocal values yields wave vectors clustered together in a circular pattern. With time, the natural convection is performed and the temperature transportation takes place. This evanescence phenomenon is observed due to the movement of fluid particles from cooler region to higher region. Hence, a natural convection is taking place. The non-linearity just makes the simulation more real-world.



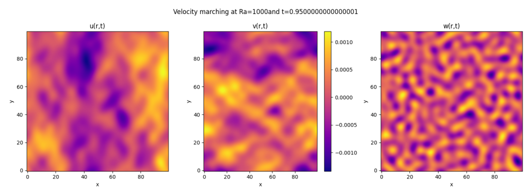
(a) at $t = 0$ secs



(b) at $t = 0.2$ secs



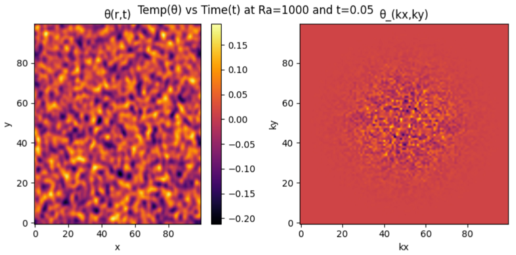
(c) at $t = 0.5$ secs



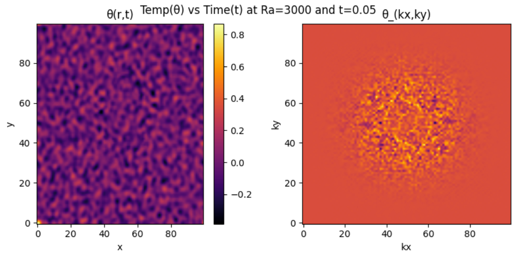
(d) at $t = 0.95$ secs

Figure 3.5: Velocity propagation is presented here for 3 components of velocity $u(r,t), v(r,t), w(r,t)$. We choose to take Rayleigh number at $Ra = 1000$ which is $Ra > R_c$ and the timestep is chosen at $t = 0.001$ seconds.

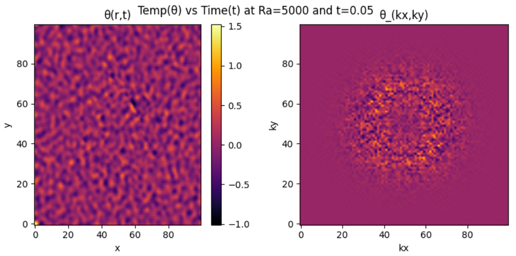
In Figure 3.5, we aim to present how three components of velocity induced by non-linear temperature gradient flows with time. Like we have seen in 3.4, the velocity is being distributed randomly in real space and due to natural convection, we see such contours fading away due to the fact that velocity is driven by the temperature difference in natural convection.



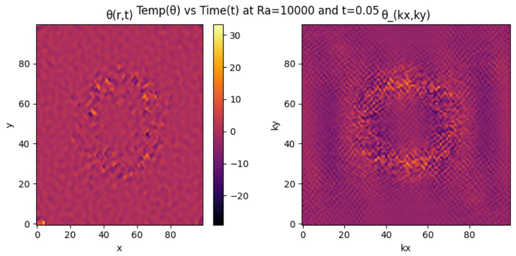
(a) at $t = 0.5, Ra = 1000$



(b) at $t = 0.5, Ra = 3000$



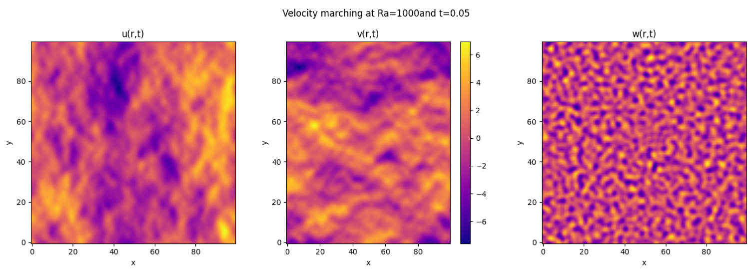
(c) at $t = 0.5, Ra = 5000$



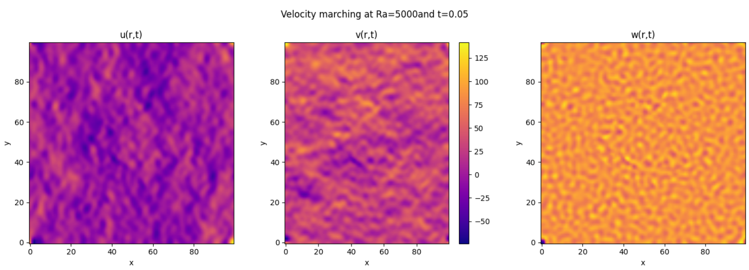
(d) at $t = 0.5, Ra = 10000$

Figure 3.6: Temperature at different Rayleigh numbers $Ra = 1000, 3000, 5000, 10000$ at $t = 0.5$

In 3.6, we see that higher the Rayleigh number, faster the propagation at a constant time rate. This is also physically interpretable as higher the Rayleigh number, higher is the buoyancy force compared to viscous force. This means that the fluid is more prone to convection and turbulence, and the flow is likely to become more complex and unstable. Also, this means a higher Rayleigh number may signify more rapid mixing and heat transfer in fluids, as well as the possibility of more chaotic and unpredictable fluid behavior. This phenomenon is again observed in velocity field 3.7.



(a) at $t = 0.5, Ra = 1000$



(b) at $t = 0.5, Ra = 5000$

Figure 3.7: Velocity at different Rayleigh numbers $Ra = 1000, 5000$ at $t = 0.5$

4

Conclusion

This project takes some of the fourier concepts and Galerkin numerical method to solve partial differential equations and we specifically solve the Rayleigh-Bénard convection using these methods. We first start by explaining some simple examples as exercises on how to use Galerkin method to solve a partial differential equation. This is mainly done by choosing some boundary suitable ansatz functions. Then, we also demonstrate how we can interpolate any function from its discrete samples using Fourier transform. Also, we explain how to calculate and code higher order derivatives using fourier method. We then derive some mathematical basis for solving the explicit and implicit Heat diffusion solution. Comparison between analytical and numerical methods have been also plotted here.

Since we are trying to solve a heat transport problem, the fundamentals of heat diffusion and solving it numerically has been explained. The derivation that guides the Galerkin method using orthogonality is also presented along with the corresponding algorithm. We present the heat flow profile in 1D and 3D for different heat sources.

Next, we start with the basic introduction to Rayleigh-Bénard problem and mentioned a derivation that leads to a coupled differential equation. For derivation, we considered 2D case and using Boussinesq approximation, derived the governing equations for our problem and how we solve the ODEs using linear algebra. Critical Rayleigh number is crucial factor for deciding the behavior of the heat convection and thus we derived linear and realistic relationship between R_c and k_c a.k.a critical wave number. The realistic Rayleigh number derivation included solving some integrations over the sample space.

Non-linearity is an important aspect in convection and also, static and dynamic simulations are important to observe the heat transport phenomenon. Thus, we perform two different simulation routine to run static and dynamic simulations. The ansatz choices are also different as in non-linear dynamic simulation with time marching, we need to have some more conditions in the boundary to fulfil like the gradient at boundaries should be zero. A complete analytical derivation along with implementation algorithm and plots have been presented in this project. Finally, we show how different Rayleigh numbers influence the simulations.

Bibliography

[1] Bengt Fornberg. The pseudospectral method; comparisons with finite differences for the elastic wave equation. *Geophysics*, 52(4):483–501, 1987.

[2] Andreas Greiner Lars Pastewka. Spectral simulation mit python, 2021.

[3] Michael Bestehorn. Computational physics. In *Computational Physics*. De Gruyter, 2018.

[4] Bengt Fornberg. *A practical guide to pseudospectral methods*. Number 1. Cambridge university press, 1998.

[5] Bruce A Finlayson and Laurence Edward Scriven. The method of weighted residuals—a review. *Appl. Mech. Rev*, 19(9):735–748, 1966.

[6] UW-Madison Department of Mathematics. Derive the onset of instability for rayleigh–bénard convection.