UCLouvain

epl

École polytechnique de Louvain

Numerical simulation of a nuclear spent fuel pool

Author: **Dorian Barré**
Supervisors: **Yann Bartosiewicz, Joauma Marichal**
Readers: **Yann Bartosiewicz, Matthieu Duponcheel, Joauma Marichal**
Academic year 2023–2024
Master [120] in Mechanical Engineering

Acknowledgments

I would like to thank Professor Yann Bartosiewicz for his guidance and insightful feedback, which helped me structure this text.
I am grateful to Bernard Van Renterghem for his assistance in resolving issues related to the use of the clusters.
Thank you to Professor Matthieu Duponcheel for his support with the cluster but also to be a part of my jury.
Finally, I would like to extend my sincere thanks to Joauma Marichal for the calls and meetings throughout the year, for her valuable advice on my work, and for the time taken dedicated to this endeavor.

Abstract

Nuclear spent fuel pools plays are an important part of every nuclear site. They are used as storage facilities for fuel assemblies after they have been removed from reactors. In designed condition, these pools are fed with the adequate mass flow rate of cool water. In case of a loss of cooling accident, the water present in the pool is the only mean to keep the fuel rods cold enough to maintain their integrity. Understanding the flow dynamics, heat transfer and evaporation rate in such configuration is crucial for security purposes.
Lots of numerical studies are conducted using direct numerical simulations (DNS) or large eddie simulation (LES), on simplified model, to gain insight into these phenomena. These simulations are quite expensive, limit the domain size/Rayleigh number (Ra) of the flow. Indeed, as the Ra increase, namely the parameter that quantifies the intensity of thermal driving in convection, the turbulence increase and resolution needs become harder to meet.
In this master thesis, the goal is to use the Reynolds-averaged Navier-Stokes equations (RANS) and compare the results to the ones found in the literature obtained by DNS or LES as well as with experimental data. We want to simulate flows with an evaporation model coupled with a domain decrease, to take into account the water lost to the environment.
All simulation are performed using @ Ansys Fluent 24R1.

Computational resources have been provided by the *Consortium des Équipements de Calcul Intensif* (CÉCI), funded by the *Fonds de la Recherche Scientifique de Belgique* (F.R.S.-FNRS) under Grant No. 2.5020.11 and by the Walloon Region.

Contents

Acknowledgments **i**

Abstract **ii**

1 Introduction **1**
1.1 Nuclear spent fuel pool 1
1.2 Loss of cooling accident 3
1.3 Outline 4

2 Numerical set up **5**
2.1 Geometry, boundary and initial conditions 5
2.2 Evaporation model 8
2.3 Decent 11
2.3.1 Dynamic mesh 12
2.4 Equations 14
2.4.1 Assumptions 15
2.4.2 Solver 16
2.4.3 Definition of a few output 17
2.5 Turbulence models 18
2.5.1 Boussinesq Approach 19
2.5.2 Reynolds Stress Approach 22
2.6 Mesh 23
2.6.1 Set up 24
2.6.2 Results 25
2.6.3 Conclusion 28

3 Rayleigh-Bénard convection **29**
3.1 Set up 29
3.2 Results 30
3.3 Flow visualisations 32

4 Evaporative flux **34**
4.1 Set up 34
4.2 Results 34
4.3 Conclusion 38

5 Evaporative flux and decent **39**
5.1 Comparison to another method 39
5.1.1 Set up 39
5.1.2 Results 40
5.1.3 Conclusion 43
5.2 New simulations 44
5.2.1 Set up 44
5.2.2 Results 44
5.3 Conclusion 48

Conclusion **49**

A User defined functions **51**

B Fluent case set up **58**

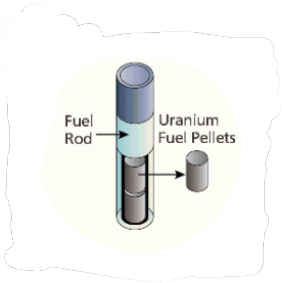
C Launching simulations on CÉCI clusters **62**

Chapter 1

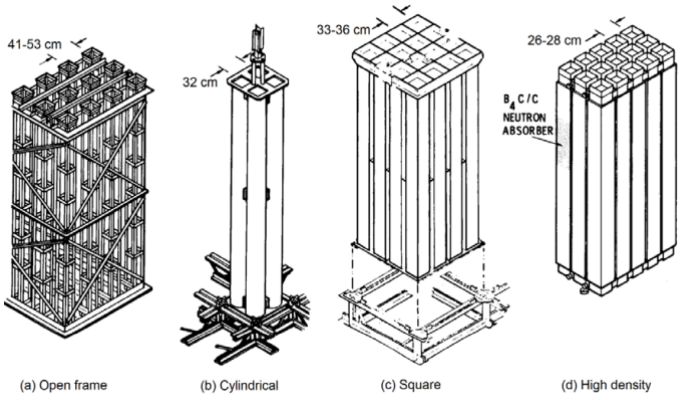
Introduction

1.1 Nuclear spent fuel pool

Spent fuel pools (SFP) are large, accident-hardened, water-retaining structures that are used to temporarily store irradiated nuclear fuel [1]. The fuel, for example enriched uranium-235 in the case of a Pressurized-Water Reactor (PWR), is sintered into pellets and loaded into long metal rods that are bundled together into a fuel assembly, as shown in Figure 1.1(a-b). After use in the reactor, they still exhibit a significant amount of residual heat that needs to be evacuated before dispatching them to long-term storage facilities.



(a) Fuel pellet stacked into metallic rod
Source: United States Nuclear Regulatory Commission



(b) Various type of fuel assembly [1]

Figure 1.1

The primary function of these pools is to store spent fuel after core off-load until the residual power is sufficiently low to allow transport of the fuel to intermediate storage. They also serve for refueling operations, stocking fresh fuel while awaiting the discharge of the spent fuel. Refueling operations are quite often an automated process. A schematic of the connection between the reactor containment building and the pool is shown in Figure 1.2 for a PWR.

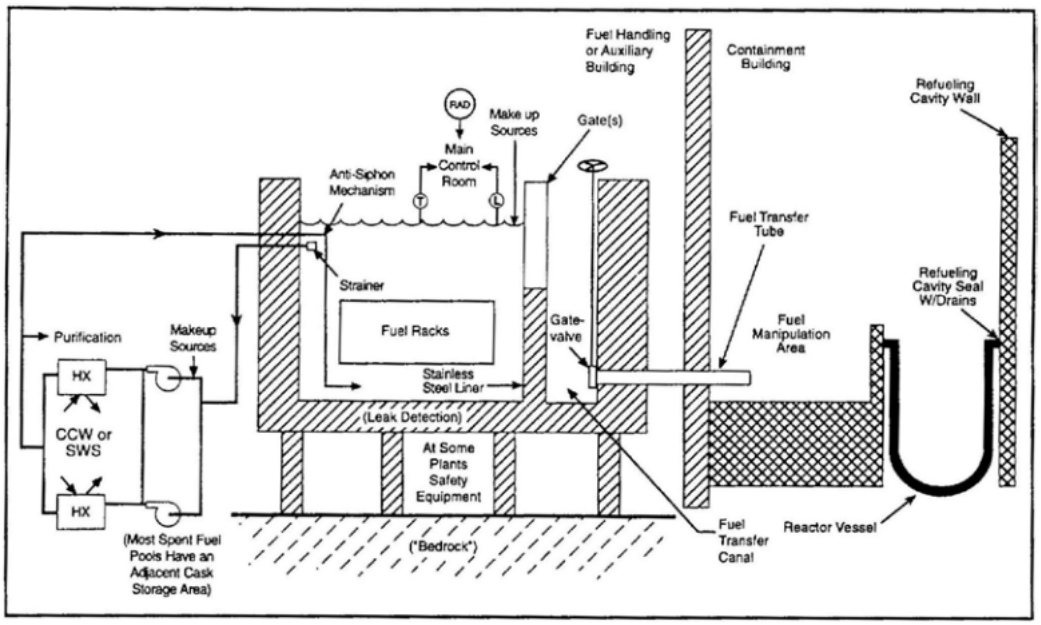


Figure 1.2: Generic SFP design for PWRs [2]

Nuclear spent fuel pool design

SFP are either close to the reactor, often in the same building as the reactor but outside its containment building (see Figure 1.2), or away from the reactor, in large facilities. The latter type is used for interim storage. The residual heat to be evacuated is much less than that of at-reactor pools and is thus less critical. We will be focusing on at-reactor pools as they are the most critical.

SFP are typically 12 meters deep, with storage racks of around 4 meters in height containing the fuel assemblies near the bottom. Their width can vary according to the needs of each individual site, but the general footprint of such a pool is around 100 m^2. They are designed to store several times the number of fuel assemblies present in the reactor core. The maximum heat load provided by the fuel assemblies is about 0.3% of the reactor thermal power, i.e., a few dozen megawatts.

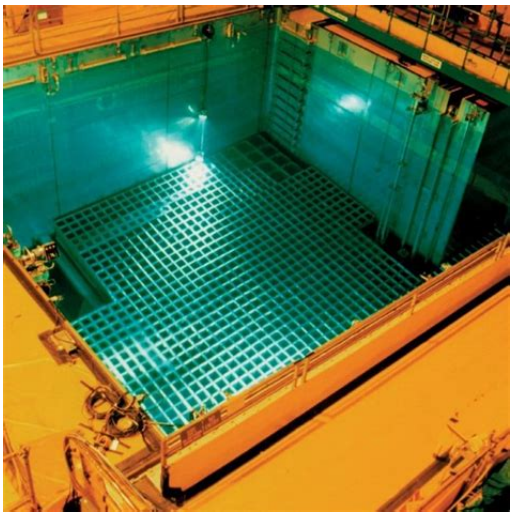


Figure 1.3: Typical spent-fuel-pool

1.2 Loss of cooling accident

"Due to the robustness of the structures, SFP severe accidents have long been regarded as highly improbable events, where there would be more than adequate time for corrective operator action. The Fukushima Daiichi nuclear accident that followed after the Great East Japan Earthquake on March 11, 2011, has renewed international interest in the safety of spent nuclear fuel stored in SFP under prolonged loss of cooling conditions."[1]

During the Fukushima accident, the loss of AC and DC power effectively shut down the secondary cooling systems. Moreover, pool water-level and temperature instrumentation became inoperable. Improvised instrumentation and aerial observations were used to monitor pool conditions [3]. Water levels were of great concern because of the possible radiation emissions that would occur if criticality was reached.

Efforts have been made to enhance nuclear safety and security following this incident. Specifically, spent fuel pools have been redesigned to withstand seismic events, and redundant and diverse cooling systems have been implemented. Additionally, monitoring systems for water level and temperature have been revisited.
Increased funding has been allocated for research on various components related to nuclear safety to better understand the phenomena involved and to develop improved models. Extensive research and simulations are being conducted to understand the behavior of spent fuel pools, including evaporation rates and water temperature dynamics.
Different approaches are undertaken, such as a 1D model used to estimate evaporative water losses driven by thermal heating of the pool water from radioactive decay in the stored spent fuel using the steady-state energy-balance model [3].
3D numerical simulations of turbulent natural convection in an air–water system with evaporation across the free surface are conducted to evaluate flow structures, evaporative mass flux, and turbulence properties [4].
Similar simulations are performed with a ventilated air flow above the water interface to develop a correlation for the evaporative heat flux based on a general ocean circulation model [5].
Other evaporation rate correlations are derived using a single fluid model based on far-field conditions [6].
The latter will be the basis of this master thesis. We will perform 3D numerical simulations in a single fluid model.

1.3 Outline

- **Chapter 2:** We will present the geometry, recall the evaporative heat flux established by Hay *et al.* (2021), and describe the method for achieving domain decrease. The equations solved in Fluent, along with their respective hypotheses and models, are discussed, as well as the chosen mesh.
- **Chapter 3:** We will briefly present results under the classical system of Rayleigh-Bénard convection in a cubic cell, for further comparison.
- **Chapter 4:** Here, we will compare and validate the implementation of the evaporative heat flux in Fluent and the model used against the results of Hay *et al.* (2021).
- **Chapter 5:** First a comparison with another method will be made, then, exploring new territory, we will investigate the addition of domain decrease to capture water losses and its effect on the simulation outcomes. Higher Rayleigh flows are also presented.
- **Conclusion:** We will review and discuss the results found. Possible improvements will be presented.
- **Appendix:** The user-defined functions for initializing the temperature field, applying the non-homogeneous heat flux at the interface, and implementing domain decrease are provided here. Additionally, a brief tutorial is included to guide you through setting up a simulation-ready case. The necessary journal and bash script for running the simulations on the CÉCI cluster are also included.

Chapter 2

Numerical set up

In this section, we will go over the geometry used in the simulations and the mathematical formulations of the problem. We will present the turbulence model used along with the mesh. We will also present the hypotheses and simplifications made.

2.1 Geometry, boundary and initial conditions

Nuclear spent fuel pools typically contain various equipment along their walls, including lights, tools, cables, and piping. A common characteristic of these pools is their rectangular cuboid shape. Additionally, the fuel assemblies are located at the bottom of the pools, making it the primary heat source, while the free surface acts as the heat sink.
A simplified geometry that captures these characteristics is a cube with an aspect ratio $\Gamma = \frac{W}{H} = 1$ (see Figure 2.1 for the definition of H and W). We could also use a rectangular cuboid with a different aspect ratio, but we will maintain $\Gamma = 1$ throughout the document (except for one comparison were that will be explicitly mention).
We imposed various boundary conditions to study their effect on the flow. The bottom wall is held at a fixed temperature, chosen according to the type of regime we want to achieve. This simulates the heat source provided by the assemblies. (The flow around the assemblies is not modeled directly but could be addressed using a porous media approach). A no-slip boundary condition is imposed at the bottom of the domain.
Regarding the side walls, they are considered thermally insulated and thus adiabatic. A no-slip boundary condition is also imposed on all four side walls.
For the top wall, the boundary condition will vary depending on the type of simulation we perform. Three cases of interest will be investigated.

- The first case has a no-slip wall at a fixed temperature lower than that of the bottom wall. This case is referred to as Rayleigh-Bénard convection (RBC).
- The next case features a free surface with an imposed evaporative flux to model the evaporation of water occurring in SFP. The evaporation model used is the one presented by Hay *et al.* (2021) [6], which is described later. This configuration is referred to as FS and will be used as a comparison to existing LES and DNS simulations. For both RBC and FS, H is constant and equal to W.
- Finally, the last case is the most realistic. It has a free surface with an imposed evaporative flux, similar to FS, but includes water loss through evaporation. As a result, the total water mass inside the pool decreases. Consequently, the top wall descends as the domain loses water, making the height of the domain a function of time. We refer to this case as free slip evaporation (FSE).

Mathematically, those boundary conditions write:

- Bottom wall: $u = v = w = 0, T = T_{\text{low}}$
- Sides wall: $u = v = w = 0, \frac{\partial T}{\partial n} = 0; n = x, z$
- Top wall (interchangeably called interface):
  - RBC: $u = v = w = 0, T = T_{\text{int}}$, H=W=const.
  - FS: $\frac{\partial u}{\partial y} = \frac{\partial w}{\partial y} = 0, v = 0, \frac{\partial T}{\partial y} = \frac{1}{k}(\dot{q}''_{conv} + \dot{q}''_{evap})$, H=W=const.
  - FSE[†]: $\frac{\partial u}{\partial y} = \frac{\partial w}{\partial y} = 0, v = 0, \frac{\partial T}{\partial y} = \frac{1}{k}(\dot{q}''_{conv} + \dot{q}''_{evap})$, H=H(t).

[†] $\dot{q}''_{conv}$ and $\dot{q}''_{evap}$ are define below

Regarding the initial conditions, a linear temperature profile is imposed in the domain to achieve faster convergence. The temperature ranges from T_{low} at the bottom to T_{int} at the top. In the case of RBC, T_{int} is simply the temperature imposed at the upper wall. For FS and FSE, an estimate is made. If the case is a reproduction of previous simulations, the mean and area-averaged temperature obtained at the upper wall will be used. Otherwise, an initial guess will be made, and a preliminary simulation will be run for a short time to obtain a better estimate. The full simulation is then conducted with this updated estimate. The temperature profile is set using a User-Defined Function (UDF) which imposes the following function:

$$T(x, y, z, t = 0) = \frac{T_{\text{int}} - T_{\text{low}}}{H} y + T_{\text{low}}; \tag{2.1}$$

Implementation can be found in Appendix A.4.
The fluid initial state is quiescent and pressure is set to 101325 Pa in the whole domain.

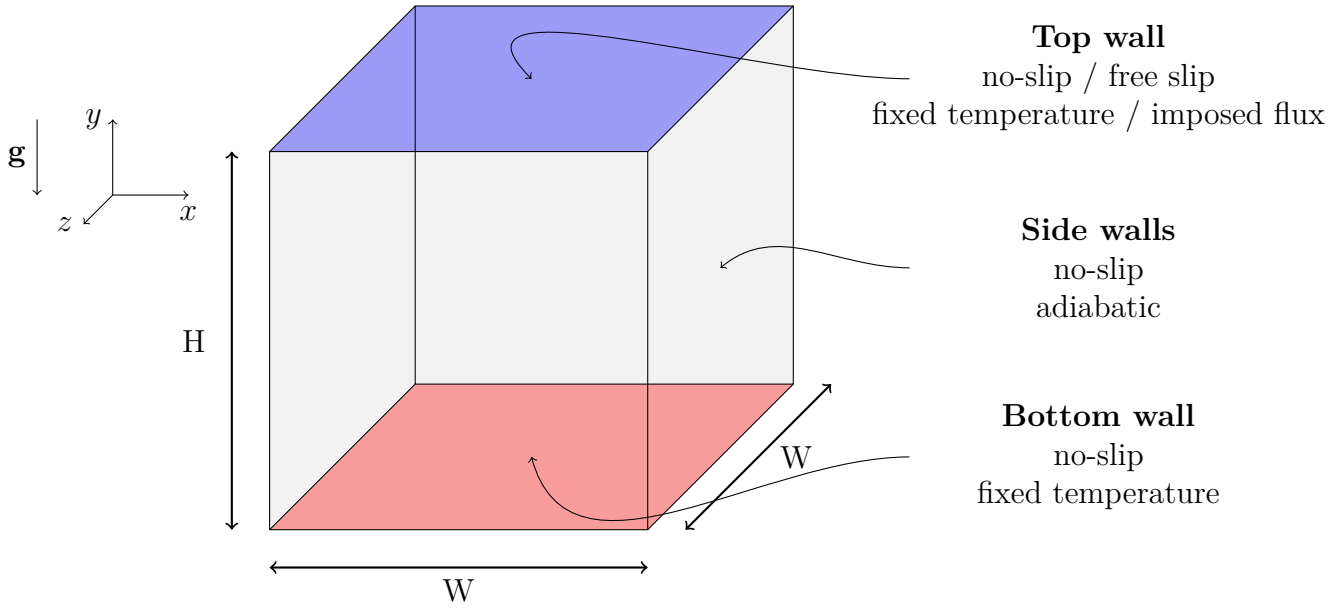


Figure 2.1: Schematic of the studied domain

2.2 Evaporation model

In order to increase the model realness, an evaporative model will be added to the upper surface. The model used is entirely based on the one detailed in Hay *et al.* (2021) [6] and recall here.
The heat flux boundary condition is calculated at each time-step and is dependent only on the interface temperature, noted T_{int}, and on fixed ambient conditions, presented in Table 2.1.

We find both the evaporative $\dot{q}''_{\text{evap}}$ and convective $\dot{q}''_{\text{conv}}$ heat losses at the water-side of the interface and use them to apply on the upper wall a heat flux of the form,

$$\dot{q}'' = \dot{q}''_{\text{conv}} + \dot{q}''_{\text{evap}} = \dot{q}''_{\text{conv}} + \dot{m}'' h_{\text{lv}} \tag{2.2}$$

where $\dot{m}''$ is the evaporative mass flux, h_{lv} is the latent heat of evaporation, and $\dot{q}''$ is the heat flux at the water-side of the interface.

T_∞ [K]	p_0 [Pa]	$p_{v,\infty}$ [Pa]	ρ_∞ [kg m^{-3}]
298.15	101325	1270	1.18

Table 2.1: Gaseous mixture properties far from the interface (at ∞). T_∞ is the ambient temperature, p_0 the ambient pressure, and $p_{\text{v},\infty}$ and ρ_∞ are the water vapour partial pressure and gaseous mixture density far from the interface. Data taken from [6]

The water vapour partial pressure $p_{v,\text{int}}$ at the interface is found[†] from the international equation for the vapor pressure of Wagner and Pruss (1993) (Eq.(2.5) in [7]).

$$\ln\left(\frac{p_{\text{v,int}}}{p_{\text{c}}}\right) = \frac{T_{\text{c}}}{T_{\text{int}}}\left(a_1\vartheta + a_2\vartheta^{1.5} + a_3\vartheta^3 + a_4\vartheta^{3.5} + a_5\vartheta^4 + a_6\vartheta^{7.5}\right) \tag{2.3}$$

with $\vartheta = 1 - T_{\text{int}}/T_{\text{c}}$, $T_{\text{c}} = 647.096$[K], $p_{\text{c}} = 22.064$[MPa], $a_1 = -7.85951783$, $a_2 = 1.84408259$, $a_3 = -11.7866497$, $a_4 = 22.6807411$, $a_5 = -15.9618719$, $a_6 = 1.80122502$.
The vapour mass fraction at the interface, $Y_{\text{v,int}}$, is then found from,

$$Y_{\text{v,int}} = \frac{p_{\text{v,int}} M_w}{p_{\text{v,int}} M_{\text{w}} + (p_0 - p_{\text{v,int}}) M_{\text{a}}} \tag{2.4}$$

where M_{w} and M_{a} are respectively the molar masses of water and air. The gaseous mixture density at the interface, ρ_{int}, is calculated from the ideal-gas equation of state,

$$\rho_{\text{int}} = \frac{M_{\text{w}} p_{\text{v,int}}}{RT} + \frac{M_{\text{a}}(p_0 - p_{\text{v,int}})}{RT} \tag{2.5}$$

[†]I used a different formula as presented by Hay *et al.*, but of the same form

$p_{\mathrm{v},\infty}$ and ρ_∞ of Table 2.1 are calculated in the same manner.

Next, we need to evaluate film properties. They are estimated from mean values of the interface and those at ∞, that is,

$$T_\mathrm{f} = \frac{T_\mathrm{int} + T_\infty}{2} \quad \text{and} \quad p_{\mathrm{v,f}} = \frac{p_{\mathrm{v,int}} + p_{\mathrm{v},\infty}}{2} \tag{2.6}$$

$Y_{\mathrm{v,f}}$ and ρ_f are calculated using Eqs.(2.4,2.5), respectively, with the corresponding film pressure vapor.

The required film dynamic viscosity is found from the kinetic theory of gases assuming a two-component mixture,

$$\mu_\mathrm{f} = \sum_{i=1}^{2} \frac{x_i \mu_i}{x_1 \phi_{i1} + x_2 \phi_{i2}} \tag{2.7}$$

with

$$\phi_{ij} = \frac{\left[1 + (\mu_i/\mu_j)^{1/2} (M_j/M_i)^{1/4}\right]^2}{\sqrt{8}\left[1 + (M_i/M_j)^{1/2}\right]} \tag{2.8}$$

where x_i is the molar fraction of component i, $x_\mathrm{w} = \frac{p_{\mathrm{v,f}}}{p_0}$ and $x_\mathrm{a} = 1 - x_\mathrm{w}$.
Values for air and vapour dynamic viscosities are found from polynomial fits.

The film mass diffusivity for the binary mixture of air and vapour, D_f is calculated as,

$$D_\mathrm{f} = 1.87 \times 10^{-10} \frac{T_\mathrm{f}^{2.072}}{p_0} \tag{2.9}$$

where the thermodynamic pressure, p_0, is in units of atmospheres, i.e. equal to 1 atm. The film diffusion coefficients are used to find the Schmidt number, Sc, and we evaluate the concentration Rayleigh number, Ra_c, as

$$Ra_\mathrm{c} = \frac{\mathrm{g}(\rho_\infty - \rho_\mathrm{int})\mathrm{W}^3}{D_\mathrm{f}\mu_\mathrm{f}} \tag{2.10}$$

with W being the characteristic length-scale, the width of the domain. We use a Sherwood number, Sh, correlation for natural convection flow

$$Sh = 0.23 Sc^{0.333} Ra_\mathrm{c}^{0.321} \quad \text{with} \quad Sc = \frac{\mu_\mathrm{f}}{\rho_\mathrm{f} D_\mathrm{f}} \tag{2.11}$$

The mass flux can be estimated by

$$\dot{m}'' = Sh \frac{\rho_\mathrm{f} D_\mathrm{f}}{W} \log(1 + B_\mathrm{m}) \tag{2.12}$$

with

$$B_\mathrm{m} = \frac{Y_{\mathrm{v},\infty} - Y_{\mathrm{v,int}}}{Y_{\mathrm{v,int}} - 1} \quad \text{and} \quad Y_{\mathrm{v},\infty} = \frac{p_{\mathrm{v},\infty} M_w}{p_{\mathrm{v},\infty} M_\mathrm{w} + (p_0 - p_{\mathrm{v},\infty}) M_\mathrm{a}} \tag{2.13}$$

Finally, we find h_{lv} from a Watson relation,

$$h_{\mathrm{lv}} = 2256.4 \left(\frac{T_{\mathrm{c}} - T_{\mathrm{int}}}{T_{\mathrm{c}} - 373.15} \right)^{0.283} \tag{2.14}$$

The evaporative heat flux is now known. Next, the convective heat flux is found by assuming that heat transfer away from the interface is proportional to the difference between the temperature of the interface, T_{int}, and the ambient gas-side temperature, T_{∞}, with the heat transfer coefficient, h, as the proportionality coefficient. We thus write

$$\dot{q}''_{\mathrm{conv}} = h(T_{\mathrm{int}} - T_{\infty}) \tag{2.15}$$

where h is estimated from the following correlation for a horizontal flat surface, warmer than the ambient air above,

$$\frac{h\mathrm{W}}{k_{\mathrm{f}}} = Nu_{\mathrm{t}} = 0.54 Ra_{\mathrm{t}}^{0.25} \tag{2.16}$$

with $\kappa_{\mathrm{f}} = \frac{k_{\mathrm{f}}}{\rho_{\mathrm{f}} c_{p,\mathrm{f}}}$ the fill thermal diffusivity and k_{f} and $c_{p,\mathrm{f}}$, the film thermal conductivity and heat capacity, respectively, are found in a similar manner as μ_{f}.

The gas side thermal Rayleigh number Ra_{t} is define as:

$$Ra_{\mathrm{t}} = \frac{\mathrm{g}\beta_{\infty}(T_{\mathrm{int}} - T_{\infty})\mathrm{W}^3}{\kappa_{\mathrm{f}}\nu_{\mathrm{f}}} \tag{2.17}$$

where $\beta_{\infty} = 3.34 \times 10^{-3}$ K^{-1} is the thermal expansion coefficient of the air at T_{∞}.

Implementation in Fluent

The heat flux boundary condition depending on the interface temperature is implemented in Fluent through a user defined function (UDF).
The UDF iterate over every face of the designated boundary (top wall), retrieving the temperature at each specific face, and then determine the corresponding heat flux according to the evaporation model.
Implementation can be found in Appendix A.2.

2.3 Decent

As evaporation occurs at the upper wall, the system loses water. Consequently, the total mass of water gradually decreases, and the computational domain shrinks. This domain reduction is managed using the dynamic mesh feature of Fluent. A prescribed velocity is imposed on the upper wall through a user-defined function (UDF).

The motion imposed on the upper wall is directly derived from the evaporation model. The UDF is called at every time step and loops through each face of the wall to retrieve the corresponding face temperatures. Using these temperatures, the evaporation model determines the evaporative mass flow rate per unit area. With the mass flow rate per unit area known, we can determine the volume of water lost at every face considering its density, face area, and the time step duration.
The total volume of water lost is obtained by summing the volumes lost at each face. This total volume, when divided by the wall surface area, yields the height reduction that occurred over the time step. In other words, the velocity at which the wall is descending.

The procedure described above is summarized in the pseudo-code of Algorithm 1 and illustrated in Figure 2.2.
Implementation can be found in Appendix A.6.

Algorithm 1 Calculation of upper wall downward velocity

procedure DESCENT(Δt) ▷ Δt is the time-step
  $VOL_{out} \leftarrow 0$ ▷ Initialize VOL_{out}, the volume of water lost
  for all faces on boundary **do** ▷ Loop through all boundary faces
    get face area A
    get face temperature T
    get mass flow rate per unit area $\dot{m}''(T)$
    get density ρ
    $VOL_{out} \leftarrow VOL_{out} + \frac{A \cdot \Delta t \cdot \dot{m}''}{\rho}$
  end for
  return $-\frac{1}{\Delta t} \cdot \frac{VOL_{out}}{W^2}$ ▷ Return the (downward) vertical velocity of the top wall
end procedure

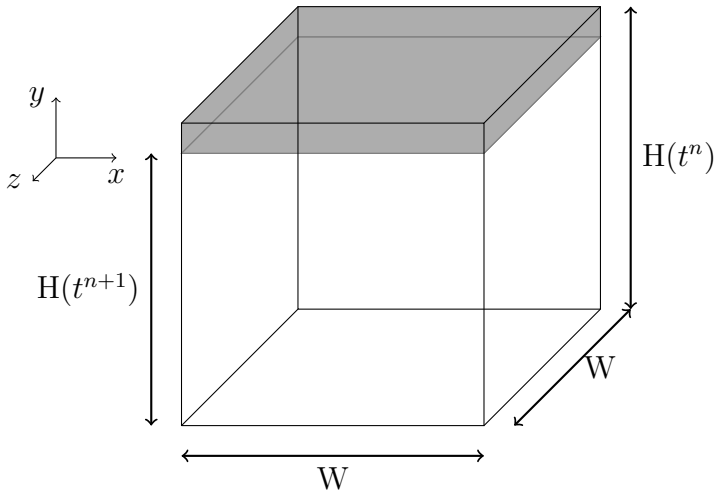


Figure 2.2: Volume reduction schematic. Not to scale

2.3.1 Dynamic mesh

The documentation and the math behind the methods used can be found in Section 12.6.2.1 Smoothing Methods, Fluent User's Guide. Some of the explanations and figures below are taken form there as well.

To account for deformation of the domain, we use the dynamic mesh feature of Fluent. We have chosen Spring-Based Smoothing method.

When smoothing is used to adjust the mesh of a zone with a moving and/or deforming boundary, the interior nodes of the mesh move, but the number of nodes and their connectivity do not change. In this way, the interior nodes "absorb" the movement of the boundary.

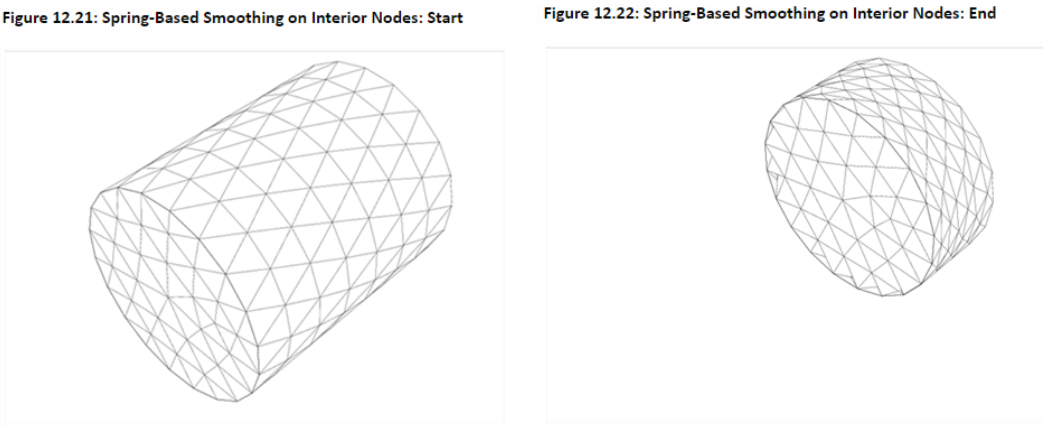
For spring-based smoothing, the edges between any two mesh nodes are idealized as a network of interconnected springs. The initial spacing of the edges before any boundary motion constitute the equilibrium state of the mesh. A displacement at a given boundary node generate a force proportional to the displacement along all the springs connected to the node.

The force on each node is expressed using Hook's Law. At equilibrium, after an iterative process, the net force on a node due to all the connected springs is zero. This condition gives the new position of each node.

Note that we could have also used a Diffusion-Based Smoothing method. It is however computationally more expensive than the spring-based smoothing. The Diffusion-based method allow for larger deformations, which we are not interested in. Moreover, spring-based smoothing method can be used when the boundary of the cell zone moves predominantly in one direction and the motion is predominantly

normal to the boundary zone. Both conditions are exactly what we are trying to do in the domain decent.

Movement is prescribed at the top wall. We considered it in Fluent as a rigid wall, with a motion imposed by the UDF. The side walls are deforming walls, so as to keep the domain rectangular. They adapt to the conditions imposed by the upper wall.
The spring stiffness is adjusted through the Spring Constant Factor, which is set to 0 in the mesh smoothing parameters dialog box. A value of 0 indicates that there is no damping on the springs, and boundary node displacements have more influence on the motion of the interior nodes. This results in an uniform compression of each node, as illustrated by Figure 2.5. Other parameters are left at their default settings.



(a) (b)

Figure 2.3: Effect of spring-based smoothing method

2.4 Equations

See Sections 1.2. and 5.2. in Fluent Theory Guide

The physics is governed by the three conservative laws: conservation of mass, momentum, and energy. The Navier-Stokes equation formulated as in the Fluent User's Guide are recalled here.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = S_m \tag{2.18}$$

$$\frac{\partial}{\partial t}(\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla p + \nabla \cdot (\boldsymbol{\tau}) + \rho \mathbf{g} + \mathbf{F} \tag{2.19}$$

$$\frac{\partial}{\partial t}\left[\rho\left(e + \frac{\mathbf{u}\cdot\mathbf{u}}{2}\right)\right] + \nabla \cdot \left[\rho \mathbf{u}(h + \frac{\mathbf{u}\cdot\mathbf{u}}{2})\right] = \nabla \cdot (k_{eff}\nabla T + \boldsymbol{\tau} \cdot \mathbf{u}) + S_h \tag{2.20}$$

where

- ρ is the fluid density
- k_{eff} is the fluid effective thermal conductivity
- μ the fluid molecular viscosity
- $\mathbf{u} = (u, v, w)$ is the velocity fields
- S_m is the mass source
- S_h is the energy source
- $\mathbf{g} = (0, -\mathrm{g}, 0)$ is the gravity acceleration vector
- $\mathbf{F}$ is external body force
- $\boldsymbol{\tau} = \mu\left[\left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right) - \frac{2}{3}\nabla \cdot \mathbf{u}\boldsymbol{I}\right]$ is the viscous stress tensor
- e stands for the internal energy of the fluid and is written as $e = h - \frac{p}{\rho}$.
  h is the sensible enthalpy computed as:

  $$h = \int_{T_{\mathrm{ref}}}^{T} c_p \, dT$$

  with $T_{\mathrm{ref}} = 298.15[\mathrm{K}]$ and c_p the fluid heat capacity at constant pressure

2.4.1 Assumptions

- Equations (2.18, 2.19, 2.20) are written without any assumptions concerning the fluid properties ρ, c_p, k, μ. However, these equations can be simplified considerably using the Boussinesq approximation. This approximation is valid when the variability in density and other fluid properties due to temperature is only moderate.

  Because variations in fluid properties are small, they can be ignored and are treated as constant. One important exception exists: the variation of density is taken into account only in the buoyancy terms $\rho\mathbf{g}$. The reason to keep this variation is that the acceleration due to buoyancy can be quite large, even larger than the accelerations due to the inertial terms [8].

  Also, the Boussinesq approximation allows for faster convergence in a simulation compared to a case where density is set up as a function of temperature.
  The buoyancy term is approximated as:

$$(\rho - \rho_0)\mathrm{g} \approx -\rho_0\beta(T - T_{\mathrm{ref}})\mathrm{g} \tag{2.21}$$

  where ρ_0 is the (constant) density of the flow, taken at T_{ref}. T_{ref} is the reference temperature (operating temperature in Fluent) and β is the thermal expansion coefficient. (Section 7.3.1.2. The Boussinesq Model, Fluent User's Guide)
  So density is constant and equal to ρ_0 everywhere except in the buoyancy term[†], and c_p, k and μ are constant everywhere.

- No source of mass S_m nor momentum S_h is implicitly implemented. The dynamic mesh feature automatically handles the update of the volume mesh at each time step based on the new positions of the boundaries. The mass is automatically updated as well.
- The flow is considered incompressible.

[†]Later on, we will just refer to the constant density as ρ and omit its subscript

2.4.2 Solver

Ansys Fluent code is based on the finite volume method, using a co-located approach, thus requiring an interpolation scheme.
Pressure and velocity are coupled using a semi-implicit method for pressure linked-equations (SIMPLE).
The pressure interpolation scheme chosen is the PRESTO! scheme, as it is recommended by the user guide for high-Rayleigh-number natural convection problems. It is used to get the pressure at cell faces.

The momentum, energy and other scalar (turbulent kinetic energy, specific dissipation rate, Reynolds stress, ...) equations are treated with a second-order upwind scheme. When using second order upwind scheme, the face value of a quantity ϕ_f is computed as:

$$\phi_f = \phi + \nabla\phi \cdot \vec{r} \tag{2.22}$$

where ϕ and $\nabla\phi$ are the cell-centered value and its gradient in the upstream cell, and $\vec{r}$ is the displacement vector from the upstream cell centroid to the face centroid. (Section 24.3.3. Evaluation of Gradients and Derivatives, Fluent Theory guide)
The gradients required in this formula are computed using the Least Squares Cell-Based Gradient Evaluation method.

The method to calculate the mass flux is left as auto-selected in the solution method dialog box. This, however, remains as the Rhie-Chow, momentum-based method, regardless of the turbulence model chosen. By default, the Auto Select option is enabled next to the Flux Type list so that Fluent automatically selects the most robust flux formulation based on the flow models and mesh attributes. (Sections 35.3.2. Mass Flux Types, Fluent User's Guide)

The discretized equations are then integrated using a first-order implicit scheme.

2.4.3 Definition of a few output

Some useful quantities are defined below, as they are used to present the output of the various simulations. This is to avoid repetition in the later sections.

The time average of a quantity ϕ is denoted by $\langle\phi\rangle$, and the time and area average over the horizontal plane is denoted by $\langle\phi\rangle_{xz}$. Area averages are calculated directly in Fluent using the Area-Weighted Average function, defined as:

$$\frac{1}{A}\int \phi \, dA = \frac{1}{A}\sum_{i=1}^{n} \phi_i |A_i| \tag{2.23}$$

The area-weighted average of a quantity is computed by dividing the summation of the product of the selected field variable and facet area by the total area of the surface. (Section 26.3.1.3. Area-Weighted Average, Ansys Fluent 24R1 Help)
We note that this function can only be called on a specific plane. Thus, the profile of a quantity $\langle\phi\rangle_{xz}$ will not be a continuous function of y.

Temperature is normalized as $\theta = \frac{T-T_{\text{ref}}}{\Delta T}$.

Velocities are normalized by the free fall velocity u_{ff}. The horizontal velocity magnitude is defined as $\langle\bar{u}\rangle_{xz} = \sqrt{\langle u\rangle_{xz}^2 + \langle w\rangle_{xz}^2}$.

Lengths $\mathcal{L}$ are normalized by the domain's initial height H (which is equal to W) and are denoted by $\hat{\mathcal{L}} = \frac{\mathcal{L}}{\text{H}}$.

The Nusselt number, is calculated by dimensionless scaling of the heat flux at the upper wall. This flux is negative, as it is a cooling wall. The corresponding Nusselt number is thus negative[†], but its sign is of zero importance. If the flux was evaluated at the bottom wall, values would have been the same except positive.

$$Nu = \frac{\langle q_{\text{w}}\rangle}{k\frac{\Delta T}{H}} \tag{2.24}$$

with q_{w} the heat flux at the upper wall.
In the context of Large Eddy Simulation, Peng *et al.* [9] showed negligible differences between the integrated Nusselt number $Nu(y)$ (which accounts for molecular diffusive heat transfer, sub-grid scale heat flux, and resolved turbulent heat flux) and the Nusselt number evaluated at the wall. For this reason, this method will be used to evaluate the Nusselt number in further simulations.

[†]From now on, Nusselt number will only be given in absolute value

2.5 Turbulence models

The complete resolution of Equations (2.18, 2.19, 2.20), known as Direct Numerical Simulation (DNS), is highly precise as it resolves all turbulent scales. However, it requires a very fine mesh. Kaczorowski & Xia (2013) [10] conducted a highly resolved DNS study on the small-scale properties of turbulent Rayleigh–Bénard convection in the core region of a cubic cell. Table 1 in [10] highlights the significant resolution requirements for such simulations. The averaging time decreased as the Rayleigh number Ra, and consequently, the number of grid points, increased. Indeed, the mean grid width $h = (\Delta x \Delta y \Delta z)^{1/3}$ is a decreasing function of the Rayleigh number. See [11] and [12] for resolution requirement correlations in DNS. This demonstrates the high computational cost of such simulations, making them impractical for industrial applications.

To avoid resolving all flow scales, averaging or filtering operations can be employed. Filtering is at the core of Large Eddy Simulations (LES). It filters the Navier-Stokes equations so that only the effects of the small-scale fluctuations are modeled by the sub-grid-scale turbulence model, while still resolving the motion of large scales (eddies).

In Reynolds-Averaged Navier-Stokes (RANS) simulations, an averaging operator is applied to the instantaneous flow quantities, separating them into their mean and fluctuating values. Only the equations governing mean-flow quantities are solved, while all fluctuations are modeled. This averaging operator is known as the Reynolds decomposition and is expressed for an instantaneous quantity ϕ as:

$$\phi = \overline{\phi} + \phi^{'} \tag{2.25}$$

where $\overline{\phi}$ is the mean component and $\phi^{'}$ the fluctuating component.

This averaging is often considered a time average for statistically steady flow but can be defined as an ensemble average for unsteady flow. The lower-frequency unsteadiness of the mean quantities is thus retained.

When applying the Reynolds decomposition to the Navier-Stokes equations, they remain in the same general form, with the velocities and other solution variables now representing ensemble-averaged (or time-averaged) values (Section 4.1.1, Reynolds (Ensemble) Averaging, Fluent Theory Guide). However, an additional term appears in the momentum equations: the Reynolds stress.

$$-\rho\overline{u_i^{'}u_j^{'}} \tag{2.26}$$

This terms needs to be modeled to close the set of equations. Two approaches will be used.

2.5.1 Boussinesq Approach

The Boussinesq approach assumes a linear relation between the stress tensor and the deformation tensor by using an effective "eddy viscosity" as a scalar quantity [13]. In short, the Reynolds stress is modeled as:

$$-\rho\overline{u_i^{'}u_j^{'}} = \mu_t \left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i} \right) - \frac{2}{3} \left(\rho k + \mu_t \frac{\partial \overline{u}_l}{\partial x_l} \right) \delta_{ij} \tag{2.27}$$

where μ_t is the eddy viscosity, also known as turbulent viscosity, and k is the turbulence kinetic energy. The goal now is to evaluate the turbulent viscosity.

The advantage of this approach is the relatively low computational cost associated with the computation of the turbulent viscosity. One or two transport equations are needed for the computation of μ_t.
However, μ_t is a scalar quantity, making the model isotropic. This assumption limits the model's accuracy when representing flows with strong anisotropic turbulence [14].

Out of the two-equation models proposed by Ansys Fluent, the $k-\omega$ SST model of Menter [15] has been chosen.

The $k-\omega$ models are well established and known for their robustness and numerical stability. Improvements from the original model have decreased its free-stream sensitivity. This new model is known as the Baseline (BSL) $k-\omega$. The SST model, which retains all the improvements of the BSL model, also accounts for the principal turbulent shear stress. The SST model leads to a significant improvement for all flows involving adverse pressure gradients and recirculation zones (see Figure 4 of [15]). Large-Scale Circulation (LSC) plays an important role in turbulent convection in a cube. It is associated with secondary circulation zones and boundary layer separation. Having a model known for better capturing recirculation, albeit in a different flow configuration, is a step toward better simulations of these flows. Therefore, using the SST model for Rayleigh-Bénard convection, with its capability to handle complex flow phenomena such as recirculation and adverse pressure gradients, is promising and could potentially yield accurate results. Moreover, we want to perform wall-resolved simulations, and $k-\omega$ models were designed to be applied throughout the boundary layer, provided that the near-wall mesh resolution is sufficient.

The model transport equations for k and ω are not recalled here but can be found in Section 4.4.2, "Baseline (BSL) $k-\omega$ Model," in the Fluent Theory Guide. Additional improvements from the BSL to the SST model are described in Section 4.4.3, "Shear-Stress Transport (SST) $k-\omega$ Model," in the Fluent Theory Guide. However, each option available in the Graphical User Interface (GUI), as shown in Figure 2.4, will be discussed.

- **k-omega Options:** Low-Re corrections
  Low Reynolds corrections were not used in the viscous model. It is not recommended and not necessary to use them when performing wall-resolved simulations. Low Reynolds number modifications have been proposed by Wilcox for the $k-\omega$ model to better match DNS data close to the wall. These corrections also affect the laminar-turbulent transition (Section 4.4, "Standard, BSL, and SST $k-\omega$ Models," Fluent Theory Guide). The simulations presented here will be for a Rayleigh number $Ra \gtrapprox 10^7$. Soft and hard turbulence will be encountered [16], so transition is not of interest.

- **Near-Wall Treatment:** Correlation
  The boundary conditions for ω depend on the first grid point y^+ values. Two distinct formulations for these boundary conditions are defined depending on this value: the low Reynolds number formulation for $y^+ < 5$ and the high Reynolds number formulation for $y^+ > 30$. Creating a mesh with only one type everywhere might not be feasible for some specific cases. Therefore, Fluent developed a y^+-insensitive near-wall treatment, which smoothly blends the two formulations. The goal of this treatment is to develop wall boundary conditions that are as insensitive to near-wall mesh resolution as possible for the wall shear stress and wall heat flux, assuming a sufficient overall resolution of the boundary layer. Nevertheless, we aim to perform wall-resolved simulations. Thus, our mesh will have y^+ values around 1. This near-wall treatment using correlations (default) will not introduce any significant modeling differences. The choice of correlation instead of tabulated values does not make a significant difference when using a fine mesh ($y^+ \approx 1$); see Figure 4.18 in Section 4.18.3, "y^+-Insensitive Near-Wall Treatment for ω-based Turbulence Models," Fluent Theory Guide.

- **Options:** Buoyancy Effects: Full
  Inclusion of the full buoyancy effects is recommended by Wei & Chen (2019) [17] when using the (RNG) $k-\varepsilon$ models. In this case, it includes the effects of buoyancy in the transport equations for both k and ω.

- **Options:** Production Limiters
  This option is activated by default. It limits the production of turbulent kinetic energy (TKE) in the vicinity of stagnation points, where the flow experiences high velocity gradients. This is a known phenomenon associated with the Boussinesq approach, as the production of kinetic energy is proportional to the square of the strain rate magnitude, as formulated in this option. In the Kato-Launder formulations, the production of TKE is proportional to the magnitude of the strain rate times the magnitude of the vorticity rate. The flow field near a stagnation point is nearly irrotational, thus limiting the TKE production. In a simple shear flow, this product is equal to the first formulation.

Other effect, such as viscous heating, curvature and corners corrections are not relevant to our problem, and are thus not selected.

Model constants are left at their default values, and no user-defined functions for turbulent viscosity and Prandtl numbers were implemented.

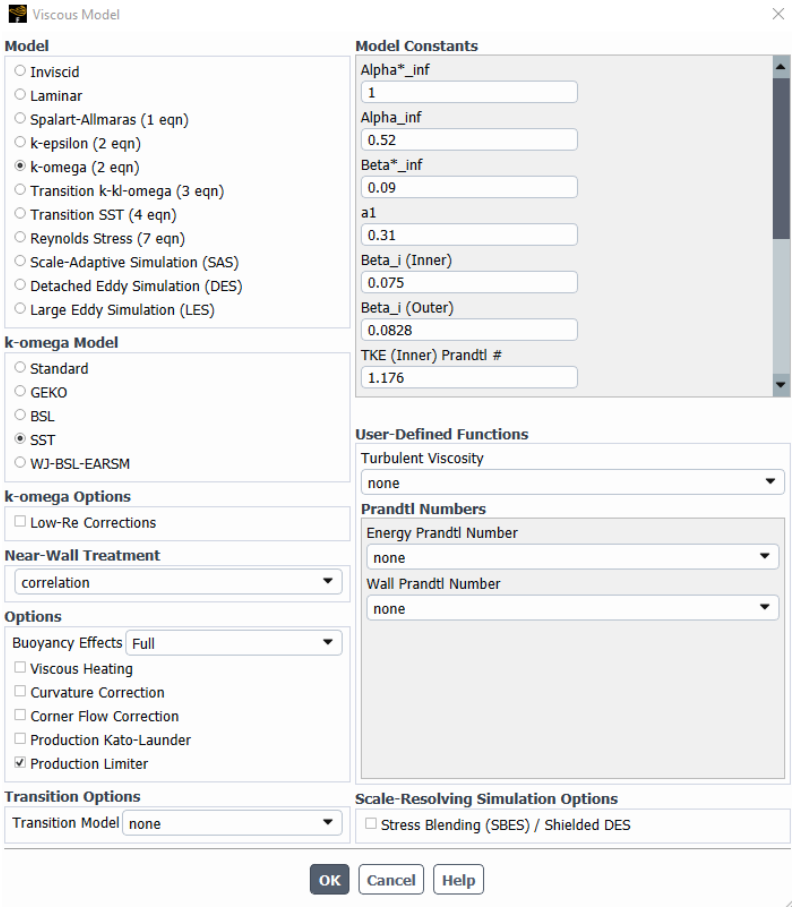


Figure 2.4: Set up of the $k-\omega$ SST viscous model

2.5.2 Reynolds Stress Approach

Chen & Chao (1997) [18] observed better agreement with experimental data using the Reynolds Stress Model (RSM) to predict airflows in a strongly turbulent buoyant plume, compared to the two-equation eddy viscosity models. These simpler models fail to capture turbulence levels and could not account for the anisotropic turbulence in the plume. This leads to poor predictions of mean flow velocity and temperature profiles.
It is thus interesting to compare the simpler model to a higher-order model such as the RSM.

In this model, all six Reynolds stresses are directly computed by solving their respective transport equations. An additional equation for the dissipation rate is needed, bringing the total to seven additional transport equations to solve. Several terms in these transport equations are directly solved. However, some terms, such as turbulent diffusion, buoyancy production, pressure strain, and dissipation, need to be modeled.

We chose to use the stress-BSL model for the same reason we chose the $k-\omega$ SST over $k-\varepsilon$ models. The equation used to model the dissipation rate ε in the RSM uses the specific dissipation rate ω, which is found using the same transport equation used in the $k-\omega$ SST model.

The near-wall treatment of the RSM also uses correlations for boundary conditions, as discussed in the Boussinesq approach. Full buoyancy effects are also selected.

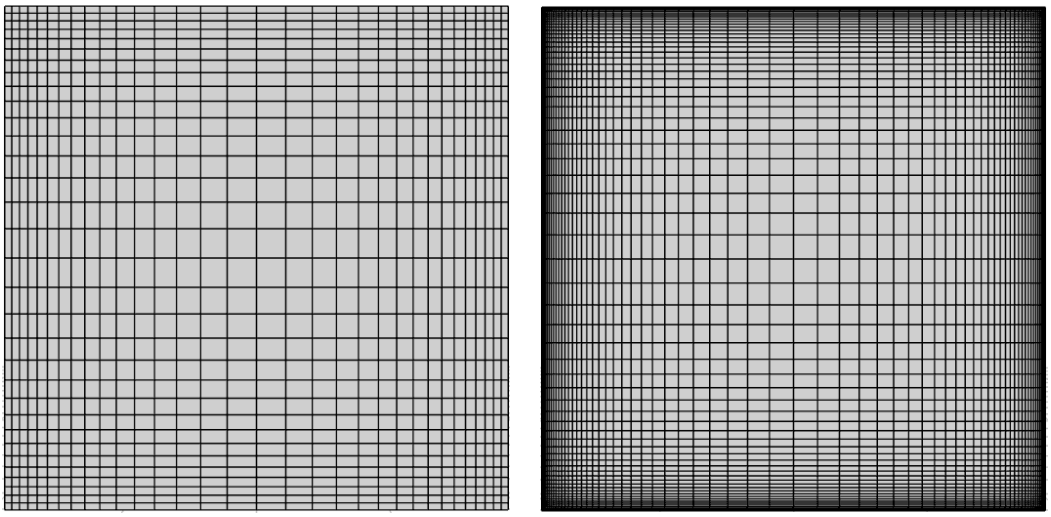
As stated in the Fluent Theory Guide, the RSM might not always yield results that are clearly superior to the simpler models in all classes of flows to justify the additional computational expense. For this reason, the $k-\omega$ SST model will be selected by default unless otherwise stated.

2.6 Mesh

The kinematic and thermal boundary layers are crucial in the thermal convection system, particularly for global heat transport across the fluid layer. For instance, the Nusselt number Nu is linked to the thickness of the thermal boundary layers at the top and bottom because the majority of the temperature drop occurs within these layers, where heat is primarily transported through conduction [19].

We thus want to perform wall-resolved simulations to eliminate additional modeling, obtain more accurate results, and completely capture the boundary layers.
To perform wall-resolved simulations, we need a sufficiently fine mesh with a few points in the viscous sub-layer. This corresponds to having a mesh with the dimensionless variable $y^+ \approx 1$ associated with the first grid point.

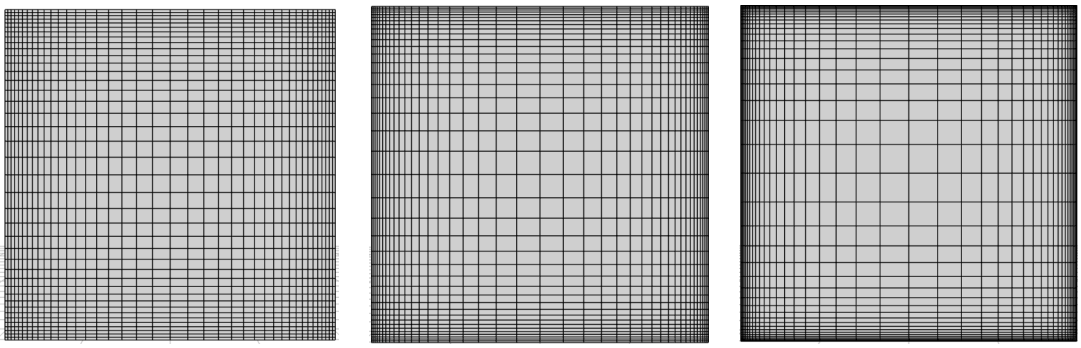
The goal is to determine if the choice of mesh is appropriate and if we can achieve a mesh-independent solution as the mesh is refined. The meshes used are structured meshes with simple edge sizing based on a given number of divisions. A bias is applied to achieve refinement close to the walls. The bias factor is kept relatively small to ensure a slow cell growth rate from the walls to the center. A total of five different refinements will be investigated, changing the number of cells and the bias factor.



(a) 32^3 cells (b) 64^3 cells

Figure 2.5: Coarsest and finest mesh, bias factor of 1.1, side wall view

The bias factor is also investigated on a rather coarse mesh (40^3 cells). This is to determine if we can achieve good results with a mesh that is fine enough near the wall but uses larger cells in the center of the domain. A larger bias factor will help to achieve finer mesh resolution close to the wall while using fewer cells overall. In other words, comparing the results obtained with different bias factors will tell us if we can accurately model the flow by properly capturing the boundary layers while maintaining a coarser mesh in the bulk of the domain.



(a) Bias : 1.1 (b) Bias : 1.15 (c) Bias : 1.2

Figure 2.6: Bias factor comparison, 40^3 cells, side wall view

2.6.1 Set up

The case of Rayleigh–Bénard Convection (RBC) is studied in a cubic domain with a height $H = 0.0267$[m]. Simulation parameters are taken from Marichal & Papalexandris (2021) [20] and are recapitulated in Table 2.2. The bottom and upper walls are at temperatures of 358.15 K and 348.15 K, respectively, leading to a temperature difference $\Delta T = 10$[K]. The reference temperature is the mean of the temperatures at the top and bottom walls.

T_{ref} [K]	ρ [kg m^{-3}]	$\beta \times 10^4$ [K^{-1}]	$\mu \times 10^4$ [Pa s]	k [m^2 s^{-1}]	c_p [J kg^{-1} K^{-1}]
353.15	972	6.36	3.54	0.667	4200

Table 2.2: Thermo-physical properties of water at T_{ref}. Density ρ, thermal expansion coefficient β, dynamic viscosity μ, thermal conductivity k and heat capacity c_p of water.

The corresponding Rayleigh and Prandtl number are:

$$Ra \triangleq \frac{\mathrm{g}\beta\Delta T \mathrm{H}^3}{\nu\alpha} = 2 \times 10^7 \quad \text{and} \quad Pr \triangleq \frac{\nu}{\alpha} = 2.23 \tag{2.28}$$

with $\nu = \frac{\mu}{\rho}$ and $\alpha = \frac{k}{\rho c_p}$, the kinematic viscosity and thermal diffusivity, respectively.

Simulations were performed for a total of 550 seconds, corresponding to approximately 840 free-fall times. The free-fall time is determined from the free-fall velocity, $u_{\mathrm{ff}} \triangleq \sqrt{\mathrm{g}\beta\Delta T\mathrm{H}}$, using the following expression: $t_{\mathrm{ff}} \triangleq \mathrm{H}/u_{\mathrm{ff}} = 0.654$.

According to Fluent's recommendation, we should use a time step $\Delta t \approx \frac{t_{\mathrm{ff}}}{4}$ to avoid divergence problems. However, a much smaller time step is chosen: $\Delta t = 0.05$. This allows for fewer iterations per time step until convergence is reached.

2.6.2 Results

Nusselt number evolution during the simulation using different meshes is shown in Figures 2.7 and 2.8. In Figure 2.7, we can notice that the Nusselt number settles as simulation time increases for every case, and as the mesh is refined, the value converges to the solution given by the finest mesh (64^3 cells). Since the difference between a mesh with 56^3 cells and 64^3 cells is already small, further refinement would not change the results. The finest mesh is thus taken as the reference case for later comparison. The average value for $t/t_{\mathrm{ff}} \geq 600$ on the finest mesh gives a Nusselt number of 18.75. The expected value was $Nu = 22.3$ (from Marichal & Papalexandris (2021) [20]), which is in very good agreement with the correlation of Niemela *et al.* [21]. Correlations from Peng *et al.* [9], $Nu = 0.162Ra^{0.286} = 19.84$, and from Hiroaki & Hiroshi (1980) [22], $Nu = 0.145Ra^{0.290} = 19.0$, are closer to our result.

Now, comparing the bias factor with 40^3 cells to the reference mesh in Figure 2.8, we can see that the mesh with a bias factor of 1.15 shows a Nusselt number close to the reference case. It shows a value even closer to the reference case than the value obtained with the 48^3 cells mesh. This indicates that if the wall regions are refined enough and the growth rate is kept reasonable, we can lower the number of cells. This is allowed because we do not have mesh constraints in the bulk, contrary to LES or DNS simulations. The mesh with 40^3 cells and a bias of 1.1 is not fine enough close to the wall. The bias factor of 1.2 is too large, leading to cells that are too large in the bulk, which corrupts the results.

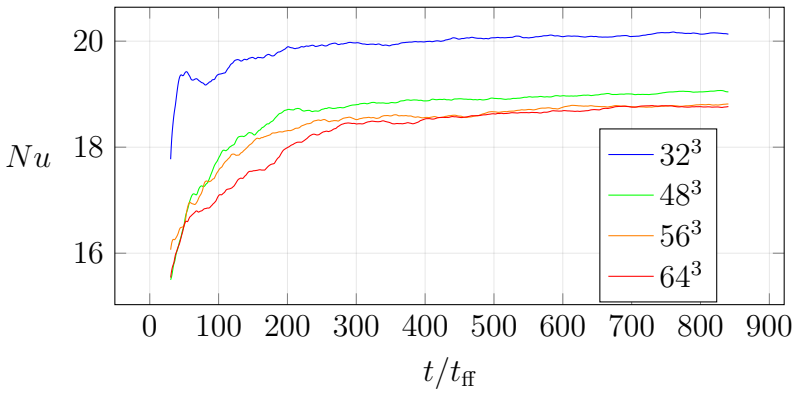


Figure 2.7: Evolution of Nu during the simulations. Bias factor set to 1.1 for all four mesh.

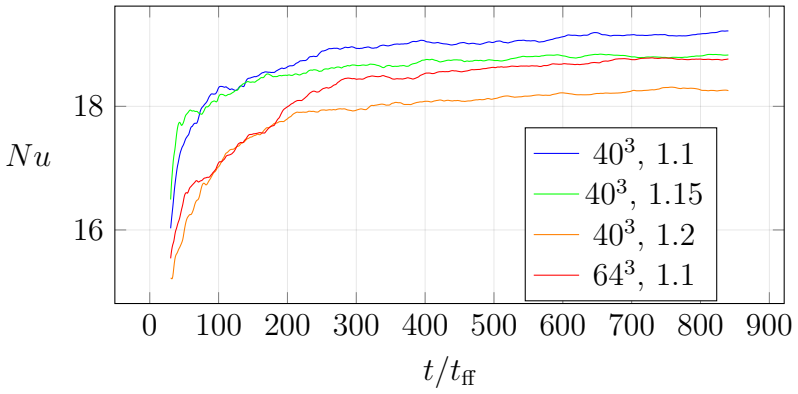


Figure 2.8: Evolution of Nu during the simulations. Bias factor comparison

The mean temperature profiles are shown in Figure 2.9. The profiles are virtually independent of the mesh. Profiles obtained with the 40^3 cells mesh are not shown, as they collapse to the same mean temperature profile. However, the profiles do not exhibit a constant temperature in the bulk, as indicated by the slight slope of the profile around $\langle\theta\rangle_{xz} = 0$.

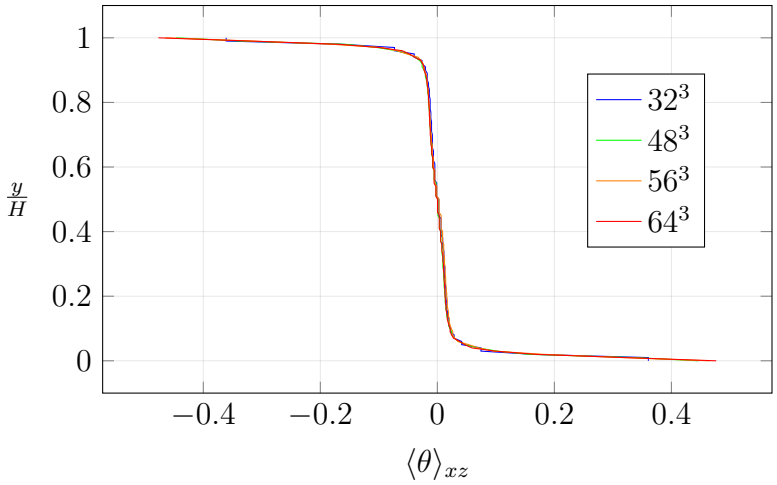


Figure 2.9: Mean temperature

Figure 2.10 shows the identification method to determine the thermal boundary layer thickness δ_{th}. The first (and last) four points of the mean temperature profile are fitted with a linear regression model. This fit is extended to $\langle\theta\rangle_{xz} = 0$. The intersection with the line $\langle\theta\rangle_{xz} = 0$ gives the boundary layer height. As explained by Zhou & Xia [19], this intersection would have been taken at the intersection of the fit and the tangent of the bulk mean temperature curve, which is vertical or close to it.
Taking the lower boundary layer, we find a normalized thickness of 0.035. This is within the range of the formula presented in [19].

$$\frac{\delta_{th}}{H} = (0.65 \pm 0.70) Ra^{-0.33 \pm 0.03} \tag{2.29}$$

The lower and upper bounds of this formula are 0.0145 and 0.0487, respectively.

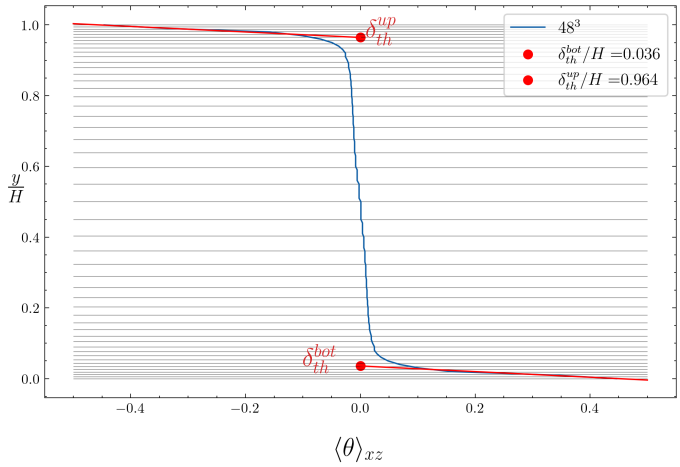
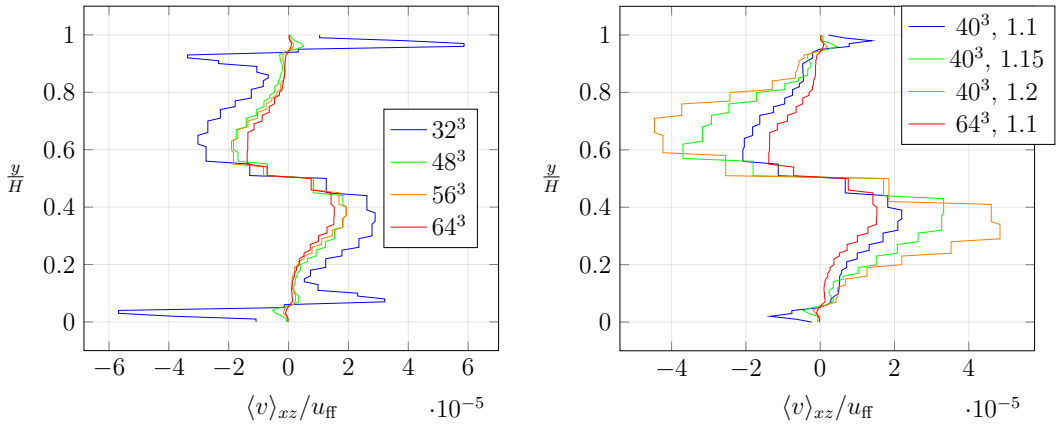


Figure 2.10: Boundary layer resolution for 48^3 cells mesh. Horizontal lines show cells faces. Verticals faces are not plotted for clarity purpose.

Moreover, Figure 2.10 shows that 5 cells are inside the thermal boundary layer thickness, both the lower and upper bounds, confirming the proper resolution of this mesh.

Figure 2.11-a shows the mean and area-averaged vertical velocity along the height of the domain. It also shows that as the mesh is refined, the profiles collapse to the one obtained with the finest mesh. It clearly shows that 32^3 cells is not enough and leads to an overestimation of the velocity almost everywhere, except in the middle of the domain. In Figure 2.11-b, we see that profiles with the smallest bias factor perform best in the middle of the domain, with a lower overestimate of the velocity magnitude. In contrast, a high bias factor helps to be more accurate near the walls, as the cells are smaller.

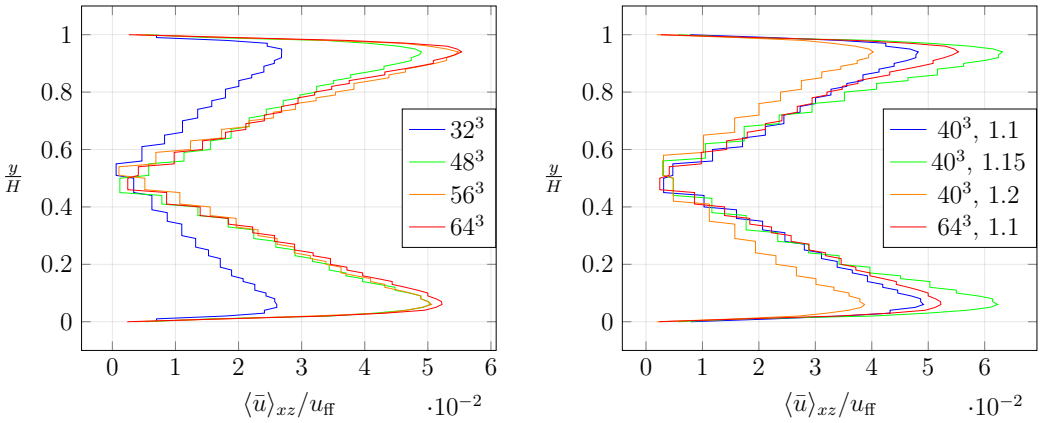


(a) Bias factor set to 1.1 for all four mesh

(b) Bias factor comparison

Figure 2.11: Mean vertical velocity

Figure 2.12-a shows again a good collapse as mesh refinement occurs. Figure 2.12-b show that high bias factor do not perform better that lower one.



(a) Bias factor set to 1.1 for all four mesh

(b) Bias factor comparison

Figure 2.12: Horizontal velocity magnitude

2.6.3 Conclusion

The goal was to find the appropriate mesh to perform later simulations. A higher bias factor helped reduce the number of cells required to predict the Nusselt number but was not the most suitable for accurately capturing the velocity profiles, at least with this number of cells. A good compromise between the number of cells and accurate results is the mesh with 48^3 cells and a bias factor of 1.1. It has captured all profiles correctly, has a Nusselt number close to the reference case, and is in agreement with some correlations. It also has 5 points in the thermal boundary layer (the 64^3 cells mesh had 9 points).

We note, however, that this mesh is appropriate for this particular Rayleigh number. As *Ra* increases, further refinement will be necessary.

Chapter 3

Rayleigh-Bénard convection

This chapter is dedicated to briefly present results under the classical system of Rayleigh-Bénard convection in a cubic cell. This configuration is labelled RBC in Section 2.1. Results will be used as a comparison for further modifications of the system, i.e., when adding the evaporative heat flux and the domain descent, discussed in the following chapters.

3.1 Set up

Table 3.1 provides the thermo-physical properties of water at different T_{ref}, taken from Hay *et al.* (2021) [6], in order to have a similar set up. Heat capacities are evaluated using a polynomial fit at the reference temperature. Thermal conductivities are deduced from the heat capacity calculated as aforementioned, and making use of the data presented in [6].
The fluid properties for each case `case_name`x are determined using the corresponding data from this table. For instance, `case_name`1 uses the properties listed under case numbering 1, `case_name`2 corresponds to case numbering 2, and so forth. Same will be applicable for future simulations in the next chapters.

case numbering	T_{ref} [K]	ρ [kg m^{-3}]	$\beta \times 10^4$ [K^{-1}]	$\mu \times 10^4$ [Pa s]	k [m^2 s^{-1}]	c_p [J kg^{-1} K^{-1}]	Pr
1	319.5	989.6	4.3	5.78	0.637	4177.87	3.8
2	332.6	983.5	5.1	4.68	0.650	4181.91	3.0
3	341.7	978.6	5.6	4.11	0.660	4187.24	2.6
4	350.4	973.5	6.1	3.65	0.665	4193.96	2.3
5	359.3	967.9	6.5	3.29	0.671	4202.34	2.1

Table 3.1: Thermo-physical properties of water at different T_{ref}. Density ρ, thermal expansion coefficient β, dynamic viscosity μ, thermal conductivity k, heat capacity c_p and the corresponding Prandtl number Pr of water.

The domain is a cube of size H = 0.045[m] for RBC cases, and of size H = 0.45[m] for bigRBC cases. Boundary conditions are defined in Section 2.1, and the temperature

used to initialize the temperature field are taken from Table 3.2.

A mesh with 56^3 cells and a bias factor of 1.1 will be used for all RBC simulations. It is a finer mesh than the one recommended to simulate the case of RBC for $Ra = 2 \times 10^7$ in Section 2.6. The reason for this finer mesh is due to the higher Prandtl number (for case RBC1 and RBC2) and the higher Ra number (case RBC4 and RBC5). Both of these parameters lead to finer thermal boundary layers.
For the three bigRBC cases, a mesh with 80^3 cells and a bias factor of 1.15 is selected. This bias factor was chosen to ensure proper wall resolution while limiting the number of cells. Moreover, as shown in Section 2.6, Figure 2.8, this bias factor allowed for a coarser mesh while still predicting the Nusselt number correctly.

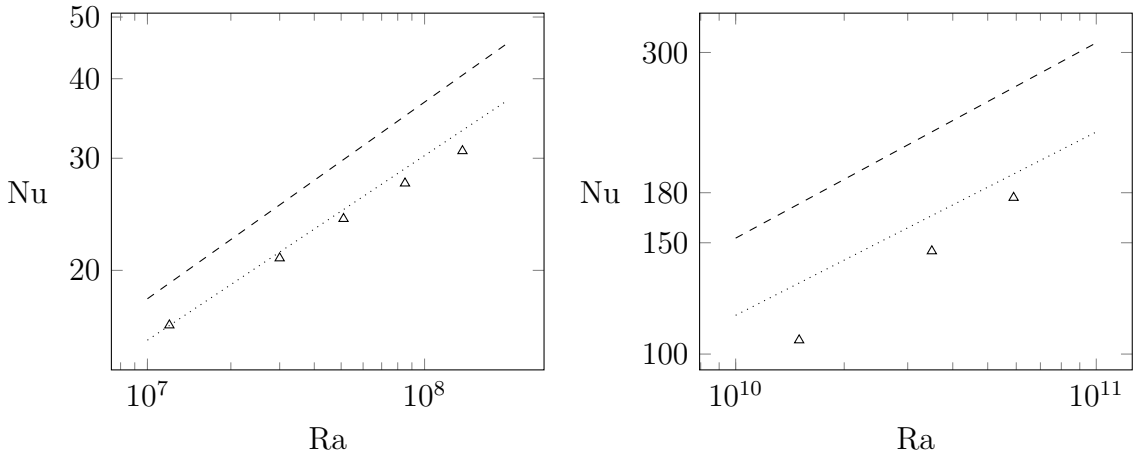
3.2 Results

The averaging time t_{avg} in free fall time units is given for each case. The main output of interest, presented in Table 3.2, is the time- and area-averaged Nusselt number Nu, evaluated at the upper wall.

case	T_{low} [K]	T_{int} [K]	ΔT [K]	Ra	Nu	t_{avg}	y^+_{max}
RBC1	320.95	318.13	2.8	1.2×10^7	16.4	694	0.33
bigRBC1		317.55	3.4	1.5×10^{10}	105.2	425	0.38
RBC2	335.15	330.24	4.9	3.0×10^7	20.9	701	0.57
bigRBC2		329.30	5.8	3.5×10^{10}	145.5	536	0.53
RBC3	345.15	338.22	6.9	5.1×10^7	24.1	703	0.92
bigRBC3		337.13	8.0	5.9×10^{10}	176.8	682	0.92
RBC4	355.15	345.59	9.6	8.5×10^7	27.4	704	1.04
RBC5	365.65	352.42	13.2	13.7×10^7	30.8	753	1.35

Table 3.2: Imposed bottom and upper wall temperature T_{low} and T_{int}, respectively. Temperature difference ΔT, Rayleigh number Ra and Nusselt number Nu of the flow. Averaging time t_{avg} and maximum y^+ value on the whole domain.

The maximum y^+ value across the entire domain for each case serves as a quick check to ensure that the mesh is fine enough to perform wall-resolved simulations. Every case has a value close to or smaller than 1, which aligns with the general recommendation for wall-resolved simulations [23].



(a) RBC

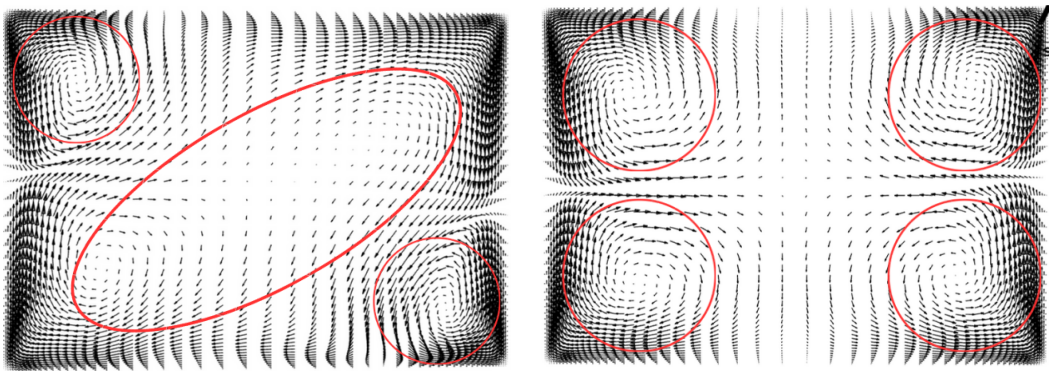
(b) bigRBC

Figure 3.1: Plot of Nu vs. Ra
Dashed line (- - -): $Nu = 0.124Ra^{0.309}$ from Niemela *et al.* (2000) and dotted line (·····): $Nu = 0.145Ra^{0.290}$ from Hiroaki & Hiroshi (1980). Data points are marked with (△).

Nusselt data did not follow the expected correlation of Niemela *et al.* (2000) [21], developed for a wide range of flows ($Ra \sim 10^6$-10^{17}) in the RBC setup, as discussed in Section 2.6. However, predictions of Nusselt values for RBC cases by Marichal & Papalexandris (2022) [20] were in good accordance with this correlation. A better fit to our results is provided by the correlation of Hiroaki & Hiroshi (1980) [22], which was developed for Rayleigh numbers ranging from 3×10^7 to 4×10^9 in a horizontal water layer heated from below.

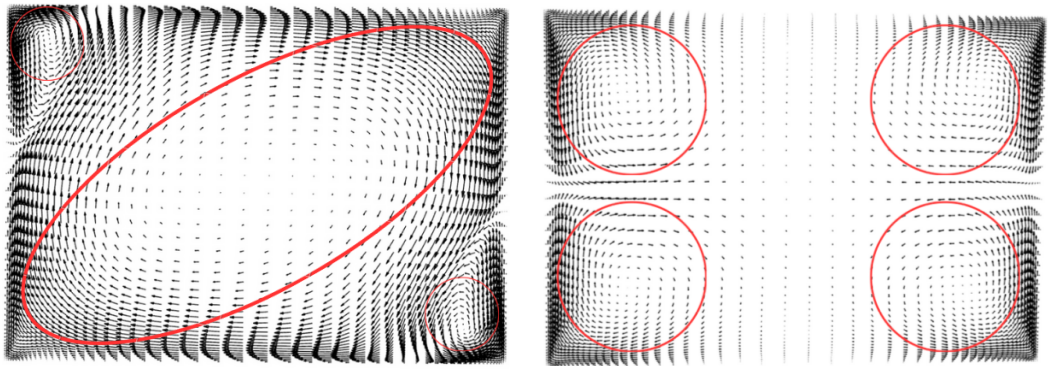
In Figure 3.1a, we observe a divergence of the data points relative to the correlations as the Rayleigh number increases. As shown in Table 3.2, the maximum y^+ value also increases with Ra. To verify that this divergence from the correlations is not related to mesh resolution, the highest Ra case, RBC5, was reproduced with a finer mesh of 80^3 cells, resulting in a $y^+_{\max} = 0.38$. The resulting Nusselt number was 30.5, compared to 30.8 with the coarser mesh, indicating that this divergence from the correlation is not due to an increase in y^+. This additional simulation also confirms that the initial mesh resolution (56^3 cells) was sufficient.
For bigRBC cases, as shown in Figure 3.1b, the results did not follow any of the established correlations.

3.3 Flow visualisations



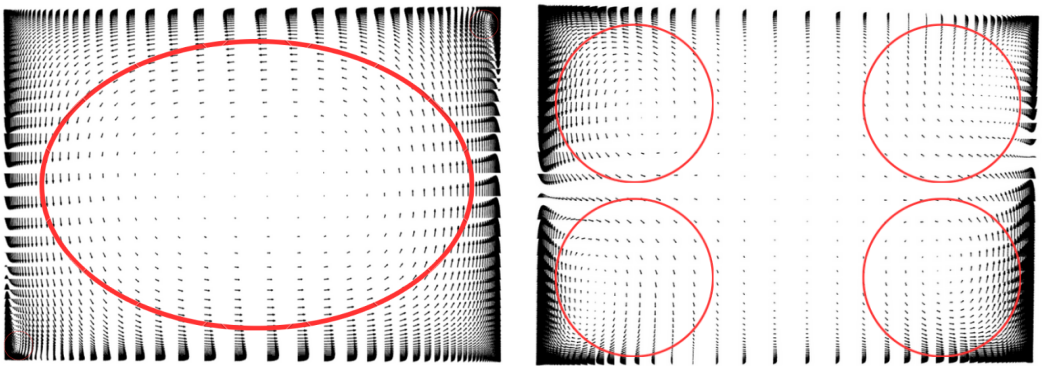
(a) (b)

Figure 3.2: Velocity vector fields on diagonal planes for RBC1 case, with identification of circulation zone in red. $Ra = 1.2 \times 10^7$



(a) (b)

Figure 3.3: Velocity vector fields on diagonal planes for RBC5, with identification of circulation zone in red. $Ra = 13.7 \times 10^7$



(a) (b)

Figure 3.4: Velocity vector fields on diagonal planes for big3RBC, with identification of circulation zone in red. $Ra = 5.9 \times 10^{10}$

Figure 3.2-3.3-3.4 (a) shows the velocity vector field in the plane containing the large-scale circulation, while (b) displays the velocity vector field in the plane orthogonal to it. In sub-figures (a), the large-scale circulation is highlighted by the large ellipse, with associated contra-rotating smaller circulation zones. As the Rayleigh number increases, the prominence of the large-scale circulation also increases, while the size of the contra-rotating circulation zones decreases.
On the orthogonal plane shown in Figure 3.2-3.3-3.4 (b), four circulation zones are observed, each approximately the same size.
For further comparison, the time-averaged streamlines for the RBC5 case are presented below.

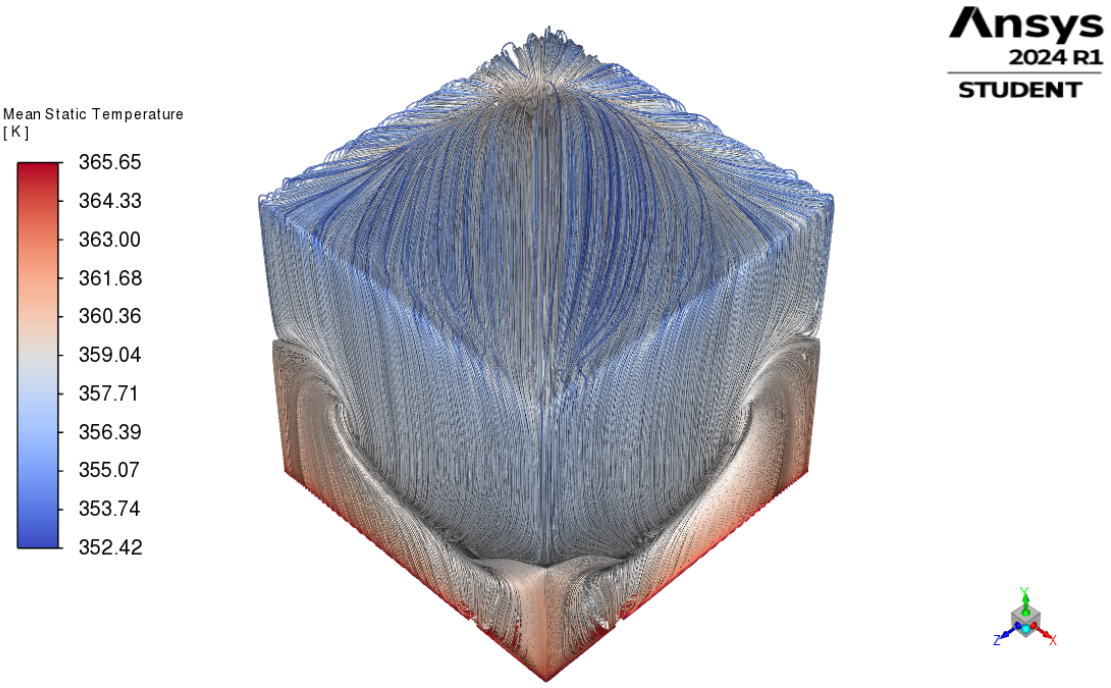


Figure 3.5: Time-average streamline color by the time-average temperature

Chapter 4

Evaporative flux

This chapter is dedicated to evaluate the model against previous results obtained mainly by LES, but also compared to one DNS from the paper by Hay *et al.* (2021) [6]. Simulation parameters will be taken from their study, and each case originally named LES"x" will be renamed FS"x" for clarity purposes. Only simulations with a domain of aspect ratio $\Gamma = 1$ will be considered.

4.1 Set up

The geometry is a cube of height H = 0.045[m]. No-slip boundary conditions are applied to the bottom and side walls, and a free-slip condition is prescribed at the upper wall. Regarding thermal boundary conditions, a fixed temperature T_{low} is imposed at the bottom wall. Side walls are adiabatic. The evaporative flux $\dot{q}'' = \dot{q}''_{\text{conv}} + \dot{q}''_{\text{evap}}$ is imposed at the upper wall. This case was referred to as FS in Section 2.1, thus the choice to rename the cases as such.

The same mesh as used for the RBC cases in Chapter 3 is taken, i.e., a mesh with 56^3 cells and a bias factor of 1.1.

Fluid properties are taken from Table 3.1, and the interface temperature used in the linear profile for the initial condition is taken from the results found in [6].

4.2 Results

The averaging time t_{avg} is taken to be approximately 700 t_{ff}. This is slightly more than double the time used in [6] (except for FS5). Performing longer simulations is not an issue in this configuration, since the mass and volume are not changing. The interface temperature will thus reach a statistical equilibrium, as will the evaporative mass flux $\dot{m}''$. Such an averaging time ensures that the flow has come to statistical equilibrium. This can be checked by evaluating the fluxes at the bottom and upper walls and ensuring they do not differ significantly.

In Table 4.1, we present the imposed bottom wall temperature T_{low} and the corresponding time- and area-averaged interface temperature. The temperature difference between the two walls is denoted as ΔT and is used in the definition of the Rayleigh number Ra of the flow.
The Nusselt number is evaluated at the top wall using Equation 2.24.
The time- and area-averaged evaporative mass flux is also provided. Data from [6] will be referred to as the reference data to avoid repetitive citations and are included in Table 4.1 in parentheses for quick comparison.

The interface temperatures are in very good agreement with the reference data. For FS1, the temperature is a bit higher than expected but lower than the reference data obtained by DNS (only this case was done with DNS).
Only the highest Ra case (FS5) shows a noticeable deviation in ΔT. This deviation is also noticeable for the evaluation of the evaporative mass flux $\dot{m}''$. The difference in the result for $\dot{m}''$ needs to be relativized. The highest difference, which is for FS5, is 0.04 $\mathrm{g\,m^{-2}\,s^{-1}}$. For such an approximate model, this is not a significant difference.

case	T_{low} [K]	T_{int} [K]	ΔT [K]	$Ra \times 10^{-7}$	Nu	$\dot{m}'' \times 10^3$ ($\mathrm{kg\,m^{-2}\,s^{-1}}$)
FS1	320.95	318.13 (318.05)	2.8 (2.9)	1.2 (1.2)	24.8 (24.2)	0.36 (0.36)
FS2	335.15	330.24 (330.15)	4.9 (5.0)	3.0 (3.1)	32.0 (31.7)	0.86 (0.86)
FS3	345.15	338.22 (338.25)	6.9 (6.9)	5.1 (5.1)	36.8 (36.8)	1.47 (1.46)
FS4	355.15	345.59 (345.55)	9.6 (9.6)	8.5 (8.3)	41.7 (41.7)	2.37 (2.36)
FS5	365.65	352.42 (352.95)	13.2 (12.7)	13.7 (13)	46.1 (47.4)	3.73 (3.77)

Table 4.1: Imposed bottom temperature T_{low}, resulting time- and area-average interface temperature T_{int}, temperature difference ΔT and Ra of the flow. Global Nusselt number Nu and the evaporative mass flux $\dot{m}''$. Result in parenthesis are the ones obtained by Hay *et al.* (2021).

An experiment was conducted by Martin & Migot (2019) [24] at the Institut de Radioprotection et de Sûreté Nucléaire (IRSN) using insulated stainless steel tanks of varying shapes and sizes, filled with water and heated from below. During the experiment, they measured the water mass, bulk water temperature, free-surface temperature (using an infrared camera), ambient air temperature, and relative humidity.
The heating was controlled to avoid boiling, and measurements were taken during

both the heating and cooling phases. The resulting data allowed them to plot the evaporative mass flux against the free-surface temperature, which Hay *et al.* (2021) used to validate the evaporation model.

Figure 4.1 was reproduced from [6] using @PlotDigitizer[1] to obtain the experimental curve and error bar measurements, as we were unable to access the paper with the experiment. Our results are plotted as colored crosses, whereas the reference data are plotted using black symbols (refer to the legend of Figure 4.1). Additional points are plotted as black crosses. These refer to simulation setups with fluid properties from Table 5.2 but with a lower T_{low}[2]. Although these additional points are not useful for comparison with the reference data, they provide extra data points, which again, show good agreement with the experimental data.

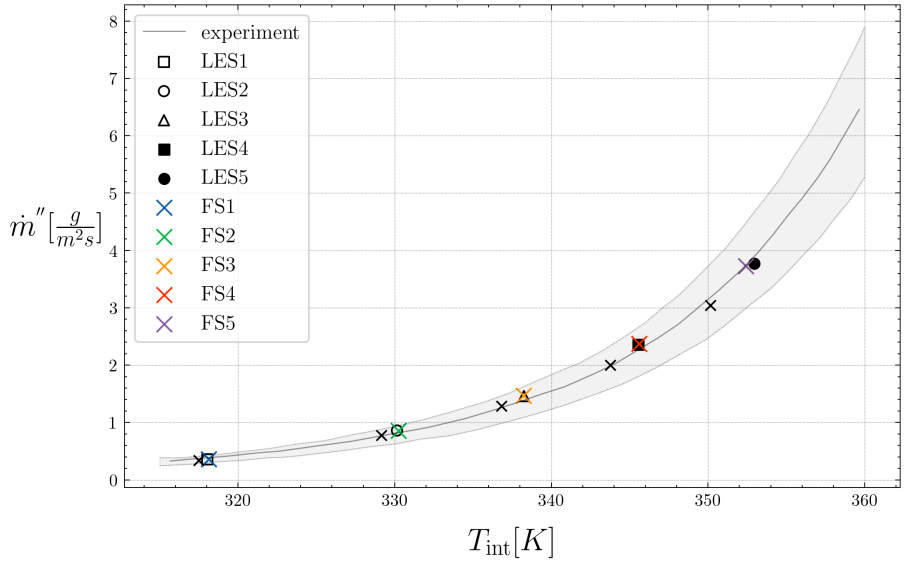


Figure 4.1: Comparison to LES predictions of Hay *et al.* (2021) and experiment of Martin & Migot (2019). Our results are plotted as crosses.

Simulations were also performed under the RBC setup from Chapter 3, with T_{low} and T_{int} fixed from Table 4.1 at the lower and upper walls, respectively. T_{int} was taken from the results of FS simulations to ensure the same Rayleigh number. The Nusselt numbers for these RBC cases are lower than those for the FS cases, as expected, due to the presence of a hydrodynamic boundary layer at the top [25]-[26]. The obtained Nusselt numbers were 16.4, 20.9, 24.1, 27.4, and 30.8.

In Figure 4.2, the global Nusselt number of each case is plotted against its respective Rayleigh number. Under the FS setup, the data fit relatively well with the correlation proposed by Hay & Papalexandris (2020) [25], which specifically accounts for the free-slip and evaporative heat flux boundary conditions.

[1] https://plotdigitizer.com/

[2] This was a mistake, where T_{ref} and T_{low} were confused

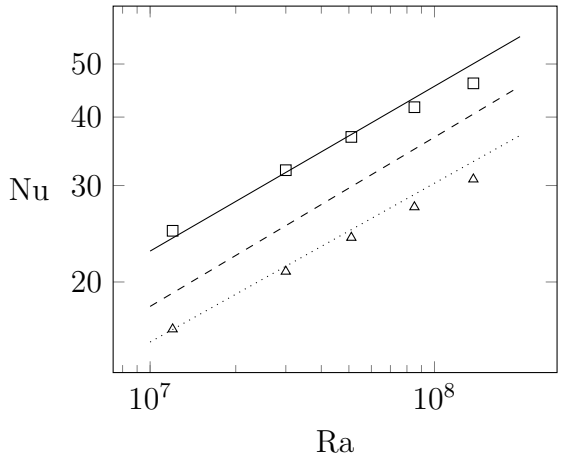


Figure 4.2: Plot of Nu vs. Ra
Solid line (——): $Nu = 0.178Ra^{0.301}$ from Hay & Papalexandris (2020), dashed line (- - -): $Nu = 0.124Ra^{0.309}$ from Niemela *et al.* (2000) and dotted line (·····): $Nu = 0.145Ra^{0.290}$ from Hiroaki & Hiroshi (1980). FS data points are marked with (▫), RBC data points are marked with (▵).

The highest Rayleigh number case, FS5, was also tested with the Reynolds Stress Turbulence Model (RSM) in both RBC and FS configurations. No distinct changes in the evaluation of the Nusselt number were noticed. For FS5, we obtained the same value of 46.1, and for RBC, we obtained 31.03 compared to the 30.8 obtained with the $k - \omega$ SST model.

Time- and area-averaged velocity profiles for the lowest and highest Rayleigh number cases are plotted in Figure 4.3. We clearly see the increase in velocity magnitude as the Rayleigh number increases. The figure also shows an increase in magnitude when the free-slip boundary conditions are applied.
The red and green solid lines in Figure 4.3 represent the velocity magnitude for FS5. The green curve was obtained using the RSM model, showing a slight increase in magnitude close to the lower wall. Aside from this, both turbulence models predict a similar velocity profile.

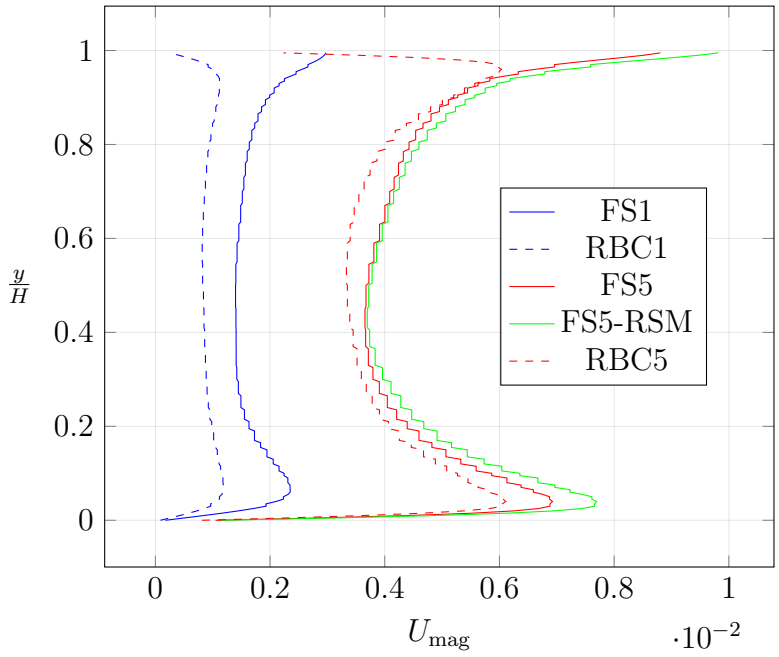


Figure 4.3: Time- and area-average velocity profile. Blue curves: $Ra = 1.2 \times 10^7$, red and green: $Ra = 13.7 \times 10^7$

4.3 Conclusion

We reproduced simulations done by LES and compared various results to assess the effectiveness and correct implementation of the heat flux boundary condition in Fluent. A comparison with the classical Rayleigh-Bénard convection was performed to highlight the differences between the two setups.

- The interface temperature T_{int}, evaporative mass flux $\dot{m}''$, and global Nusselt number Nu were predicted correctly under the FS setup and showed very good agreement with reference cases and, when applicable, to experimental data.
- Under the RBC setup, the global heat transfer Nu was consistently under-predicted.
- No sufficient improvement when using RSM were shown to justify its use against simpler model.

Chapter 5

Evaporative flux and decent

Simulations by Hay *et al.* were not performed for a long enough time to observe significant water loss (less than 3% according to [4]). Because of this, there was no need for a computational domain that varies with time.

In this chapter, simulations will be run for a longer duration to observe significant evaporation, with the dynamic mesh feature enabled. Adding the domain descent will allow simulations to be run for a longer time while remaining physically representative.

5.1 Comparison to another method

Let us first compare a simulation done with the evaporative heat flux and domain descent to the results obtained by an air–water simulation with evaporation across the free surface, as presented by Carlier & Papalexandris (2024) [4].

5.1.1 Set up

The initial geometry is a cuboid with width W=4.5cm and height H_0=2.25cm. The aspect ratio of this domain is thus 2. The bottom wall is at a temperature of $T_{\text{low}} = 315.25$ [K]. The upper wall, interchangeably called the interface, is at a temperature of $T_{\text{int}} = 313.5$ [K], based on their initial guess. The simulation is run for a bit more than 1000 free fall times, as theirs.
Fluid properties used in Fluent are provided in Table 5.1.

case	T_{ref} [K]	ρ [kg m^{-3}]	$\beta \times 10^4$ [K^{-1}]	$\mu \times 10^4$ [Pa s]	k [m^2 s^{-1}]	c_p [J kg^{-1} K^{-1}]	Pr
CAR	314.37	991,8	3.85	6.42	0.642	4177.7	4.18

Table 5.1: Same legend as Table 5.1

The mesh has $48 \times 32 \times 48$ cells in the x, y, and z directions, using approximately 3.4

times fewer points than the coarsest mesh used by Carlier & Papalexandris (2024). The maximum y^+ value is 0.42 and is located on the side walls.

5.1.2 Results

Three cases were investigated by Carlier & Papalexandris (2024) [4], where the initial aspect ratios of the gas column were modified, while the initial aspect ratio of the water column was kept at 2.

Case	T_{low} [K]	T_{int} [K]	ΔT [K]	$\dot{m}'' \times 10^3$ ($\text{kg m}^{-2}\,\text{s}^{-1}$)
CAR	315.25	313.21	2.04	0.24

Table 5.2: Imposed bottom temperature T_{low}, resulting time- and area-average interface temperature T_{int} and evaporative mass flux $\dot{m}''$, the corresponding temperature difference ΔT.

The interface temperature T_{int} is lower than the predicted temperature of each of their cases, and the evaporative mass flux is slightly higher, resulting in a faster domain decrease, as seen in Figure 5.1. This is expected, as a higher evaporative mass flux leads to greater enthalpy loss due to evaporation, thus resulting in a lower temperature [4]. Our results are still in accordance with the experimental data of Martin & Migot but are very close to its lower end.

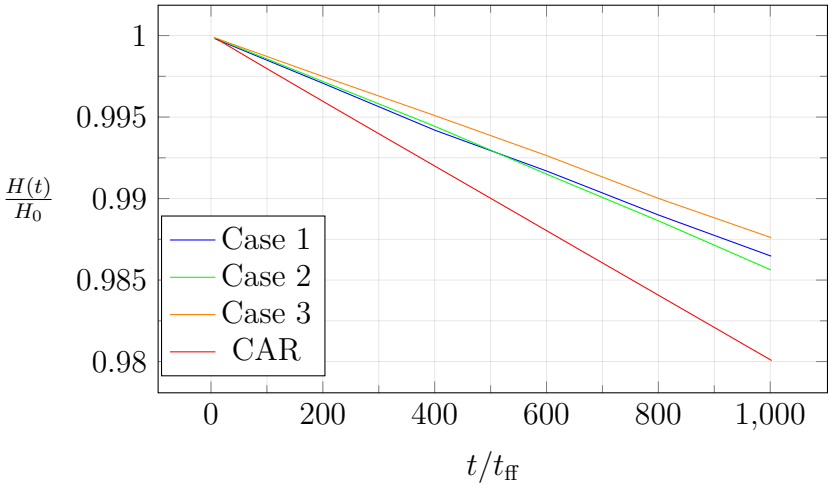


Figure 5.1: Rates of descent of the interface. Lines labeled Case 1, Case 2, and Case 3 were extracted from [4] for comparison using @PlotDigitizer.

Before showing the Ra-Nu plot of the simulation, we present the methodology used to compute the Ra and Nu, as the height H is now a function of time. We use the time- and area-average interface temperature and heat flux, T_{int} and q_{int}, respectively, so that the only variable left depending on time is the domain's height. We then define the Rayleigh and Nusselt numbers as

$$Ra = \frac{\mathrm{g}\beta\Delta T(\mathrm{H}(t))^3}{\nu\alpha} \quad \text{and} \quad Nu = \frac{-q_{\text{int}}}{k\Delta T/(\mathrm{H}(t))} \tag{5.1}$$

where $\Delta T = T_{\text{low}} - T_{\text{int}}$
The continuous curve of Ra-Nu for this simulation is plotted in Figure 5.2, alongside its discrete averaging, divided into five steps. The curved arrow indicates the direction of evolution as time progresses.

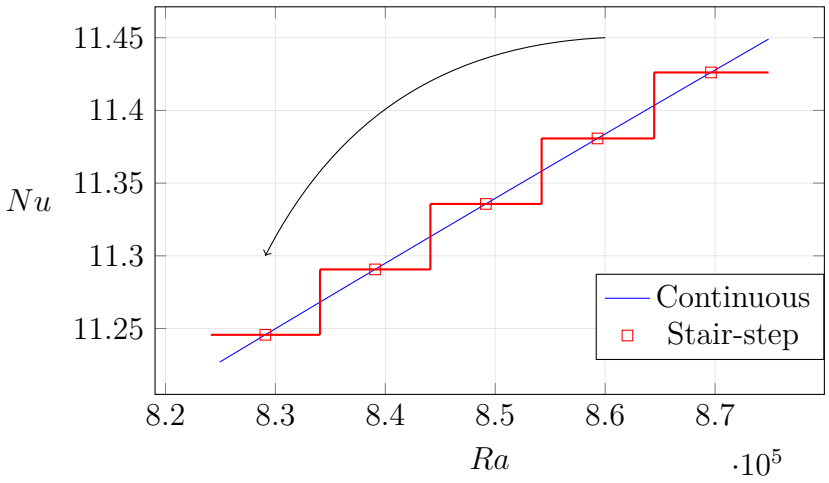
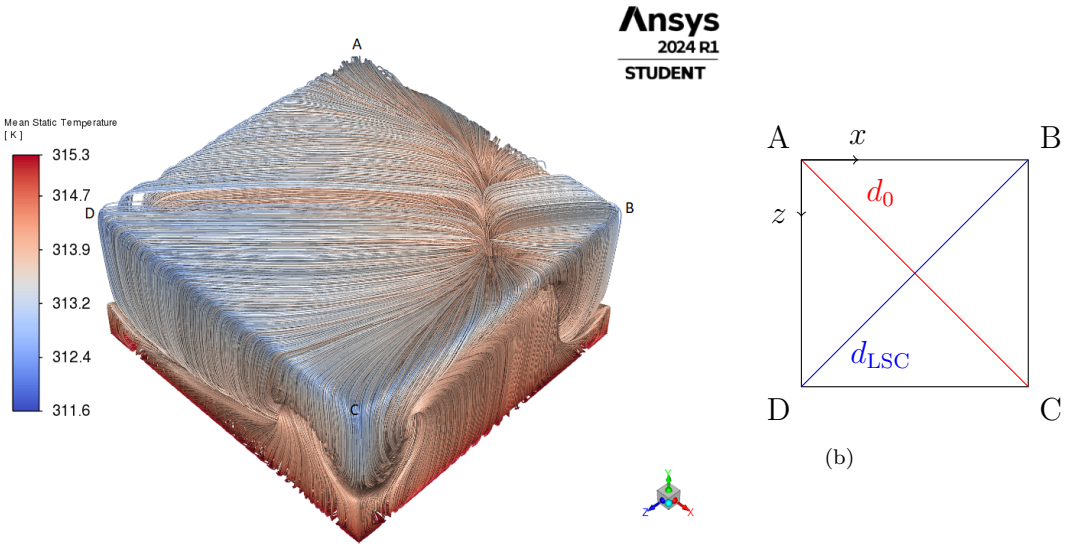


Figure 5.2: Discrete averaging of Nusselt and Rayleigh number.

The streamlines in Figure 5.3a indicate the presence of one steady convection roll or large-scale circulation (LSC). It is well aligned along one diagonal d_{LSC}, as expected for a cubic cell (Sun *et al.* (2008) [27]), and occupies almost the entire cell. The convection pattern appears very symmetric.



(a) (b)

Figure 5.3: (a) Streamlines colored by the time-averaged temperature, (b) Top view showing the two diagonal planes

Figure 5.4 consolidates the symmetric tendency of the velocity field. It also shows a similar temperature field to that in Figure 12 of [4], with the edge of the right upper corner (D) being colder than its diagonally opposite corner (B). We can clearly see the highest temperature spot where the fluid from the bottom rises to the top, located in the lower left corner (B).

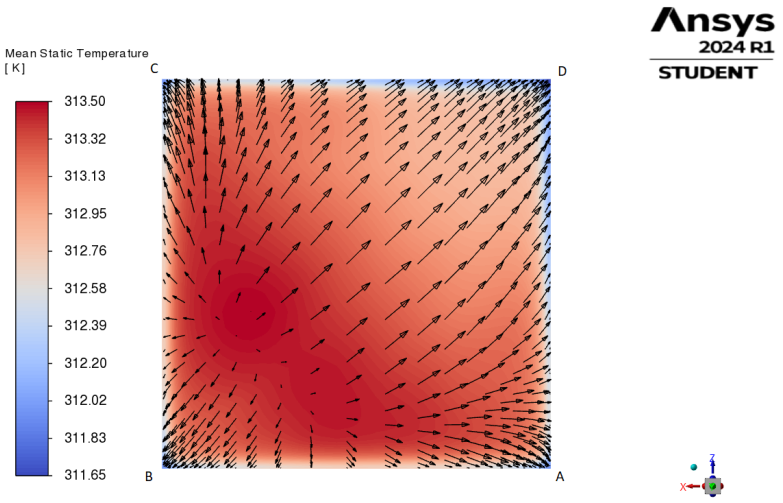
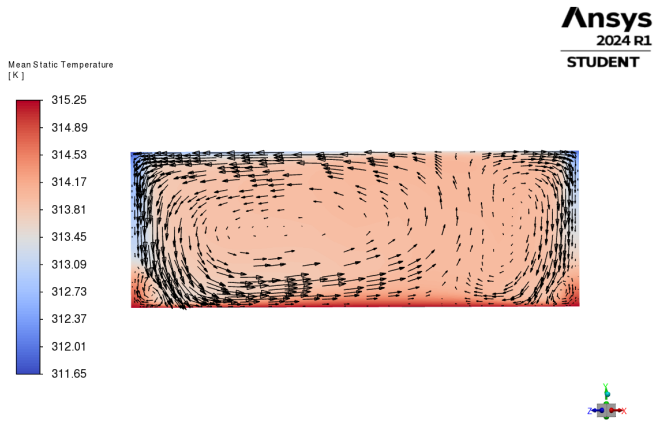


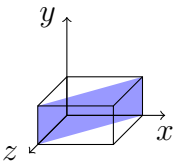
Figure 5.4: Temperature distribution at the interface and superimposed vector plots of the time-averaged velocity

The flow pattern, the large-scale circulation (LSC), and its contra-rotating cells are depicted in Figure 5.5. We also notice additional vortical structures, mostly located in the corners of the cells. In Figure 5.5a, we observe the two main circulations in

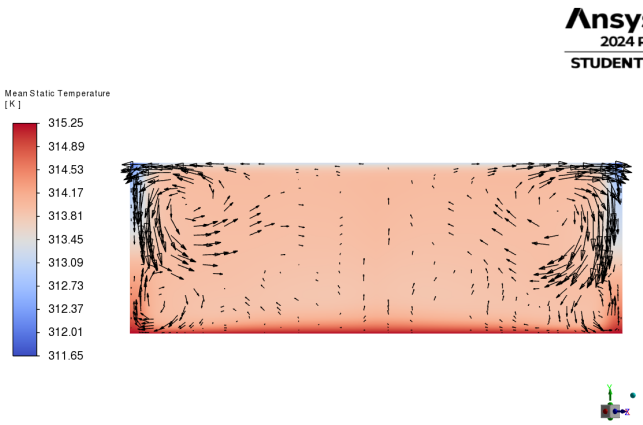
the LSC plane (one dominant) and, in the lower corner, small vortical structures. On the other hand, four structures are contained in the orthogonal plane.



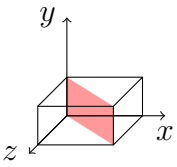
(a)



(b) LSC plane d_{LSC}



(c)



(d) Orthogonal plane d_0

Figure 5.5: (a,c) color plots of the time-averaged temperature and superimposed vector plots of the time-averaged velocity, (b,d) Corresponding diagonal plane

5.1.3 Conclusion

We have reproduced a simulation to compare the evaporation model of Hay *et al.* (2021) [6] and its implementation in Fluent coupled with domain descent against a two-fluid numerical model. Predictions for temperature and evaporative mass flux differ slightly, while remaining in accordance with both the comparison simulations and experimental data. The rate of descent is impacted by the evaporative mass flux and also shows some differences. Nusselt number predictions are slightly lower than the $Nu = 12$ predicted by Carlier & Papalexandris (2024). Flow structures are consistent with their results.

5.2 New simulations

5.2.1 Set up

In the following, cases similar to those in Chapter 4 will be performed, with the addition of domain descent to account for water losses. Two geometries will be used: first, a cube of the same size as in Chapter 4, i.e., $H_0 = 0.045$ [m], to reproduce similar Rayleigh flows. These cases will be called FSE"x", according to the boundary conditions associated with the naming, as detailed in Section 2.1. The other geometry is 10^3 times bigger, i.e., $H_0 = 0.45$ [m]. Cases using this geometry will be called bigFSE"x". The "x" refers to the case numbering in Table 3.1 for the associated fluid properties.
The same mesh used in Chapter 4 is applied to the FSE"x" cases, whereas a mesh with 80^3 cells and a bias factor of 1.15 is used for the bigFSE"x" cases. This bias factor was chosen to ensure proper wall resolution while limiting the number of cells.

5.2.2 Results

The time- and area-averaged interface temperature T_{int} and evaporative mass flux for their respective cases are presented in Table 5.3. The corresponding temperature difference ΔT, used in the calculation of the Rayleigh number, is also provided. The averaging time is given in free fall units, and the percentage of volume lost for each case is included, as well as the maximum y^+ value in the domain.
In addition, the time-averaged descent rate $\dot{H}$, in millimeters per hour (for better interpretation), is also reported. It is not directly related to the Rayleigh number but rather to the interface temperature, so as the evaporative mass flux.

case	T_{low} [K]	T_{int} [K]	ΔT [K]	$\dot{m}'' \times 10^3$ [kg m^{-2} s^{-1}]	ΔV [%]	t_{avg}	y^+_{max}	$\dot{H}$ [$\frac{\text{mm}}{\text{h}}$]
FSE1	320.95	318.07	2.9	0.36	4.64	2986	0.39	1.31
bigFSE1		317.55	3.4	0.31	0.17	433	0.59	1.14
FSE2	335.15	330.15	5.0	0.87	5.15	1968	0.57	3.16
bigFSE2		329.30	5.8	0.73	0.25	380	0.86	2.68
FSE3	345.15	337.87	7.3	1.40	5.14	1519.3	0.87	5.17
bigFSE3		337.13	8.0	1.25	0.33	366	0.94	4.52
FSE4	355.15	345.24	9.9	2.25	5.13	1142	1.42	8.36
bigFSE4		344.26	10.9	1.96	0.26	218	1.02	7.24

Table 5.3: Imposed bottom temperature T_{low}, resulting time- and area-average interface temperature T_{int} and evaporative mass flux $\dot{m}''$, temperature difference ΔT and the percentage of volume lost ΔV. Corresponding averaging time t_{avg} in free fall time unit and maximum y^+ in the domain.

Around 5% of the volume has been lost for the FSE cases, whereas less than 1% of the volume has been lost for the bigFSE cases. Unfortunately, we could not extend the simulation time further, as these simulations were quite time-consuming. For such cases, using the simpler FS setup (without the descent) might have been equivalent.
The same procedure described in Section 5.1.2 is used to calculate the Nusselt and Rayleigh numbers the flow. Results for the FSE1 case are plotted in Figure 5.6, along with a comparison to the correlation proposed by Hay & Papalexandris (2020) [25].

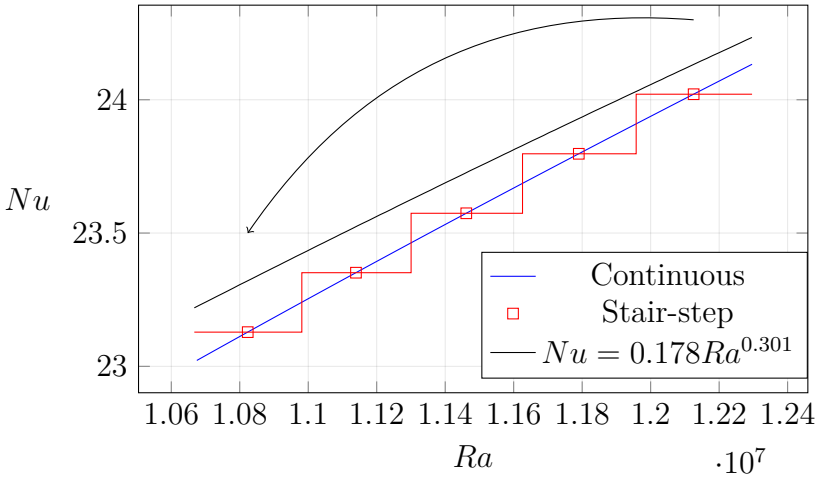
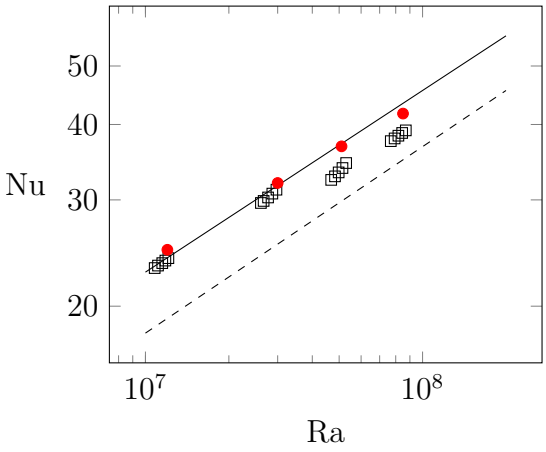
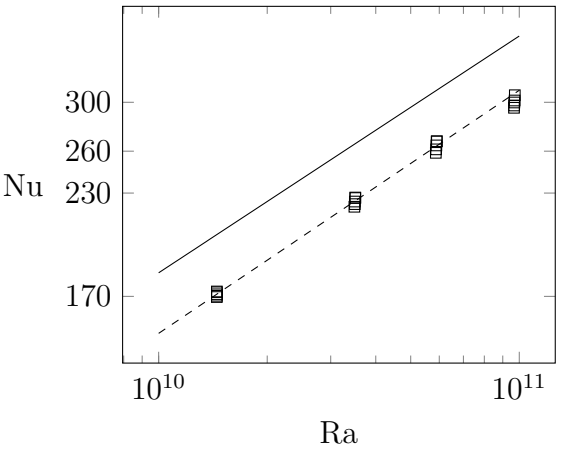


Figure 5.6: Discrete averaging of Nusselt and Rayleigh number. Case: FSE1

This averaging is performed for all cases in Table 5.3, and the results are reported in Figure 5.7a for FSE cases and in Figure 5.7b for bigFSE cases. In Figure 5.7a, noticeable variations in the Nusselt and Rayleigh numbers are observed, resulting from the $\sim 5\%$ domain decrease. We also observe a similar divergence from the correlation as seen in the RBC setup, shown in Figure 3.1a. As in the previous setup, this divergence is associated with an increase in the Rayleigh number. In this case, the divergence might also be linked to the increase in the rate of descent of the domain.
As for the plot of the bigFSE cases in Figure 5.7b, it is hard to explain why the data seem to collapse onto the correlation of Niemela *et al.* (2000), which was developed for an RBC setup. Perhaps the correlation of Hay & Papalexandris (2020), which was developed for a Rayleigh number range of 7.5×10^6 to 1.3×10^8, needs further development for higher Rayleigh numbers.



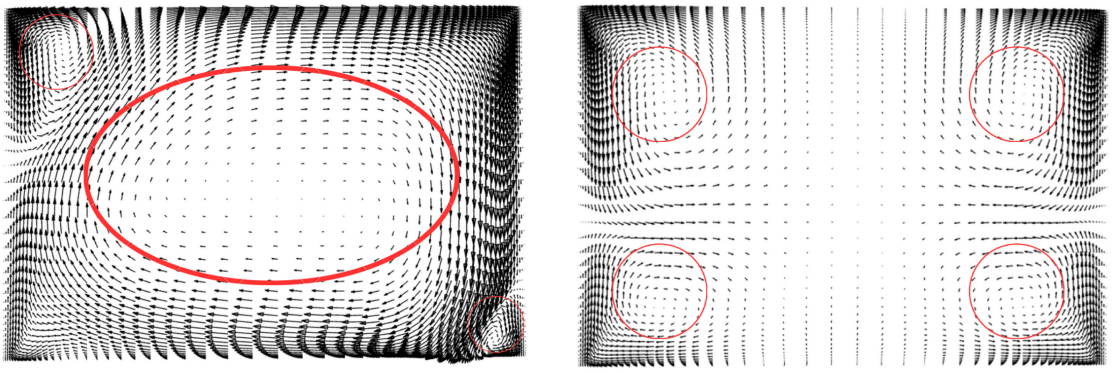
(a) FSE. Red dots are corresponding simulation of Chapter 4, under FS set up.



(b) bigFSE

Figure 5.7: Plot of Nu vs. Ra
Solid line (——): $Nu = 0.178Ra^{0.301}$ from Hay & Papalexandris (2020), dashed line (- - -): $Nu = 0.124Ra^{0.309}$ from Niemela *et al.* (2000).

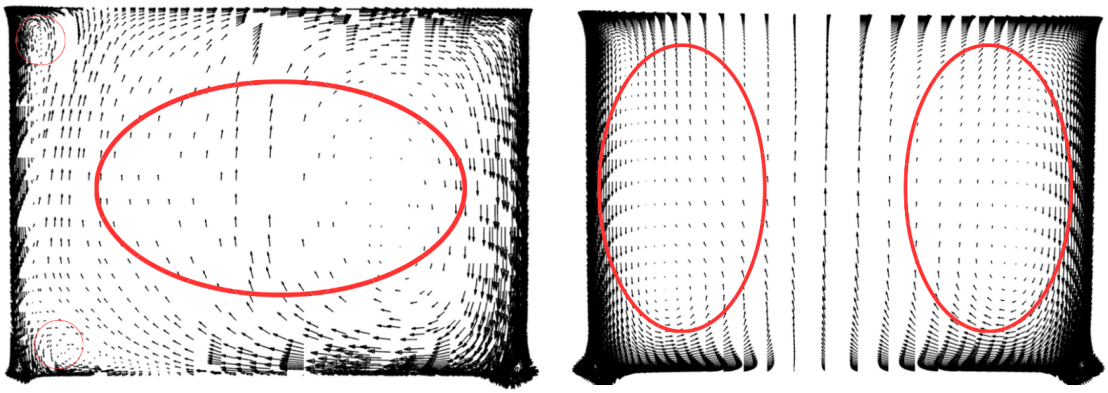
Similar flow structures to those obtained under the RBC setup (see Section 3.3) are recovered for the same range of Rayleigh numbers. See Figure 5.9.



(a) (b)

Figure 5.8: Velocity vector fields on diagonal planes for FSE1, with identification of circulation zone in red.

However, for higher Rayleigh flows, it is more difficult to identify such zones. In Figure 5.9, the red marks are only there to help visualize the main flow characteristics but do not match the circulation zones as well as in the other vector fields shown above. Nonetheless, we can still identify the large-scale circulation in Figure 5.9a. On the orthogonal plane for bigFSE1, Figure 5.9b, we do not observe the four vortical structures in the corners but rather two contra-rotating zones, which differ from the RBC case with a similar Rayleigh number.



(a) (b)

Figure 5.9: Velocity vector fields on diagonal planes for bigFSE1, with identification of circulation zone in red.

To appreciate the instantaneous patterns of the flow, iso-surfaces of instantaneous temperature superimposed with the corresponding velocity vector field for the FSE1 case are shown in Figure 5.10. We notice the ascending and descending plumes, which are aligned with the sense of rotation of the large-scale circulation. This illustrates how thermal plumes rise from the heated bottom surface and cooler fluid descends from the top, driving the convection currents that characterize this system.

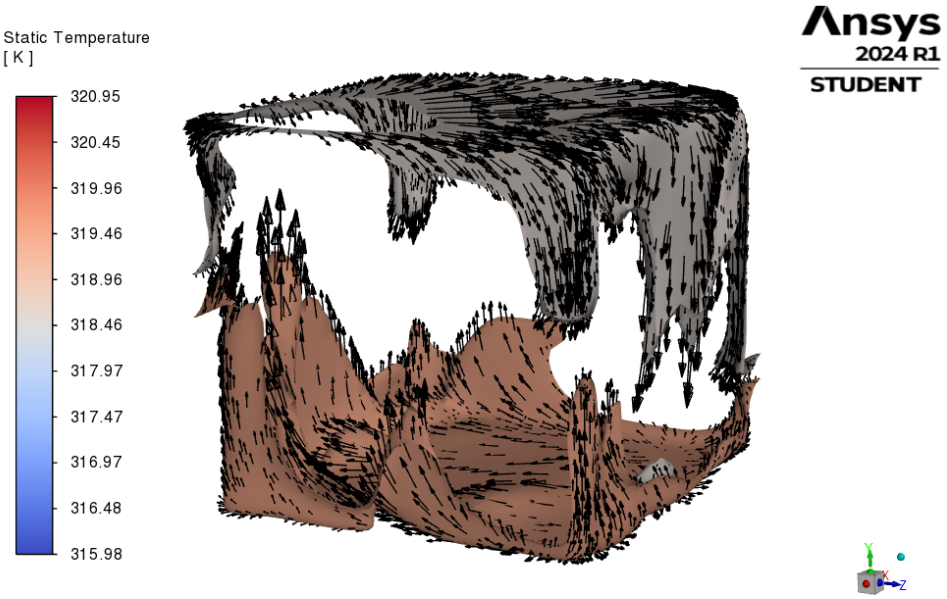


Figure 5.10: Iso-surfaces of instantaneous temperature superpose with the corresponding velocity vector field. Case: FSE1

5.3 Conclusion

We first compared our simulation setup to another method that models evaporation in an air-water system. The results differ slightly but remain consistent with the compared approach, previous results, and experimental data.
Subsequently, new simulations were performed over a range of Rayleigh numbers previously tested under the FS setup (10^7-10^8), as well as over a higher range (10^{10}-10^{11}). The goal was to run simulations with an evaporative heat flux, coupled with domain descent (FSE setup), long enough to observe a volume change. The lower Rayleigh range simulations lost approximately 5% of volume, which is satisfactory, whereas the volume loss in higher Rayleigh flow cases is practically negligible. Longer simulations should have been run to justify the use of a domain descent.
Results were compared to correlations, with some agreement for the lower range of the Ra flows. Surprisingly, higher Ra flows showed a collapse to the classical correlation of Niemela *et al.* (2000).
Finally, flow structures were compared to the classical case of Rayleigh-Bénard convection in a cubic cell.

Conclusion

A simplified model of a nuclear spent fuel pool, in the context of a loss of cooling accident—specifically, a cubic cell heated from below—was introduced.
The goal of this work was to simulate flows using an evaporation model coupled with domain decrease, to account for water loss to the environment.
We first presented various flows under the classical setup of Rayleigh-Bénard convection. Nusselt number estimations were consistently under-predicted, especially as the Rayleigh number increased; however, flow structures matched results found in the literature.
Next, a more realistic setup was introduced. The top wall was modeled as a free-slip surface with a non-uniform heat flux to represent evaporative cooling. Results were compared with existing LES data and experimental data, showing very good agreement.
Finally, domain descent was incorporated into the setup to account for water loss to the environment. We first compared our setup to an air–water simulation with evaporation across the free surface, conducted by Carlier & Papalexandris (2024). Although the two different setups yielded different results, they were close to one another and in agreement with experimental data. Subsequently, previous simulations were run for a longer duration with descent, achieving a 5% volume reduction. Nusselt number predictions deviated from the correlation of Hay & Papalexandris (2020) as the Rayleigh number increased. Additionally, higher Rayleigh number flows were simulated, with their Nusselt numbers aligning with the Rayleigh-Bénard convection correlation of Niemela *et al.* (2000).

Improvement

Some modifications to the setup could enhance global performance. These include:

- Using hyperbolic tangent stretching in the mesh. This type of stretching maintains a more constant resolution in the bulk, providing better control over the bulk resolution compared to a bias factor applied to all cells.
- Initializing simulations with the evaporative heat flux and domain descent (FSE) using the last time step from similar simulations with only the evaporative flux (FS) could help mitigate transient effects. Similarly, data sampling could be delayed until after some time to further minimize transient effects.

- Optimizing the usage of cluster memory for faster simulations.

Other interesting possibilities for resolving such flows include using wall functions for high Rayleigh number flows, which would reduce grid requirements. Additionally, Detached Eddy Simulation (DES), where RANS models are employed in the boundary layer and LES treatment is applied in the bulk, could offer higher fidelity simulations for higher Rayleigh flows [28].

I would like to acknowledge the assistance of ChatGPT in refining the grammar and orthography of this document. Additionally, ChatGPT provided support in generating and formatting plots, which contributed to the clarity and presentation of the results. Furthermore, assistance was also received in developing and troubleshooting user-defined functions.

Appendix A

User defined functions

```
#include "udf.h"
#include "math.h"
#include <stdio.h>
#include <stdlib.h>

#define T_cold 318.05   // upper wall temperature estimate  TO CHANGE FOR THE CASE
       used in DEFINE_INIT
#define T_hot 320.95    // lower wall temperature estimate  TO CHANGE FOR THE CASE
       used in DEFINE_INIT
#define W 0.045          // cube size                        TO CHANGE FOR THE CASE
        used in DEFINE_INIT, DEFINE_PROFILE, DEFINE_CG_MOTION
#define Beta 4.3e-4     // thermal expansion coef           TO CHANGE FOR THE CASE
       used in routine, MASSOUT

// DO NOT CHANGE
// Gaseous mixture properties far from the interface, used in routine, MASSOUT
#define T_inf 298.15        // ambient temperature
#define p0 101325.0         // ambient pressure
#define pv_inf 1270.0       // water vapour partial pressure far from the interface
#define rho_inf 1.18        // gaseous mixture density far from the interface
#define beta_inf 3.34*1e-3  // thermal expansion coefficient of the air at T_inf
#define pc 22.064e6         // 2.2a  (Wagner and Pruss 2002)
#define Tc 647.096          // 2.2b  (Wagner and Pruss 2002)
```

Listing A.1: Import, far field conditions and simulation parameters

```
double routine(double T)
{
    // for eq. numbering refer to Hay et al. 2021

    //eq12
    double Tr = T/Tc;
    double tau = 1.0 - Tr;
    // 2.5  J. Phys. Chem. Ref. Data, Vol. 31, No. 2, 2002 (Wagner and Pruss 2002)
    double a1=-7.85951783, a2=1.84408259, a3=-11.7866497, a4=22.6807411, a5
    =-15.9618719, a6=1.80122502;
    double pv_int = pc*exp((a1*tau + a2*pow(tau, 1.5) + a3*pow(tau, 3.0) + a4*pow(
    tau, 3.5) + a5*pow(tau, 4.0) + a6*pow(tau, 7.5))/Tr); // 2.5  (Wagner and Pruss
     2002)

    //eq13
    double Mw = 18.01528e-3;
    double Ma = 28.9647e-3;
    double Yv_int = (pv_int*Mw)/(pv_int*Mw + (p0-pv_int)*Ma);

    //eq14
    double R = 8.314;
    double rho_int = (Mw*pv_int + Ma*(p0 - pv_int))/(R*T);

    double Tf = (T+T_inf)/2;
    double pv_f = (pv_int+pv_inf)/2;
    // double Yv_f = pv_f*Mw/(pv_f*Mw + (p0 - pv_f)*Ma);
    double rho_f = (Mw*pv_f + Ma*(p0 - pv_f))/(R*Tf);

    //eq15
    double xw = pv_f/p0;
    double xa = 1.0 - xw;

    //eq16
    double mu_w = 0.000133401 + -1.5184768e-06*Tf +  6.786288e-09*Tf*Tf + -1.323448
    e-11*Tf*Tf*Tf + 9.706705e-15*Tf*Tf*Tf*Tf;
    double mu_a = -4.843e-7 + 8.507e-8*T - 9.81833e-11*T*T + 1.0152e-13*T*T*T -
    4.98623e-17*T*T*T*T;
    // The data used to make the fit was taken from: A heat transfer Textbook, John
    . H. Lienhard. (Given by J. Marichal)

    double phi_aa = 1.0;
    double phi_aw = pow( (1 + sqrt((mu_a/mu_w) * sqrt(Mw/Ma)) ), 2.0 )/sqrt(8*(1 +
    Ma/Mw));
    double phi_wa = pow( (1 + sqrt((mu_w/mu_a) * sqrt(Ma/Mw)) ), 2.0 )/sqrt(8*(1 +
    Mw/Ma));
    double phi_ww = 1.0;
    double mu_f = (xa*mu_a)/(xa*phi_aa + xw*phi_aw) + (xw*mu_w)/(xa*phi_wa + xw*
    phi_ww);
    double nu_f = mu_f/rho_f;

    // double lambda_int = 0.6065*(-1.48445 + 4.12292*(T/298.15) - 1.63866*(T
    /298.15)*(T/298.15)); // eq 4 https://srd.nist.gov/jpcrdreprint/1.555963.pdf

    double Tstar=647.3, rho_star=317.7;
    double Tp = T/Tstar, rhop=rho_f/rho_star;
    double lbda_0 = sqrt(Tp)* (0.0102811 + 0.0299621*Tp + 0.0156146*pow(Tp,2.0) -
    0.00422464*pow(Tp,3.0) ); // eq 3.20 https://srd.nist.gov/jpcrdreprint
    /1.555718.pdf
    double lbda_1 = - 0.397070+ 0.400302*rhop +  1.060000*exp( -0.171587*pow(rhop +
     2.392190, 2.0) ); // eq 3.22
    double DTp = (1.0-Tp)+0.00308976;
    double Q = 2.0 + 0.0822994*pow(DTp,-0.6);
    double S = 10.0932*pow(DTp, -0.6);
    double lbda_2 = (0.0701309*pow(Tp,-10.0) + 0.0118520)*pow(rhop, 1.8)*exp
    (0.642857*(1.0 - pow(rhop, 2.8)))
                  + 0.00169937 * S * pow(rhop, Q) * exp(Q/(1.0+Q)) * (1.0 - pow(
    rhop, 1.0+Q))
```

```
                + -1.0200 * exp(-4.11717*pow(Tp,1.5) + - 6.17937*pow(rhop, -5.0))
    ; // eq 3.23
    double lambda_w = lbda_0 + lbda_1 + lbda_2; // eq 3.17

    double Tc_air = 132.52;
    double C1=33.9729025, C2=-164.702679, C3=262.108546, C4=-21.5346955, C5
    =-443.455815, C6=607.339582, C7=-368.790121, C8=111.296674, C9=-13.4122465;
    double Tr_air = Tf/Tc_air;
    double lambda_a = (C1/Tr_air
                     + C2*pow(Tr_air, -2./3.)
                     + C3*pow(Tr_air, -1./3.)
                     + C4
                     + C5*pow(Tr_air,  1./3.)
                     + C6*pow(Tr_air,  2./3.)
                     + C7*Tr_air
                     + C8*pow(Tr_air,  4./3.)
                     + C9*pow(Tr_air,  5./3.)
                     )*4.358e-3; // eq5 https://srd.nist.gov/jpcrdreprint/1.555749.
    pdf

    phi_aw = pow( (1 + sqrt(lambda_a/lambda_w * sqrt(Mw/Ma))),2.0 )/sqrt(8*(1 + Ma/
    Mw));
    phi_wa = pow( (1 + sqrt(lambda_w/lambda_a * sqrt(Ma/Mw))),2.0 )/sqrt(8*(1 + Mw/
    Ma));
    double lambda_f = (xa*lambda_a)/(xa*phi_aa + xw*phi_aw) + (xw*lambda_w)/(xa*
    phi_wa + xw*phi_ww);


    double cp_w = (-203.6060 + 1523.290*(T/1000) - 3196.413*pow((T/1000),2.0) +
    2474.455*pow((T/1000),3.0) + 3.855326*pow((T/1000),-2.0))/Mw; // https://
    webbook.nist.gov/cgi/cbook.cgi?ID=C7732185&Units=SI&Mask=7&Type=JANAFL&Table=on
    #JANAFL
    double cp_a = 0.0;
    double N[12] = {0.0, 3.490888032, 2.395525583e-6, 7.172111248e-9, -23.11541310e
    -13, 0.223806688,  0.791309509, 0.212236768,  0.197938904, 3364.011, 2242.45,
    11580.4};
    double u = N[9]/Tf, v=N[10]/Tf, w=N[11]/Tf;
    for (int i = 1; i < 5; i++)
    {
        cp_a += N[i]*pow(Tf, i-1.0) + N[5]*pow(Tf, -1.5) + N[6]*u*u*exp(u)/pow((exp
    (u)-1.0), 2.0) + N[7]*v*v*exp(v)/pow((exp(v)-1.0), 2.0) + (2.0/3.0)*N[8]*w*w*
    exp(-w)/pow(((2.0/3.0)*exp(-w) +1), 2.0);
    }
    cp_a *= R/Ma; // eq18 https://srd.nist.gov/jpcrdreprint/1.1285884.pdf

    phi_aw = pow( (1 + sqrt(cp_a/cp_w * sqrt(Mw/Ma))),2.0 )/sqrt(8*(1 + Ma/Mw));
    phi_wa = pow( (1 + sqrt(cp_w/cp_a * sqrt(Ma/Mw))),2.0 )/sqrt(8*(1 + Mw/Ma));
    double cp_f = (xa*cp_a)/(xa*phi_aa + xw*phi_aw) + (xw*cp_w)/(xa*phi_wa + xw*
    phi_ww);

    double kappa_f = lambda_f/(rho_f*cp_f);

    //eq17
    double Df = 1.87e-10*pow(Tf, 2.072)/(p0/101325.0);

    //eq18
    double Rac = 9.81*(rho_inf-rho_int)*pow(W,3.0)/(Df*mu_f);

    //eq19
    double Sc = mu_f/(rho_f*Df);
    double Sh = 0.23*pow(Sc,0.333)*pow(Rac, 0.321);

    //eq21
    double Yv_inf = (pv_inf*Mw)/(pv_inf*Mw + (p0-pv_inf)*Ma);
    double Bm = (Yv_inf - Yv_int)/(Yv_int - 1);

    //eq22
```

```
    double mdot_pp = Sh*rho_f*Df/W * log(1 + Bm);

    //eq23
    double hlv = 2256.4*pow((Tc-T)/(Tc - 373.15), 0.283);

    //eq26
    double Rat = 9.81*beta_inf*(T-T_inf)*pow(W,3.0)/(kappa_f*nu_f);
    double Nu_t = 0.54*pow(Rat,0.25);

    //eq25
    double h = lambda_f*Nu_t/W;

    //eq24
    double q_conv = h*(T - T_inf);
    double q_evap = mdot_pp*hlv*1000;

    double q = q_conv + q_evap;

    return -q;
}
```

Listing A.2: Compute the evaporative heat flux

```
double MASSOUT(double T)
{
    // used in DEFINE_CG_MOTION
    //eq12
    double Tr = T/Tc;
    double tau = 1.0 - Tr;
    // 2.5  J. Phys. Chem. Ref. Data, Vol. 31, No. 2, 2002 (Wagner and Pruss 2002)
    double a1=-7.85951783, a2=1.84408259, a3=-11.7866497, a4=22.6807411, a5
    =-15.9618719, a6=1.80122502;
    double pv_int = pc*exp((a1*tau + a2*pow(tau, 1.5) + a3*pow(tau, 3.0) + a4*pow(
    tau, 3.5) + a5*pow(tau, 4.0) + a6*pow(tau, 7.5))/Tr); // 2.5  (Wagner and Pruss
     2002)

    //eq13
    double Mw = 18.01528e-3;
    double Ma = 28.9647e-3;
    double Yv_int = (pv_int*Mw)/(pv_int*Mw + (p0-pv_int)*Ma);

    //eq14
    double R = 8.314;
    double rho_int = (Mw*pv_int + Ma*(p0 - pv_int))/(R*T);

    double Tf = (T+T_inf)/2;
    double pv_f = (pv_int+pv_inf)/2;
    double rho_f = (Mw*pv_f + Ma*(p0 - pv_f))/(R*Tf);

    //eq15
    double xw = pv_f/p0;
    double xa = 1.0 - xw;

    //eq16
    double mu_w = 0.000133401 + -1.5184768e-06*Tf +  6.786288e-09*Tf*Tf + -1.323448
    e-11*Tf*Tf*Tf + 9.706705e-15*Tf*Tf*Tf*Tf;
    double mu_a = -4.843e-7 + 8.507e-8*T - 9.81833e-11*T*T + 1.0152e-13*T*T*T -
    4.98623e-17*T*T*T*T; // The data used to make the fit was taken from: A heat
    transfer Textbook, John. H. Lienhard. (From J. Marichal)

    double phi_aa = 1.0;
    double phi_aw = pow( (1 + sqrt((mu_a/mu_w) * sqrt(Mw/Ma)) ), 2.0 )/sqrt(8*(1 +
    Ma/Mw));
    double phi_wa = pow( (1 + sqrt((mu_w/mu_a) * sqrt(Ma/Mw)) ), 2.0 )/sqrt(8*(1 +
    Mw/Ma));
    double phi_ww = 1.0;
    double mu_f = (xa*mu_a)/(xa*phi_aa + xw*phi_aw) + (xw*mu_w)/(xa*phi_wa + xw*
```

```
    phi_ww);

    //eq17
    double Df = 1.87e-10*pow(Tf, 2.072)/(p0/101325.0);

    //eq18
    double Rac = 9.81*(rho_inf-rho_int)*pow(W,3.0)/(Df*mu_f);

    //eq19
    double Sc = mu_f/(rho_f*Df);
    double Sh = 0.23*pow(Sc,0.333)*pow(Rac, 0.321);

    //eq21
    double Yv_inf = (pv_inf*Mw)/(pv_inf*Mw + (p0-pv_inf)*Ma);
    double Bm = (Yv_inf - Yv_int)/(Yv_int - 1);

    //eq22
    double mdot_pp = Sh*rho_f*Df/W * log(1 + Bm);

    return mdot_pp;
}
```

Listing A.3: Compute the evaporative mass flux

```
DEFINE_INIT(my_init_func,d)
{
    cell_t c;
    Thread *t;
    real xc[ND_ND];

    /* loop over all cell threads in the domain  */
    thread_loop_c(t,d)
    {
        /* loop over all cells  */
        begin_c_loop_all(c,t)
        {
            C_CENTROID(xc,c,t);
            C_T(c,t) = (T_cold-T_hot)/W * xc[1] + T_hot;
        }
        end_c_loop_all(c,t)
    }
}
```

Listing A.4: Initialization of the temperature field

```
DEFINE_PROFILE(evap_heat_flux, t, i)
{
    face_t f;
    real mdot_local = 0.0;
    real mdot_global = 0.0;

    #if !RP_HOST
        Thread* thread;
        real NV_VEC(A);
        thread = Lookup_Thread(t->domain, t->id);

        begin_f_loop(f, thread)
        {
            F_PROFILE(f, t, i) = routine(F_T(f, t));

            F_AREA(A, f, thread); // Get the face area vector
            real area_mag = NV_MAG(A); // Magnitude of the face area vector
            real mass_flow_rate = MASSOUT(F_T(f, t));

            mdot_local += area_mag * mass_flow_rate;
        }
        end_f_loop(f, thread)
    #endif /* !RP_HOST */

    /* Perform a global summation of mdot_local across all nodes */
    #if RP_NODE
        // Message("Node %d: Local mdot = %f\n", myid, mdot_local);
        mdot_global = PRF_GRSUM1(mdot_local);
    #else
        mdot_global = mdot_local;
    #endif /* RP_NODE */

    /* Normalize the global mass flow rate */
    mdot_global /= (W * W);

    #if !RP_HOST
        FILE *str;
        str = fopen("UDF_output.txt", "a");
        if (I_AM_NODE_ZERO_P) /* Only node 0 writes to the file */
        {
            real current_time = CURRENT_TIME;
            fprintf(str, "Node %d: %f, %f\n", myid, current_time, mdot_global);
            fclose(str);
        }
    #endif /* !RP_HOST */
}
```

Listing A.5: Impose the evaporative heat flux

```
DEFINE_CG_MOTION(decent,dt,vel,omega,time,dtime)
{
    face_t f;
    cell_t c;
    real NV_VEC(A);
    real VOL_OUT_global = 0.0;
    real VOL_OUT_local = 0.0;
    real area_mag, mass_flow_rate, density;

    #if !RP_HOST
        /* reset velocities */
        NV_S(vel, =, 0.0);
        NV_S(omega, =, 0.0);

        Thread *t;
        if (!Data_Valid_P())
        return;

        t = DT_THREAD(dt); /* get the thread pointer for which this motion is
    defined */

        /*
        F_AREA(A,f,t): face normal vector returned from F_AREA
        NV_MAG(A): The utility NV_MAG computes the magnitude of a vector. This is
    taken as the square root of the sum of the squares of the vector components.
    https://www.afs.enea.it/project/neptunius/docs/fluent/html/udf/node111.htm
        C_R(c,t): density https://www.afs.enea.it/project/neptunius/docs/fluent/
    html/udf/node97.htm
        MASSOUT(C_T(f, t)): from "my" routine, get the mass flow rate per unit area
     of fluid vapor
        */
        begin_f_loop(f,t)
        {
            F_AREA(A,f,t); // Get the face area vector
            area_mag = NV_MAG(A); // Magnitude of the face area vector

            mass_flow_rate = MASSOUT(F_T(f, t));  // Mass flow rate per unit area
    from routine

            c = F_C0(f, t); // Get the cell pointer associated with the face

            density = C_R(c, THREAD_T0(t)); // C_R(c, t): Density at the cell

            VOL_OUT_local += dtime * area_mag * mass_flow_rate / density;
        }
        end_f_loop(f,t)
    #endif /* !RP_HOST */

    /* Perform a global summation of mdot_local across all nodes */
    #if RP_NODE
        VOL_OUT_global = PRF_GRSUM1(VOL_OUT_local);
    #else
        VOL_OUT_global = VOL_OUT_local;
    #endif /* RP_NODE */

    vel[1] = -(VOL_OUT_global/pow(W,2.0))/dtime;
}
```

Listing A.6: Impose the top wall velocity

Appendix B

Fluent case set up

This appendix is a brief tutorial on how to set up a Fluent case to perform simulations with the evaporative heat flux and a domain decrease.

Assuming the mesh (and geometry) to be already loaded in a Fluent case. Launch Fluent in solution mode. You should be greeted with the following screen:

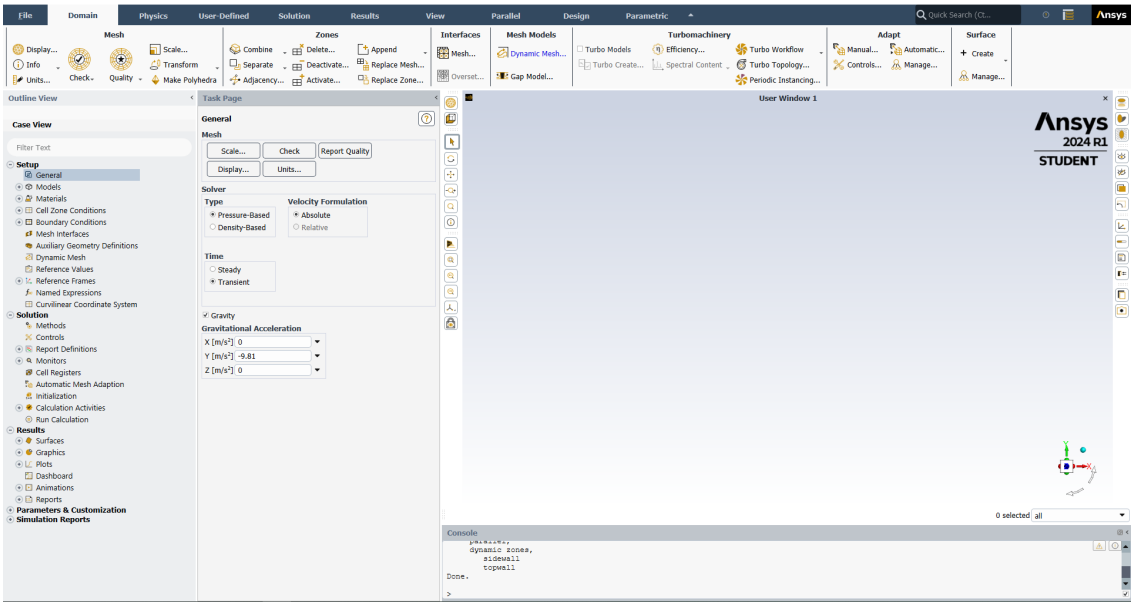


Figure B.1: Start screen

1: Activate gravity and set it in the desired direction and select Transient time formulation.

2: Inside `Physics`, under the `Operating conditions`, set the `Operating temperature` to T_{ref}. This will be the reference temperature used by the Boussinesq model.

3: Going into the `User-Defined tab`, click on `Functions/Compiled`. Window of Figure B.2 should appear.

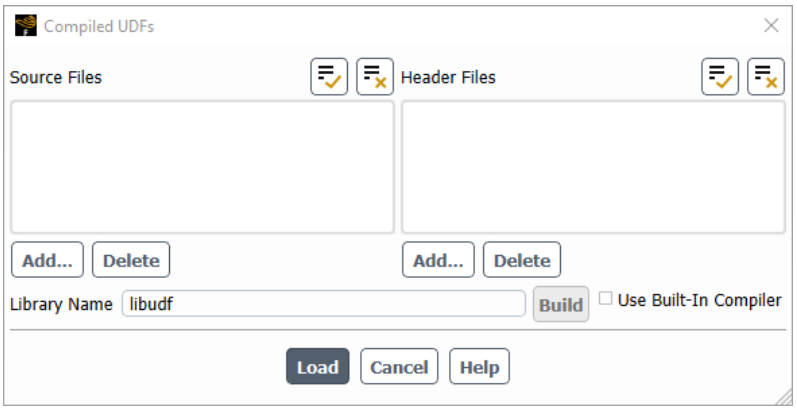


Figure B.2: Complied UDFs

Add the C file containing the UDFs. You can load multiple file. Then Build and finally, Load. The UDFs are now be linked to Fluent.

The next steps all take place in the `Outline View`.

4: Under `Models`, click on `Energy`, and ticks the box to activate the use of the energy equation.

5: Under `Models`, chose your turbulence model

6: Under `Materials/Fluid`, you can create a new fluid, here for example water-liquid. Set the density to Boussinesq, and enter the fluid properties. See Figure B.3.

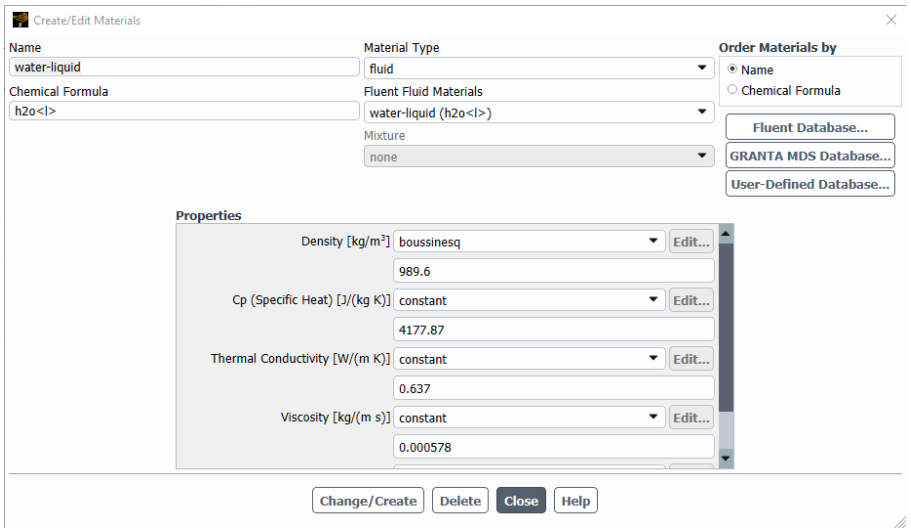


Figure B.3: Create/Edit Materials

7: Under `Cell Zone Conditions/Fluid/solid`, make sure the you use your fluid of choice. Here, water-liquid is chosen. This step is required, otherwise the default

fluid, air, will be used.

8: Under `Boundary Conditions/Wall`, find the wall corresponding to the upper wall. Since the UDFs have been complied already, you can select your custom heat flux, as on Figure B.4:

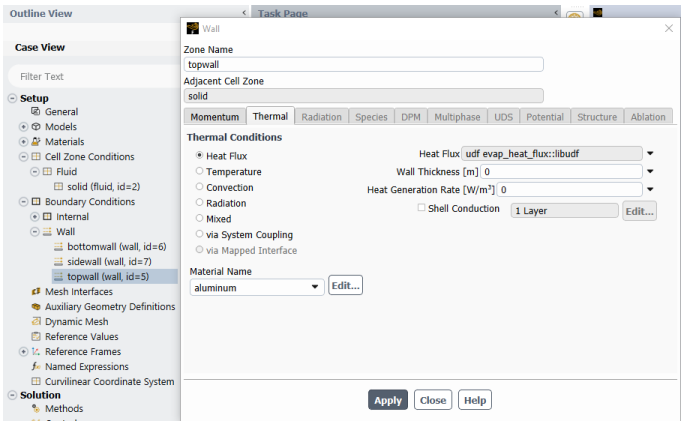


Figure B.4: Link the evaporative heat flux

Set up accordingly the rest of the boundary conditions, on all walls.

9: Under `Dynamic Mesh`, tick the box, use the Mesh Methods of your choice. Here, Spring-Based Smoothing method is chosen. Set the top wall to Rigid Body type, and located the UDF under `Motion UDF/Profile`. For the side walls, under Type, select deforming. See Figure B.5

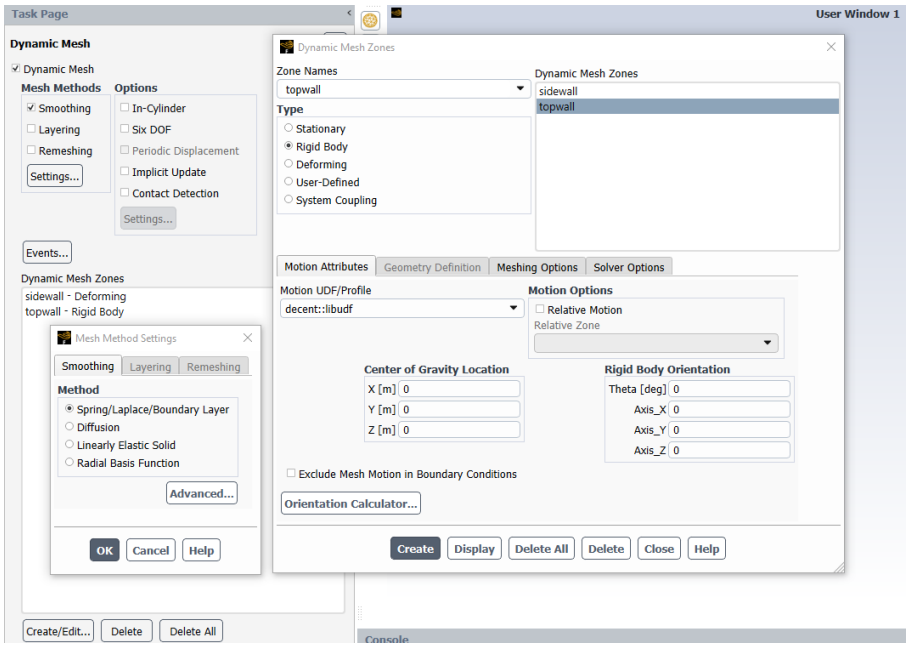


Figure B.5: Dynamic mesh

The final step is outside the `Outline view`.
10: Before initialization, go to `User-Defined Function Hooks`, and locate your temperature field initialization function, as seen on the Figure B.6.

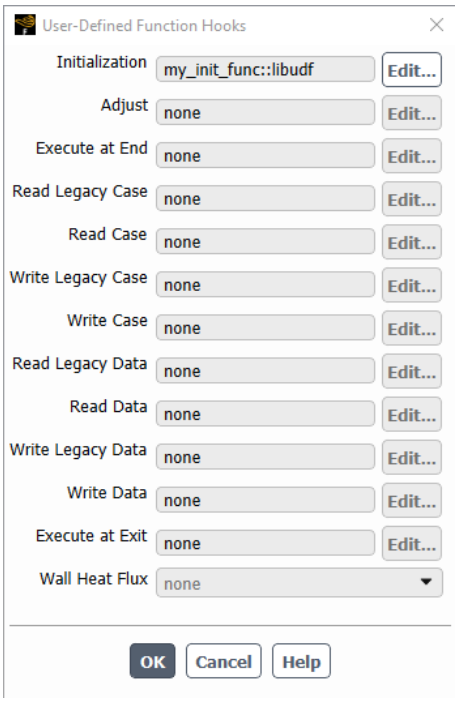


Figure B.6: Function hook

You can now initialize the case. After the methods have been chosen, residuals set up accordingly, the time step chosen (fixed) or the method for determining the time step chosen, you are now ready to run the calculations. Check the data sampling options to record time average and save the case. `Calculation Activities` The case has been initialize and is now ready to be sent on a cluster for calculation. See Appendix C.

Appendix C

Launching simulations on CÉCI clusters

After consolidating all the code snippets provided in Appendix A into a single source file for the UDF, named here `udfs.c`, and initializing the case in Fluent[†], we are ready to launch the simulation on the clusters.

We begin by connecting to the clusters and creating a new directory. Inside that directory, named here `Simulation folder`, should be the journal file, bash script, case file, and the source file for the UDF, as depicted by the directory tree of Figure C.1.

Within this folder containing the required files, we execute the following command to launch the calculations:

```
sbatch run_slurm.txt
```

```
pwd
(rpsetvar 'cache-flush/target/reduce-by-mb 12288)
(flush-cache) # this might not be the most adapted command
/define/user-defined/compiled-functions compile libudf yes myudfs.c "" ""
/define/user-defined/compiled-functions load libudf
/file/read-case CaseFile.cas.h5

/file/confirm-overwrite? no
/file/auto-save/data-frequency 100
/file/auto-save/case-frequency each-time
/file/auto-save/retain-most-recent-files yes
/file/auto-save/max-files 1
/solve/init/init
/solve/set/time-step 0.05
/solve/dual-time-iterate 40000 50
/file/write-case-data CaseFile_final.cas.h5
/exit
yes
```

Listing C.1: Bash script, `run_slurm.txt`

[†]Initialization can be entirely managed within the journal file

```
#!/bin/bash
#SBATCH --job-name=CaseName
#SBATCH --mail-user=dorian.barre@student.uclouvain.be
#SBATCH --mail-type=END,FAIL
#SBATCH --time=48:00:00
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=16
#SBATCH --mem-per-cpu=10000
#SBATCH --partition=batch
##SBATCH --exclude=lmWn[012,059]
#module purge
module load FLUENT/24.1
MACHINEFILE="nodes.$SLURM_JOBID"
# Generate Machinefile for mpich such that hosts are in the same
#   order as if run via srun
#
srun -l /bin/hostname | sort -n | awk '{print $2}' > $MACHINEFILE
fluent 3ddp  -mpi=intel -t16 -pdefault -cflush -cnf=$MACHINEFILE -g < journal_run >
     logfluent.$SLURM_JOB_ID
```

Listing C.2: Journal file, `journal_run`

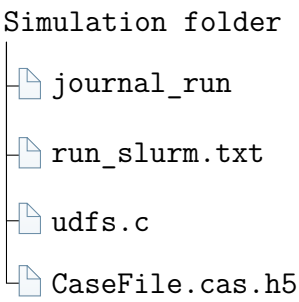


Figure C.1: Folder structure for a simulation project

Result will be written in `CaseFile_final.cas.h5`, and any extra file, coming for Fluent Report Definition will be written in `Simulation folder`.

Bibliography

[1] Nuclear Energy Agency (NEA). *Status Report on Spent Fuel Pools under Loss-of-Cooling and Loss-of-Coolant Accident Conditions: Final Report*. Paris: OECD Publishing, 2015.

[2] J.G. Ibarra et al. *Operating experience feedback report: Assessment of spent fuel cooling*. Report NUREG-1275. Washington, DC, USA: U.S. Nuclear Regulatory Commission, 1997.

[3] Committee on Lessons Learned from the Fukushima Nuclear Accident for Improving Safety et al. *Lessons Learned from the Fukushima Nuclear Accident for Improving Safety and Security of U.S. Nuclear Plants: Phase 2*. Fukushima Daiichi Nuclear Accident: Lessons Learned for Spent Fuel Storage. Washington, DC: National Academies Press (US), 2016. URL: https://www.ncbi.nlm.nih.gov/books/NBK373721/.

[4] Julien Carlier and Miltiadis V. Papalexandris. "Turbulent natural convection in an air–water system with evaporation across the free surface". In: *International Journal of Multiphase Flow* 177 (2024), p. 104873. ISSN: 0301-9322. DOI: https://doi.org/10.1016/j.ijmultiphaseflow.2024.104873. URL: https://www.sciencedirect.com/science/article/pii/S0301932224001502.

[5] Chihiro Yanagi et al. "Evaluation of Heat Loss and Water Temperature in a Spent Fuel Pit". In: *Journal of Power and Energy Systems* 6 (Jan. 2012), pp. 51–62. DOI: 10.1299/jpes.6.51.

[6] William Hay et al. "Turbulent thermal convection driven by free-surface evaporation in cuboidal domains of different aspect ratios". In: *Physics of Fluids* 33 (Jan. 2021), p. 015104. DOI: 10.1063/5.0035277.

[7] W Wagner and A Pruß. "The IAPWS Formulation 1995 for the Thermodynamic Properties of Ordinary Water Substance for General and Scientific Use". In: *Journal of Physical and Chemical Reference Data* 31 (June 2002). DOI: 10.1063/1.1461829.

[8] S. Chandrasekhar. *Hydrodynamic and Hydromagnetic Stability*. Oxford: Clarendon Press, 1961. ISBN: 978-0486640716.

[9] Sh-H Peng, L Davidson, and K Hanjalic. "Large-eddy simulation and deduced scaling analysis of Rayleigh-Benard convection up to Ra=10 9". Undefined/Unknown. In: *Journal of Turbulence* 7.66 (2007). niet eerder opgenomen, pp. 1–29. ISSN: 1468-5248.

[10] Matthias Kaczorowski and Ke-Qing Xia. "Turbulent flow in the bulk of Rayleigh–Bénard convection: small-scale properties in a cubic cell". In: *Journal of Fluid Mechanics* 722 (2013), pp. 596–617. DOI: 10.1017/jfm.2013.74.

[11] Günther Grötzbach. "Spatial resolution requirements for direct numerical simulation of the Rayleigh-Bénard convection". In: *Journal of Computational Physics* 49.2 (1983), pp. 241–264. ISSN: 0021-9991. DOI: https://doi.org/10.1016/0021-9991(83)90125-0. URL: https://www.sciencedirect.com/science/article/pii/0021999183901250.

[12] Janet D Scheel, Mohammad S Emran, and Jörg Schumacher. "Resolving the fine-scale structure in turbulent Rayleigh–Bénard convection". In: *New Journal of Physics* 15.11 (Nov. 2013), p. 113063. DOI: 10.1088/1367-2630/15/11/113063. URL: https://dx.doi.org/10.1088/1367-2630/15/11/113063.

[13] J. O. Hinze. *Turbulence*. New York: McGraw-Hill Publishing Co., 1975.

[14] G. Groetzbach. *Anisotropy and Buoyancy in Nuclear Turbulent Heat Transfer - Critical Assessment and Needs for Modelling*. Tech. rep. FZKA–7363. Germany: Forschungszentrum Karlsruhe, 2007.

[15] F. R. Menter. "Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications". In: *AIAA Journal* 32.8 (Aug. 1994), pp. 1598–1605.

[16] Bernard Castaing et al. "Scaling of hard thermal turbulence in Rayleigh-Bénard convection". In: *Journal of Fluid Mechanics* 204 (1989), pp. 1–30. DOI: 10.1017/S0022112089001643.

[17] Hongyang Wei and Yitung Chen. "Assessment of different turbulence models on the large scale internal heated water pool natural convection simulation". In: *Annals of Nuclear Energy* 131 (2019), pp. 23–38. ISSN: 0306-4549. DOI: https://doi.org/10.1016/j.anucene.2019.03.018. URL: https://www.sciencedirect.com/science/article/pii/S0306454919301434.

[18] Qingyan Chen and Nien-Tzu Chao. "Comparing Turbulence Models for Buoyant Plume and Displacement Ventilation Simulation". In: *Indoor and Built Environment* 6.3 (Aug. 2017), pp. 140–149. ISSN: 1420-326X. DOI: 10.1159/000463322. eprint: https://karger.com/ibe/article-pdf/6/3/140/2992999/000463322.pdf. URL: https://doi.org/10.1159/000463322.

[19] Quan Zhou and Ke-Qing Xia. "Thermal boundary layer structure in turbulent Rayleigh–Bénard convection in a rectangular cell". In: *Journal of Fluid Mechanics* 721 (2013), pp. 199–224. DOI: 10.1017/jfm.2013.73.

[20] Joauma Marichal and Miltiadis V. Papalexandris. "On the dynamics of the large scale circulation in turbulent convection with a free-slip upper boundary". In: *International Journal of Heat and Mass Transfer* 183 (2022), p. 122220. ISSN: 0017-9310. DOI: https://doi.org/10.1016/j.ijheatmasstransfer.2021.122220. URL: https://www.sciencedirect.com/science/article/pii/S0017931021013193.

[21] J. Niemela et al. “Turbulent convection at very high Rayleigh numbers”. In: *Nature* 404 (May 2000), pp. 837–40. DOI: 10.1038/35009036.

[22] Tanaka Hiroaki and Miyata Hiroshi. “Turbulent natural convection in a horizontal water layer heated from below”. In: *International Journal of Heat and Mass Transfer* 23.9 (1980), pp. 1273–1281. ISSN: 0017-9310. DOI: https://doi.org/10.1016/0017-9310(80)90057-5. URL: https://www.sciencedirect.com/science/article/pii/0017931080900575.

[23] F. R. Menter, R. Sechner, and A. Matyushenko. *Best Practice: RANS Turbulence Modeling in Ansys CFD*. Version 1.0. Ansys Germany GmbH, NTS, St. Petersburg, Russia, 2022. URL: https://www.ansys.com/content/dam/amp/2022/february/asset-creation/best-practices-campaign/Best%20Practice-Rans%20turbulence%20modeling%20in%20Ansys%20CFD.pdf.

[24] J. Martin and B. Migot. “Experimental study of the surface evaporation rate of a heated water pool at high temperature using infrared thermography”. In: *18th International Topical Meeting on Nuclear Reactor Thermal Hydraulics*. American Nuclear Society. Portland, Oregon, 2019, pp. 2302–2313.

[25] William A. Hay and Miltiadis V. Papalexandris. “Evaporation-driven turbulent convection in water pools”. In: *Journal of Fluid Mechanics* 904 (2020), A14. DOI: 10.1017/jfm.2020.670.

[26] W. A. Hay and M. V. Papalexandris. “Numerical simulations of turbulent thermal convection with a free-slip upper boundary”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 475.2232 (2019), p. 20190601. DOI: 10.1098/rspa.2019.0601. eprint: https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.2019.0601. URL: https://royalsocietypublishing.org/doi/abs/10.1098/rspa.2019.0601.

[27] CHAO SUN, YIN-HAR CHEUNG, and KE-QING XIA. “Experimental studies of the viscous boundary layer properties in turbulent Rayleigh–Bénard convection”. In: *Journal of Fluid Mechanics* 605 (2008), pp. 79–113. DOI: 10.1017/S0022112008001365.

[28] Younes Benarafa. “Application du couplage RANS/LES aux écoulements turbulents à haut nombre de Reynolds de l’industrie nucléaire”. Theses. Université Pierre et Marie Curie - Paris VI, Dec. 2005. URL: https://theses.hal.science/tel-00011371.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**