Rayleigh–Bénard Convection

Tutorials

May 28, 2025

Department of Aeronautics, Imperial College London, UK
Scientific Computing and Imaging Institute, University of Utah, USA

Introduction

This tutorial provides an overview of utilising the spectral/hp element framework *Nektar++* for the numerical solution of a classical fluid dynamics problem: Rayleigh–Bénard Convection (RBC). The tutorial will delve into the direct numerical simulation and linear stability analysis of a two-dimensional RBC.

Information on how to install the libraries, solvers, and utilities on your own computer is available on the webpage www.nektar.info. This tutorial assumes the reader has already completed the previous tutorial in the Flow Stability series on the channel and already has the necessary software installed.

Task 1.1

Prepare for the tutorial. Make sure that you have:

- Downloaded the tutorial files: `http://doc.nektar.info/tutorials/5.8.0/rbc/rbc.tar.gz`
  Unpack it using `unzip rbc.tar.gz` to produce a directory `rbc` with subdirectories called `tutorial` and `complete`. We will refer to the `tutorial` directory as `$NEKTUTORIAL`.

Governing equations

The dynamical equations that describe Rayleigh–Bénard Convection (RBC) under Boussinesq approximation in the non-dimensional form are

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + RaPrT\mathbf{e}_y + Pr\nabla^2 \mathbf{u}, \tag{1.1}$$

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = \nabla^2 T, \tag{1.2}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{1.3}$$

where **u** and T are the velocity and temperature fields, respectively, and $\mathbf{e}_y$ is the buoyancy direction. Here p is the modified pressure including the buoyancy term accounting for the reference temperature. Flow is controlled by two non-dimensional numbers, the Rayleigh number Ra and the Prandtl number Pr.

By decomposing the velocity, temperature and pressure in the equations (1.2-1.3) as a summation of a base flow $(\mathbf{U}, \bar{T}, P)$ and perturbation $(\mathbf{u}', T', p')$, such that

$$\mathbf{u} = \mathbf{U} + \mathbf{u}' \tag{1.4}$$

$$T = \bar{T} + T' \tag{1.5}$$

$$p = P + p', \tag{1.6}$$

and retaining first-order terms only yields the linearised equations governing the evolution of infinitesimal perturbations:

$$\frac{\partial \mathbf{u}'}{\partial t} + \mathbf{u}' \cdot \nabla \mathbf{U} + (\mathbf{U} \cdot \nabla)\mathbf{u}' = -\nabla p' + RaPrT'\mathbf{e}_y + Pr\nabla^2\mathbf{u}', \tag{1.7}$$

$$\frac{\partial T'}{\partial t} + \mathbf{u}' \cdot \nabla \bar{T} + \mathbf{U} \cdot \nabla T' = \nabla^2 T', \tag{1.8}$$

$$\nabla \cdot \mathbf{u}' = 0. \tag{1.9}$$

In Chapter 2, we will validate a case of two-dimensional RBC in a domain of aspect ratio 2.02. In Chapter 3, we will discuss the linear stability analysis for a no-slip boundary condition at top and bottom plates, and will estimate the critical value of Rayleigh number. The files contained in the `$NEKTUTORIAL` directory are as follows:

- Folder `geometry`

  - `rbc.geo` - *Gmsh* file that contains the geometry of the problem.
  - `rbc.msh` - *Gmsh* generated mesh data listing mesh vertices and elements.

- Folder `DNS/Ra_5e3_Pr_0p71`

  - `rbc-DNS.xml` - *Nektar++* session file, generated with the `$NEK/NekMesh` utility, for computing the flow for $Ra = 5000$ and $Pr = 0.71$.

- Folder `LSA/Ra_1900`

  - `rbc-LSA.xml` - *Nektar++* session file, generated with `$NEK/NekMesh` for the linear stability analysis for $Ra = 1900$ and $Pr = 0.71$.

Mesh generation

The first step is to generate a mesh that is readable by *Nektar++*. The files necessary in this section can be found in `$NEKTUTORIAL/geometry/`. To achieve this task, we use *Gmsh* in conjunction with the *Nektar++* pre-processing utility called `$NEK/NekMesh`.

Specifically, we first generate the mesh in figure 1.1 using *Gmsh* and successively, we convert it into a suitable *Nektar++* format using `$NEK/NekMesh`. For all calculations in this tutorial, we fix $L_x = 2.02$ and $L_y = 1.0$ to maintain the aspect ratio of 2.02.

Task 1.2

Convert the *Gmsh* geometry provided into the XML *Nektar++* format and with two periodic boundaries

- `rbc.msh` can be generated using *Gmsh* by running the following command:

```
gmsh -2 rbc.geo
```

- `rbc.xml` can be generated using the module `peralign` available with the pre-processing tool `$NEK/NekMesh`:

```
$NEK/NekMesh -m peralign:surf1=2:surf2=3:dir=x rbc.mesh
rbc.xml
```

where `surf1` and `surf2` correspond to the periodic physical surface IDs specified in *Gmsh* (in our case the inflow has a physical ID=2 while the outflow has a physical ID=3) and `dir` is the periodicity direction (i.e. the direction normal to the inflow and outflow boundaries - in our case x).
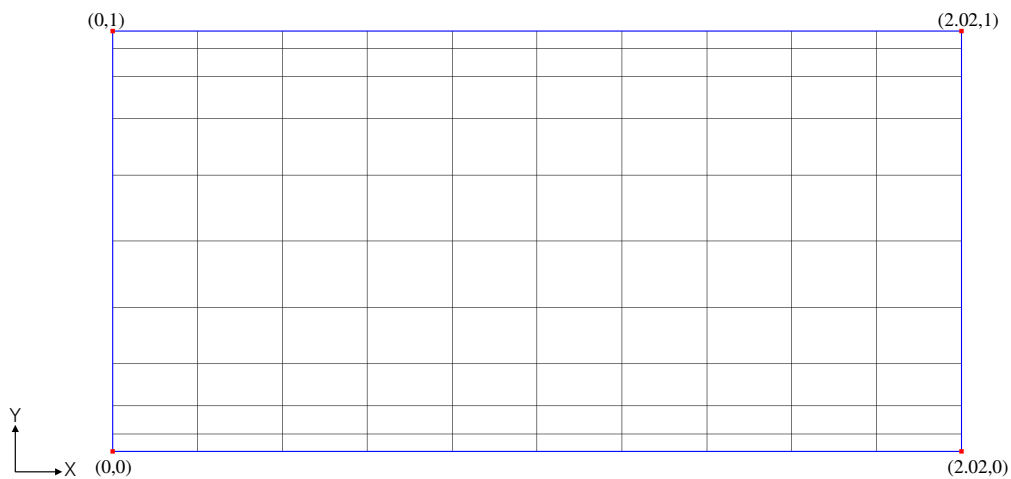

Figure 1.1 100 quadrilaterals mesh

Direct Numerical Simulation

This chapter will consider two-dimensional Rayleigh–Bénard convection in a box of aspect ratio 2.02. At the top and bottom walls, we employ a no-slip boundary condition for the velocity field and a conducting boundary condition for the temperature field. At the sidewalls, we set periodic boundary conditions for both fields. We will compare our results with Clever and Busse (J. Fluid Mech., 1974; **65**:625-645) for $Ra = 5000$, $Pr = 0.71$, and $L_x = 2\pi/3.117 = 2.02$.

In the folder `$NEKTUTORIAL/DNS/Ra_5e3_Pr_0p71` you will find the file `rbc-DNS.xml` which contains the geometry along with the necessary parameters to solve the problem. The `GEOMETRY` section is responsible for defining the mesh of the problem, and it is automatically generated, as demonstrated in the preceding task. Within the `EXPANSIONS` section, the expansion type and order are specified. An expansion basis is applied to a geometric *composite*, where *composite* refers to a collection of mesh entities. The `$NEK/NekMesh` utility always includes a default entry, and in this case, the composite `C[0]` represents the set of all elements. The `FIELDS` attribute indicates the fields for which this expansion is applicable, and the `TYPE` attribute designates the type of polynomial basis functions used in the expansion. For example,

```
1 <EXPANSIONS>
2    <E COMPOSITE="C[0]" NUMMODES="10" FIELDS="u,v,T,p"
3    TYPE="GLL_LAGRANGE_SEM" />
4 </EXPANSIONS>.
```

To finalise the problem definition and facilitate solution generation, we define a section named `CONDITIONS` within the session file. Specifically, the `CONDITIONS` section encompasses the following entries:

1. **Solver information** (`SOLVERINFO`) such as the equation, the projection type, the evolution operator, and the analysis driver to use, along with other properties. The solver properties are specified as quoted attributes and have the form

```
1 <SOLVERINFO>
2     <I PROPERTY="[STRING]" VALUE="[STRING]" />
3     ...
4 </SOLVERINFO>
```

Task 2.1

In the `SOLVERINFO` section of `rbc-DNS.xml`:

Note: The bits to be completed are identified by . . . in this file.

- set the `EvolutionOperator` to `Nonlinear` in order to select the equations (1.2)–(1.3),

- set the `Projection` property to `Continuous` in order to select the continuous Galerkin approach,

- set the `Driver` to `Standard` in order to perform standard time-integration.

2. The **parameters** (`PARAMETERS`) are specified as name-value pairs:

```
1 <PARAMETERS>
2     <P> [KEY] = [VALUE] </P>
3     ...
4 </PARAMETERS>
```

Task 2.2

Declare the two parameters `Ra`, that represents the Rayleigh number, and `Pr`, which represents the Prandtl number. Now set the Rayleigh number to 5000 and the Prandtl number to 0.71 - i.e.

```
<P> Ra = 5000 </P>
<P> Pr = 0.71 </P>
```

3. The declaration of the **variable(s)** (`VARIABLES`).

```
1 <VARIABLES>
2         <V ID="0"> u </V>
3         <V ID="1"> v </V>
4         <V ID="2"> T </V>
5         <V ID="3"> p </V>
6 </VARIABLES>
```

4. Defining **boundary regions** (`BOUNDARYREGIONS`) involves specifying composites from the `GEOMETRY` section and the associated conditions applied to these boundaries (`BOUNDARYCONDITIONS`). The format for **boundary conditions** imposed on a

region must be in accordance with the conditions outlined for each variable in the `VARIABLES` section to establish a well-posed problem. The `REF` attribute associated with a boundary condition region ought to align with the `ID="[INDEX]"` of the intended **boundary region** outlined in the `BOUNDARYREGIONS` section. For example, to model a hot bottom plate corresponding to `REF="0"`, we set the following condition:

```
1 <REGION REF="0">
2     <D VAR="u" VALUE="0" />
3     <D VAR="v" VALUE="0" />
4     <D VAR="T" VALUE="1" />
5     <N VAR="p" USERDEFINEDTYPE="H" VALUE="0" />
6 </REGION>
```

> **Task 2.3**
> Model the top-cooled plate, which corresponds to `REF="1"` in the session file.

5. The specification of (time- and) space-dependent functions (`FUNCTION`), involves defining functions in terms of x, y, z, and t, encompassing aspects like initial conditions and forcing functions. The `VARIABLES` denote the components of the particular function in a specified direction.

```
1 <FUNCTION NAME="[NAME]">
2     <E VAR="[VARIABLE_1]" VALUE="[EXPRESSION]"/>
3     <E VAR="[VARIABLE_2]" VALUE="[EXPRESSION]"/>
4     ...
5 </FUNCTION>
```

> **Task 2.4**
> Define a function called `InitialConditions`. The initial condition can be derived from the solution of the conduction equation for the temperature field in the absence of fluid motion. Therefore, the initial condition will be as follows:
>
> $$u = 0 \tag{2.1}$$
> $$v = 0 \tag{2.2}$$
> $$T = 1 - y \tag{2.3}$$
> $$p = 0 \tag{2.4}$$

Next, we apply the buoyancy force in the vertical direction, with gravity acting downward. This is implemented by defining the following function as the body force.

```
1 <FUNCTION NAME="BodyForce">
2     <E VAR="u" VALUE="0" EVARS="uvT">
```

```
3          <E VAR="v" VALUE="Ra*Pr*T" EVARS="uvT" />
4          <E VAR="T" VALUE="0" EVARS="uvT" />
5   </FUNCTION>
```

Note that for using the body force you need the following additional tag outside the section CONDITIONS:

```
1   <FORCING>
2       <FORCE TYPE="Body">
3           <BODYFORCE> BodyForce </BODYFORCE>
4       </FORCE>
5   </FORCING>
```

We also wish to monitor the simulation progress. To achieve this, we plan to record flow quantities at the location $(x, y) = (1.0, 0.5)$. For this purpose, we incorporate a filter called HistoryPoints in the session file as follows:

```
1       <FILTERS>
2           <FILTER TYPE="HistoryPoints">
3               <PARAM NAME="OutputFile">rbc</PARAM>
4               <PARAM NAME="OutputFrequency">10</PARAM>
5               <PARAM NAME="Points">
6                   1.0 0.5 0
7               </PARAM>
8           </FILTER>
9       </FILTERS>
```

This completes the specification of the problem.

Task 2.5

Run the solver by typing the following command on the command line:

```
$NEK/IncNavierStokesSolver rbc-DNS.xml
```

Now, after completing the calculation, we wish to verify if the solution has reached a steady state. To do this, we plot the time history of $v(t)$ at $(x, y) = (1.0, 0.5)$, displayed in figure 2.1. As evident from the plot, the solution has reached a steady state. It is worth noting that you can use any plotting software for this analysis. Next, we want to visualise the steady solution. Specifically, we need to convert the .fld file into a format readable by a visualisation post-processing tool. In this tutorial we decided to convert the .fld file into a VTK format and to use the open-source visualisation package called *Visit*.

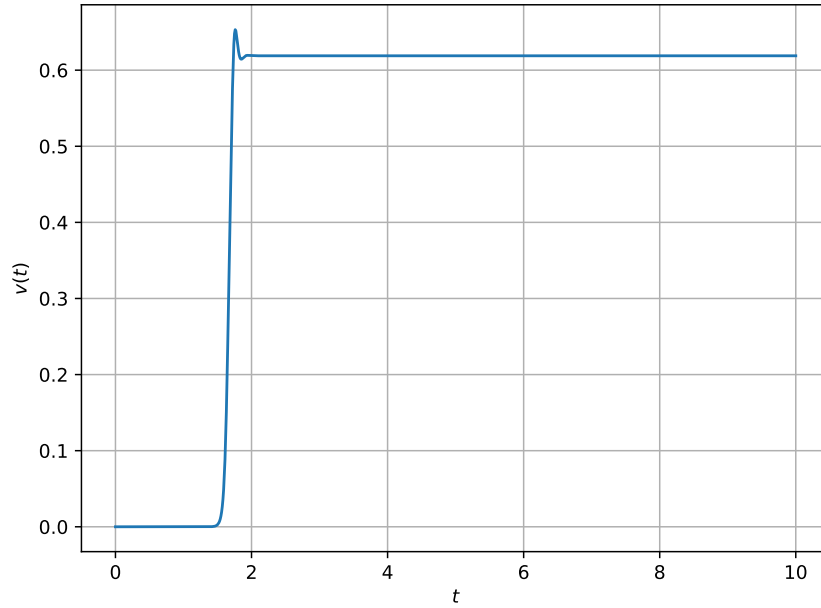**Figure 2.1** Time history of $v(t)$ velocity component measured at $(x, y) = (1.0, 0.5)$.

> **Task 2.6**
>
> Convert the file:
>
>   `$NEK/FieldConvert rbc-DNS.xml rbc-DNS.fld rbc-DNS.vtu`
>
> Now open *Visit* and use File ->Open, to select the VTK file. Next, you can plot different fields.

Figure 2.2 displays the density plots of the temperature field and the velocity vector field in the $x - y$ plane.

Nusselt number

To validate our results, we will calculate the Nusselt number (Nu), representing the ratio of the total heat flux (convective plus conductive) to the conductive heat flux at the plate. The Nusselt number is expressed as follows[1]:

$$Nu = -\frac{1}{L_x} \int \frac{\partial T}{\partial y} dx. \tag{2.5}$$

For the calculation of the Nusselt number, we will use the `FieldCovert` and `NekMesh` utility. We will follow the following steps:

---

[1] Gary A. Glatzmaier, *Introduction to Modeling Convection in Planets and Stars: Magnetic Field, Density Stratification, Rotation*, Princeton University Press, 2013.
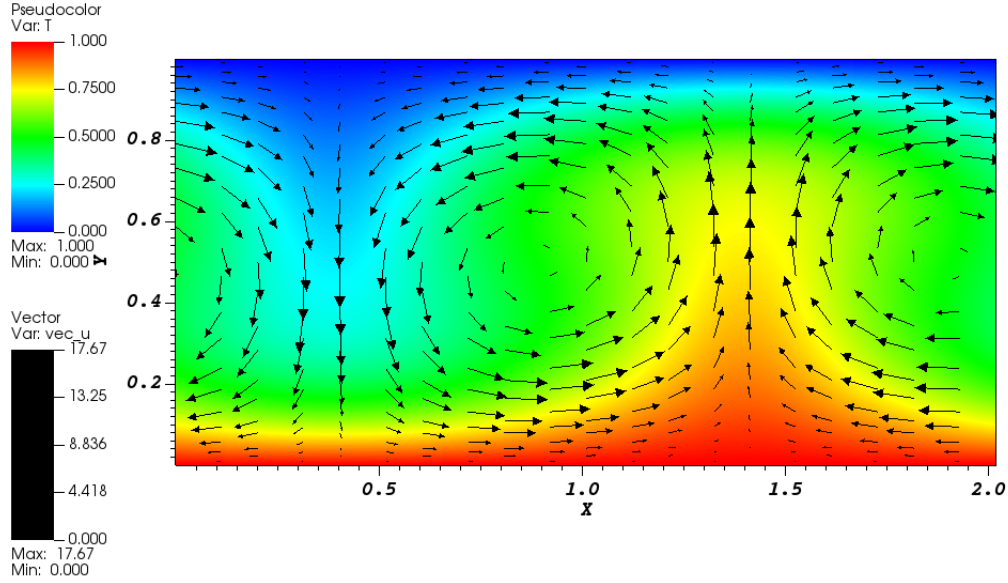
Figure 2.2 Converged solution for $Ra = 5000$ and $Pr = 0.71$.

1. Extract the wall boundary surface in a `xml` file using the `NekMesh` utility as:

```
NekMesh -m extract:surf=1 rbc-DNS.xml rbc_surf.xml
```

> **Tip**
> Remember to change the `NUMODES` and the polynomial basis functions according to the session file in `rbc-DNS.xml`.

2. Compute $\frac{\partial T}{\partial y}$, and remove all unnecessary fields.

```
FieldConvert -m gradient rbc-DNS.xml rbc-DNS.fld temp_1.fld
FieldConvert -m removefield:fieldname="u,v,T,p,u_x,u_y,\\
v_x,v_y,T_x,p_x,p_y" rbc-DNS.xml temp_1.fld dT_dy.fld
```

3. Next, extract $\frac{\partial T}{\partial y}$ at the top boundary.

```
FieldConvert -m extract:bnd=0 rbc-DNS.xml dT_dy.fld dT_dy_bnd.fld
```

4. Finally, we will compute $\frac{1}{L_x} \int \frac{\partial T}{\partial y} dx$ using `mean` utility.

```
FieldConvert -m mean rbc_surf.xml dT_dy_bnd_b0.fld stdout
```

Above command gives the following output:

```
Domain length : 2.02
Integral (variable T_y) : -4.26191
Mean (variable T_y) : -2.10986
```

Now since

$$\frac{1}{L_x}\int \frac{\partial T}{\partial y}dx = -2.10986 \implies Nu = -\frac{1}{L_x}\int \frac{\partial T}{\partial y}dx = 2.10986.$$

Our calculation of the Nusselt number match very well with that of Clever and Busse (J. Fluid Mech., 1974; **65**:625-645), which is 2.112.

Task 2.7

In the file `rbc-DNS.xml` change the values of Ra and Pr, and again run the solver to compute the Nusselt number (your results should match with Table (2.1).

Ra	Pr	Δt	Nu
10^4	0.71	5×10^{-4}	2.65179
3×10^3	7.0	10^{-3}	1.66236

Table 2.1 Nusselt number as a function of Rayleigh number and Prandtl number. Above results matches very well with the Table 1 of Clever and Busse (J. Fluid Mech., 1974; **65**:625-645).

This completes the tutorial.

Linear Stability Analysis

In this chapter, we will compute the leading eigenvalues and eigenvectors for Rayleigh–Bénard Convection (RBC). We have the analytical solution for the base flow, which is simply the solution of the conduction equation for the temperature field in the absence of fluid motion (the same solution used as the initial condition in the previous chapter). Therefore, $\mathbf{U} = 0$ and $\bar{T} = 1 - y$. Our goal is to reproduce the value of the critical Rayleigh number, $Ra_c = 1707.762$, first predicated by Chandrasekhar[1].

First, we will perform linear stability analysis for $Ra = 1900$ and $Pr = 0.71$. In the `$NEKTUTORIAL/LSA/Ra_1900` folder there is a file called `rbc-LSA.xml`. This file is similar to `rbc-DNS.xml`, but contains additional instructions to perform the linear stability analysis.

> **Task 3.1**
>
> Configure the following additional `SOLVERINFO` options which are related to the stability analysis.
>
> 1. set the `EvolutionOperator` to `Direct` in order to activate the linearised governing equations (1.8)–(1.9).
>
> 2. set the `Driver` to `ModifiedArnoldi` in order to use the *ModifiedArnoldi* eigenvalue analysis.

---

[1]S. Chandrasekhar, *Hydrodynamic and Hydromagnetic Stability*, Clarendon, 1968.

Task 3.2

Set the parameters for the Arnoldi algorithm.

- `kdim=8`: dimension of Krylov-space,

- `nvec=1`: number of requested eigenvalues,

- `nits=500`: number of maximum allowed iterations,

- `evtol=1e-7`: accepted tolerance on the eigenvalues and it determines the stopping criterion of the method.

Next, we need to define the `InitialConditions`. We will use the random initial condition, which is implemented by using following within the `CONDITIONS` section.

```
1 <FUNCTION NAME="InitialConditions">
2     <E VAR="u" VALUE="awgn(0.0001)" />
3     <E VAR="v" VALUE="awgn(0.0001)" />
4     <E VAR="T" VALUE="awgn(0.0001)" />
5     <E VAR="p" VALUE="0" />
6 </FUNCTION>
```

Task 3.3

Define a function called `BaseFlow`. Set the following function for the base flow:

$$U = 0 \tag{3.1}$$
$$V = 0 \tag{3.2}$$
$$\bar{T} = 1 - y \tag{3.3}$$
$$P = 0 \tag{3.4}$$

Task 3.4

Run the solver to perform the analysis

    $NEK/IncNavierStokesSolver rbc-LSA.xml

The simulation should converge in 8 iterations, and the terminal screen should look similar to the one below at the end:

```
EV: 0 1.13325      0             1.25088      0
Writing: "rbc-LSA_eig_0.fld" (0.015323s, XML)
```

Task 3.5

Plot the leading eigenvector in *Paraview* or *VisIt*. This should look like the solution shown in figures 3.1.
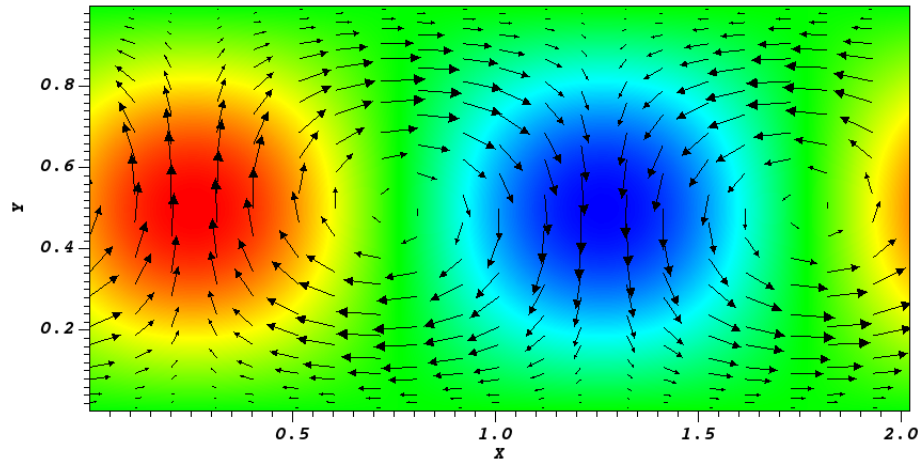

Figure 3.1 The velocity vector field $\mathbf{u}'$ on top of the density plots of T' for the leading eigenmode.

3.1   Estimate the value of critical Rayleigh number

Task 3.6

Now again run the linear stability analysis for $Ra = 1600, 1700, 1800$. Value of growth rates for these simulation are given in Table 3.1.

Ra	σ
1600	-0.731021
1700	-0.0519525
1800	0.608177

Table 3.1 Growth rate of the most unstable mode as a function of Rayleigh number.

Task 3.7

Use linear interpolation to estimate the Rayleigh critical. You should get $Ra_c = 1707.87$, which makes a good agreement with the value obtained by Chandrasekhar (1961).

For linear interpolation you can use following Python script:

```
In [1]: Ra = np.array([1600, 1700, 1800, 1900])

In [2]: sigma = np.array([-0.731021, -0.0519525, 0.608177, 1.25088])

In [3]: np.lnterp(0, sigma, Ra)

Out[3]: 1707.8700467105318
```

This completes the tutorial.