

**École polytechnique de Louvain**

# **Numerical study of the natural circulation cooling stability of spent fuel pools**

Analysis of turbulent natural convection and free surface evaporation driven by internal heating

Author: **Matthieu MARTENS**

Supervisor: **Miltiadis PAPALEXANDRIS**

Readers: **Anis BOUSBIASALAH, Vincent LEGAT**

Academic year 2024–2025

Master [120] in Mechanical Engineering

*They did not know it was impossible so they did it*  
Mark Twain

# Abstract

This thesis focuses on spent nuclear fuel assemblies in deactivation pools. It studies the consequences of a loss-of-cooling accident (LOCA) in such pools, *i.e.* a scenario where the cold water inlet and hot water outlet stop abruptly, as was the case with the Fukushima disaster in 2011. The physical phenomena studied in this work are, on the one hand, the behaviour of the flow induced by the heating of the spent nuclear fuel and, on the other hand, the evaporation at the free surface of the pool during the early stage of the accident, when the spent fuel assemblies are still under water and not yet exposed to the ambient air.

The evaporation at the free surface was modelled on the basis of a non-zero Neumann boundary condition on the temperature field. This model was developed by William Hay in his doctoral thesis, *Evaporation-driven turbulent convection in water pools heated from below*, in 2021.

In this thesis, the analysis of the convective behaviour of the flow was first performed using full-scale two-dimensional geometry. Then, a three-dimensional geometry was developed, but at a reduced scale due to limited computational resources. For each of the two geometries, four cases were investigated, based on different distributions of thermal power at the bottom of the pool (*i.e.* different arrangements of “hot” and “cold” assemblies). These assemblies were modelled as a porous medium inside of which head losses were taken into account.

The simulations were carried out using Code\_Saturne, an open source code developed by EDF. These simulations were performed using the Oberbeck-Boussinesq approximation, with the exception of an additional simulation, for each geometry and for comparison purposes, that was carried out using the dilatable flow algorithm of Code\_Saturne, which is akin to the low-Mach number approximation.

It was found that, for the same total thermal power at the bottom of the pool, assemblies with lower power should be placed at the centre and surrounded by higher-power assemblies. This configuration is advantageous compared to the opposite arrangement, where higher-power assemblies are placed at the centre and lower-power assemblies surround them. The former setup delays the time before assemblies become exposed to ambient air by 5 to 7% compared to the latter.

# Acknowledgements

My sincere thanks go first to Professor Miltiadis Papalexandris who backed me up all the way through my Master's thesis. He guided me, encouraged me and he gave me sound advice. I valued the fact that he was very much available.

I would also like to express my gratitude to Doctor Vincent Deledicque and Doctor Anis Bousbiasalah, my industrial supervisors, for their professional support.

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. My many thanks also to Mr Pieter Heremans from CÉCI who was always available and very helpful.

I would like to thank Professor Vincent Legat as well, he agreed to be the reader for this Master's thesis.

On a more personal level, I would like to thank my family for their support and patience and finally I am grateful to Olivier Verdickt for lending me a computer enabling me to have virtual machines so as to be able to use Linux environments in order to use Code\_Saturne.



# Symbols

| Symbol           | Description                                     | Units              |
|------------------|-------------------------------------------------|--------------------|
| $\beta$          | Thermal expansion coefficient                   | $K^{-1}$           |
| $\epsilon$       | Turbulent dissipation rate                      | $m^2 s^{-3}$       |
| $\partial\Omega$ | Boundary of $\Omega$                            | -                  |
| $\zeta$          | Bulk viscosity                                  | $Pa s$             |
| $\eta$           | Concentration expansion coefficient             | $m^3 kg^{-1}$      |
| $\kappa$         | Thermal diffusivity                             | $m^2 s^{-1}$       |
| $\lambda$        | Thermal conductivity                            | $W m^{-1} K^{-1}$  |
| $\mu$            | Dynamic viscosity                               | $Pa s$             |
| $\nu$            | Kinematic viscosity                             | $m^2 s^{-1}$       |
| $\rho$           | Density                                         | $kg m^{-3}$        |
| $\tau_{wall}$    | Wall shear stress                               | $Pa$               |
| $\phi$           | Porosity                                        | -                  |
| $\omega$         | Specific dissipation rate                       | $s^{-1}$           |
| $\Omega$         | Three-dimensional domain                        | -                  |
| $B_m$            | Mass transfer driving force                     | -                  |
| $C$              | Concentration                                   | $kg m^{-3}$        |
| $c_p$            | Specific heat capacity                          | $J kg^{-1} K^{-1}$ |
| $dS$             | Differential surface element                    | $m^2$              |
| $dV$             | Differential volume element                     | $m^3$              |
| $\mathcal{D}$    | Mass diffusivity                                | $kg m^{-2} s^{-1}$ |
| $e$              | Internal energy                                 | $J kg^{-1}$        |
| $e_t$            | Total energy                                    | $J kg^{-1}$        |
| $f_D$            | Darcy friction factor                           | -                  |
| $h$              | Enthalpy                                        | $J kg^{-1}$        |
| $h_h$            | Convective heat transfer coefficient            | $W m^{-2} K^{-1}$  |
| $h_h^*$          | Volumetric convective heat transfer coefficient | $W m^{-3} K^{-1}$  |
| $h_{lv}$         | Latent heat of evaporation                      | $J kg^{-1}$        |
| $h_m$            | Convective mass transfer coefficient            | $m s^{-1}$         |
| $H_a$            | Height of the assemblies                        | $m$                |
| $H_{bulk}$       | Height of the bulk region of the pool           | $m$                |
| $H_C$            | Height of the computational domain              | $m$                |
| $H_{gap}$        | Height of the gap in the pool                   | $m$                |
| $H_{pool}$       | Height of the pool                              | $m$                |
| $H_S$            | Height of the free-surface                      | $m$                |
| $k$              | Turbulent kinetic energy                        | $m^2 s^{-2}$       |
| $L_c$            | Characteristic length of a transfer surface     | $m$                |

|                         |                                                |                                  |
|-------------------------|------------------------------------------------|----------------------------------|
| $\dot{m}_{evap}''$      | Evaporative mass flux                          | $\text{kg m}^{-2} \text{s}^{-1}$ |
| $n_r$                   | Number of rods inside an assembly              | -                                |
| $p$                     | Thermodynamic pressure                         | Pa                               |
| $P_a$                   | Power of an assembly                           | W                                |
| $q_f'''$                | Volumetric heat source for the fluid           | $\text{W m}^{-3}$                |
| $q_s'''$                | Volumetric heat source for the solid           | $\text{W m}^{-3}$                |
| $T$                     | Temperature                                    | K                                |
| $u^+$                   | Normalized velocity                            | -                                |
| $u_\tau$                | Friction velocity                              | $\text{m s}^{-1}$                |
| $W_a$                   | Width of an assembly                           | m                                |
| $W_{gap}$               | Width of the gap in the pool                   | m                                |
| $W_{pool}$              | Width of the pool                              | m                                |
| $W_{wall}$              | Width of a wall between two assemblies         | m                                |
| $y$                     | Normal distance to the wall                    | m                                |
| $y^+$                   | Dimensionless distance from the wall           | -                                |
| $Y$                     | Mass fraction                                  | -                                |
| <hr/>                   |                                                |                                  |
| $\mathbf{g}$            | Gravity vector                                 | $\text{m s}^{-2}$                |
| $\hat{\mathbf{n}}$      | Outward unit vector                            | -                                |
| $\mathbf{q}''$          | Heat-flux vector                               | $\text{W m}^{-2}$                |
| $\mathbf{u}$            | Velocity field                                 | $\text{m s}^{-1}$                |
| $\mathbf{v}$            | Velocity vector                                | $\text{m s}^{-1}$                |
| $\mathbf{x}$            | Position vector                                | m                                |
| <hr/>                   |                                                |                                  |
| $\mathbf{I}$            | Unit tensor                                    | -                                |
| $\mathbf{K}$            | Velocity head loss tensor                      | $\text{s}^{-1}$                  |
| $\mathbf{R}$            | Reynolds stress tensor                         | $\text{kg m}^{-1} \text{s}^{-2}$ |
| $\mathbf{S}$            | Strain rate tensor                             | $\text{s}^{-1}$                  |
| $\boldsymbol{\sigma}$   | Cauchy stress tensor                           | Pa                               |
| $\boldsymbol{\sigma}^v$ | Viscous stress tensor (with bulk viscosity)    | Pa                               |
| $\boldsymbol{\tau}$     | Viscous stress tensor (without bulk viscosity) | Pa                               |
| <hr/>                   |                                                |                                  |
| Gr                      | Thermal Grashof number                         | -                                |
| $\text{Gr}_c$           | Concentration Grashof number                   | -                                |
| Le                      | Lewis number                                   | -                                |
| Nu                      | Nusselt number                                 | -                                |
| Pr                      | Prandtl number                                 | -                                |
| Ra                      | Thermal Rayleigh number                        | -                                |
| $\text{Ra}_c$           | Concentration Rayleigh number                  | -                                |
| Re                      | Reynolds number                                | -                                |
| Ri                      | Richardson number                              | -                                |
| Sc                      | Schmidt number                                 | -                                |
| Sh                      | Sherwood number                                | -                                |

In this document, we adopt the following notation conventions:

- Scalar quantities are denoted by a lowercase or uppercase letter (*e.g.*  $u$  or  $U$ ).
- Vector quantities are denoted by a bold lowercase letter (*e.g.*  $\mathbf{u}$ ).
- Tensor quantities are denoted by a bold uppercase or a bold Greek letter (*e.g.*  $\boldsymbol{\sigma}$ ).

Regarding the material derivative, the  $D/Dt$  notation is used. Moreover, we recall that the material derivative of a quantity  $f$  is given by:

$$\frac{Df}{Dt} \equiv \frac{\partial f}{\partial t} + \mathbf{u} \cdot \nabla f \quad (1)$$

Moreover, the gravity vector is defined as follows:

$$\mathbf{g} = -g\hat{\mathbf{e}}_z \quad (2)$$

with  $g = 9.81 \text{ m/s}^2$ .

# Contents

|                                                     |            |
|-----------------------------------------------------|------------|
| <b>Abstract</b>                                     | <b>i</b>   |
| <b>Acknowledgements</b>                             | <b>ii</b>  |
| <b>Symbols</b>                                      | <b>iii</b> |
| <b>Introduction</b>                                 | <b>1</b>   |
| <b>State of the art</b>                             | <b>5</b>   |
| <b>1 Mathematical and physical basis</b>            | <b>7</b>   |
| 1.1 Key mathematical theorems . . . . .             | 7          |
| 1.2 Governing equations for laminar flows . . . . . | 8          |
| 1.2.1 Mass equation . . . . .                       | 8          |
| 1.2.2 Momentum equation . . . . .                   | 8          |
| 1.2.3 Energy equation . . . . .                     | 9          |
| 1.2.4 Resulting governing equations . . . . .       | 10         |
| 1.3 Governing equations in porous media . . . . .   | 12         |
| 1.4 Turbulence . . . . .                            | 12         |
| 1.4.1 General concepts . . . . .                    | 12         |
| 1.4.2 RANS approach . . . . .                       | 14         |
| 1.4.3 Turbulence models . . . . .                   | 16         |
| 1.4.4 Law of the wall . . . . .                     | 18         |
| 1.5 The Oberbeck-Boussinesq approximation . . . . . | 21         |
| 1.6 Dimensionless analysis . . . . .                | 22         |
| 1.6.1 Dimensionless governing equations . . . . .   | 22         |
| 1.6.2 Heat and mass transfer analogy . . . . .      | 24         |
| <b>2 Numerical setup</b>                            | <b>26</b>  |
| 2.1 Computational domain . . . . .                  | 26         |
| 2.2 Initial and boundary conditions . . . . .       | 27         |
| 2.3 Modelling the evaporation . . . . .             | 27         |
| 2.3.1 Latent heat losses . . . . .                  | 28         |
| 2.3.2 Convective heat losses . . . . .              | 29         |
| 2.3.3 Computational aspect . . . . .                | 30         |
| 2.3.4 Free surface descent . . . . .                | 30         |
| 2.4 Fluid properties . . . . .                      | 32         |
| 2.4.1 Water properties . . . . .                    | 32         |
| 2.4.2 Air properties . . . . .                      | 33         |

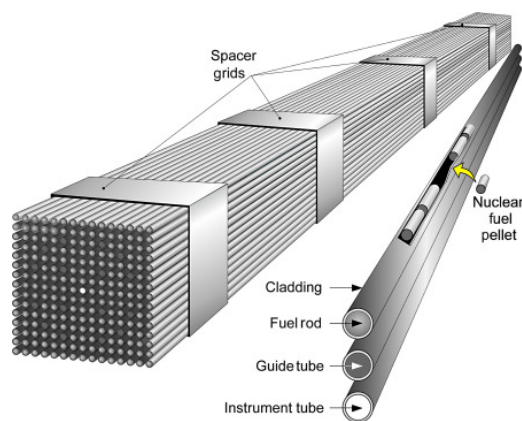
|          |                                                                  |           |
|----------|------------------------------------------------------------------|-----------|
| 2.4.3    | Water vapour properties . . . . .                                | 34        |
| 2.4.4    | Film properties . . . . .                                        | 35        |
| 2.5      | Porosity and head loss due to the assemblies . . . . .           | 37        |
| 2.6      | Heat dissipation of an assembly . . . . .                        | 40        |
| 2.7      | Geometries and case studies . . . . .                            | 42        |
| 2.7.1    | Domain dimensions and meshes . . . . .                           | 42        |
| 2.7.2    | Definition of case studies . . . . .                             | 45        |
| 2.8      | Solving the Navier-Stokes equations . . . . .                    | 46        |
| 2.8.1    | Time discretisation of transport equations . . . . .             | 46        |
| 2.8.2    | Pressure-based velocity-pressure solver . . . . .                | 47        |
| 2.8.3    | Choice of the turbulence model . . . . .                         | 48        |
| 2.8.4    | Stability conditions . . . . .                                   | 49        |
| 2.9      | User subroutines . . . . .                                       | 50        |
| <b>3</b> | <b>Numerical results and discussion</b>                          | <b>52</b> |
| 3.1      | Two-dimensional geometry . . . . .                               | 52        |
| 3.1.1    | Preliminary considerations . . . . .                             | 52        |
| 3.1.2    | Numerical results - Case 1 . . . . .                             | 53        |
| 3.1.3    | Comparison with the other cases . . . . .                        | 58        |
| 3.1.4    | Non-Oberbeck-Boussinesq flow . . . . .                           | 64        |
| 3.1.5    | Numerical performances . . . . .                                 | 65        |
| 3.2      | Three-dimensional geometry . . . . .                             | 65        |
| 3.2.1    | Preliminary considerations . . . . .                             | 65        |
| 3.2.2    | Numerical results - Case 1 . . . . .                             | 66        |
| 3.2.3    | Comparison with the other cases . . . . .                        | 70        |
| 3.2.4    | Non-Oberbeck-Boussinesq flow . . . . .                           | 73        |
| 3.2.5    | Numerical performances . . . . .                                 | 74        |
| 3.3      | Results discussion . . . . .                                     | 74        |
|          | <b>Conclusion</b>                                                | <b>76</b> |
| <b>A</b> | <b>Head losses inside the assemblies</b>                         | <b>78</b> |
| A.1      | Computation of the friction coefficients . . . . .               | 78        |
| A.2      | Geometric parameters . . . . .                                   | 80        |
| <b>B</b> | <b>Additional figures for the two-dimensional geometry</b>       | <b>81</b> |
| B.1      | Non-Oberbeck-Boussinesq flow . . . . .                           | 81        |
| <b>C</b> | <b>Additional figures for the three-dimensional geometry</b>     | <b>85</b> |
| C.1      | Descent of the free-surface across the different cases . . . . . | 85        |
| C.2      | Non-Oberbeck-Boussinesq flow . . . . .                           | 86        |
|          | <b>Bibliography</b>                                              | <b>91</b> |

# Introduction

After a decline in the use of nuclear energy following the Fukushima disaster in 2011, nuclear programs are now being revived in many countries. In Belgium, on 15 May 2025, Parliament repealed the law that had planned the phase-out of nuclear power [1]. In 2023, nuclear energy provided approximately 9% of the world's electricity, generated by around 440 reactors operating across the globe [2]. In Europe, the country with the largest number of nuclear power plants is France, with 57 reactors in operation, accounting for about 70% of its energy mix [3]. All of these reactors are Pressurized Water Reactors (PWRs), with the exception of the power plant at Flamanville.

In the present work, numerous data related to French PWRs, made publicly available by *Électricité de France* (EDF) and the *Autorité de Sûreté Nucléaire et de Radioprotection* (ASNR), are used as a foundation.

In a PWR, the nuclear fuel used in the reactors mainly comes from enriched natural uranium. This fuel takes the form of small pellets, stacked one on top of another and inserted into what is known as a *fuel rod*. Several of these fuel rods are then grouped together to form a *fuel assembly*, which is approximately 4 meters long. Such a fuel assembly is shown in Figure 1.

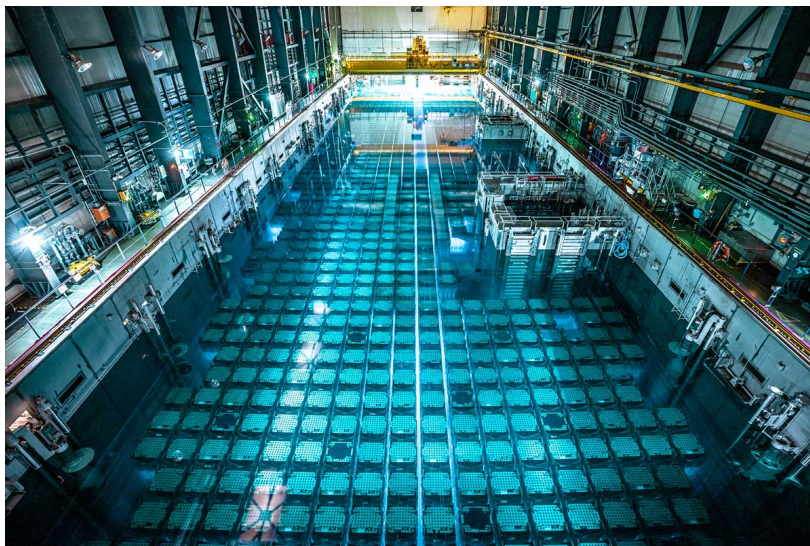


**Figure 1:** Typical fuel assembly. This figure is taken from [4].

These assemblies undergo three cycles of 12 to 18 months each in the reactor. At the end of each cycle, the reactor is shut down and the oldest one-third of assemblies is removed and replaced, ensuring that at any given time, the pool contains one-third new fuel, one-third intermediate, and one-third old fuel. Although fission reactions stop once the reactor is shut down, fuel assemblies are still highly radioactive due to the presence of unstable fission products. As a result, they continue to release a considerable amount of energy in the form of heat. Therefore, these assemblies are typically left in the reactor

for 3 to 4 days before being taken out of the core and placed in Spent Fuel Pools (SFPs), located on the power plant site, where they remain for 2 to 3 years [5, 6]. To give order of magnitudes, a PWR1300, which is a reactor producing 1300 MW of electrical power, generates 3817 MW of thermal power<sup>1</sup> [7]. Immediately after shutdown, the residual power is about 7% of the nominal power [5], or approximately 265 MW of thermal power. Since a PWR1300 contains 193 fuel assemblies in the core [7], this corresponds to an average of about 1.35 MW of thermal power per assembly, which is enormous. However, because some fission products have very short half-lives, this thermal power drops significantly during the first few days. After the typical time of 3 to 4 days in the reactor following shutdown, the residual thermal power per assembly drops below 100 kW.

In spent fuel pools, the assemblies are typically gathered in *racks*. These racks stabilize the assemblies and facilitate their handling. In addition, they prevent cross-flow between adjacent assemblies. Figure 2 shows a spent fuel pool, highlighting the arrangement of the assemblies within different racks.



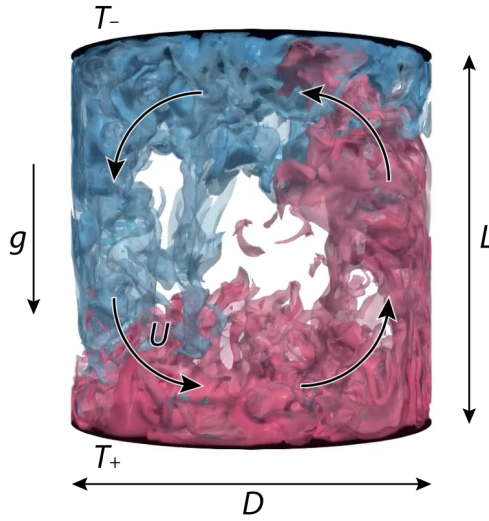
**Figure 2:** *Spent fuel pool at La Hague facility. This is not a pool on a reactor site, but an end-of-life treatment centre for spent fuel. However, these pools are fundamentally the same. This figure is taken from [8].*

To remove the heat generated by the spent fuel assemblies, spent fuel pools are equipped with dedicated cooling systems. This continuous cooling process is essential to maintain safe water temperatures and to ensure that the fuel assemblies remain fully submerged, thus providing both effective heat removal and radiation shielding. In this thesis, we focus on the consequences of a failure of such cooling systems. This situation is known as a loss-of-cooling accident (LOCA). Particularly, we consider the first phase of this event, when the spent fuel assemblies are still under water and not yet exposed to the ambient air. In the absence of a cooling system, a temperature gradient develops between the bottom of the pool, where the assemblies generate heat, and the upper part of the pool, *i.e.* the free surface. This gradient causes instabilities in the fluid, which tends to set it in motion. Convection cells, similar to Rayleigh-Bénard cells, are therefore observed.

---

<sup>1</sup>The thermal efficiency of such reactors is around 34 to 35%.

Rayleigh-Bénard Convection (RBC) is a type of natural convection and describes the flow occurring in a closed box that is homogeneously heated from below and cooled from above. These flows are driven by the density differences between the lighter hot fluid, which wants to rise, and the heavier cold fluid, which wants to sink. Therefore, gravity plays a crucial role in such flows. The strength of the thermal driving is characterized by the Rayleigh number, the formal definition of which will be given later. Under a certain threshold of the Rayleigh number, heat transfer only occurs through conduction in the fluid and no fluid motion is observed. Above this threshold, the density gradient becomes large enough to induce instabilities in the fluid and thus set it in motion. This motion is characterized by specific structures known as cells that can take various shapes depending, among others, on the geometry and the aspect ratio, defined as the ratio of the width of the domain over its height [9]. Figure 3 shows a typical RBC with a sole convective cell clearly apparent.



**Figure 3:** Formation of a Rayleigh-Bénard convection cell inside a cylindrical domain heated from below and cooled from above, with aspect ratio  $D/L = 1$ . This figure is taken from [9].

Evaporation at a free surface corresponds to the phase change of the liquid state to the vapour state. This phenomenon occurs when more molecules leave the interface from the water side than molecules enter from the gas side. The rate at which molecules escape depends on the kinetic energy of the liquid molecules, which in turn is related to the water temperature. It is important to note that the evaporation rate depends on the temperature at the free surface, rather than on the bulk temperature, as was long assumed [10]. Moreover, evaporation is a mass transfer phenomenon that depends on the ambient air conditions. In fact, the evaporation rate is governed by gas side transport processes. Water vapour molecules must first diffuse through a concentration boundary layer, and are then transported away by convection.

In this thesis, Code\_Saturne (v. 8.1.3), an open source software, developed by EDF will be used to perform the different numerical simulations. Some data and dimensions regarding the deactivation pools were provided by BelV, a Belgian private foundation affiliated with the *Agence Fédérale de Contrôle Nucléaire* (AFCN). Dr. Anis Bousbiasalah and Dr. Vincent Deledicque, from BelV, closely followed the development of the present work.



The structure of this work is such that we give an introductory description of the state of the art. Then, in Chapter 1, the formulation of the governing equations of fluid mechanics, as implemented in Code\_Saturne, will be derived from the conservation laws. Moreover, the physics of turbulence will be reviewed and a dimensional analysis will be performed. Chapter 2 establishes the numerical setup. In particular, the evaporation model, and the modelling of the assemblies are given. The different case studies are also provided. Lastly, Chapter 3 presents the numerical results and discuss the flow behaviours as well as the influence of each case study on the evaporation rates.

# State of the art

The study of the consequences of the failure of the cooling systems in spent nuclear fuel deactivation pools became critical after the Fukushima disaster in 2011. These studies focus on natural convection and evaporation and are still in progress. They should contribute to the continuous improvement of the safety of spent fuel storage.

Natural convection, on the one hand, and evaporation, on the other hand, have already been studied and modelled outside the context of deactivation pools. Natural convection consists of flows induced by density gradients, themselves provoked by temperature gradients for instance. Such flows are present in nature and are key for numerous problems widely investigated in oceanography, climatology, geophysics, and astrophysics [9]. They also occur in technological applications such as in boilers and power plants, where they were studied specifically [11]. These flows present fundamental similarities with the Rayleigh-Bénard Convection, with specific differences highlighted in each field of investigation. As regards evaporation, this topic is particularly relevant, as growing demand of freshwater resources for industrial or agricultural needs make it essential to accurately predict evaporation rates from inland water bodies such as lakes, reservoirs and life-stock ponds [12, 13, 14].

After 2011, the study of those two physical phenomena has been extended to the field of spent fuel-pools.

In the scientific field, a post-Fukushima project, named the DENOPI project, was launched in 2013 by the ASNR to collect experimental data for the validation of numerical simulation codes. This project is structured along three axes. The first axis is dedicated to the study of large convective structures within the pool as well as boiling phenomena on the basis of reduced-scale pool models. This axis studies the early stage of a loss-of-cooling accident, where the assemblies are still surrounded by a large amount of water. The second axis studies thermo-hydraulic phenomena on the scale of an assembly where the latter is partially or totally exposed to air. Finally, the last axis considers the oxidation of zirconium present in the assemblies [15].

At UCLouvain, William Hay (2021) [10] performed a numerical study of turbulent thermal convection driven by free-surface evaporation in spent-fuel pools. In his thesis, geometric simplifications were made. Specifically, only the upper volume of the pool was modelled, *i.e.* above the spent-fuel assemblies. Therefore, he considered cuboidal domains, with different aspect ratios, with a homogeneously heated wall at the bottom. As regards the free surface, he developed a model for quantifying free-surface evaporation rates and heat losses due to this phenomenon. This was achieved by implementing a non-zero Neumann boundary condition for the temperature.

Later, in her thesis, Victoria Hamtiaux (2024) [16] studied natural convection within cuboidal domains where spent fuel assemblies were explicitly considered. The latter were modelled as porous media with internal volumetric heat generation. However, evaporation at the free surface was not taken into account.

At present, Julien Carlier and Miltiadis Papalexandris are studying turbulent natural convection in air-water systems with evaporation at the free surface. More specifically, the dynamics between the liquid and gaseous phases and the free surface are under investigation [17].

The present Master's thesis is a continuation of the work of Gauthier Rubbers (2023) [18] and Michael Magain (2024) [19]. In our work, only the flow in the liquid phase is solved, the evaporation at the free surface is modelled by a non-zero Neumann boundary condition, and the assemblies are considered as a porous medium at the bottom of the pool.

# Chapter 1

## Mathematical and physical basis

The aim of this chapter is to develop the various mathematical and physical aspects required in the present study.

First, the formulation of the governing equations of fluid mechanics implemented in Code\_Saturne will be briefly derived. Then, the equations will be generalized to account for porosity. These developments closely follow the Theoretical and Practical User's guides provided by EDF [20, 21], while some additional physical interpretations are drawn from [22] and [23]. Following this, the concept of turbulence will be introduced, and the conservation laws will be extended to this concept. In addition, the Boussinesq approximation, which accounts for density variations only in the buoyancy term, will also be introduced. Lastly, the governing equations will be provided in dimensionless form and the analogy between heat and mass transfers will be discussed.

### 1.1 Key mathematical theorems

Before proceeding with the main developments, two fundamental mathematical theorems must be introduced.

Firstly, the Reynolds transport theorem describes the material derivative<sup>1</sup> of the integral of a field variable  $f$ , which can be tensor-, vector- or scalar-valued, over a domain  $\Omega$  that deforms over time. It is stated as follows:

$$\frac{D}{Dt} \left( \int_{\Omega(t)} f(\mathbf{x}, t) dV \right) = \int_{\Omega(t)} \frac{\partial}{\partial t} f(\mathbf{x}, t) dV + \int_{\partial\Omega(t)} f(\mathbf{x}, t) \mathbf{v} \cdot \hat{\mathbf{n}} dS \quad (1.1)$$

where  $\mathbf{x}$  is the position vector,  $\mathbf{v}$  the velocity of the boundary of  $\Omega$ , denoted  $\partial\Omega$ , with an outward unit vector  $\hat{\mathbf{n}}$ , and  $dV$  and  $dS$  represent the differential volume and surface elements, respectively.

Secondly, the divergence theorem states that the integral of the divergence of a continuously differentiable vector field  $\mathbf{f}$  is equal to the flux of the vector field through a closed surface. It reads:

$$\int_{\Omega(t)} \nabla \cdot \mathbf{f}(\mathbf{x}, t) dV = \int_{\partial\Omega(t)} \mathbf{f}(\mathbf{x}, t) \cdot \hat{\mathbf{n}} dS \quad (1.2)$$

---

<sup>1</sup>Physically, the material derivative expresses the rate of change of a quantity along streamlines.

## 1.2 Governing equations for laminar flows

### 1.2.1 Mass equation

Mass conservation states that the mass within a fluid volume<sup>2</sup> remains constant, as no fluid particles can enter or leave this volume. Therefore, one can write:

$$\frac{D}{Dt} \left( \int_{\Omega(t)} \rho dV \right) = 0 \quad (1.3)$$

By consecutively applying (1.1) and (1.2) to (1.3) and considering the density  $\rho$  and the fluid velocity field  $\mathbf{u}$  as continuously differentiable functions, we can derive the mass conservation law in integral form:

$$\int_{\Omega(t)} \left( \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) \right) dV = 0 \quad (1.4)$$

Since this law holds for an arbitrary fluid volume, then the integrand must be zero. This gives the local form of the mass balance law, also referred as the *continuity equation* in the literature:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (1.5)$$

### 1.2.2 Momentum equation

According to Newton's second law, the rate of change of the momentum of a body equals the sum of the forces acting on it. In general, a distinction is made between the volumetric forces, which act inside the body, and the surface forces, which act on material surfaces, either within the body or at its boundary. Volumetric forces are typically due to the gravity  $\mathbf{g}$ , while surface forces are expressed in terms of stresses and cast inside the Cauchy stress tensor  $\boldsymbol{\sigma}$ , so that we have:

$$\frac{D}{Dt} \left( \int_{\Omega(t)} \rho \mathbf{u} dV \right) = \underbrace{\int_{\partial\Omega(t)} \boldsymbol{\sigma} \cdot \hat{\mathbf{n}} dS}_{\text{surface forces}} + \underbrace{\int_{\Omega(t)} \rho \mathbf{g} dV}_{\text{volumetric forces}} \quad (1.6)$$

Applying (1.1) and (1.2) to the left-hand side of (1.6), and (1.2) to the first term of the right-hand side of (1.6), and noting that this expression holds for any fluid volume, one can obtain the momentum balance law in local form:

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g} \quad (1.7)$$

where  $\otimes$  stands for the dyadic product.

Regarding the stresses, it is important to distinguish the equilibrium stresses and the dissipative stresses. Equilibrium stresses produce reversible work while dissipative ones produce irreversible work. The latter are due to friction, and thus to the viscosity of

---

<sup>2</sup>A fluid volume, or material volume, consists of a region of space that moves and deforms according to the motion of the fluid particles.

the fluid. Moreover, for simple fluids<sup>3</sup>, the only equilibrium stress is the thermodynamic pressure  $p$ . Therefore, by letting  $\boldsymbol{\sigma}^v$  designate the viscous stresses, the stress tensor  $\boldsymbol{\sigma}$  can be decomposed as follows:

$$\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\sigma}^v \quad (1.8)$$

For Newtonian fluids, the viscous stresses are linear with respect to the velocity gradients and can be written as [22]:

$$\boldsymbol{\sigma}^v = 2\mu\mathbf{S} + \left(\zeta - \frac{2}{3}\mu\right) \text{tr}(\mathbf{S})\mathbf{I} \quad (1.9)$$

with  $\mu$  the dynamic molecular viscosity,  $\zeta$  the volume or bulk viscosity and  $\mathbf{S}$  the strain rate tensor, defined as:

$$\mathbf{S} = \frac{1}{2} ((\nabla\mathbf{u}) + (\nabla\mathbf{u})^T) \quad \text{and} \quad \text{tr}(\mathbf{S}) = \nabla \cdot \mathbf{u} \quad (1.10)$$

Finally, one can consider an additional term to take into account head losses so that (1.7) becomes [20]:

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla \cdot (\rho\mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot \boldsymbol{\sigma}^v + \rho\mathbf{g} - \rho\mathbf{K}\mathbf{u} \quad (1.11)$$

where  $\mathbf{K}$  is the velocity head loss tensor, which, by definition, is a symmetric positive tensor so that its contribution to the kinetic energy balance is negative. Strictly speaking, (1.11) corresponds to the so-called Navier-Stokes equations.

As a final note, it should be emphasized that, for isotropic materials, such as water, the principle of conservation of angular momentum implies that the stress tensor is symmetric, *i.e.*  $\boldsymbol{\sigma} = \boldsymbol{\sigma}^T$ .

### 1.2.3 Energy equation

The first law of thermodynamics states that the variation of total energy inside a fluid volume, which is by definition a closed system, is equal to the sum of the work of the external forces acting on the fluid and the heat that the fluid exchanges with its exterior. The total energy is simply the sum of the internal energy  $e$  and the kinetic energy:

$$e_t = e + \frac{1}{2}\mathbf{u} \cdot \mathbf{u} \quad (1.12)$$

Moreover, we know from classical mechanics that the work performed by a force is given by the dot product of the force and the displacement vector. Therefore, the infinitesimal work per unit time is given by the dot product of the force with the velocity field. This allows us to write the energy conservation principle for a continuous medium:

$$\frac{D}{Dt} \left( \int_{\Omega(t)} \rho e_t dV \right) = \int_{\Omega(t)} \rho \mathbf{g} \cdot \mathbf{u} dV + \int_{\partial\Omega(t)} (\boldsymbol{\sigma}\mathbf{u}) \cdot \hat{\mathbf{n}} dS + \int_{\partial\Omega(t)} \mathbf{q}'' \cdot \hat{\mathbf{n}} dS \quad (1.13)$$

where  $\mathbf{q}''$  is the heat-flux vector.

---

<sup>3</sup>Simple fluids are fluids that do not support shear stresses at equilibrium.

By introducing the decomposition (1.8) of the stress tensor  $\boldsymbol{\sigma}$  in (1.13), and by adopting a similar approach to the previous conservation laws, we can apply (1.1) followed by (1.2) to the left-hand side of (1.13), and then apply (1.2) to the second and last terms on the right-hand side of (1.13), which allows us to get the energy balance law in local form:

$$\frac{\partial(\rho e_t)}{\partial t} + \nabla \cdot (\rho \mathbf{u} e_t) = -\nabla \cdot (p \mathbf{u}) + \rho \mathbf{g} \cdot \mathbf{u} + \nabla \cdot (\boldsymbol{\sigma}^v \mathbf{u}) - \nabla \cdot \mathbf{q}'' \quad (1.14)$$

The energy balance law can also be written in terms of the internal energy  $e$ . By inserting (1.12) in (1.14) and then taking the dot product of the momentum balance law in local form (1.7) with the velocity field  $\mathbf{u}$ , and inserting this result in (1.14), we get:

$$\rho \frac{De}{Dt} = -p \nabla \cdot \mathbf{u} + \boldsymbol{\sigma}^v : \mathbf{S} - \nabla \cdot \mathbf{q}'' \quad (1.15)$$

However, in this study, the focus will be placed on the temperature field. Therefore, one must further develop (1.17) in order to express it in terms of temperature. First, we will express the energy equation in terms of enthalpy, defined as:

$$h \equiv e + \frac{1}{\rho} p \quad (1.16)$$

By applying the material derivative to  $h$ , introducing this result into (1.17), and then recognizing that the continuity equation can be made to appear, we obtain:

$$\rho \frac{Dh}{Dt} = \frac{Dp}{Dt} + \boldsymbol{\sigma}^v : \mathbf{S} - \nabla \cdot \mathbf{q}'' \quad (1.17)$$

We can now introduce the temperature  $T$  by using some thermodynamics relationships. For a pure substance, Maxwell's relations give [20]:

$$dh = c_p dT + \frac{1}{\rho} (1 - \beta T) dp \quad (1.18)$$

with  $c_p$  the specific heat capacity at constant pressure and  $\beta$  the thermal expansion coefficient. Finally, by inserting (1.18) in (1.17), by using Fourier's law of heat conduction to model the heat-flux vector and by introducing a volumetric heat source term for the fluid  $q_f'''$ , the energy conservation law reads:

$$\rho c_p \frac{DT}{Dt} = \beta T \frac{Dp}{Dt} + \boldsymbol{\sigma}^v : \mathbf{S} + \nabla \cdot (\lambda \nabla T) + q_f''' \quad (1.19)$$

where  $\lambda$  is the thermal conductivity. The source term  $q_f'''$  will be modelled later.

## 1.2.4 Resulting governing equations

So far, we have derived the governing equations for Newtonian fluids in their most general form. Given that this document focuses on the study of natural convection, it is clear that the Mach number will remain very low throughout the domain. Consequently, we can consider the flow as incompressible.

We will now present the system of equations implemented in Code\_Saturne, solved within the framework of the *unsteady dilatable flow algorithm*. By *dilatable flow*, we refer

to a flow in which the fluid density can vary, typically with temperature, while this variation remains small and is independent of pressure. This algorithm is slightly different from the *Low-Mach algorithm*, but we will return to this distinction later [20].

First, for incompressible flows, the bulk viscosity is neglected [20], which simplifies the expression of the viscous stresses (1.9) as follows:

$$\boldsymbol{\sigma}^v|_{\zeta=0} \equiv \boldsymbol{\tau} = 2\mu\mathbf{S} - \frac{2}{3}\mu\text{tr}(\mathbf{S})\mathbf{I} \quad (1.20)$$

In this case, one can show that the second term on the right-hand side of (1.19) satisfies the following equality [24]:

$$\boldsymbol{\sigma}^v|_{\zeta=0} : \mathbf{S} \equiv \boldsymbol{\tau} : \mathbf{S} = 2\mu\mathbf{S}^D : \mathbf{S}^D = \mu S^2 \quad (1.21)$$

with  $\mathbf{S}^D$  the deviatoric part of the strain rate tensor, defined as follows:

$$\mathbf{S}^D = \mathbf{S} - \frac{1}{3}\text{tr}(\mathbf{S})\mathbf{I} \quad (1.22)$$

The term  $\mu S^2$  in (1.21) is sometimes referred to as *viscous dissipation heating* and describes the degradation of mechanical energy into thermal energy, which is considered negligible in free convection flows [24]. This quantity is neglected in Code\_Saturne and will therefore no longer be included in (1.19).

Finally, in the context of the dilatable flow algorithm, the term  $\beta T$  in (1.19) is neglected, which leads to the disappearance of the pressure term in the energy conservation equation.

Accordingly, the system of equations is given by:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (1.23)$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot \left( 2\mu\mathbf{S} - \frac{2}{3}\mu\text{tr}(\mathbf{S})\mathbf{I} \right) + \rho \mathbf{g} - \rho \mathbf{K} \mathbf{u} \quad (1.24)$$

$$\frac{\partial (\rho c_p T)}{\partial t} + \nabla \cdot (\rho c_p \mathbf{u} T) = \nabla \cdot (\lambda \nabla T) + q_f''' \quad (1.25)$$

Note that (1.25) is obtained from (1.19) by making the continuity equation appear, allowing some terms to be eliminated, thereby obtaining a conservative form.

We can now state that the difference between the dilatable flow algorithm and the Low-Mach number algorithm is the absence of a pressure term in equation (1.25) [25]. Moreover, in the Low-Mach number algorithm, the bulk viscosity is not necessarily neglected, and the resulting viscous pressure is incorporated into the pressure term in equation (1.24). These two algorithms are thus analogous. In what follows, we will refer to system (1.23)–(1.25) as the dilatable flow model, or simply dilatable flow, to remain consistent with the code documentation. Regarding the Low-Mach number algorithm, the reader is referred to the literature for its specific features.

To conclude the derivation of the governing equations, we point out that an equation of state for the density is required in order to close the system of equations (1.23)–(1.25). The latter will be introduced in Section 2.4.1.



## 1.3 Governing equations in porous media

As mentioned in the state of the art, the assemblies of spent nuclear fuel rods are modelled as porous media. Therefore, the governing equations (1.23)-(1.25) have to be adapted to account for porosity. Introducing the porosity  $\phi$ , defined as the ratio of the volume of water to the volume of the porous medium, the system of equations becomes:

$$\frac{\partial(\phi\rho)}{\partial t} + \nabla \cdot (\phi\rho\mathbf{u}) = 0 \quad (1.26)$$

$$\frac{\partial(\phi\rho\mathbf{u})}{\partial t} + \nabla \cdot (\phi\rho\mathbf{u} \otimes \mathbf{u}) = -\phi\nabla p + \nabla \cdot \left( 2\phi\mu\mathbf{S} - \frac{2}{3}\phi\mu\text{tr}(\mathbf{S})\mathbf{I} \right) + \phi\rho\mathbf{g} - \phi\rho\mathbf{K}\mathbf{u} \quad (1.27)$$

$$\frac{\partial(\phi\rho c_p T)}{\partial t} + \nabla \cdot (\phi\rho c_p \mathbf{u} T) = \nabla \cdot (\phi\lambda\nabla T) + q_f''' \quad (1.28)$$

Equations (1.26)–(1.28) can therefore be regarded as the governing equations for the entire domain. Indeed, by considering  $\phi = \phi(\mathbf{x})$ , with  $\mathbf{x}$  being the position vector, we have  $\phi = 1$  in the pure fluid region, allowing us to recover equations (1.23)–(1.25), while in the porous medium,  $\phi < 1$ , leading to the use of the aforementioned system.

Note that to avoid overloading the equations, the porosity term  $\phi$  will no longer be explicitly mentioned in the remainder of this chapter. However, all the principles that will be developed can be generalized to include porosity.

## 1.4 Turbulence

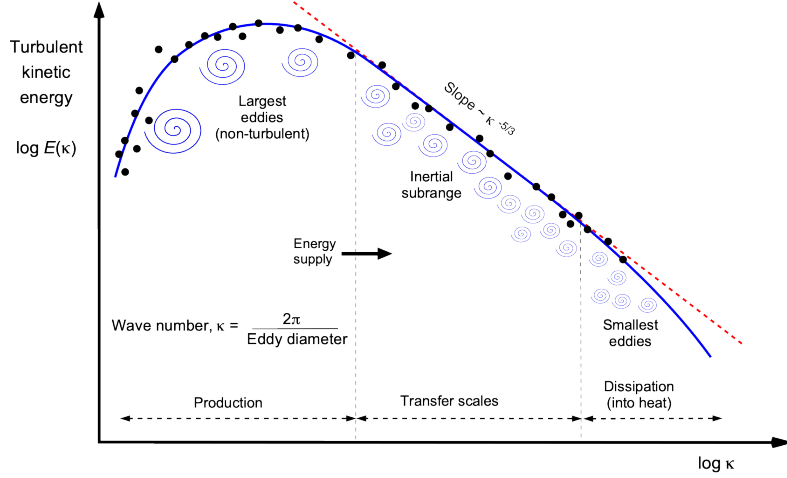
### 1.4.1 General concepts

Turbulence is the phenomenon of rapid, unsteady and random variations of the flow quantities. This contrasts with laminar flow, where the fluid moves in smooth, orderly layers, with well-defined and predictable trajectories. Turbulence arises when flow instabilities become strong enough to overcome viscous damping. Typically, the onset of turbulence is defined using the dimensionless Reynolds number. A sufficiently high Reynolds number indicates the presence of turbulence. However, for natural convection flows, the Reynolds number is not suitable. Instead, the Grashof number determines the flow regime [26, 27]. We will return to this in Section 1.6.

Turbulent flow is characterized by the presence of a multitude of vortical structures of different length scales. These vortical structures are called *eddies*. Every turbulent flow undergoes a cascade process, involving a transfer of turbulent kinetic energy,  $k$ , from larger eddies to smaller eddies. Dissipation of kinetic energy to heat through the action of molecular viscosity occurs at the scale of the smallest eddies [28]. The latter is known as the Kolmogorov's scale. The distribution of turbulent kinetic energy across different length scales is quantified using the energy spectrum function  $E(\kappa)$ , where  $\kappa$  is the eddy wave number, which is inversely proportional to the length scale  $l_e$  of the eddy:  $l_e = 2\pi/\kappa$ . Therefore, the turbulent kinetic energy, which is the kinetic energy per unit mass of the fluctuating turbulent velocity, is given by:

$$k = \int_0^\infty E(\kappa) d\kappa \quad (1.29)$$

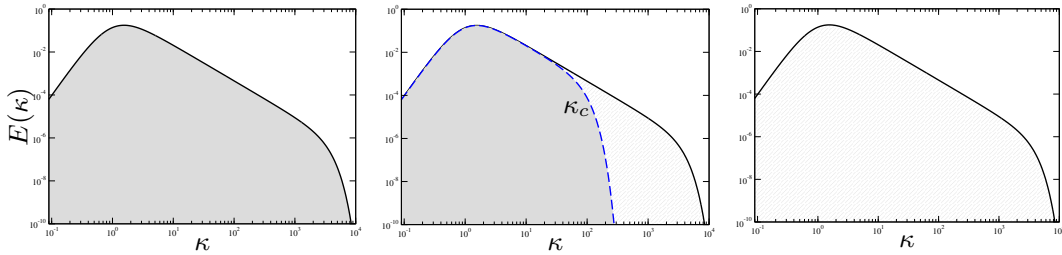
The energy spectrum  $E(\kappa)$  can be derived using dimensional analysis, leading to results that are consistent with measurements. This spectrum is illustrated in Figure 1.1.



**Figure 1.1:** *Turbulent energy spectrum. This figure is taken from [29].*

Following the presentation of the physical aspects of turbulence, we can now focus on the numerical aspect of this phenomenon and discuss its resolution and modelisation.

There are three main approaches for turbulence resolution: Direct Numerical Simulation (DNS), Large Eddy Simulation (LES), and Reynolds-Averaged Navier-Stokes (RANS). In DNS, all turbulence scales are directly resolved, thereby capturing the full range of flow fluctuations. LES resolves only the large-scale aspects of turbulence, while the smaller-scale aspects are modelled. Finally, RANS does not resolve any of the turbulence scales. Instead, it relies entirely on models to account for their effects. These approaches are illustrated in Figure 1.2.



**Figure 1.2:** *Resolved turbulence length scales for DNS, LES and RANS (from left to right). Resolved scales are shown in gray, while modeled scales are in white. This figure is adapted from [30].*

As we will see later, the geometry considered in this study is of significant dimensions. Therefore, to maintain computational complexity at an accessible level<sup>4</sup>, the RANS approach was selected and hence will be the only one described in the following sections.

<sup>4</sup>Performing a DNS or LES simulation requires a significantly more refined mesh than RANS to capture the smallest turbulence scales, which substantially increases the computational complexity.

### 1.4.2 RANS approach

Statistical turbulence modelling, or RANS modelling, treats turbulence as a deterministic chaotic phenomenon<sup>5</sup>. Since the variables are considered random, this approach is designed to produce numerical solutions for the mean fluid motion, based on an ensemble average [31].

Each flow variable  $f$  (such as velocity, pressure, temperature, etc.) is decomposed into a sum of a mean part  $\bar{f}$  and a fluctuating part  $f'$ :

$$f = \bar{f} + f' \quad (1.30)$$

As mentioned above, the Reynolds average is defined as the ensemble average:

$$\bar{f}(\mathbf{x}, t) = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{n=1}^N f_n(\mathbf{x}, t) \quad (1.31)$$

By invoking the ergodicity hypothesis, ensemble averaging is equal to time averaging, so that, for statistically stationary flows, the Reynolds average becomes:

$$\bar{f}(\mathbf{x}) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} f(\mathbf{x}, t) dt \quad (1.32)$$

However, (1.32) requires defining an infinite period  $T$ , which is not feasible for any flow in practice. Therefore, to form the time average  $\bar{f}$  of a quantity, a time  $T$  that is very long compared to the maximum period  $T_1$  of the fluctuations  $f'$  is selected. This causes no issues in practice [28].

Nonetheless, as we will see later, our study considers statistically non-stationary quantities, implying that  $\bar{f} = \bar{f}(\mathbf{x}, t)$ . In this situation, the Reynolds average can be defined in a similar way as in the case of statistically stationary flows:

$$\bar{f}(\mathbf{x}, t) = \frac{1}{T} \int_t^{t+T} f(\mathbf{x}, t) dt \quad (1.33)$$

where one must select  $T$  so that  $T_1 \ll T \ll T_2$ , with  $T_2$  the characteristic time scale of the slow variations in the flow that we do not wish to regard as belonging to the turbulence. This methodology forms the basis of URANS (Unsteady Reynolds-Averaged Navier-Stokes), which is the default RANS method implemented in Code\_Saturne<sup>6</sup>.

Applying the Reynolds average operator on the Navier-Stokes equations (1.23) and (1.24) yields:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \bar{\mathbf{u}}) = 0 \quad (1.34)$$

$$\frac{\partial (\rho \bar{\mathbf{u}})}{\partial t} + \nabla \cdot (\rho \bar{\mathbf{u}} \otimes \bar{\mathbf{u}}) = -\nabla \bar{p} + \nabla \cdot \left( 2\mu \bar{\mathbf{S}} - \frac{2}{3} \mu \text{tr}(\bar{\mathbf{S}}) \right) + \rho \mathbf{g} - \rho \mathbf{K} \bar{\mathbf{u}} - \nabla \cdot (\rho \mathbf{R}) \quad (1.35)$$

---

<sup>5</sup>A deterministic phenomenon evolves according to known physical laws and is not random. In our case, these laws are the Navier-Stokes equations. However, due to their highly non-linear nature, they exhibit chaotic behaviour, meaning they are highly sensitive to initial conditions.

<sup>6</sup>This information is not directly available in the official Code\_Saturne documentation. However, it has been confirmed by a developer on the user forum [32].

In (1.35), there is an additional term, in which  $\mathbf{R}$  appears.  $\mathbf{R}$  is the Reynolds stress tensor, which is by definition the covariance tensor of the fluctuating velocity field:

$$\mathbf{R} \equiv \overline{\mathbf{u}' \otimes \mathbf{u}'} \quad (1.36)$$

Because of  $\mathbf{R}$ , the system of equations described in (1.34)-(1.35) is not closed anymore. Therefore, one has to model the Reynolds stress tensor.

In the present study, we will adopt the Boussinesq hypothesis on turbulence [30]. This hypothesis assumes that the effective viscosity of the fluid is simply the sum of the molecular viscosity  $\mu$  and a turbulent viscosity  $\mu_t$ . Under this assumption, the Reynolds stress tensor  $\mathbf{R}$  is aligned with the strain rate tensor of the mean flow  $\bar{\mathbf{S}}$  and reads:

$$\rho \mathbf{R} = \frac{2}{3} \rho \bar{k} \mathbf{I} - 2 \mu_t \bar{\mathbf{S}}^D \quad (1.37)$$

where  $\mu_t$  needs to be modelled while  $\bar{k}$  is the turbulent kinetic energy, which was introduced in (1.29) and is now defined by:

$$\bar{k} = \frac{1}{2} \text{tr}(\mathbf{R}) \quad (1.38)$$

These types of models are known in the literature as Eddy Viscosity Models (EVM).

To conclude this section on the Reynolds averaging of the governing equations, the temperature equation must also be considered. Similarly to the turbulent viscosity  $\mu_t$ , a turbulent thermal conductivity  $\lambda_t$  is defined and added to the molecular thermal conductivity  $\lambda$ . Consequently, the system of equations (1.23)-(1.24) becomes [20, 33]:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \bar{\mathbf{u}}) = 0 \quad (1.39)$$

$$\frac{\partial (\rho \bar{\mathbf{u}})}{\partial t} + \nabla \cdot (\rho \bar{\mathbf{u}} \otimes \bar{\mathbf{u}}) = -\nabla \bar{p}_{dyn} + \nabla \cdot \left( 2(\mu + \mu_t) \bar{\mathbf{S}}^D \right) + (\rho - \rho_0) \mathbf{g} - \rho \mathbf{K} \bar{\mathbf{u}} \quad (1.40)$$

$$\frac{\partial (\rho c_p \bar{T})}{\partial t} + \nabla \cdot (\rho c_p \bar{\mathbf{u}} \bar{T}) = \nabla \cdot ((\lambda + \lambda_t) \nabla \bar{T}) + \bar{q}_f''' \quad (1.41)$$

where the turbulent thermal conductivity is derived from the turbulent viscosity as follows:

$$\lambda_t = \frac{\mu_t c_p}{\text{Pr}_t} \quad (1.42)$$

with  $\text{Pr}_t$  the turbulent Prandtl number, which is assumed constant. Moreover, one can see that the thermodynamic pressure  $\bar{p}$  in (1.40) was replaced by the dynamic pressure  $\bar{p}_{dyn}$ . This manipulation is performed to account for the hydrostatic pressure, thereby introducing a factor  $(\rho - \rho_0)$  in front of  $\mathbf{g}$ , with  $\rho_0$  being the reference density. The dynamic pressure is defined as follows:

$$\bar{p}_{dyn} = \bar{p}_{eff} - \rho_0 \mathbf{g} \cdot \mathbf{x} = \bar{p} + \frac{2}{3} \rho \bar{k} - \rho_0 \mathbf{g} \cdot \mathbf{x} \quad (1.43)$$

The *effective pressure* is a quantity with the dimensions of pressure, defined by incorporating the turbulent kinetic energy within the pressure gradient. This form is presented here because  $\bar{p}_{eff}$  sometimes appears in the literature.

### 1.4.3 Turbulence models

As we have just seen, modelling turbulence using Boussinesq's hypothesis introduces a turbulent dynamic viscosity that is added to the molecular dynamic viscosity. The goal now is to model this turbulent dynamic viscosity. To achieve this, various models can be found in the literature. These models consist of systems of transport equations for different quantities related to turbulence, such as turbulent kinetic energy, turbulent dissipation rate, or the specific dissipation rate. These transport equations must then be solved in addition to the governing equations in order to determine the turbulent viscosity [31]. Five of these turbulence models are available in Code\_Saturne, namely, the  $k - \epsilon$ ,  $k - \epsilon$  with linear production,  $k - \omega$  SST, Spalart Allmaras and the elliptic blending  $Bl - v^2 - k$  model [20]. For the sake of brevity, only two turbulence models will be presented below, namely, the  $k - \epsilon$  model and the  $k - \omega$  SST model.

#### $k - \epsilon$ model

Initially proposed by Launder and Spalding [34], this model involves the turbulent kinetic energy  $k$  and the turbulent dissipation rate  $\epsilon$ . It is a two-equation model with one transport equation for  $k$  and another for  $\epsilon$ . Since the exact equations relating  $k$  and  $\epsilon$  involve numerous unknown or unmeasurable quantities, certain coefficients are introduced and must be calibrated. In this model, the turbulent dynamic viscosity is given by:

$$\mu_t = C_\mu \frac{\bar{k}^2}{\bar{\epsilon}} \quad (1.44)$$

The transport equation for  $k$  reads:

$$\rho \frac{\partial \bar{k}}{\partial t} + \nabla \bar{k} \cdot (\rho \bar{\mathbf{u}}) - \nabla \cdot \left( \left( \mu + \frac{\mu_t}{\sigma_k} \right) \nabla \bar{k} \right) = \mathcal{P} + \mathcal{G} - \rho \bar{\epsilon} \quad (1.45)$$

and the transport equation for  $\epsilon$  is given by:

$$\rho \frac{\partial \bar{\epsilon}}{\partial t} + \nabla \bar{\epsilon} \cdot (\rho \bar{\mathbf{u}}) - \nabla \cdot \left( \left( \mu + \frac{\mu_t}{\sigma_\epsilon} \right) \nabla \bar{\epsilon} \right) = C_{\epsilon_1} \frac{\bar{\epsilon}}{\bar{k}} (\mathcal{P} + (1 - C_{\epsilon_3}) \mathcal{G}) - \rho C_{\epsilon_2} \frac{\bar{\epsilon}^2}{\bar{k}} \quad (1.46)$$

where  $\mathcal{P}$  and  $\mathcal{G}$  stand for the production term created by mean shear and the production term created by gravity effects, respectively. They are given by:

$$\mathcal{P} = 2\mu_t \bar{\mathbf{S}}^D : \bar{\mathbf{S}}^D - \frac{2}{3} \rho \bar{k} \text{tr}(\bar{\mathbf{S}}) \quad (1.47)$$

$$\mathcal{G} = \frac{1}{\rho} \frac{\mu_t}{\text{Pr}_t} \nabla \rho \cdot \mathbf{g} \quad (1.48)$$

The value of the constant coefficients implemented in Code\_Saturne are given in Table 1.1 [20].

| $C_\mu$ | $C_{\epsilon_1}$ | $C_{\epsilon_2}$ | $\sigma_k$ | $\sigma_\epsilon$ |
|---------|------------------|------------------|------------|-------------------|
| 0.09    | 1.44             | 1.92             | 1          | 1.3               |

**Table 1.1:** *Coefficients of the  $k - \epsilon$  model.*

Note that  $C_{\epsilon_3} = 0$  if  $\mathcal{G} \geq 0$  and  $C_{\epsilon_3} = 1$  if  $\mathcal{G} \leq 0$ .

### $k - \omega$ SST model

The  $k - \omega$  model is a two-equation turbulence model, with one transport equation for the turbulent kinetic energy  $k$  and another one for the specific dissipation rate  $\omega$ . This model was initially proposed by Wilcox [35]. However, we will focus here on the  $k - \omega$  Shear Stress Transport (SST) model proposed by Menter, which is the one implemented in Code\_Saturne [36, 37]. This model enables a transition from a  $k - \omega$  formulation near the walls to a  $k - \epsilon$  formulation in the fully turbulent regions. This model therefore avoids the problems associated with the  $k - \epsilon$  formulation, which is known to not perform well in the very near-wall zone, by using the  $k - \omega$  formulation, which offers good resolution of turbulent boundary layers [31]. In this approach, the turbulent dynamic viscosity is defined as:

$$\mu_t = \frac{a_1 \rho \bar{k}}{\max(a_1 \omega, \sqrt{2 \bar{\mathbf{S}} : \bar{\mathbf{S}}} F_2)} \quad (1.49)$$

where the function  $F_2$  is given by:

$$F_2 = \tanh(\Phi_2^2) \quad \text{with} \quad \Phi_2 = \max\left(\frac{2\sqrt{\bar{k}}}{\beta^* \bar{\omega} y}, \frac{500\mu}{\rho \bar{\omega} y^2}\right) \quad (1.50)$$

with  $y$  the normal distance to the wall. The transport equation for  $k$  reads:

$$\rho \frac{\partial \bar{k}}{\partial t} + \nabla \bar{k} \cdot (\rho \bar{\mathbf{u}}) - \nabla \cdot ((\mu + \sigma_k \mu_t) \nabla \bar{k}) = \mathcal{P} - \beta^* \rho \bar{k} \bar{\omega} \quad (1.51)$$

where  $\mathcal{P}$  is defined in (1.47) and the transport equation for  $\omega$  is given by:

$$\rho \frac{\partial \bar{\omega}}{\partial t} + \nabla \bar{\omega} \cdot (\rho \bar{\mathbf{u}}) - \nabla \cdot ((\mu + \sigma_\omega \mu_t) \nabla \bar{\omega}) = \alpha \mathcal{P} - \beta \rho \bar{\omega}^2 + 2(1 - F_1) \frac{\sigma_{\omega_2} \rho}{\bar{\omega}} (\nabla \bar{k}) \cdot (\nabla \bar{\omega}) \quad (1.52)$$

The constant coefficients implemented in Code\_Saturne are given in Table 1.2<sup>7</sup>.

| $a_1$ | $\beta^*$ | $\alpha_1$ | $\alpha_2$ | $\beta_1$ | $\beta_2$ | $\sigma_{k_1}$ | $\sigma_{k_2}$ | $\sigma_{\omega_1}$ | $\sigma_{\omega_2}$ |
|-------|-----------|------------|------------|-----------|-----------|----------------|----------------|---------------------|---------------------|
| 0.31  | 0.09      | 5/9        | 0.44       | 3/40      | 0.0828    | 0.85           | 1              | 0.5                 | 0.856               |

**Table 1.2:** Coefficients of the  $k - \omega$  SST model.

Finally, if we denote by  $\psi$  the constant coefficients  $\alpha$ ,  $\beta$ ,  $\sigma_k$  and  $\sigma_\omega$ , their values can be computed as follows:

$$\psi = F_1 \psi_1 + (1 - F_1) \psi_2 \quad (1.53)$$

where the function  $F_1$  is defined by:

$$F_1 = \tanh(\Phi_1^4) \quad \text{with} \quad \Phi_1 = \min\left[\max\left(\frac{\sqrt{\bar{k}}}{\beta^* \bar{\omega} y}, \frac{500\mu}{\rho \bar{\omega} y^2}\right), \frac{4\sigma_{\omega_2} \bar{k}}{\text{CD}_{k\omega} y^2}\right] \quad (1.54)$$

with  $\text{CD}_{k\omega}$  the positive portion of the cross-diffusion term of (1.52) and it reads:

$$\text{CD}_{k\omega} = \max\left(2 \frac{\sigma_{\omega_2} \rho}{\bar{\omega}} (\nabla \bar{k}) \cdot (\nabla \bar{\omega}), 10^{-20}\right) \quad (1.55)$$

---

<sup>7</sup>These coefficients are not directly provided in the documentation but can be found in the source code.

### 1.4.4 Law of the wall

As mentioned earlier, there is a wide variety of turbulence models. Some of these models are inherently unable to capture near-wall phenomena effects, such as the  $k - \epsilon$  model, while others apply natural boundary conditions<sup>8</sup> at the walls for each quantity, such as the  $k - \omega$  model.

Turbulence models can be divided into two main categories. The first category, known as the *high-Reynolds* models, requires a mesh size near the walls large enough for the first computational points to be placed in a fully turbulent zone, thus removing the influence of the wall, as in the case of the  $k - \epsilon$  model. The second category, known as *low-Reynolds* models, requires a very fine mesh near the walls in order to capture all phenomena related to the wall, as for the  $k - \omega$  model [31].

However, in practice, it is not always feasible to use a sufficiently refined mesh near the wall. Therefore, one may wonder whether it is possible to adapt turbulence models in order to avoid the need for consequent near-wall resolution.

To illustrate the situation, let us consider the velocity field. If the mesh is sufficiently refined and the first computational point is close enough to the wall, then a linear evaluation at this point of the velocity gradient poses no problem. In this case, a turbulence model applying natural boundary conditions would be capable of capturing all phenomena related to the near-wall region. On the other hand, if the first computational point is too far from the wall, the error introduced by a linear evaluation of the gradient becomes significant<sup>9</sup>. In this case, accurately estimating wall friction is essential to correct the linearly constructed gradient. This is achieved using wall functions. Hence, a high-Reynolds model, such as the  $k - \epsilon$  model, inherently requires a near-wall law, whereas a low-Reynolds model, like the  $k - \omega$  model, will only require a near-wall law if the mesh is not sufficiently refined. Recall that the  $k - \omega$  SST model allows a transition from  $k - \omega$  near the wall to  $k - \epsilon$  in the fully turbulent zone. Therefore, this model will also require a near-wall law if the mesh is not refined enough.

We will now present such a wall function. To do so, let us first consider the simplified momentum balance in the first boundary cell [20]:

$$(\mu + \mu_t) \frac{\partial \bar{u}}{\partial y} = \tau_{wall} \quad (1.56)$$

where  $y$  is the normal distance to the wall,  $\bar{u}$  the velocity projected parallel to the wall and  $\tau_{wall}$  the wall shear stress. Let us now introduce the friction velocity  $\bar{u}_\tau$ , the normalized velocity  $\bar{u}^+$  and the dimensionless distance from the wall  $y^+$ , defined by:

$$\bar{u}_\tau = \sqrt{\frac{\tau_{wall}}{\rho}}; \quad \bar{u}^+ = \frac{\bar{u}}{\bar{u}_\tau}; \quad y^+ = \frac{y \bar{u}_\tau}{\nu} \quad (1.57)$$

with  $\nu$  the kinematic viscosity. Using (1.57), (1.56) can be written as:

$$\left(1 + \frac{\mu_t}{\mu}\right) \frac{\partial \bar{u}^+}{\partial y^+} = 1 \quad (1.58)$$

---

<sup>8</sup>By "natural boundary conditions", we refer to the actual boundary conditions (zero velocity if the wall is stationary, zero turbulent kinetic energy, etc.).

<sup>9</sup>The velocity gradient at the wall is far from being linear.

In the previous section, we introduced two-equation EVMs. However, Prandtl originally proposed an algebraic model with zero transport equations, based on a mixing length  $l_m$  [38]. According to this model, the turbulent viscosity is expressed as:

$$\nu_t = l_m^2 \left| \frac{\partial \bar{u}}{\partial y} \right| \quad (1.59)$$

where the mixing length read  $l_m = \kappa y$  with  $\kappa$  being the Von Kármán constant, set to 0.42 in Code\_Saturne [20]. Based on this model, one can solve (1.58) in the very near-wall region dominated by laminar effects (*i.e.*,  $\nu \gg \nu_t$ ), as well as in the near-wall region dominated by turbulence (*i.e.*,  $\nu \ll \nu_t$ ), leading to:

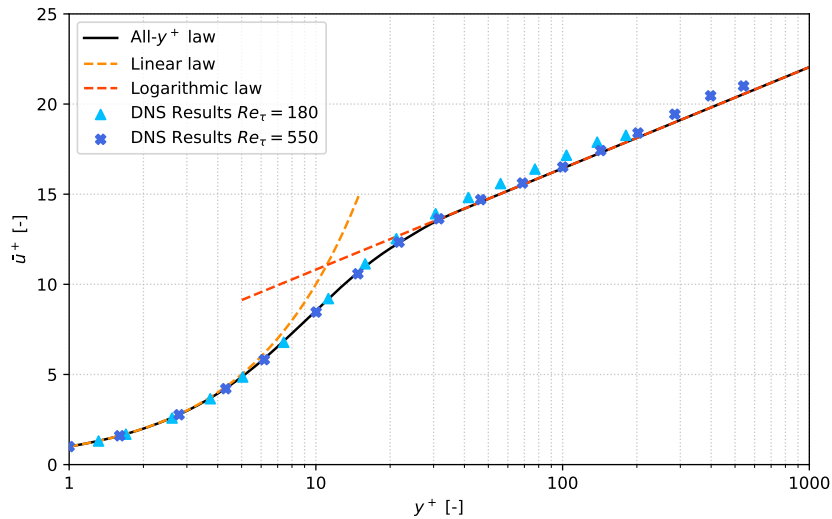
$$\bar{u}^+ = \begin{cases} y^+ & \text{if } \nu \gg \nu_t \\ \frac{1}{\kappa} \ln(y^+) + C & \text{if } \nu \ll \nu_t \end{cases} \quad (1.60)$$

with  $C$  a constant, set to 5.2 in Code\_Saturne [20]. However, a clear delimitation of the laminar and turbulent dominance zones should be expressed in terms of  $y^+$ . Prandtl and Kármán observed that for flow between two plates, these limits were well defined [39]. This led to what is now known as the *universal law of the wall* which, in practice, is used for all types of geometries. These limits read:

$$\nu \gg \nu_t \text{ if } y^+ < 5 \quad (1.61)$$

$$\nu \ll \nu_t \text{ if } y^+ > 30 \quad (1.62)$$

Hence, we can observe the existence of a buffer zone, located for  $y^+$  values between 5 and 30, where the dimensionless velocity profile  $\bar{u}^+$  is neither linear nor logarithmic. However, having a region where  $\bar{u}^+$  is not clearly defined is unacceptable. The first method used in Code\_Saturne to address this issue is to apply the linear profile if  $\bar{u}^+ < 1/\kappa$  and the logarithmic profile otherwise but this approach is fundamentally coarse in the buffer zone. Subsequently, a second option is to implement a function that is valid for any  $y^+$ . Such a function is depicted in Figure 1.3.



**Figure 1.3:** Universal wall law. Linear, logarithmic, and all- $y^+$  relations for  $\bar{u}^+$ . The added points are DNS results and can be found at [40].



In Code\_Saturne, the mathematical formulation of this function is given by:

$$u^+(y^+) = f_{blend}(y^+) \cdot y^+ + \left( \frac{1}{\kappa} \ln(y^+) + C \right) \cdot \left( 1 - \exp \left( - \left( \frac{y^+}{y_0} \right)^n \right) \right) \cdot (1 - f_{blend}(y^+)) \quad (1.63)$$

with  $n = 2.25$ ,  $y_0 = 14.5$  and  $f_{blend}$  a blending function given by:

$$f_{blend}(y^+) = \exp \left( -\frac{1}{4} \chi (y^+)^3 \right) \quad (1.64)$$

where  $\chi$  is a constant set to  $1.0674 \times 10^{-3}$ . Note that in Figure 1.3,  $Re_\tau$  is a local Reynolds number defined in the very near-wall region, given by [30]:

$$Re_\tau = \frac{\delta u_\tau}{\nu} \quad (1.65)$$

with  $\delta$  the thickness of the turbulent boundary layer. Note that from a dimensional analysis, one can show that this Reynolds number scales with the ratio of  $\nu_t$  and  $\nu$ . This is precisely the origin of the terms *high-Reynolds* and *low-Reynolds* used to describe turbulence models and the near-wall mesh refinement, as discussed earlier.

To conclude this section, we want to point out that in (1.57), the velocity  $\bar{u}$  is made dimensionless solely by the wall shear stress  $\tau_{wall}$ . This approach is known in the literature as the one-scale friction velocity wall function. However, as in many numerical solvers, Code\_Saturne makes the velocity  $\bar{u}$  dimensionless using both the wall shear stress  $\tau_{wall}$  and the turbulent kinetic energy  $\bar{k}$ . This method is referred to as the two-scale friction velocity wall function. Consequently, the friction velocity is given by:

$$\bar{u}_\tau = \frac{\tau_{wall}}{\rho u_k} \quad (1.66)$$

with  $u_k$  the friction velocity based on the turbulent kinetic energy in the first cell, defined as:

$$u_k = \sqrt{g \frac{\nu \bar{u}}{y} + (1 - g) \sqrt{C_\mu \bar{k}}} \quad (1.67)$$

where the value of  $C_\mu$  depends on the selected turbulence model and  $g$  is a blending factor defined by:

$$g = \exp \left( -\frac{\sqrt{\bar{k}} y}{11\nu} \right) \quad (1.68)$$

In the above, we only developed wall laws for the velocity field. However, wall laws are also used for the temperature field. These will not be developed in this report in order not to overload it. The interested reader is invited to read [41].

## 1.5 The Oberbeck-Boussinesq approximation

The Oberbeck-Boussinesq approximation is a frequently used simplification of the Navier-Stokes equations when considering buoyancy-driven flows. This approximation assumes that density variations are negligible except when they appear in the buoyancy term (i.e. the term multiplied by gravity  $\mathbf{g}$ ). The key aspect of the Oberbeck-Boussinesq approximation is that gravity is strong enough to set the fluid in motion due to density differences. These density differences are assumed to be proportional to temperature variations. This approach significantly reduces the complexity of the governing equations, making it particularly useful for modelling natural convection flows, which are the focus of this study. Mathematically, each occurrence of the density  $\rho$  is replaced by the reference density  $\rho_0$  except in the gravity term where it is expressed as:

$$\rho(T) = \rho_0 - \rho_0\beta(T - T_0) \quad (1.69)$$

where  $T_0$  is the reference temperature and  $\beta$  is the thermal expansion coefficient, defined by:

$$\beta = -\frac{1}{\rho} \left( \frac{\partial \rho}{\partial T} \right)_p \quad (1.70)$$

Expression (1.69) is obtained by writing the Taylor series for  $\rho$  as a function of  $T$ , considering the pressure  $p$  to be constant, and keeping only the first two terms of the series. Since  $\rho$  is kept constant except in the gravity term, it is clear that the continuity equation (1.23) implies that the velocity field is divergence-free:

$$\nabla \cdot \mathbf{u} = 0 \quad (1.71)$$

For the momentum equation, it is useful to perform a manipulation with the pressure gradient to simplify the expression. Consequently, a new pressure  $P$  is defined as follows:

$$P = \frac{p - \rho_0 \mathbf{g} \cdot \mathbf{x}}{\rho_0} \quad (1.72)$$

so that (1.24) becomes:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla P + \nu \nabla^2 \mathbf{u} - \beta(T - T_0)\mathbf{g} - \mathbf{K} \mathbf{u} \quad (1.73)$$

Note that the energy equation (1.25) does not simplify further, except that the density  $\rho$  is replaced by the reference density  $\rho_0$ .

It is important to emphasize that the Oberbeck-Boussinesq approximation is not applicable in all situations. First, this assumption is only meaningful when the flow dynamics are primarily governed by buoyancy forces, which corresponds to a high Richardson number [24, 27]. The definition of this dimensionless number will be given in Section 1.6. Secondly, this approximation is only valid when the density variation is small. Mathematically, this condition reads:

$$\frac{\Delta \rho}{\rho_0} \ll 1 \quad (1.74)$$

The validity of this condition for our application will be detailed in Section 2.4.1.

## 1.6 Dimensionless analysis

### 1.6.1 Dimensionless governing equations

We will now perform a dimensional analysis of the governing equations. For simplicity, we will consider the system of equations under the Oberbeck-Boussinesq approximation, with no pressure losses nor volumetric heat source. This methodology might seem counter-intuitive, but since we are conducting a dimensional study, it will allow us to highlight the most important conclusions. All geometric quantities are made dimensionless using the characteristic length  $l_0$ , velocities are made dimensionless using the reference velocity  $u_0$ , and time is made dimensionless using both  $l_0$  and  $u_0$ . The pressure  $P$  and the temperature  $T$ , are made dimensionless as follows:

$$\hat{P} = \frac{P - P_0}{\rho_0 u_0^2} \quad \text{and} \quad \hat{T} = \frac{T - T_0}{T_1 - T_0} \quad (1.75)$$

where we used the hat symbol " $\hat{\phantom{x}}$ " to denote dimensionless quantities, the subscript "0" to denote reference quantities and where  $T_1$  can be taken as the temperature at the bottom of the domain, for instance where the assemblies are located, so that  $T_1 - T_0$  denotes the temperature difference across the water column. Therefore, equations (1.70), (1.73) and (1.25) become [24]:

$$\hat{\nabla} \cdot \hat{\mathbf{u}} = 0 \quad (1.76)$$

$$\frac{D\hat{\mathbf{u}}}{D\hat{t}} = -\hat{\nabla}\hat{P} + \frac{1}{\text{Re}}\hat{\nabla}^2\hat{\mathbf{u}} - \frac{\text{Gr}}{\text{Re}^2} \left( \frac{\mathbf{g}}{g} \right) \hat{T} \quad (1.77)$$

$$\frac{D\hat{T}}{D\hat{t}} = \frac{1}{\text{RePr}}\hat{\nabla}^2\hat{T} \quad (1.78)$$

Note that we chose to write the left-hand side of (1.76)-(1.78) in terms of total derivatives for clarity. Moreover, Re denotes the Reynolds number, Pr the Prandtl number and Gr the Grashof number. Their definitions are the following:

$$\text{Re} = \frac{l_0 u_0}{\nu}; \quad \text{Pr} = \frac{c_{p0} \mu}{\lambda} = \frac{\nu}{\kappa}; \quad \text{Gr} = \frac{g\beta(T_1 - T_0)l_0^3}{\nu^2} \quad (1.79)$$

with  $\kappa$  the thermal diffusivity. However, in this study, the focus is on natural convection. Thus, one may wonder how to define the reference velocity  $u_0$ . Indeed, this velocity does not appear to be explicitly defined at first. It is therefore practical to consider it as [27]:

$$u_{0_N} = \sqrt{g\beta(T_1 - T_0)l_0} \quad (1.80)$$

where the subscript "N" indicates a quantity related to natural convection. Hence, we can express the Reynolds number in terms of this reference velocity, which yields:

$$\text{Re}_N = \sqrt{\text{Gr}} \quad (1.81)$$

Consequently, (1.77) and (1.78) become:

$$\frac{D\hat{\mathbf{u}}}{D\hat{t}} = -\hat{\nabla}\hat{P} + \frac{1}{\sqrt{\text{Gr}}}\hat{\nabla}^2\hat{\mathbf{u}} - \left( \frac{\mathbf{g}}{g} \right) \hat{T} \quad (1.82)$$

$$\frac{D\hat{T}}{D\hat{t}} = \frac{1}{\sqrt{\text{Gr}}\text{Pr}}\hat{\nabla}^2\hat{T} \quad (1.83)$$

Equations (1.82) and (1.83) allow for an interpretation of the Grashof number. Indeed, in view of (1.79), this number is defined as the ratio between buoyancy forces and viscous forces. Consequently, if the Grashof number is low, the buoyancy forces will not be sufficient to set the fluid in motion so that there is no fluid motion. On the other hand, as mentioned in Section 1.4.1, turbulence arises when flow instabilities become strong enough to overcome viscous damping. Therefore, when the Grashof number is high, the magnitude of viscous forces becomes negligible compared to the magnitude of buoyancy forces, leading to the development of a turbulent flow. This transition typically occurs around a Grashof number of  $10^9$  [27].

In Section 1.5, we discussed the Richardson number. Its definition is the following:

$$Ri = \frac{Gr}{Re^2} = \frac{g\beta(T_1 - T_0)l_0}{u_0^2} \quad (1.84)$$

In view of (1.77) and (1.84), the meaning of this number is straightforward. A low value indicates that the flow is primarily driven by forced convection while a high value indicates that the flow is driven by natural convection. Indeed, if  $Ri \ll 1$ , the buoyancy term in (1.77) is negligible. On the other hand, if  $Ri \gg 1$ , the buoyancy term cannot be neglected.

To conclude this section, we introduce the Rayleigh number  $Ra$ . In practice, the Grashof number is not directly used to quantify buoyancy-driven flows. Instead, it is used through the Rayleigh number:

$$Ra = GrPr = \frac{g\beta(T_1 - T_0)l_0^3}{\nu\kappa} \quad (1.85)$$

This number is extremely useful as it allows to reduce the number of dimensionless groups in (1.82) and (1.83). For conventional fluids such as air and water, the Prandtl number typically ranges between 0.7 and 7. Therefore, the Rayleigh number marking the onset of turbulent flow is expected to differ by at most one order of magnitude from the Grashof number that characterizes this transition, given the aforementioned relatively narrow range of Prandtl numbers. This is the reasoning adopted by Incropera *et al.*, who estimate the onset of turbulence around  $Ra \approx 10^9$  [26].

However, in practice, this transition is far less abrupt and depends on multiple factors, such as the Prandtl number and the aspect ratio of the domain. In recent decades, the Rayleigh number range has been divided into sub-regimes to better describe the flow characteristics [42, 9]:

- Around  $Ra \approx 2 \times 10^5$ , the flow becomes turbulent. This regime, sometimes called the *soft turbulence regime*, is characterized by coherent convective structures and laminar boundary layers.
- Around  $Ra \approx 4 \times 10^7$ , the flow enters the *hard turbulence regime*, where turbulence in the bulk intensifies, while boundary layers remain laminar.
- Finally, around  $Ra \approx 10^{14}$ , the *ultimate regime* is reached, marked by intense turbulence in the bulk and by turbulent boundary layers.

### 1.6.2 Heat and mass transfer analogy

As mentioned in the state of the art, a major aspect of this study is the evaporation of water at the free surface. This process involves not only heat transfer but also mass transfer. As will be detailed in Section 2.3, evaporation will be modelled by considering a thin layer, referred to as a film, composed of a binary mixture of air and water vapour. Therefore, some mathematical models have to be developed accordingly.

In the case of a mixture of miscible fluid species, one can write the mass equation for each species. In absence of chemical reactions and using Fick's law to model the diffusion flux, the mass balance of species  $i$  reads [24]:

$$\frac{\partial(\rho Y_i)}{\partial t} + \nabla \cdot (\rho Y_i \mathbf{u}) = -\nabla \cdot (\rho \mathcal{D}_i \nabla Y_i) \quad (1.86)$$

with  $Y_i$  and  $\mathcal{D}_i$  the mass fraction and the mass diffusivity of species  $i$ , respectively, and  $\rho$  the density of the mixture. However, density does not only vary with temperature. Indeed, the concentration of the different species can also lead to density variations. Consequently, the momentum equation has to be modified in order to take into account the variations of the species concentration. One can use an approach similar to the Oberbeck-Boussinesq approximation for the temperature variations, so that the expression of the density in the gravity term becomes [24, 43]:

$$\rho(T, C_i) = \rho_0 - \rho_0 \beta (T - T_0) - \rho_0 \eta (C_i - C_{i_0}) \quad (1.87)$$

where  $C_i$  is the concentration of species  $i$ , defined by  $\rho Y_i$ ;  $C_{i_0}$  the reference concentration of species  $i$  and  $\eta$  the concentration expansion coefficient. The latter is defined in an analogous manner as the thermal expansion coefficient  $\beta$ :

$$\eta = -\frac{1}{\rho} \left( \frac{\partial \rho}{\partial C_i} \right)_p \quad (1.88)$$

Expression (1.87) is obtained by writing a double Taylor series expansion of  $\rho$  as a function of  $T$  and  $C_i$ . Using (1.87), (1.73) becomes [24]:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla P + \nu \nabla^2 \mathbf{u} - \beta (T - T_0) \mathbf{g} - \eta (C_i - C_{i_0}) \mathbf{g} \quad (1.89)$$

To obtain the dimensionless system of equations, we introduce the following quantities:

$$\hat{Y}_i = \frac{Y_i - Y_{i_0}}{Y_{i_1} - Y_{i_0}} \quad \text{and} \quad \hat{C}_i = \frac{C_i - C_{i_0}}{C_{i_1} - C_{i_0}} \quad (1.90)$$

where  $Y_{i_1}$  and  $C_{i_1}$  can for instance be taken as the mass fraction and concentration of species  $i$  at the top of the film, respectively. The dimensionless system of equations for combined heat and mass transfer, by writing the left-hand side of (1.86) in terms of material derivatives, is then given by [24]:

$$\hat{\nabla} \cdot \hat{\mathbf{u}} = 0 \quad (1.91)$$

$$\frac{D\hat{\mathbf{u}}}{D\hat{t}} = -\hat{\nabla} \hat{P} + \frac{1}{\text{Re}} \hat{\nabla}^2 \hat{\mathbf{u}} - \frac{\text{Gr}}{\text{Re}^2} \left( \frac{\mathbf{g}}{g} \right) \hat{T} - \frac{\text{Gr}_c}{\text{Re}^2} \left( \frac{\mathbf{g}}{g} \right) \hat{C}_i \quad (1.92)$$

$$\frac{D\hat{Y}_i}{D\hat{t}} = \frac{1}{\text{ReSc}} \hat{\nabla}^2 \hat{Y}_i \quad (1.93)$$

$$\frac{D\hat{T}}{D\hat{t}} = \frac{1}{\text{RePr}} \hat{\nabla}^2 \hat{T} \quad (1.94)$$

In (1.93),  $Sc$  denotes the Schmidt number while in (1.92),  $Gr_c$  denotes the concentration Grashof number. They are defined as follows:

$$Sc = \frac{\mu}{\rho \mathcal{D}} = \frac{\nu}{\mathcal{D}} \quad \text{and} \quad Gr_c = \frac{g\eta(C_{i_1} - C_{i_0})l_0^3}{\nu^2} \quad (1.95)$$

Similarly to the thermal Rayleigh number  $Ra$ , a concentration Rayleigh number  $Ra_c$  can be defined as [43]:

$$Ra_c = Gr_c Sc = \frac{g\eta(C_{i_1} - C_{i_0})l_0^3}{\nu \mathcal{D}} \quad (1.96)$$

However, in practice, (1.96) is rarely used. Instead, a definition of the concentration Rayleigh number based on the density difference of the whole mixture across the domain of interest is directly employed [26]:

$$Ra_c = \frac{g(\rho_1 - \rho_0)l_0^3}{\mu \mathcal{D}} \quad (1.97)$$

As we can see from system (1.91)-(1.94), the equations describing heat and mass transfer exhibit strong similarities. This similarity is widely used in practice. Indeed, conducting measurements related to heat transfer presents numerous challenges. Consequently, a more straightforward approach is to consider an analogous mass transfer process, which is easier to measure. This allows the development of correlations between the different dimensionless numbers related to mass transfer, which can then be converted into correlations for the dimensionless numbers associated with heat transfer [44]. Table 1.3 highlights these various analogies. Note that we also added the Nusselt number  $Nu$  and the Sherwood number  $Sh$  for completeness. They are used to characterize a heat and mass transfer between a fluid and an interface, respectively. The symbols  $h_h$  and  $h_m$  denote the convective heat transfer coefficient and the convective mass transfer coefficient, respectively.

| Heat transfer system |                                            | Mass transfer system |                                                              |
|----------------------|--------------------------------------------|----------------------|--------------------------------------------------------------|
| Prandtl number       | $Pr = \frac{\nu}{\kappa}$                  | Schmidt number       | $Sc = \frac{\nu}{\mathcal{D}}$                               |
| Grashof number       | $Gr = \frac{g\beta\Delta TL^3}{\nu^2}$     | Grashof number       | $Gr_c = \frac{gL^3}{\nu^2} \frac{\Delta\rho}{\rho}$          |
| Rayleigh number      | $Ra = \frac{g\beta\Delta TL^3}{\nu\kappa}$ | Rayleigh number      | $Ra_c = \frac{gL^3}{\nu\mathcal{D}} \frac{\Delta\rho}{\rho}$ |
| Nusselt number       | $Nu = \frac{h_h L}{\lambda}$               | Sherwood number      | $Sh = \frac{h_m L}{\mathcal{D}}$                             |

**Table 1.3:** *Dimensionless groups for analogy systems.*

Finally, the Lewis number is defined as the ratio of thermal diffusivity to mass diffusivity:

$$Le = \frac{Sc}{Pr} = \frac{\kappa}{\mathcal{D}} \quad (1.98)$$

This dimensionless number is used when heat and mass transfer occur simultaneously. It relates the thickness of the thermal boundary layer to that of the concentration boundary layer.

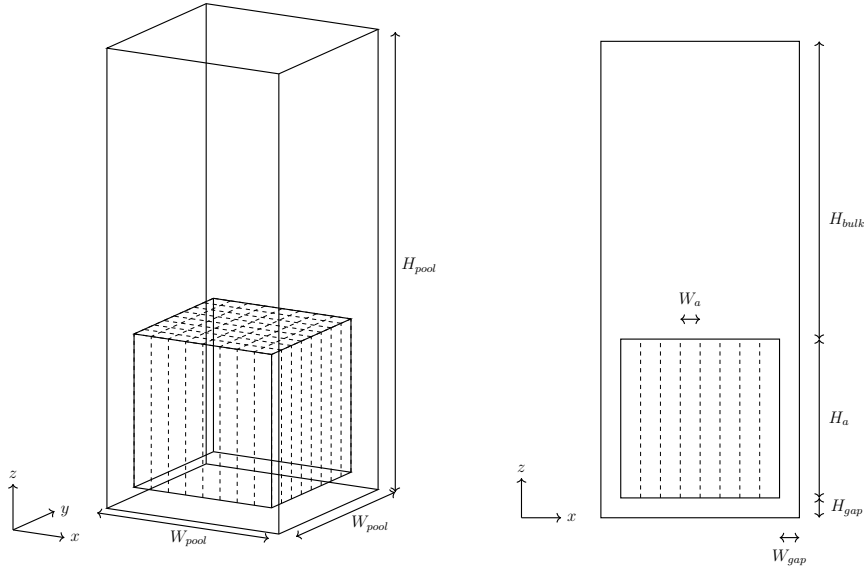
# Chapter 2

## Numerical setup

This chapter provides the models for evaporation, head loss in the assemblies, and heat dissipation from the assemblies. We then provide the dimensions of the computational domain and present the mesh generation process. Next, we describe how the solver handles the Navier-Stokes equations, followed by a discussion of the turbulence model selection. Finally, a brief description of the implemented user subroutines is given.

### 2.1 Computational domain

Figure 2.1 illustrates the geometry of the problem. The computational domain consists of a parallelepiped representing the pool, with a rack of spent fuel assemblies located at the bottom. In this study, we will consider a single rack composed of 8x8 assemblies. The fluid region above the assemblies will be referred to as the *bulk*, while the space between the rack and the pool walls is termed the *gap*. The letter  $H$  denotes a height, while  $W$  denotes a width. Finally, the subscript "a" refers to an assembly. Since two distinct geometries will be considered, the specific dimensions will be provided later.



**Figure 2.1:** Computational domain. On the left, a 3D view of the geometry; on the right, a cross-section in the  $x$ - $z$  plane. The figure is not to scale. Attention is drawn to the coordinate system used, where the  $z$ -coordinate increases from the bottom to the top of the domain.

## 2.2 Initial and boundary conditions

In normal operating conditions, the water temperature cannot exceed 45°C, while the ambient air temperature cannot exceed 26°C [45]. In this study, the initial water temperature is set to 30°C, corresponding to the normal operating conditions of the La Hague pools [46]. For the air, the temperature is set to 25°C. Additionally, the water is considered initially quiescent (*i.e.* motionless everywhere).

Regarding the boundary conditions, the four lateral walls and the bottom of the pool are considered adiabatic (*i.e.* no heat flux). The upper boundary, representing the free surface of the water, is modelled as a shear-free surface with a non-zero Neumann boundary condition to account for evaporation. Therefore, the free surface is the only boundary of the domain allowing heat exchange with the surroundings.

## 2.3 Modelling the evaporation

For specified thermal conditions at the free surface and for given ambient conditions, the evaporation rate is determined by the gas-side transport phenomena, as the flow of air above the pool dictates the convective behaviour of heat and mass transfer at the interface. However, this study focuses on the thermal mixing on the water side, specifically the mixing of water driven by natural convection induced by the heat generated by the fuel rods at the bottom of the pool. As a result, only the water flow within the pool is simulated, and the air above is not explicitly considered. Consequently, the evaporation at the free surface needs to be modelled.

In this study, we will use the mathematical model developed by Hay [10] to quantify the evaporation rate at the free surface. As previously mentioned, this model treats the free surface as a shear-free surface, where a non-zero Neumann boundary condition is applied to account for the heat flux across the interface. It is important to stress that this model assumes a quiescent air side environment, which is physically reasonable in the context of deactivation pools.

By performing an energy balance, it is evident that the heat added to the interface from the water side,  $\dot{q}_{added}''$ , is equal to the heat lost to the ambient gas side environment,  $\dot{q}_{lost}''$ . Since, according to the model assumptions, there is no fluid motion at the free surface, heat transfer from the water side ( $\dot{q}_{added}''$ ) occurs solely by conduction [26]. On the other hand, heat losses to the gas side can occur through convection ( $\dot{q}_{conv}''$ ) and latent heat losses due to evaporation ( $\dot{q}_{evap}''$ ). Consequently, the energy balance is written as:

$$-\lambda_{w,int} \left. \frac{\partial T}{\partial z} \right|_{w,int} = \dot{q}_{conv}'' + \dot{q}_{evap}'' \quad (2.1)$$

with  $\lambda_{w,int}$  the thermal conductivity of water at the mean interface temperature  $T_{int}$ . Note that in this section, the terms *interface* and *free surface* will refer to the same concept and will be used interchangeably. Moreover, the thermal conductivity  $\lambda_{w,int}$  will be retained on the left-hand side of (2.1) to remain consistent with the implementation in Code\_Saturne, but we will elaborate on this later.



As mentioned in Section 1.6.2, the evaporation model developed by Hay [10] assumes that, on the gas side, the transition from the interface conditions to those far from the surface occurs within a finite thickness layer, referred to here as a *film*. This film consists of a binary mixture of water vapour and air. It is now appropriate to introduce some notations. The subscripts "w", "v", and "a" refer to physical properties related to water, water vapour, and air, respectively. The subscripts "int", "f", and " $\infty$ " denote properties at the interface, within the film, and in the air far from the interface, respectively.

We will now develop the expressions for the two terms on the right-hand side of 2.1. Note that thermophysical properties such as density, heat capacity, dynamic viscosity, and thermal conductivity of water, water vapour, and air, in the film or far from the interface, will be provided in Section 2.4 to avoid overloading the current developments. The kinematic viscosity  $\nu$  and thermal diffusivity  $\kappa$  can be computed from these properties.

### 2.3.1 Latent heat losses

By expanding the term  $\dot{q}_{evap}''$ , we get:

$$\dot{q}_{evap}'' = \dot{m}_{evap}'' h_{lv} \quad (2.2)$$

with  $\dot{m}_{evap}''$  the evaporative mass flux and  $h_{lv}$  interface-temperature-dependent latent heat of evaporation. The latter can be calculated using Watson's relation [47]:

$$h_{lv} = h_{ref} \left( \frac{T_c - T_{int}}{T_c - T_{ref}} \right)^{0.38} \quad (2.3)$$

where  $T_c$  is the critical temperature and the subscript "ref" denotes properties at the normal boiling point. For water, we have  $T_c = 647.096$  K [48],  $T_{ref} = 373.15$  K and  $h_{ref} = 2257$  kJ/kg [26]. The evaporative mass flux, on the other hand, is given by [10]:

$$\dot{m}_{evap}'' = \text{Sh} \frac{\rho_f \mathcal{D}_f}{L_c} \log(1 + B_m) \quad (2.4)$$

In this expression,  $L_c$  denotes the characteristic length of the transfer surface, defined as the ratio of the transfer surface area to the transfer surface perimeter<sup>1</sup> [49],  $B_m$  the mass transfer driving force and  $\mathcal{D}_f$  the film mass diffusivity for the binary mixture of air and vapour. The latter is given by [50]:

$$\mathcal{D}_f = 1.87 \times 10^{-10} \frac{T_f^{2.072}}{p_0} \quad (2.5)$$

with  $p_0$  the pressure far from the interface, in units of atmospheres. The film temperature  $T_f$  is taken as the average of the temperatures at the interface and far from the interface ( $T_f \approx (T_{int} + T_\infty)/2$ ). On the other hand, the mass transfer driving force reads [51]:

$$B_m = \frac{Y_{v,\infty} - Y_{v,int}}{Y_{v,int} - 1} \quad (2.6)$$

with  $Y_v$  the mass fraction of water vapour.

---

<sup>1</sup>In our case, the characteristic length is given by  $L_c = W_{pool}/4$  (see Figure 2.1).

The mass fraction of water vapour can be determined from the water vapour pressure as follows:

$$Y_{v,\star} = \frac{p_{v,\star}M_w}{p_{v,\star}M_w + (p_0 - p_{v,\star})M_a} \quad (2.7)$$

where the subscript " $\star$ " may refer to the interface ("*int*"), the film ("*f*"), or the gas side far from the interface (" $\infty$ ") and  $M_w$  and  $M_a$  are respectively the molar masses of water and air. Furthermore, we emphasize that in (2.7),  $p_0$  is expressed in Pascals. To evaluate the water vapour pressure, one can use the following expression [48]:

$$p_v = p_c \exp \left( \frac{T_c}{T} (a_1\tau + a_2\tau^{1.5} + a_3\tau^3 + a_4\tau^{3.5} + a_5\tau^4 + a_6\tau^{7.5}) \right) \quad (2.8)$$

with  $p_c = 22.064$  MPa, the critical pressure of water and  $\tau = 1 - T/T_c$ . The value of the different coefficients is given in Table 2.1.

| $a_1$       | $a_2$      | $a_3$       | $a_4$      | $a_5$       | $a_6$      |
|-------------|------------|-------------|------------|-------------|------------|
| -7.85951783 | 1.84408259 | -11.7866497 | 22.6807411 | -15.9618719 | 1.80122502 |

**Table 2.1:** Coefficients in (2.8).

Finally, the Sherwood number can be computed using the following correlation [52]:

$$\text{Sh} = 0.23\text{Sc}^{0.33}\text{Ra}_c^{0.321} \quad (2.9)$$

where the Schmidt number and concentration Rayleigh number are given by:

$$\text{Sc} = \frac{\mu_f}{\rho_f \mathcal{D}_f} \quad \text{and} \quad \text{Ra}_c = \frac{g(\rho_\infty - \rho_{int})L_c^3}{\mu_f \mathcal{D}_f} \quad (2.10)$$

The density  $\rho_{int}$  in (2.10) denotes the density of the binary mixture of water vapour and air at the interface and is given by:

$$\rho_{int} = \frac{p_{v,int}M_w}{RT_{int}} + \frac{(p_0 - p_{v,int})M_a}{RT_{int}} \quad (2.11)$$

As a final remark, it was assumed that  $Le = 1$ , indicating that the thicknesses of the concentration and thermal boundary layers above the interface are equal. Therefore, one can use a gas-side concentration Rayleigh number defined using properties based on the local water vapour mass fraction and temperature [10].

### 2.3.2 Convective heat losses

The convective heat flux can be expressed as follows:

$$\dot{q}_{conv}'' = h_h(T_{int} - T_\infty) \quad (2.12)$$

where  $h_h$  is the heat transfer coefficient. The latter can be estimated from the following correlation for a horizontal flat surface that is warmer than the ambient air above [53]:

$$\frac{h_h L_c}{\lambda_f} = \text{Nu} = \text{Sh} = 0.54\text{Ra}^{0.25} \quad (2.13)$$

In (2.13), we used the fact that  $\text{Nu} = \text{Sh}$  since we consider that  $Le = 1$  [54]. Consequently, and developing the expression for the thermal Rayleigh number, we get:

$$\dot{q}_{conv}'' = 0.54\lambda_f \left( \frac{g\beta_\infty}{\kappa_f \nu_f L_c} \right)^{0.25} (T_{int} - T_\infty)^{1.25} \quad (2.14)$$

### 2.3.3 Computational aspect

It is important to note that in our simulations, the evaporative mass flux in (2.4), the latent heat losses in (2.2), and the convective heat losses in (2.14) are computed locally, *i.e.* for each cell located on the free surface. Consequently, the boundary condition in (2.1) is also defined locally. To enable a simplified analysis, the free-surface temperature, the evaporative mass flux, and both types of heat fluxes are subsequently area-averaged. For a given quantity  $\phi$ , the area-averaged value is computed as follows:

$$\langle \phi \rangle_S = \frac{1}{A} \int_S \phi dA = \frac{1}{A} \sum_{i \in \mathcal{N}} \phi_i |A_i| \quad (2.15)$$

where  $\mathcal{N}$  is the set of all cells on the free surface. We emphasize here that  $\langle \phi \rangle$  refers strictly to an area-averaged quantity, not a time-averaged one. In what follows, the notation  $\langle \phi \rangle_{xy}$  will be used to denote an area-averaged quantity over the free surface.

As for the total evaporated mass or volume over the entire free surface, the contribution from each cell is simply summed. From this point onwards, we will refer to this quantity directly, without introducing any specific notation.

### 2.3.4 Free surface descent

As water evaporates, the free surface gradually lowers. To model this descent, the entire geometry must be remeshed. However, at each computation step, the amount of evaporated volume is minimal. Therefore, performing a remeshing procedure at every time step would result in significant computational costs with little to no noticeable effect on the behaviour of the flow. Consequently, we chose to reduce the height of the computational domain, *i.e.* the height of the mesh, only when 0.5% of the remaining volume has evaporated. As a result, at any given moment, we distinguish between the height of the computational domain  $H_C$  and the height of the free surface  $H_S$ , which represents the free-surface level that would result from the actual evaporated volume. In order to fix ideas, let  $t^n$  represent the time after  $n$  time steps. We suppose that the solution is known at  $t^n$ . Also, let  $\Delta t^n$  be the current time step, *i.e.*  $\Delta t^n = t^{n+1} - t^n$ . The volume that has evaporated during the current time step is given by:

$$V_{evap}^{t^n} = \sum_{i \in \mathcal{N}} \Delta t^n \frac{\dot{m}_{evap,i}'' A_i}{\rho_{int,i}} \quad (2.16)$$

where  $\mathcal{N}$  is the set of all cells on the free surface. A cumulative evaporated volume is then defined, which sums the evaporated volumes at each time step:

$$V_{evap}^\Sigma = \sum_{t^n} V_{evap}^{t^n} \quad (2.17)$$

Since the volume of water inside the pool  $V_w$  is known, the fraction of evaporated volume  $f_{evap}$  can be calculated as:

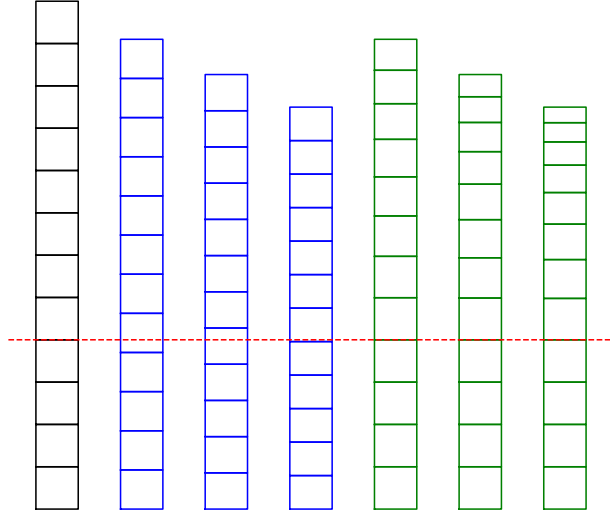
$$f_{evap} = \frac{V_{evap}^\Sigma}{V_w} \quad (2.18)$$

If  $f_{evap}$  is less than 0.005, then the fraction of evaporated volume has not reached the threshold fraction, and therefore no remeshing is performed. However, if  $f_{evap}$  is greater than 0.005, a remeshing procedure must be applied. This procedure will be explained in what follows.

We now consider that the geometry must be remeshed. Accordingly, the height of the computational domain must be multiplied by a *shrink* factor  $f_{shrink} = 1 - f_{evap}$ . In other words,  $H_C^{t^{n+1}} = f_{shrink} \cdot H_C^{t^n}$ . The most straightforward implementation would be to multiply the  $z$ -coordinate of each vertex by this factor, distributing the deformation uniformly over the entire mesh height in order to avoid significantly altering the aspect ratio of certain cells. However, this approach does not take into account the presence of the assemblies at the bottom of the pool. As we will detail later, these assemblies are characterized by a porosity profile, a volumetric heat generation profile, and head losses. Numerically, we therefore want to avoid modifying these zones. Consequently, we choose to deform only the part of the mesh above the assemblies and to introduce a progressive deformation starting from the free surface. To clarify, let  $z$  denote the coordinate of a vertex before deformation, and  $z'$  its coordinate after deformation. We can then write:

$$z' = \begin{cases} z & \text{if } z < z_{a,top} \\ [1 - (1 - f_{shrink})h(s)] \cdot z & \text{else} \end{cases} \quad \text{where } s = \frac{z - z_{a,top}}{H_C^{t^n} - z_{a,top}} \quad (2.19)$$

with  $z_{a,top}$  the  $z$ -coordinates of the top of the assemblies,  $H_C^{t^n}$  the height of the computational mesh after  $n$  time steps and  $h(s)$  a function to be specified. The idea behind introducing this function is that it equals 0 just above the assemblies (no deformation) and 1 at the free surface (maximum deformation). This function must be monotonically increasing, and its slope dictates the local deformation rate of the cells. Therefore,  $h(s) = s^k$  seems like a straightforward choice. In our case, we chose  $k = 2$  to avoid excessive deformation of the cells near the free surface. Figure 2.2 illustrates the deformation obtained after three consecutive remeshing steps using this method.



**Figure 2.2:** In black, a simplified representation of the undeformed mesh. In blue, the deformation obtained by multiplying the  $z$ -coordinates of all vertices by  $f_{shrink}$ . In green, the deformation following the procedure detailed above. The red dashed line indicates the top of the assemblies.

At the end of the deformation,  $V_{evap}^\Sigma$  is reset to zero,  $H_C$  is updated, and the mesh is recalculated with the new vertex coordinates to update various quantities, such as the water volume  $V_w$ . From a purely numerical perspective, the coordinates of the cell centres, their volumes, the areas of the various cell faces, and other related quantities are obviously recalculated as well.

## 2.4 Fluid properties

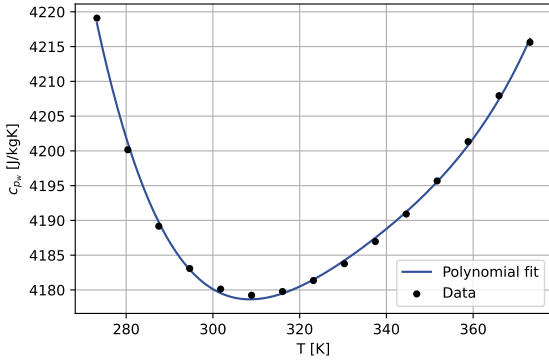
We will now provide the thermophysical properties of water, water vapour, and air, as well as the procedure to determine the thermophysical properties of the binary mixture in the film. In general, for a temperature-dependent property  $\phi$ , a 4th-order polynomial is used to model this quantity:

$$\phi(T) = \sum_{i=0}^4 \alpha_i T^i \quad \text{for } T \in [273.15; 373.15]\text{K} \quad (2.20)$$

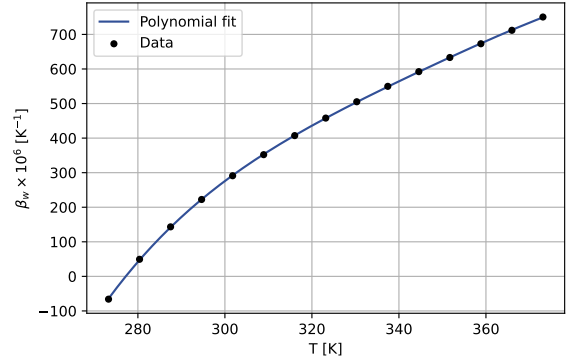
The CoolProp database and library were used to obtain the thermophysical property data needed to fit these polynomials [55].

### 2.4.1 Water properties

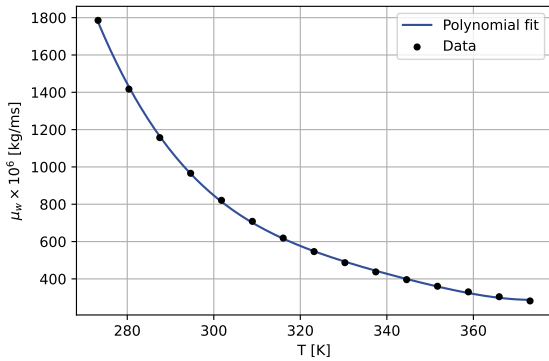
The polynomial fits for the heat capacity, the thermal expansion coefficient, the dynamic viscosity and the thermal conductivity of water are represented in Figure 2.3. The coefficients used for these fits are given in Table 2.2. Those polynomial fits constitute the equations of state that allow us to close the system of equations under the Oberbeck-Boussinesq approximation given in (1.71) and (1.73).



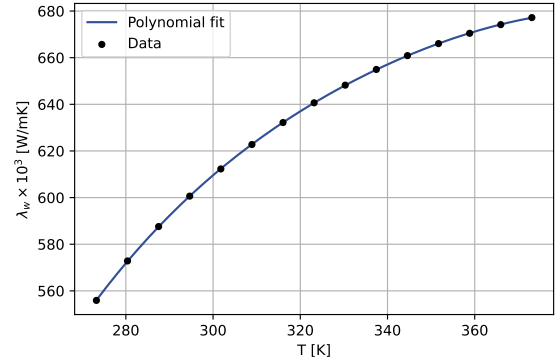
(a) Heat capacity



(b) Thermal expansion coefficient



(c) Dynamic viscosity



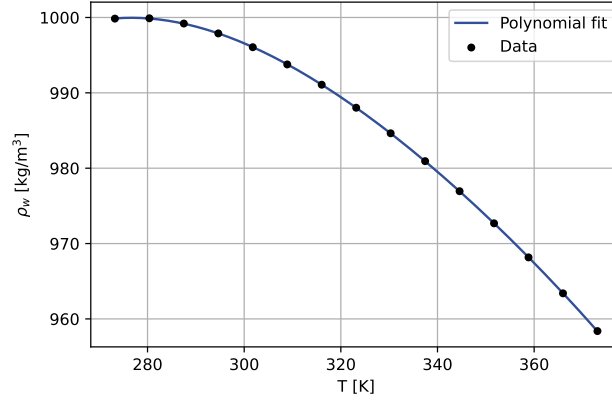
(d) Thermal conductivity

**Figure 2.3:** Thermophysical properties of water as a function of temperature.

|            | $c_{p_w}$ [J/kgK]       | $\beta_w$ [K <sup>-1</sup> ] | $\mu_w$ [Ns/m <sup>2</sup> ] | $\lambda_w$ [W/mK]       | $\rho_w$ [kg/m <sup>3</sup> ] |
|------------|-------------------------|------------------------------|------------------------------|--------------------------|-------------------------------|
| $\alpha_0$ | $4.105 \times 10^4$     | $-8.876 \times 10^{-2}$      | $4.604 \times 10^{-1}$       | -6.607                   | $-1.138 \times 10^3$          |
| $\alpha_1$ | $-4.376 \times 10^2$    | $1.005 \times 10^{-3}$       | $-5.325 \times 10^{-3}$      | $7.667 \times 10^{-2}$   | $2.396 \times 10^1$           |
| $\alpha_2$ | 1.948                   | $-4.293 \times 10^{-6}$      | $2.321 \times 10^{-5}$       | $-3.143 \times 10^{-4}$  | $-9.930 \times 10^{-2}$       |
| $\alpha_3$ | $-3.858 \times 10^{-3}$ | $8.223 \times 10^{-9}$       | $-4.507 \times 10^{-8}$      | $5.911 \times 10^{-7}$   | $1.826 \times 10^{-4}$        |
| $\alpha_4$ | $2.872 \times 10^{-6}$  | $-5.922 \times 10^{-12}$     | $3.288 \times 10^{-11}$      | $-4.266 \times 10^{-10}$ | $-1.293 \times 10^{-7}$       |

**Table 2.2:** Coefficients for the thermophysical properties of water.

If the Oberbeck-Boussinesq approximation is not applied, an equation of state is required for the density in order to close the system given in (1.23)-(1.25). Similarly to the other properties, a polynomial fit is used for this purpose. This fit is shown in Figure 2.4, and the coefficients are provided in Table 2.2.



**Figure 2.4:** Water density as a function of temperature.

This equation of state will allow us to assess the validity of the condition stated in (1.74) for the Oberbeck-Boussinesq approximation. Recalling that the initial water temperature is taken at 303.15 K (see Section 2.2) and that the temperature can go up to a maximum of 373.15 K, we obtain:

$$\frac{\Delta\rho}{\rho_0} \approx 0.037 \ll 1 \quad (2.21)$$

Therefore, we can consider the Oberbeck-Boussinesq approximation as valid.

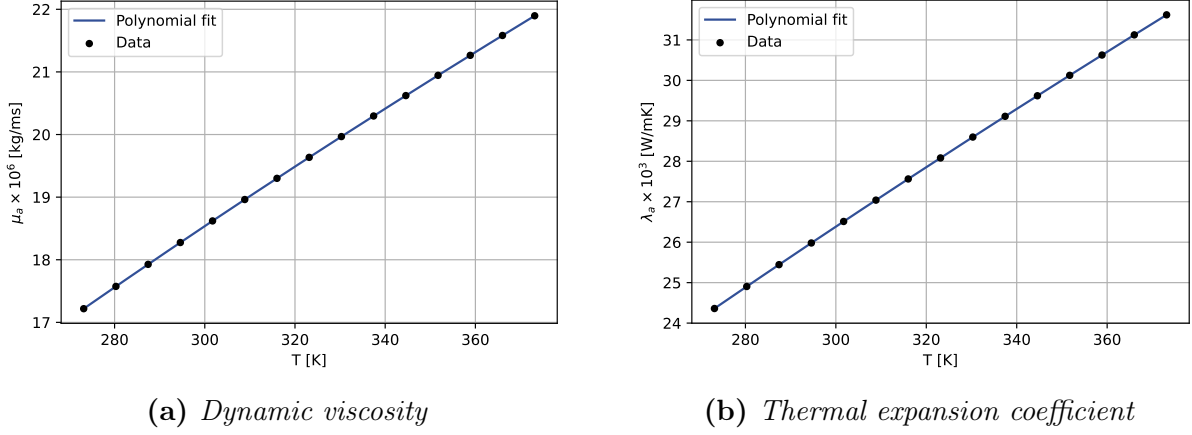
## 2.4.2 Air properties

In Section 2.2, we set the ambient air temperature to 298.15 K. Additionally, we consider a relative humidity of 40% and an ambient pressure of one atmosphere. Using these conditions, the partial pressure of water vapour far from the interface and the air density can be calculated [10]. Finally, considering air as a perfect gas, the thermal expansion coefficient is simply given by  $\beta_\infty = 1/T_\infty$ . These values are listed in Table 2.3.

| $T_\infty$ [K] | $p_0$ [Pa] | $p_{v,\infty}$ [Pa] | $\rho_\infty$ [kg/m <sup>3</sup> ] | $\beta_\infty$ [K <sup>-1</sup> ] |
|----------------|------------|---------------------|------------------------------------|-----------------------------------|
| 298.15         | 101325     | 1270                | 1.18                               | $3.354 \times 10^{-3}$            |

**Table 2.3:** Gaseous mixture properties far from the interface.

Regarding the other thermophysical properties of air, the heat capacity is assumed constant and taken at  $1007 \text{ J/kgK}^2$ . For the dynamic viscosity and the thermal conductivity, polynomial fits are shown in Figure 2.5, with the corresponding coefficients listed in Table 2.4.



**Figure 2.5:** Thermophysical properties of air as a function of temperature.

|            | $\mu_a \text{ [Ns/m}^2\text{]}$ | $\lambda_a \text{ [W/mK]}$ |
|------------|---------------------------------|----------------------------|
| $\alpha_0$ | $-4.594 \times 10^{-7}$         | $-1.201 \times 10^{-3}$    |
| $\alpha_1$ | $8.474 \times 10^{-8}$          | $1.166 \times 10^{-4}$     |
| $\alpha_2$ | $-9.654 \times 10^{-11}$        | $-1.099 \times 10^{-7}$    |
| $\alpha_3$ | $9.790 \times 10^{-14}$         | $1.071 \times 10^{-10}$    |
| $\alpha_4$ | $-4.689 \times 10^{-17}$        | $-4.931 \times 10^{-14}$   |

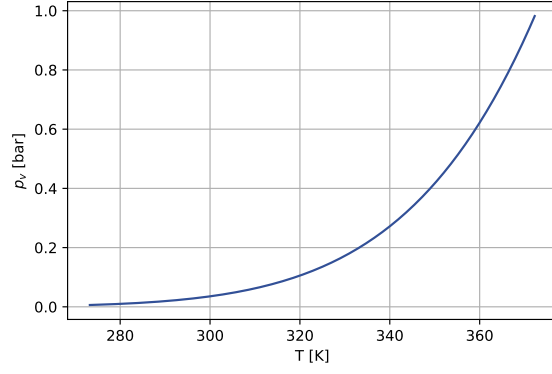
**Table 2.4:** Coefficients for the thermophysical properties of air.

### 2.4.3 Water vapour properties

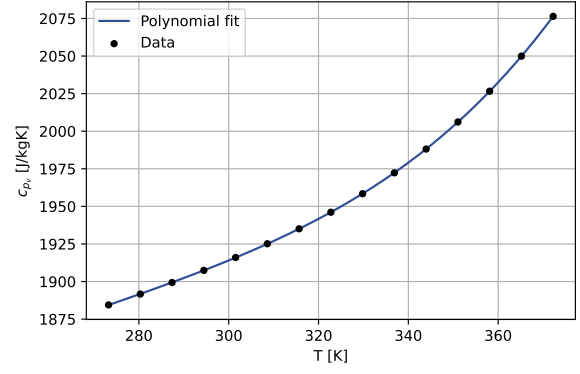
To determine the thermophysical properties of a fluid, one must specify its thermodynamic state. For instance, temperature and pressure can be provided. In Section 2.4.1, the properties of water were evaluated at various temperatures and at a pressure of 1 atmosphere. Given that water is an almost incompressible fluid, this approach is appropriate. Similarly, the properties of air in Section 2.4.2 were also determined at 1 atmosphere. This choice is entirely valid since the pressure  $p_0$  remains constant.

However, for water vapour, we cannot assume a fixed pressure over the entire temperature range. Since water vapour is considered saturated, its properties must be determined using the saturation pressure corresponding to the given temperature. To this end, we can use (2.8). The partial pressure of water vapour, along with the polynomial fits for the heat capacity, the dynamic viscosity, and the thermal conductivity are then shown in Figure 2.6, with the corresponding coefficients provided in Table 2.5.

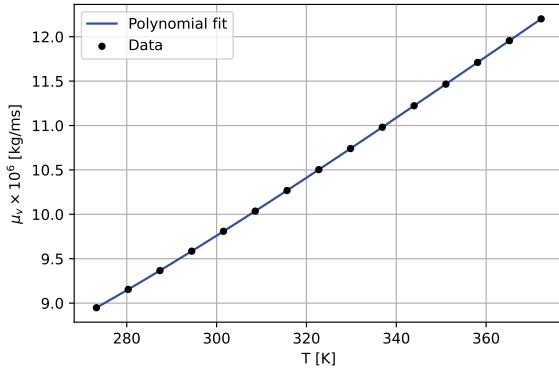
<sup>2</sup>Indeed, over the temperature range from  $0^\circ\text{C}$  to  $100^\circ\text{C}$ , the heat capacity of air varies by only 0.49%.



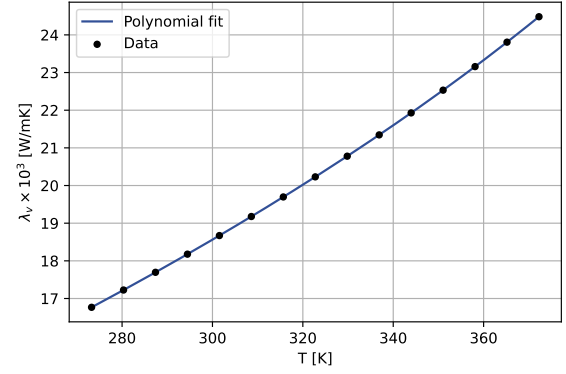
(a) *Partial pressure*



(b) *Heat capacity*



(c) *Dynamic viscosity*



(d) *Thermal expansion coefficient*

**Figure 2.6:** *Partial pressure and thermophysical properties of water vapour as a function of temperature.*

|            | $c_{p_v}$ [J/kgK]       | $\mu_v$ [Ns/m <sup>2</sup> ] | $\lambda_v$ [W/mK]      |
|------------|-------------------------|------------------------------|-------------------------|
| $\alpha_0$ | $3.296 \times 10^3$     | $1.836 \times 10^{-5}$       | $2.484 \times 10^{-2}$  |
| $\alpha_1$ | $-2.567 \times 10^1$    | $-1.551 \times 10^{-7}$      | $-2.484 \times 10^{-4}$ |
| $\alpha_2$ | $1.585 \times 10^{-1}$  | $7.056 \times 10^{-10}$      | $1.486 \times 10^{-6}$  |
| $\alpha_3$ | $-4.193 \times 10^{-4}$ | $-1.162 \times 10^{-12}$     | $-3.337 \times 10^{-9}$ |
| $\alpha_4$ | $4.168 \times 10^{-7}$  | $7.154 \times 10^{-16}$      | $3.037 \times 10^{-12}$ |

**Table 2.5:** *Coefficients for the thermophysical properties of water vapour.*

Note that with CoolProp, it is necessary to explicitly specify the gaseous phase, as using the saturation pressure corresponds to the curve separating the liquid and gaseous phases in the phase diagram.

#### 2.4.4 Film properties

We will now present the method used to determine the thermophysical properties within the film, which are necessary for the developments in Section 2.3. This approach follows the evaporation model described in [10].



### Film density

The film density is directly computed using the following relation:

$$\rho_{int} = \frac{p_{v,f} M_w}{RT_f} + \frac{(p_0 - p_{v,f}) M_w}{RT_f} \quad (2.22)$$

where the partial pressure of water vapour inside the film  $p_{v,f}$  is computed using (2.8).

### Film heat capacity

The heat capacity in the film is simply obtained as a mass-weighted average using the film mass fraction  $Y_{v,f}$ :

$$c_{p_f} = Y_{v,f} c_{p_{v,f}} + (1 - Y_{v,f}) c_{p_{a,f}} \quad (2.23)$$

where  $Y_{v,f}$  is computed using (2.7), and  $c_{p_{v,f}}$  and  $c_{p_{a,f}}$  denote  $c_{p_v}$  and  $c_{p_a}$  evaluated at  $T_f$ . These last two are given in Sections 2.4.3 and 2.4.2 respectively.

### Film dynamic viscosity

The dynamic viscosity for gas mixtures reads [56]:

$$\mu_f = \frac{x_{a,f} \mu_{a,f}}{x_{a,f} \xi_{11} + x_{v,f} \xi_{12}} + \frac{x_{v,f} \mu_{v,f}}{x_{a,f} \xi_{21} + x_{v,f} \xi_{22}} \quad (2.24)$$

where  $x_{a,f}$  and  $x_{v,f}$  are the molar fractions of air and water vapour in the film, respectively. Their expressions are given by:

$$x_{a,f} = \frac{1 - Y_{v,f}}{1 + \left( \frac{M_a}{M_w} - 1 \right) Y_{v,f}} \quad (2.25)$$

$$x_{v,f} = 1 - x_{a,f} \quad (2.26)$$

Finally, the expression for  $\xi_{ij}$  is given by:

$$\xi_{ij} = \frac{\left[ 1 + \left( \frac{\mu_i}{\mu_j} \right)^{1/2} \left( \frac{M_j}{M_i} \right)^{1/4} \right]^2}{\sqrt{8 \left( 1 + \frac{M_i}{M_j} \right)}} \quad (2.27)$$

with  $i$  and  $j$  being either "a" or "v". Note that  $Y_{v,f}$  is once again computed via (2.7) while  $\mu_{a,f}$  and  $\mu_{v,f}$  denote  $\mu_a$  and  $\mu_v$  evaluated at  $T_f$ . These last two are given in Sections 2.4.3 and 2.4.2 respectively.

### Film thermal conductivity

The following approximate formula for the thermal conductivity of gas mixtures will be used [57]:

$$\lambda_f = \frac{\lambda_{a,f}}{1 + G_{12} \frac{x_{v,f}}{x_{a,f}}} + \frac{\lambda_{v,f}}{1 + G_{21} \frac{x_{a,f}}{x_{v,f}}} \quad (2.28)$$

with  $x_{a,f}$  and  $x_{v,f}$  the molar fractions of air and water vapour in the film, which are computed using (2.25) and (2.26).

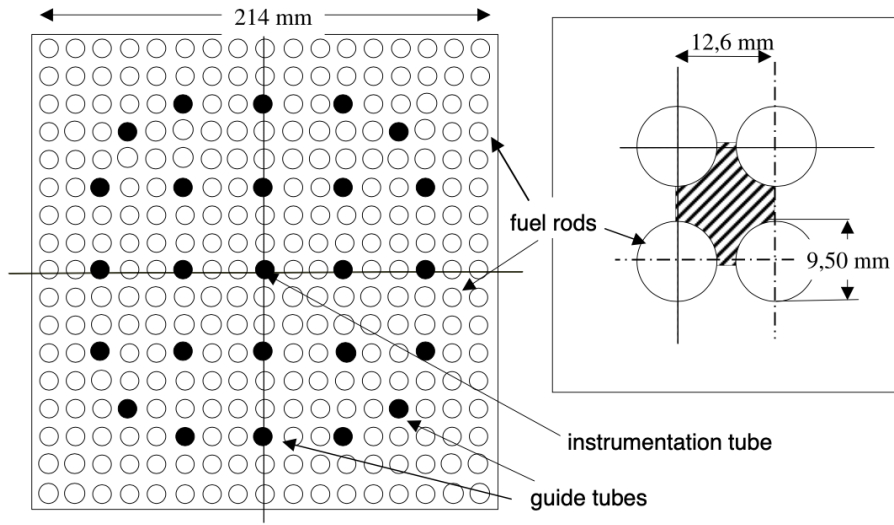
The values of  $G_{ij}$  is then computed by:

$$G_{ij} = \frac{1.065}{2\sqrt{2}} \left(1 + \frac{M_i}{M_j}\right)^{-1/2} \left[1 + \left(\frac{\mu_i M_j}{\mu_j M_i}\right)^{1/2} \left(\frac{M_i}{M_j}\right)^{1/4}\right]^2 \quad (2.29)$$

with  $i$  and  $j$  being either "a" or "v". In (2.28)  $\lambda_{a,f}$  and  $\lambda_{v,f}$  denote  $\lambda_a$  and  $\lambda_v$  evaluated at  $T_f$ . These last two are given in Sections 2.4.3 and 2.4.2 respectively.

## 2.5 Porosity and head loss due to the assemblies

A typical French PWR fuel assembly consists of 264 fuel rods, 24 guide tubes, and one instrumentation tube [7]. The transverse cross-section of one such assembly is shown in Figure 2.7.



**Figure 2.7:** Fuel assembly. Transverse arrangement of the fuel rod lattice. On the left, a transverse cross-section; on the right, a detailed view of a typical cell. This figure is adapted from [7].

When the assemblies are immersed at the bottom of a deactivation pool, the space between the rods is filled with water. Instead of using an extremely refined mesh around the assemblies to model the 289 rods and then solving the equations within this complex geometry, which would significantly increasing computational times, a common approach is to model an assembly as a porous medium. A portion of the total volume of the assembly, also known as the porous volume  $V_p$ , is divided into a fluid volume  $V_f$  and a solid volume  $V_s$ . The porosity is then defined as:

$$\phi = \frac{V_f}{V_p} \quad (2.30)$$

With reference to Figures 2.1 and 2.7, we have  $W_a = 214$  mm and  $d_t = 9.5$  mm, with  $W_a$  the width of an assembly and  $d_r$  the diameter of a rod. Consequently, the porosity of a typical assembly is given by:

$$\phi_a = \frac{V_p - V_s}{V_p} = \frac{H_a W_a^2 - n_r \pi (d_r/2)^2 H_a}{H_a W_a^2} \approx 0.55 \quad (2.31)$$

where  $n_r$  is the number of rods inside an assembly and  $H_a$  is the height of an assembly, set to 4 m [7].

An assembly can be represented as a porous medium, but it is important to keep in mind that in reality, the fluid experiences friction against the various rods, resulting in head losses. Consequently, one must model these head losses. This can be done through the velocity head loss tensor  $\mathbf{K}$  in (1.27), which, due to the anisotropic nature of the geometry of an assembly, is defined as:

$$\mathbf{K} = \begin{pmatrix} K_{xx} & 0 & 0 \\ 0 & K_{yy} & 0 \\ 0 & 0 & K_{zz} \end{pmatrix} \quad (2.32)$$

We must now determine an expression for  $K_{xx}$ ,  $K_{yy}$  and  $K_{zz}$ . We will begin by determining  $K_{zz}$ , which represents the head loss in the axial direction of the rods. To achieve this, we will follow the model developed by Todreas *et al.* [58, 59]. The mechanical energy loss over a length  $L$  per unit fluid volume due to friction is then expressed similarly to that of flow in a pipe:

$$\frac{\Delta p}{L} = \frac{f_D}{D_{h,b}} \frac{\rho V_b^2}{2} \quad (2.33)$$

where  $f_D$  is the Darcy friction factor, which will be developed for the particular geometry of an assembly,  $D_{h,b}$  is the hydraulic diameter of the cross-section occupied by the fluid around the rod bundle, and  $V_b$  is the average axial velocity over this cross-section. The friction factor is then defined as proportional to the Reynolds number of the bundle  $Re_b$ , up to a bundle friction coefficient  $C_{f,b}$  that still needs to be determined:

$$f_D = \frac{C_{f,b}}{Re_b^n} \quad \text{with} \quad Re_b = \frac{\rho V_b D_{h,b}}{\mu} \quad (2.34)$$

The coefficient  $C_{f,b}$  and the exponent  $n$  in (2.34) will then depend on the flow regime, whether it is laminar or turbulent so that  $f_D$  is defined piecewise by:

$$f_D = \begin{cases} \frac{C_{f,b,L}}{Re_b} & \text{for } Re_b \leq Re_{b,L} \\ \frac{C_{f,b,L}}{Re_b}(1 - \Psi)^\gamma + \frac{C_{f,b,T}}{Re_b^{0.18}}\Psi^\gamma & \text{for } Re_{b,L} < Re_b < Re_{b,T} \\ \frac{C_{f,b,T}}{Re_b^{0.18}} & \text{for } Re_b \geq Re_{b,T} \end{cases} \quad (2.35)$$

where  $Re_{b,L}$  denotes the end of the laminar regime,  $Re_{b,T}$  the beginning of the turbulent regime, and  $\Psi$  is an intermittency factor defined as:

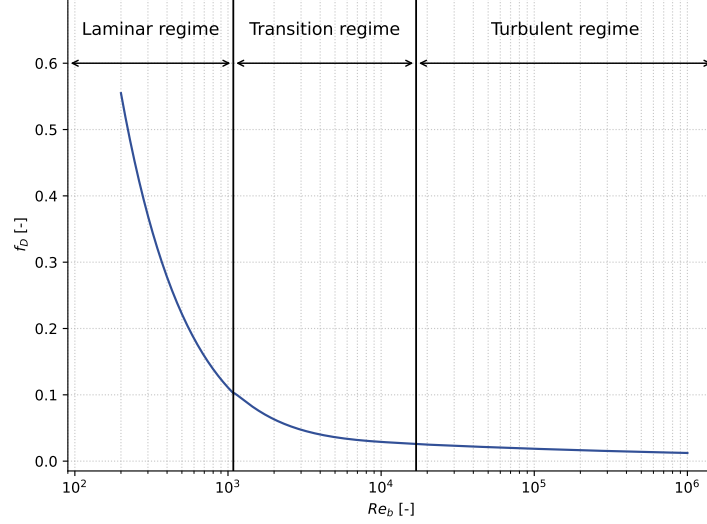
$$\Psi = \frac{\log_{10}(Re_b) - \log_{10}(Re_{b,L})}{\log_{10}(Re_{b,T}) - \log_{10}(Re_{b,L})} \quad (2.36)$$

For an assembly with the dimensions depicted in Figure 2.7, the different values constants in 2.35 are given in Table 2.6. The exponent  $\gamma$  in (2.35) is adjusted to minimize the root mean square of the prediction error in the transition regime [59]. The procedure used to determine the other constants is detailed in Appendix A.

| $C_{f,b,L}$ | $C_{f,b,T}$ | $Re_{b,L}$ | $Re_{b,T}$ | $\gamma$ |
|-------------|-------------|------------|------------|----------|
| 110.971     | 0.148       | 1083.27    | 16920.81   | 0.33     |

**Table 2.6:** Value of the different constants for the head loss model.

In the case of purely natural convection flow within the assemblies, it is unlikely that the flow reaches a fully turbulent regime. However, transitional flow may still occur. Since the expression for  $f_D$  in this transitional regime depends on  $Re_{b,T}$ , we have chosen to implement the complete model. Figure 2.8 depicts the expression of  $f_D$  in (2.35) as a function of  $Re_b$ .



**Figure 2.8:** Darcy friction factor for the axial flow inside a rod bundle as a function of the bundle Reynolds number.

With the Darcy friction factor now determined, we can obtain the  $K_{zz}$  component of the head loss tensor  $\mathbf{K}$  based on expressions (2.33) and (1.27):

$$K_{zz} = 0.5 \frac{f_D}{D_{h,b}} |V_b| \quad (2.37)$$

Note that in equation (2.37), the porosity is not accounted for because it is included in equation (1.27). However, when the components of the head loss tensor  $\mathbf{K}$  are defined via a user subroutine in Code\_Saturne, the porosity is not considered in (1.27) and must therefore be included in (2.37). Practically, this simply means that (2.37) must be multiplied by the porosity  $\phi$ .

Regarding the terms  $K_{xx}$  and  $K_{yy}$ , which represent the head losses in the transverse direction of the rods, limited experimental data is available, and no applicable correlation for our situation has been found. Therefore, a value 1000 times that of  $K_{zz}$  has been arbitrarily chosen:

$$K_{xx} = K_{yy} = 1000 K_{zz} \quad (2.38)$$

This assumption implies that no cross-flow will occur within the assemblies. While this is not entirely accurate, it remains a reasonable approximation because each assembly is physically separated by walls, meaning that, in practice, cross-flow is negligible. Note that this approximation has been used in [16]. We will also see later that this assumption allows for a simplification of the geometry used.

To conclude this section, we emphasize that, as currently defined, the porosity within the assemblies is set to  $\phi_a$ , while it instantaneously transitions to a porosity of one outside the assemblies. This abrupt change can lead to discontinuities or numerical instabilities in view of equations (1.26)-(1.28). To avoid such problems, a smooth porosity profile was implemented. In this approach, the porosity gradually decreases from unity at the assembly boundaries to the assigned value  $\phi_a$  within the assemblies over a transition distance denoted as  $\lambda_\phi$ . A hyperbolic tangent profile was chosen for the porosity distribution, defined as:

$$\phi(z) = \phi_a + \frac{1 - \phi_a}{2} \left[ 2 - \tanh\left(\frac{2\pi(z - z^- - \lambda_\phi/2)}{\lambda_\phi}\right) + \tanh\left(\frac{2\pi(z - z^+ + \lambda_\phi/2)}{\lambda_\phi}\right) \right] \quad (2.39)$$

where  $\lambda_\phi$  was arbitrarily taken at 3.5% of  $H_a$  and  $z^-$  and  $z^+$  denote the lower and upper boundaries of the assemblies, respectively. Furthermore, it is important to note that this porosity profile was established only along the  $z$ -direction, since flow at the boundaries of the assemblies is solely possible in this direction, as the assemblies are physically separated by walls.

## 2.6 Heat dissipation of an assembly

The objective of this section is to develop a model to quantify the heat dissipation of the various assemblies. To begin with, let us consider a single fuel rod. Since heat generation is directly related to the neutron flux within the fuel rod, the heat profile is expected to be non-uniform along the axial direction. Indeed, as the neutron flux follows a sinusoidal distribution, the heat flux can also be considered sinusoidal. In contrast, the radial dependency can be neglected [60]. However, as the fuel undergoes burnup, meaning that it gradually consumes fissile material while generating energy, the heat generation profile tends to flatten in the central region over time [61].

We will then assume that the heat dissipation of an entire assembly can be approximated by that of a single fuel rod. Consequently, the volumetric heat generation within the solid part of the assemblies (*i.e.* the fuel rods, as opposed to the fluid-filled region), denoted as  $q_s'''$ , will have the following form for a sinusoidal profile:

$$q_s'''(z) = A \sin\left(\frac{\pi z}{H_a}\right) \quad (2.40)$$

whereas for a flattened profile, due to burnup, a parabolic form will be used:

$$q_s'''(z) = B \left[ -\left(z - \frac{H_a}{2}\right)^n + \left(\frac{H_a}{2}\right)^n \right] \quad (2.41)$$

where the constants  $A$  and  $B$  have to be determined, as well as the exponent  $n$  in (2.41). To do so, we can, by definition of the volumetric heat generation, establish a relationship between  $A$  and  $B$  and the power  $P_a$  of an assembly. Indeed, we have:

$$P_a = \int_{V_s} q_s'''(z) dV = \int_{V_p} (1 - \phi_a) q_s'''(z) dV = (1 - \phi_a) W_a^2 \int_0^{H_a} q_s'''(z) dz \quad (2.42)$$

where it was assumed that the porosity  $\phi_a$  is constant in the assemblies for simplicity.

Consequently, by substituting expressions (2.40) and (2.41) into (2.42), one can easily solve the integral and explicitly determine the constants  $A$  and  $B$  as functions of  $P_a$ , which yields:

$$A = \frac{P_a}{(1 - \phi_a)W_a^2 H_a} \quad \text{and} \quad B = \frac{2^n(n+1)P_a}{n(1 - \phi_a)W_a^2 H_a^{n+1}} \quad (2.43)$$

We now need to determine a value for  $P_a$ . To do so, real data can be used, and we will rely on information from spent fuel pools of French nuclear power plants. Available to us are the maximum thermal power in the pools of various French PWR plants as well as the maximum number of assemblies that can be stored [6]. By averaging these data, the average maximum thermal power and the average maximum number of assemblies can be obtained:

$$P_{av,max} = 11.3 \text{ MW} \quad \text{and} \quad N_{av,max} = 480 \quad (2.44)$$

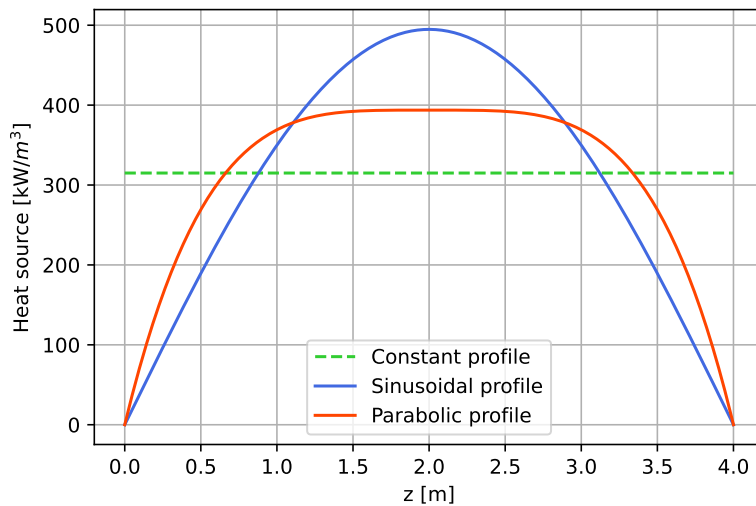
Since we know that there are 264 fuel rod in a single assembly and the geometry of a typical assembly is as described in Section 2.5, the total volume of fuel rods can be determined, which allows us to determine the average maximal volumetric heat generation inside the fuel rods:

$$q_{s,av,max}''' = \frac{P_{av,max}}{V_{tot,fuel rods}} \approx 315 \text{ kW/m}^3 \quad (2.45)$$

The power  $P_a$  in (2.42) and (2.43) of a single assembly can thus be estimated as follows:

$$P_a = (1 - \phi_a)W_a^2 H_a q_{s,av,max}''' \quad (2.46)$$

For a typical PWR assembly, this results in an estimated power of approximately 26 kW. Figure 2.9 then illustrates the sinusoidal heat profile, the parabolic profile with  $n = 4$ <sup>3</sup>, as well as a constant profile based on the value of  $q_{s,av,max}'''$ .



**Figure 2.9:** Comparison of different volumetric heat generation profiles.

<sup>3</sup>The value  $n = 4$  was chosen somewhat arbitrarily. It corresponds to the exponent that provides a profile similar to those observed in [61].

To conclude this section, it is important to emphasize that we have modelled here  $q_s'''$ , the heat generated within the fuel rods, and not the source term  $q_f'''$  in the fluid present in equation (1.28) solved in our study. Strictly speaking, an additional equation should be introduced to solve the temperature field within the fuel rods, and then determine a convective heat transfer coefficient  $h_h^*$ <sup>4</sup>, to quantify the heat transfer from the fuel rods to the fluid, as shown below:

$$\frac{\partial(\phi\rho c_p T)}{\partial t} + \nabla \cdot (\phi\rho c_p \mathbf{u}T) = \nabla \cdot (\phi\lambda\nabla T) + h_h^*(T_s - T) \quad (2.47)$$

$$\frac{\partial((1-\phi)\rho_s c_{ps} T_s)}{\partial t} = \nabla \cdot ((1-\phi)\lambda\nabla T) - h_h^*(T_s - T) + q_s''' \quad (2.48)$$

Note that in (2.47) and (2.48), the heat transfer coefficient  $h_h^*$  has to be expressed in W/m<sup>3</sup>K and its expression must be such that this coefficient is zero in the pure-fluid region, *i.e.* where  $\phi = 1$ .

In our case, for simplicity, we will not solve (2.48). Instead, we will directly use  $q_s'''$  expressed per unit volume of fluid to model  $q_f'''$ , which leads to:

$$q_f''' = \frac{1-\phi_a}{\phi_a} q_s''' \quad (2.49)$$

## 2.7 Geometries and case studies

So far, we have not provided the dimensions of the computational domain and have limited ourselves to presenting the dimensions of a typical fuel assembly used in a PWR. This was done to avoid confusing the reader in the previous developments. Indeed, as mentioned before, the present study will consider two distinctly different geometries: a two-dimensional (2D) geometry and a three-dimensional (3D) geometry.

The 2D geometry was originally developed by [18] and later reused by [19]. We will also reuse the mesh created for this geometry. However, this 2D geometry was not fixed according to the typical dimensions of PWR fuel assemblies as provided in Section 2.5. Consequently, some values, such as the porosity, will be adjusted accordingly. On the other hand, the 3D geometry has been developed based on the dimensions given in Section 2.5. The various dimensions for both geometries are provided below, along with the different case studies defined for each geometry.

### 2.7.1 Domain dimensions and meshes

#### Two-dimensional geometry

The 2D geometry is shown on the right side of Figure 2.1. The various dimensions are listed in Table 2.7.

| $H_{gap}$ | $H_a$ | $H_{bulk}$ | $H_{pool}$ | $W_{gap}$ | $W_a$ | $W_{pool}$ |
|-----------|-------|------------|------------|-----------|-------|------------|
| 0.16m     | 4m    | 8m         | 12.16m     | 0.16m     | 0.2m  | 1.992m     |

**Table 2.7:** *Dimensions of the two-dimensional geometry.*

---

<sup>4</sup>This coefficient is expressed by unit volume, not by unit surface.

It is important to mention that in the 2D geometry, each assembly is physically separated by a wall with a thickness  $W_{wall}$  of 8mm. This explains the total width of the pool, which is calculated as follows:

$$W_{pool} = 2 \times W_{gap} + 8 \times W_a + 9 \times W_{wall} = 1.992 \text{ m} \quad (2.50)$$

Regarding the mesh, the reader is invited to refer to [18] for detailed information regarding its development. Here, we will limit ourselves to a brief description, highlighting its main characteristics. The mesh consists of approximately 1.6 million cells, each with uniform dimensions of 4mm in the  $x$ -direction, 4mm in the  $z$ -direction, and 2mm in the  $y$ -direction<sup>5</sup>. Since this is a 2D geometry, there is only one cell along the  $y$ -direction.

Concerning the porosity within the assemblies, its value is slightly modified, resulting in a value of  $\phi_a = 0.5$ . For the power  $P_a$  of a single assembly, we obtain 25.2 kW. Finally, with respect to head losses, the differences in dimensions result in negligible variations in the model parameters. Therefore, the coefficients presented in Section 2.5 are retained.

### Three-dimensional geometry

The 3D geometry is shown on the left side of Figure 2.1. Given the actual dimensions of the geometry, creating a mesh with a sufficient level of refinement would result in an enormous number of cells, making the problem practically unsolvable due to the computational power at our disposal. Therefore, a choice was made to consider a 1/10 scale geometry, allowing us to maintain a cell count comparable to the 2D geometry, as will be explained below. The various dimensions are listed in Table 2.8.

| $H_{gap}$ | $H_a$ | $H_{bulk}$ | $H_{pool}$ | $W_{gap}$ | $W_a$   | $W_{pool}$ |
|-----------|-------|------------|------------|-----------|---------|------------|
| 0.02m     | 0.4m  | 0.8m       | 1.22m      | 0.02m     | 0.0214m | 0.2112m    |

**Table 2.8:** *Dimensions of the three-dimensional geometry.*

For the 3D geometry, we have, for simplicity, considered the entire 8x8 array of assemblies as a single volume at the bottom of the pool. In other words, the individual assemblies are not separated by walls. Additionally, note that the gap between the assemblies and the walls of the pool has increased from 16cm to 20cm. Hence, the total width of the pool is given by:

$$W_{pool} = 2 \times W_{gap} + 8 \times W_a = 0.2112 \text{ m} \quad (2.51)$$

Consequently, the condition imposed in (2.38), which states that head losses in the transverse direction are 1000 times higher than those in the axial direction, becomes relevant. This condition effectively prevents any transverse flow from developing within the assembly region.

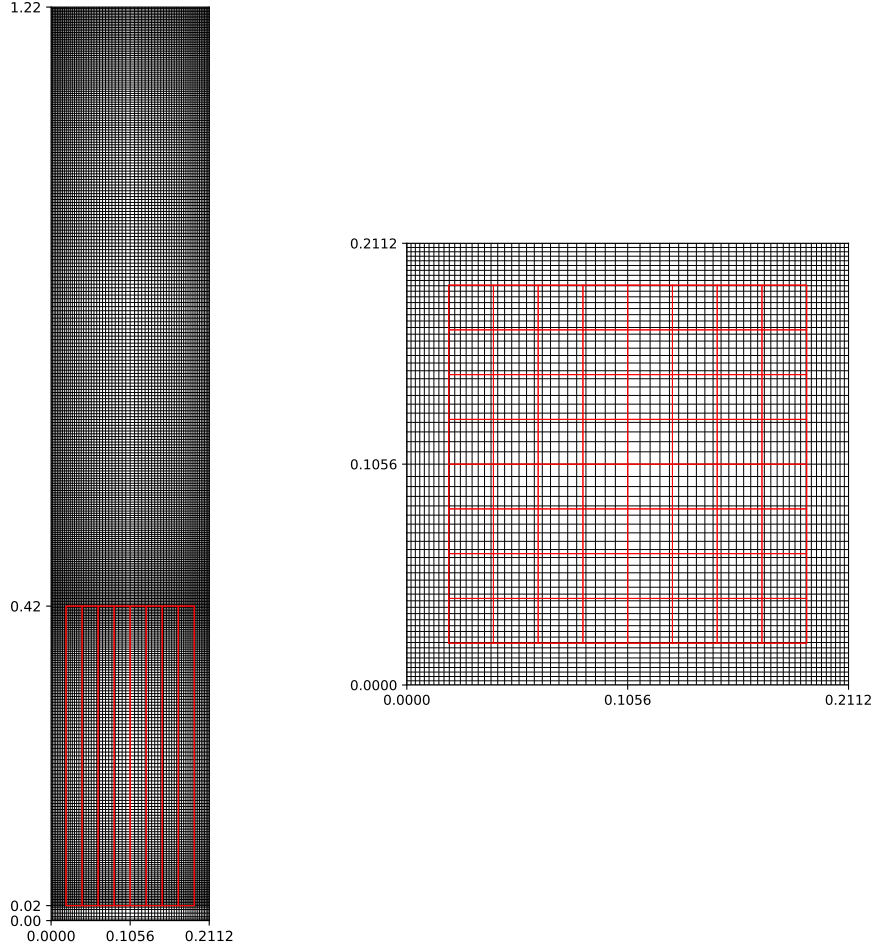
Regarding the mesh, a refinement has been applied along the pool walls, as well as above and below the assemblies. However, in RANS simulations, since all turbulence scales are modelled, there are no strict conditions for the maximum and minimum cell

---

<sup>5</sup>Although the geometry considered is two-dimensional, Code\_Saturne, like most CFD solvers, only handles three-dimensional geometries. A symmetry condition is then imposed to effectively simulate a two-dimensional configuration.



sizes. For the minimum size, the  $y^+$  coordinate of the cell closest to the wall can provide a criterion, which will be discussed later. For the maximum cell size, no specific criterion exists, but obviously, cells cannot be too large either. In our case, several meshes were tested, and the results were carefully examined. Whenever insufficient refinement was observed<sup>6</sup>, the maximum cell size was reduced accordingly. In our case, we also imposed the condition of maintaining the same total number of cells as in the 2D mesh to avoid excessive computational time. Taking all these considerations into account, we achieved a mesh with cell thicknesses of 2mm along the walls increasing to 5mm at the centre of the domain according to a geometric progression. The mesh was developed using Salome, a software co-developed by EDF. Figure 2.10 provides its  $x$ - $z$  and  $x$ - $y$  cross sections.



**Figure 2.10:** *Cross sections of the 3D mesh. Both figures are at scale. On the left, cross section in the  $x$ - $z$  plane, and on the right, cross-section in the  $x$ - $y$  plane. The separations between the different assemblies are shown in red for visualization purposes. Note that the assemblies are not physically separated in the mesh.*

Lastly, since all dimensions have been scaled down, it is worth mentioning that the porosity within the assemblies remains at  $\phi_a = 0.55$ , as stated in Section 2.5. Regarding the head loss model, it also remains unchanged because all dimensions have been scaled proportionally. However, to maintain the same volumetric heat production in the assemblies, the power of a single assembly  $P_a$  is now 26 W.

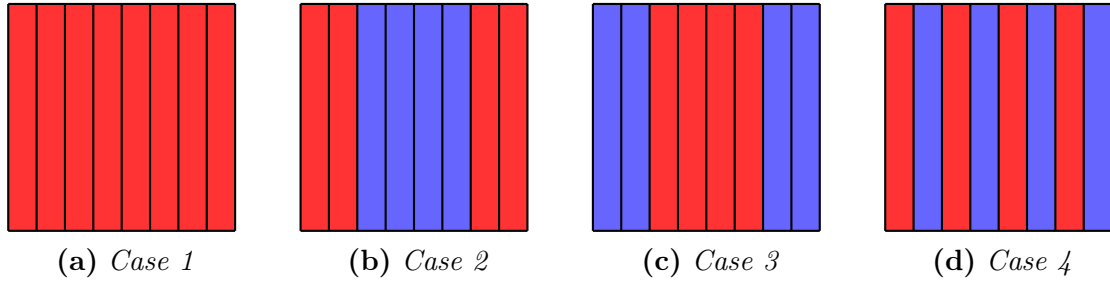
<sup>6</sup>When the cells were too large, the solution did not converge. For intermediate cell sizes, we noticed that some variables exhibited discontinuities or irregular shapes, likely due to numerical instabilities.

## 2.7.2 Definition of case studies

We will now define the different cases that will be simulated and analysed in the following Chapter. These cases differ by the arrangement of *hot* and *cold* assemblies at the bottom of the pool. By hot assembly, we mean an assembly with a higher thermal power than the cold assemblies. Note that in what follows the parabolic volumetric heat generation profile, as defined in Section 2.6, is used for every assembly in every case.

### Case studies for the two-dimensional geometry

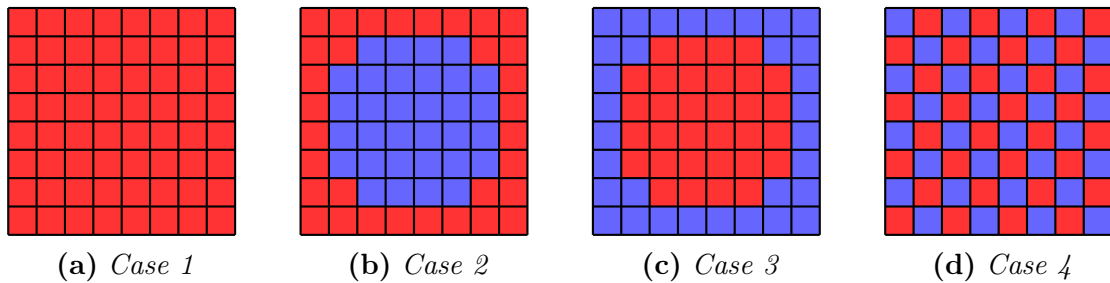
The different heat distributions considered for the two-dimensional geometry are shown in Figure 2.11. In Case 1, all assemblies are hot, with  $P_a = 25.2\text{ kW}$ , as mentioned in Section 2.6. In Cases 2 to 4, the hot assemblies have a power of  $P_{a,hot} = 33.6\text{ kW}$  while the cold assemblies have a power of  $P_{a,cold} = 16.8\text{ kW}$ , so as to preserve the same total power for all assemblies combined. Note that when the assemblies do not all have the same power, the hot assemblies have a power twice that of the cold ones.



**Figure 2.11:** Different case studies for the 2D geometry, shown in the  $x$ - $z$  plane. Hot assemblies are shown in red, while cold assemblies are shown in blue.

### Case studies for the three-dimensional geometry

The different heat distributions considered for the two-dimensional geometry are shown in Figure 2.12. Similarly to the two-dimensional geometry, Case 1 considers all assemblies as hot, with a power  $P_a = 25.2\text{ kW}$ , as mentioned in Section 2.6. For Cases 2 to 4, the hot assemblies have a power  $P_{a,hot} = 34.7\text{ W}$ , while the cold assemblies have a power  $P_{a,cold} = 17.3\text{ W}$ . This scenario also preserves the total power placed at the bottom of the pool.



**Figure 2.12:** Different case studies for the 3D geometry, shown in the  $x$ - $y$  plane. Hot assemblies are shown in red, while cold assemblies are shown in blue.

## 2.8 Solving the Navier-Stokes equations

### 2.8.1 Time discretisation of transport equations

At the beginning of a time step, Code\_Saturne computes the various physical properties of the flow, such as density, viscosity, specific heat capacity, etc. Indeed, these properties may depend on the flow variables, such as temperature, for example. Then, if necessary, the equations for the turbulent variables are solved. Next, the equations for the scalar variables are solved. Finally, all variables are updated and a new time step may begin.

To do so, time discretisation is performed using a  $\theta$ -scheme. One can distinguish an explicit first-order Euler scheme ( $\theta = 0$ ), an implicit first-order Euler scheme ( $\theta = 1$ ), and a second-order Crank-Nicolson scheme ( $\theta = 1/2$ ). For the second-order scheme, the time step  $\Delta t$  is considered constant. To fix ideas, let us consider the conservative form of the governing equations. In general, these can be written as:

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathbf{F}(U, \Phi, \varphi) = S(U, \Phi, \varphi) \quad (2.52)$$

with  $U$  representing the conserved variables (such as mass, momentum in each direction, or energy), expressed per unit volume, while  $\Phi$  denotes the physical properties appearing in the conservation equations of  $U$ , and  $\varphi$  represents any other flow variable involved. The total flux  $\mathbf{F}$  is composed of a convective flux and a diffusive flux. Finally,  $S$  represents a volumetric source term. The  $\theta$ -scheme then reads [20, 62]:

$$\frac{U^{n+1} - U^n}{\Delta t} + \nabla \cdot [\mathbf{F}(U^{n+\theta}, \Phi^{n+\theta_\Phi}, \varphi^{n+\theta}) + \mathbf{F}(U^n, \Phi^n, \varphi^n)] = [S(\Phi, \varphi)]^{n+\theta_S} \quad (2.53)$$

Equation (2.53) reveals that different  $\theta$ -schemes are applied within Code\_Saturne. Specifically, the code employs distinct schemes for different quantities:  $\theta$  for the flow variables,  $\theta_\Phi$  for the physical properties, and  $\theta_S$  for the source terms. For the physical properties, the scheme reads:

$$\Phi^{n+\theta_\Phi} = (1 + \theta_\Phi)\Phi^n - \theta_\Phi\Phi^{n-1} \quad (2.54)$$

While for the source terms, the time discretisation is expressed as:

$$[S(\Phi, \varphi)]^{n+\theta_S} = (1 + \theta_S)S(\Phi^n, \varphi^n) - \theta_S S(\Phi^{n-1}, \varphi^{n-1}) \quad (2.55)$$

Finally, the conserved variables and other flow quantities are discretised as follows:

$$U^{n+\theta} = \theta U^{n+1} + (1 - \theta)U^n \quad (2.56)$$

It should be noted that the description provided above is a simplified version of the time discretisation actually performed by Code\_Saturne. In practice, the code splits the source terms into implicit and explicit parts, which are treated differently. Here, only the explicit part has been considered. Moreover, as mentioned, the flux is further divided into convective and diffusive components. A separate scheme, denoted  $\theta_F$  is then used for the convective flux. We will not delve into these details here. The interested reader is referred to the Theoretical Guide [20].

## 2.8.2 Pressure-based velocity-pressure solver

In fluid dynamics, the Navier-Stokes equations establish a highly non-linear relationship between the pressure field  $p$  and the velocity field  $\mathbf{u}$ . For compressible flows, this relationship is managed using the continuity equation as the transport equation for density. In this case, pressure can be determined using an equation of state  $p = p(\rho, T)$ . However, for incompressible or weakly dilatable flows, where density is constant or slightly dependent on temperature, pressure and density are decoupled. This absence of a direct link introduces a constraint. The pressure field must be such that the resulting velocity field satisfies the continuity equation. Accurately maintaining this constraint is a key challenge for numerical solvers [63].

Pressure-based solvers, such as the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) algorithm, efficiently resolve this coupling using an iterative approach. These solvers are widely used in industrial applications due to their robustness, stability, and strong adherence to conservation laws. The SIMPLE algorithm has been further improved in a widely used variant called SIMPLER (Semi-Implicit Method for Pressure-Linked Equations Consistent). The "C" in SIMPLER, standing for "Consistent", indicates that this method ensures better accuracy by modifying the pressure correction equation to align with the discretisation of the momentum equations. This adjustment reduces the over-correction of velocity fields, improving convergence.

We will now describe the SIMPLER algorithm for ordered grids, which corresponds to the structure of our meshes. To keep the presentation concise, we will focus on the two-dimensional case, but the method can be easily extended to three dimensions. Our presentation closely follows the developments in [63]. In this algorithm, the pressure  $p$  and the velocity components  $u$  and  $v$  are divided into a guessed component, denoted with a superscript "\*", and a correction component, denoted with a hat symbol "^". Additionally, we denote by  $\phi$  the field quantity to be solved for. The different steps of the algorithm are then as follows:

1. Specify the boundary conditions and guess the initial values.
2. Determine the velocity and pressure gradients.
3. Calculate the pseudo velocities  $\hat{u}$  and  $\hat{v}$ :

$$a_{i,J}\hat{u}_{i,J} = \sum_{k \in \mathcal{N}} a_k u_k^* + b_{i,J} \quad (2.57)$$

$$a_{I,j}\hat{v}_{I,j} = \sum_{k \in \mathcal{N}} a_k v_k^* + b_{I,j} \quad (2.58)$$

with  $\mathcal{N}$  the neighbouring cells, the subscript " $i, J$ " denotes the left or right cell face of cell  $i$  while the subscript " $I, j$ " denotes the top and bottom cell face of cell  $j$ . The momentum source term is noted  $b$ . The coefficients  $a_k$  are coefficients computed during the pre-processing stage and each cell is assigned such a value.

4. Solve the following equation to get the pressure  $p$ :

$$a_{I,J}p_{I,J} = a_{I-1,J}p_{I-1,J} + a_{I+1,J}p_{I+1,J} + a_{I,J-1}p_{I,J-1} + a_{I,J+1}p_{I,J+1} + b_{I,J} \quad (2.59)$$

5. Set  $p^* = p$ .

6. Solve the discretised momentum equations to get  $\hat{u}$  and  $\hat{v}$  by using  $p^*$ :

$$a_{i,J}u_{i,J}^* = \sum_{k \in \mathcal{N}} a_k u_k^* + (p_{I-1,J}^* - p_{I,J}^*)A_{i,J} + b_{i,J} \quad (2.60)$$

$$a_{I,j}v_{I,j}^* = \sum_{k \in \mathcal{N}} a_k v_k^* + (p_{I,J-1}^* - p_{I,J}^*)A_{I,j} + b_{I,j} \quad (2.61)$$

where  $A_{i,J}$  and  $A_{I,j}$  denote the left or right and top or bottom cell face area of the corresponding cell, respectively.

7. Solve the pressure correction equation:

$$a_{I,J}p'_{I,J} = a_{I-1,J}p'_{I-1,J} + a_{I+1,J}p'_{I+1,J} + a_{I,J-1}p'_{I,J-1} + a_{I,J+1}p'_{I,J+1} + b_{I,J} \quad (2.62)$$

8. Evaluate the corrected velocities and pressure:

$$u_{i,J} = u_{i,J}^* + d_{i,J}(p'_{I-1,J} - p'_{I,J}) \quad (2.63)$$

$$v_{I,j} = v_{I,j}^* + d_{I,j}(p'_{I,J-1} - p'_{I,J}) \quad (2.64)$$

$$p = p^* + p' \quad (2.65)$$

where the quantities  $d$  are defined as follows:

$$d_{i,J} = \frac{A_{i,J}}{a_{i,J}} \quad \text{and} \quad d_{I,j} = \frac{A_{I,j}}{a_{I,j}} \quad (2.66)$$

9. Solve all other discretised transport equations for the quantities  $\phi$ :

$$a_{I,J}\phi_{I,J} = a_{I-1,J}\phi_{I-1,J} + a_{I+1,J}\phi_{I+1,J} + a_{I,J-1}\phi_{I,J-1} + a_{I,J+1}\phi_{I,J+1} + b_{I,J}^\phi \quad (2.67)$$

10. If  $\phi$  shows convergence, the process is stopped. Otherwise, we set  $p^* = p$ ,  $u^* = u$ ,  $v^* = v$  and  $\phi^* = \phi$  and the whole process is repeated.

### 2.8.3 Choice of the turbulence model

Choosing an appropriate turbulence model is crucial, as a poor choice can lead to entirely erroneous results. This selection must be guided by the expected flow characteristics and by the geometry and mesh used. In our case, the flow under consideration is a buoyancy-driven flow due to internal heating. Therefore, the formation of a natural flow with convective cells is expected. Hence, one must take into account that convective cells imply regions with adverse pressure gradients.

As discussed in Sections 1.4.3 and 1.4.4, different turbulence models have inherent limitations. Specifically, the  $k - \epsilon$  model is unsuitable for natural boundary conditions and performs poorly near walls. It requires the first cell<sup>7</sup> to be located in the fully turbulent region, *i.e.*  $y^+ > 30$ . However, in our case, it was observed that the  $y^+$  value of the first cell of the mesh averaged, over time<sup>8</sup>, around  $y^+ = 15$  for two-dimensional geometry and around  $y^+ = 5$  for the three-dimensional geometry. Consequently, this makes the  $k - \epsilon$

<sup>7</sup>Here, by the first cell, we mean the computation point in the first cell.

<sup>8</sup>Considering an unsteady simulation,  $y^+$  naturally varies with time.

model clearly inappropriate for both meshes. Furthermore, according to Wilcox [28], this model is also known to underperform in regions with adverse pressure gradients. On the other hand, the  $k - \omega$  model, while more accurate in the near-wall region, is also known to struggle in the presence of adverse pressure gradients. In fact, this is one of the reasons that led to the development of the  $k - \omega$  SST model [36]. The  $k - \omega$  SST model combines the strengths of the  $k - \epsilon$  and the  $k - \omega$  models while mitigating their weaknesses, making it a highly versatile choice. Given the nature of our problem, we therefore have opted for the  $k - \omega$  SST model, which is well-suited for flows with adverse pressure gradients.

It is worth recalling that the  $k - \omega$  SST model is inherently a low-Reynolds number model. As previously mentioned, the value of  $y^+$  for the first mesh cell adjacent to the wall is approximately 15 for the 2D geometry and around 5 for the 3D geometry. In the context of low-Reynolds models, accurately resolving the near-wall region requires the first cell to lie within the viscous sublayer, ideally around  $y^+ = 1$  [30]. However, this approach is computationally expensive as it demands a highly refined mesh near the walls. An alternative strategy consists in placing the first cell at  $y^+ \approx 30$  and applying an appropriate wall law. In any case, the buffer layer (defined by  $5 < y^+ < 30$ ) should ideally be avoided, as it lies in between the linear and logarithmic regions and can lead to inaccurate turbulence modelling. For this reason, the 3D mesh was specifically designed so that the first cell lies outside the buffer zone. The 2D mesh, however, does not satisfy this criterion, as the first cell lies within the buffer region. Fortunately, some wall laws, such as the one described in (1.63) and illustrated in Figure 1.3, are formulated to remain valid in the buffer region. These so-called "All  $y^+$ " wall laws remain valid across the entire near-wall region. This is the law that we adopted for both geometries<sup>9</sup>.

## 2.8.4 Stability conditions

In computational fluid dynamics, for unsteady flows, two main stability conditions must be satisfied to ensure that the numerical solution remains stable and that convergence is achieved. These are the Courant–Friedrichs–Lewy (CFL) condition, which applies to the convective terms, and the Fourier condition (also known as the von Neumann condition), which applies to the diffusive terms. Both conditions impose a constraint on the time step  $\Delta t$  used during the simulation [64]. They are expressed as:

$$\text{Co} = \max \left( \frac{u\Delta t}{\Delta x}, \frac{v\Delta t}{\Delta x}, \frac{w\Delta t}{\Delta x} \right) \leq \text{Co}_{\max} \quad (2.68)$$

$$\text{Fo} = \max \left( \frac{\nu\Delta t}{\min(\Delta x, \Delta y, \Delta z)^2}, \frac{\kappa\Delta t}{\min(\Delta x, \Delta y, \Delta z)^2} \right) \leq \text{Fo}_{\max} \quad (2.69)$$

For explicit solvers,  $\text{Co}_{\max} = 1$  and  $\text{Fo}_{\max} = 0.5$ , though much smaller values are used in practice to assure stability (*i.e.* around 0.3). Implicit solvers, on the other hand, are generally less sensitive to instabilities and can tolerate larger values. Both explicit and implicit time integration schemes are supported by Code\_Saturne, which allows for more relaxed stability constraints. According to the Code\_Saturne Best Practice Guide [65], values of  $\text{Co}_{\max} = 1$  and  $\text{Fo}_{\max} = 10$  are recommended by default. In our simulations, we used  $\text{Co}_{\max} = 0.6$  and  $\text{Fo}_{\max} = 10$ . These values were selected after testing several options, in order to minimize the total simulation time while maintaining numerical stability.

---

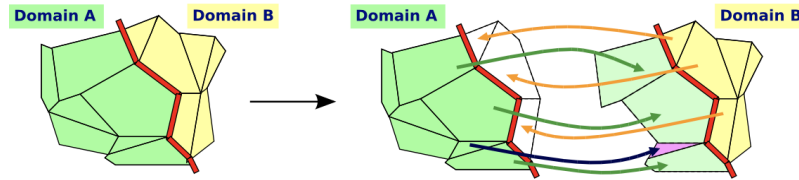
<sup>9</sup>Strictly speaking, the 3D geometry would not require such an adaptive law. However, we chose to apply it nonetheless for the sake of consistency.



## 2.9 User subroutines

Before detailing the implementation of each user-defined function, it is useful to recall some key aspects of how simulations are parallelized across multiple processors. Code\_Saturne uses the Message Passing Interface (MPI) for this purpose. The computational domain is divided into several subdomains, each assigned to a separate processor. The goal is to distribute the workload evenly by ensuring that each processor handles a roughly equal number of cells.

When two subdomains are adjacent, data from cells located at the boundary of one subdomain but belonging to the neighbouring one are duplicated. These duplicated cells are referred to as *ghost cells* and form what is called the *halo* of the subdomain. A specific feature of Code\_Saturne is that the halo is split into two distinct parts: the standard halo, which includes cells that share a face with the local subdomain and are involved in parallel boundary exchanges, and the extended halo, which contains cells connected through vertices. The extended halo is mainly used for least squares gradient reconstruction using an extended neighbourhood [66]. A schematic illustration of the domain decomposition into subdomains is provided in Figure 2.13.



**Figure 2.13:** *Decomposition of the computational domain into subdomains. Arrows represent the duplication of data into the halo of each subdomain. This figure is taken from [66].*

We now proceed to describe the various user subroutines, listed in alphabetical order.

### `cs_user_boundary_condition.c`

This subroutine computes the evaporation-related quantities at each time step, such as the evaporative mass flux, the convective and evaporative heat fluxes, the evaporated mass and volume, and the evaporated volume since the previous mesh deformation. It also sets the boundary condition at the free surface, as defined in (2.1). We can now clarify why the thermal conductivity was retained on the left-hand side of the energy balance: the boundary condition for temperature must be expressed in units of  $\text{W}/\text{m}^2$ . Moreover, the associated heat flux is considered negative when it represents a source term for the fluid [21].

### `cs_user_extra_operations.c`

This subroutine creates a file and writes the evaporation related quantities in it for post-processing purposes. It also computes area-averaged values of these quantities. The mesh deformation is also handled by this subroutine. Moreover, at the end of a simulation, the coordinates of the mesh vertices are written in the checkpoint folder<sup>10</sup> along with the evaporation related quantities.

---

<sup>10</sup>In Code\_Saturne, at the end of each simulation, every quantity, physical or not, is written in the checkpoint folder, allowing a new simulation to be restarted from the previous results.

### **cs\_user\_head\_losses.c**

This subroutine computes the components of the velocity head loss tensor inside the assemblies as defined in (2.37) and (2.38).

### **cs\_user\_initialization.c**

This subroutine allocates memory for the evaporation related quantities at the start of a simulation so that the data can be shared through the different subroutines and the different processors. In case of a restart, the checkpoint folder is read and the evaporation quantities are restored.

### **cs\_user\_mesh.c**

In case of a restart, this subroutine reads the checkpoint folder and the vertices of the mesh are restored. Then, the mesh is rebuilt from these vertices.

### **cs\_user\_parameters.c**

This subroutine sets the system of equations to be solved. In our case, it corresponds either to the Oberbeck–Boussinesq approximation or to the dilatable flow algorithm.

### **cs\_user\_physical\_properties.c**

If the system of equations to be solved is based on the Oberbeck–Boussinesq approximation, this subroutine defines the thermal expansion coefficient using a fourth-degree polynomial. The coefficients of this polynomial are provided in Table 2.2. This function also updates the subdomain’s halo to prevent any errors in the computation of the various fluxes at the subdomain boundaries.

### **cs\_user\_porosity.c**

This subroutine sets the prescribed porosity value inside the assemblies with a smooth transition along the  $z$ -direction according to the hyperbolic tangent profile defined in (2.39). This function also updates the subdomain’s halo to prevent any errors in the computation of the various fluxes at the subdomain boundaries.

### **cs\_user\_source\_term.c**

This subroutine imposes the volumetric heat source profile within the assemblies. The profile corresponds to the flattened parabolic distribution defined in (2.41). In addition, this function allows for the implementation of the different power distributions described in Section 2.7.2. This function also updates the subdomain’s halo to prevent any errors in the computation of the various fluxes at the subdomain boundaries.

### **cs\_user\_zones.c**

This subroutine defines volume zones corresponding to the fuel assemblies. It serves a purely practical and numerical purpose, making it easier to manage the assemblies in the other subroutines.



# Chapter 3

## Numerical results and discussion

In this chapter, we present and analyse the results obtained from our numerical simulations. The structure of the analysis will be the same for both the two-dimensional and three-dimensional geometries. First, we provide a detailed presentation of the results for Case 1. Then, we compare these results with those of Cases 2, 3, and 4, with a particular focus on evaporation. Finally, the results of Case 4 are compared to those obtained for the same configuration without invoking the Oberbeck–Boussinesq approximation.

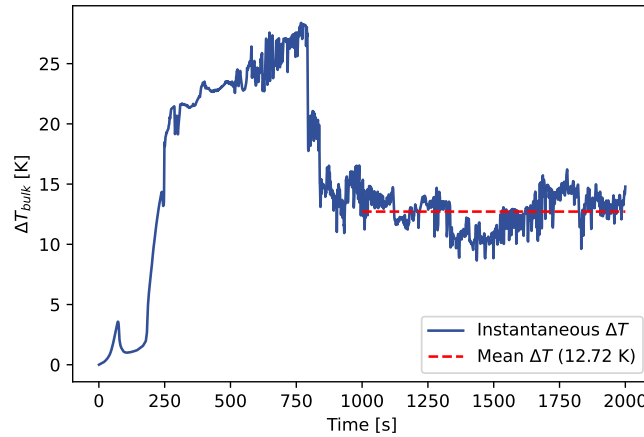
### 3.1 Two-dimensional geometry

#### 3.1.1 Preliminary considerations

Before presenting the numerical results, it is essential to characterize the flow regime by evaluating the Rayleigh number (1.85). Here, since the flow of interest is located within the pure fluid region, we compute the bulk Rayleigh number:

$$\text{Ra}_{bulk} = \frac{g\beta_0\Delta T_{bulk}H_{bulk}^3}{\nu_0\kappa_0} \quad (3.1)$$

However, this requires evaluating the temperature difference across the bulk, which is not directly known since we impose a volumetric heat source at the bottom of the pool. To estimate it, a coarser simulation was performed. The result is shown in Figure 3.1.



**Figure 3.1:** *Instantaneous and time-averaged temperature difference across the bulk.*

We must clarify here that by *coarser simulation*, we refer to a simulation using a larger reference time step. Moreover, to further reduce the computational cost, the measured quantities are not area-averaged. Instead, two measurement points were used: one located just above the assemblies and the other just below the free surface. It should also be noted that the configuration of Case 1 was used for this simulation. Nevertheless, the results shown in Figure 3.1 will be considered representative of all cases.

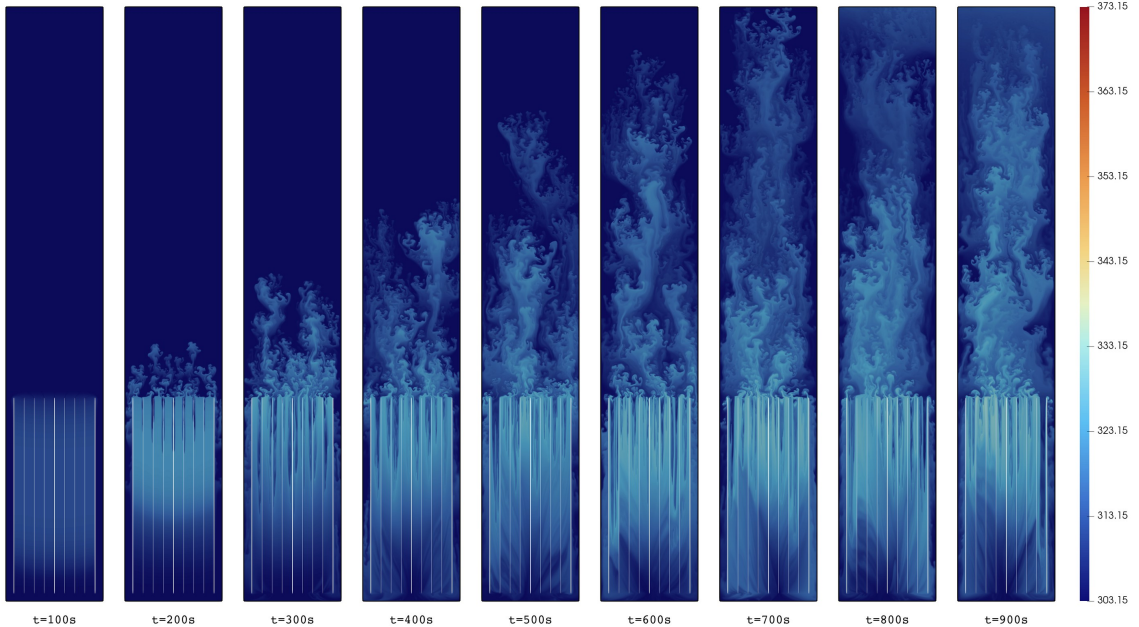
As will be shown later, a transitional period is required before the heated fluid plumes reach the free surface, which lasts approximately 800 seconds. The temperature difference was thus time-averaged over the interval ranging from 1000 s to 2000 s, yielding a mean value of 12.72 K. This average is represented by the red dashed line in Figure 3.1. Using this value, we obtain  $Ra_{bulk} \approx 1.6 \times 10^{14}$ . In view of this value, and based on the discussion in Section 1.6.1, we conclude that the flow lies at the onset of the ultimate regime, and we therefore expect a highly turbulent flow to develop in the bulk region.

Finally, we can also estimate the reference velocity of the flow, as developed in (1.80):

$$u_{0_N} = \sqrt{g\beta_0\Delta T_{bulk}H_{bulk}} \approx 0.55 \text{ m/s} \quad (3.2)$$

### 3.1.2 Numerical results - Case 1

We begin our analysis by presenting the temperature field. It is worth noting that for all results obtained with the two-dimensional geometry, regardless the case, the same temperature scale is used across all figures. This choice was made to avoid misleading the reader and to highlight the global heating of the water that will be observed later. Figure 3.2 shows the temperature evolution during the first 900 seconds.

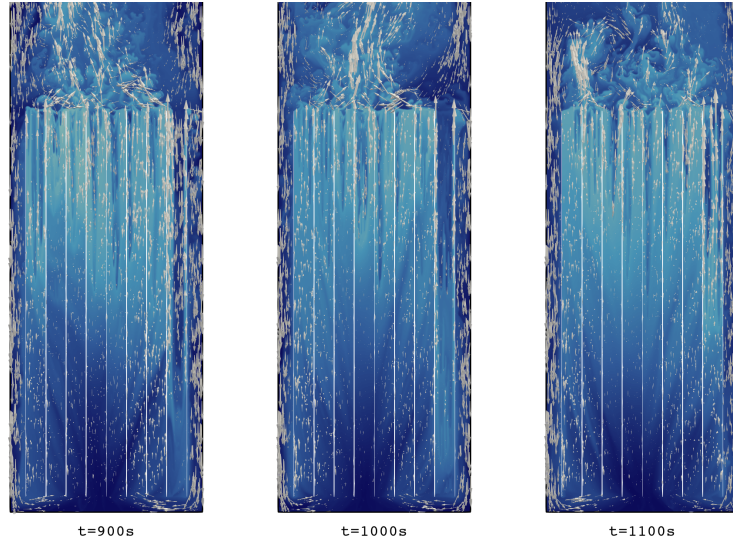


**Figure 3.2:** *Temperature field, in Kelvin, for  $t \in [100 \text{ s}; 900 \text{ s}]$ .*

We directly observe thermal plumes rising from the top of the assemblies towards the free surface. Indeed, since the density of the heated fluid decreases, it tends to rise due to

buoyancy. These plumes are characterized by intense turbulence, with numerous eddies forming along them that break them up and give them an irregular, fragmented shape. This behaviour is consistent with the flow being in the ultimate regime, as established by the Rayleigh number estimation.

Another important observation is that, in general, heat tends to accumulate at the top of the assemblies. However, some particular cases can be identified. For instance, after between 500 and 600 seconds, heat appears to be almost entirely evacuated from the leftmost assembly (Figure 3.2). At around 900 seconds, it is the rightmost assembly that appears to have released its heat. To better analyse this phenomenon, Figure 3.3 shows a zoomed-in view of the assemblies at around 900 seconds, with local velocity vectors superimposed in white.

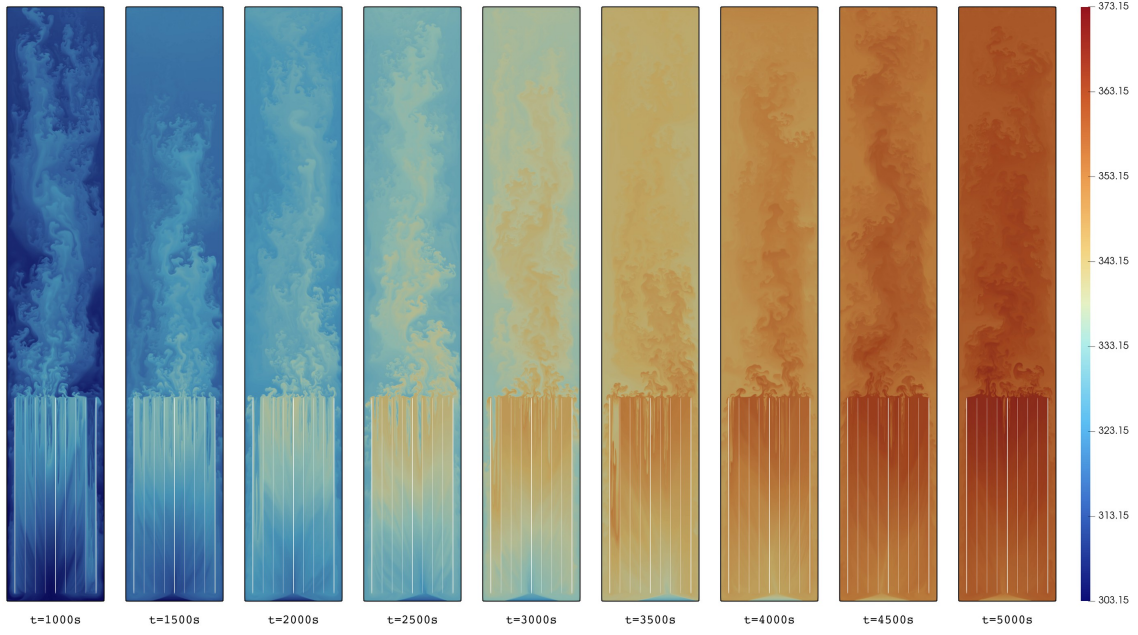


**Figure 3.3:** Zoom on the assemblies. Velocity vectors are shown in white. The colour map used here is the same as the one in Figure 3.2.

Even though convection cells do not form coherent structures and are not directly distinguishable in Figure 3.2, they do exist. Indeed, since heat is removed through the free surface due to evaporation, the hot fluid that has risen tends to cool down near the top and subsequently sinks. What we see in Figure 3.3 is that this downwards flow of cooled fluid goes, on the one hand, between the assemblies and the pool walls, and on the other hand, directly through the rightmost assembly, thereby removing heat from it. This reverse flow within the assembly is, of course, not sustained indefinitely. As we will see later, the convection cells continuously reorganize, and thus may no longer direct the flow into that particular assembly. Moreover, we see that the reverse flow weakens as it moves downwards through the assembly due to head losses. Finally, it is worth noting that these reverse flows tend to occur in a completely random manner.

At this point, the reader may wonder why we only present instantaneous results. First, we wanted to highlight the transient period before the thermal plumes reach the free surface. Secondly, the flow never reaches a steady state. As a result, presenting time-averaged results would be both meaningless and irrelevant. Indeed, since the heat supplied by the assemblies exceeds that removed by evaporation, heat accumulates in the pool. This heat accumulation will be discussed in detail and clearly demonstrated with

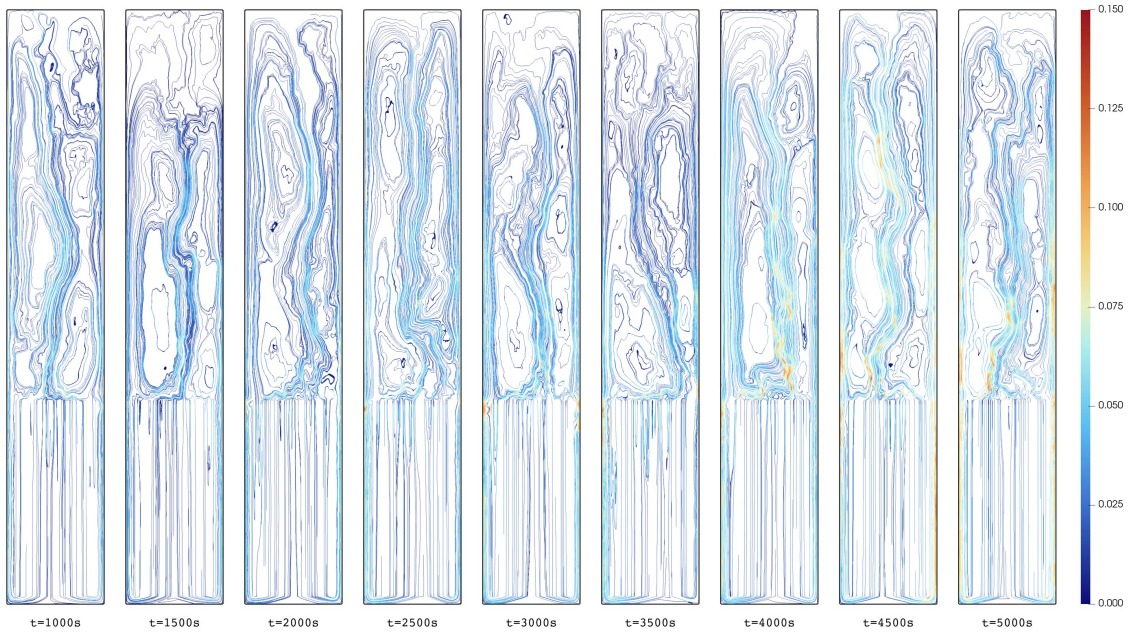
quantitative results later, when focusing on the evaporation analysis. As a result, the water temperature continuously increases, as shown in Figure 3.4.



**Figure 3.4:** *Temperature field, in Kelvin, for  $t \in [1000 \text{ s}; 5000 \text{ s}]$ .*

In practice, water temperature cannot increase indefinitely. For instance, at atmospheric pressure, water boils at approximately  $100^\circ\text{C}$ . Regarding our simulation, as soon as any point in the pool reaches  $100^\circ\text{C}$ , Code\_Saturne automatically stops the calculation. In the present case, this occurs after 5299 seconds.

To finalize the flow behaviour analysis, streamlines are presented in Figure 3.5.

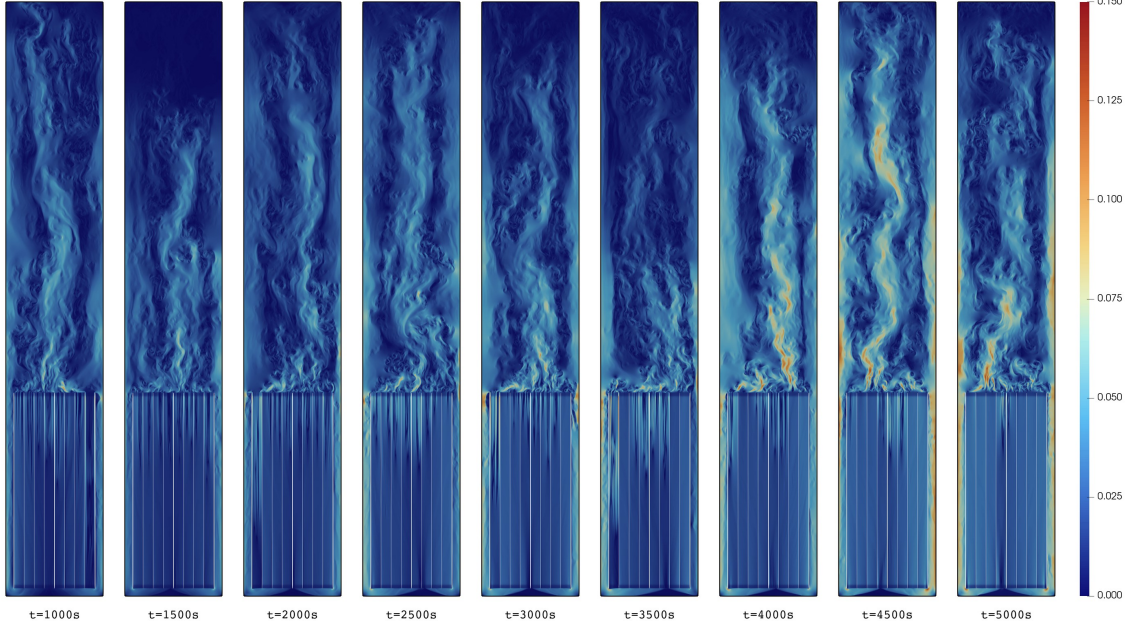


**Figure 3.5:** *Streamlines, coloured by velocity magnitude, in  $\text{m/s}$ , for  $t \in [1000 \text{ s}; 5000 \text{ s}]$ .*

As expected, we observe very chaotic convection cells that constantly rearrange. However,



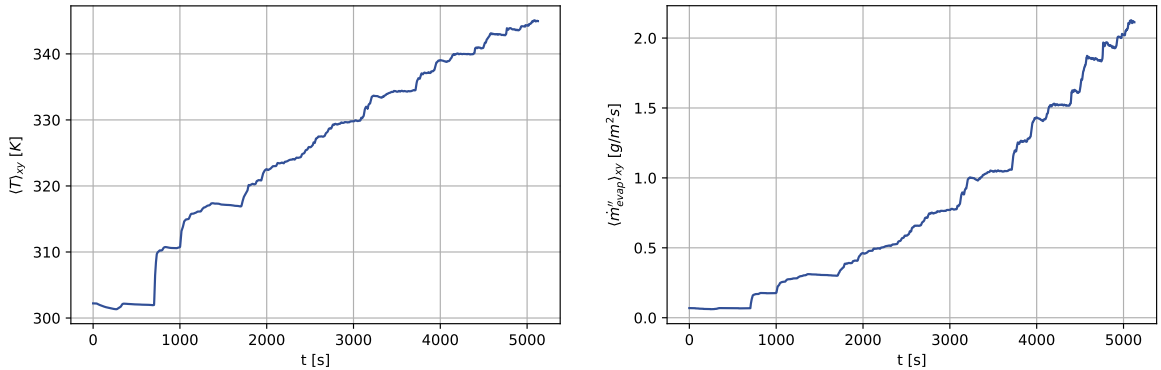
a general pattern can still be identified. Two adjacent cells extend throughout the entire height of the bulk. The cell on the left exhibits a counter-clockwise flow, while the cell on the right exhibits a clockwise flow. Moreover, the maximum velocity magnitude observed is around 0.10 m/s, which remains significantly lower than the prediction obtained from (3.2). That being said, only the values along the streamlines are shown here. Figure 3.6 therefore provides the complete velocity magnitude field.



**Figure 3.6:** *Velocity magnitude field, in m/s, for  $t \in [1000\text{ s}; 5000\text{ s}]$ .*

We observe that the maximum velocity magnitude reaches 0.15 m/s. We can therefore confidently state that (3.2) overestimates the characteristic flow velocity.

To initiate the analysis of the evaporation process, we first present the time evolution of the area-averaged free-surface temperature and area-averaged evaporative mass flux. These quantities are shown in Figure 3.7.

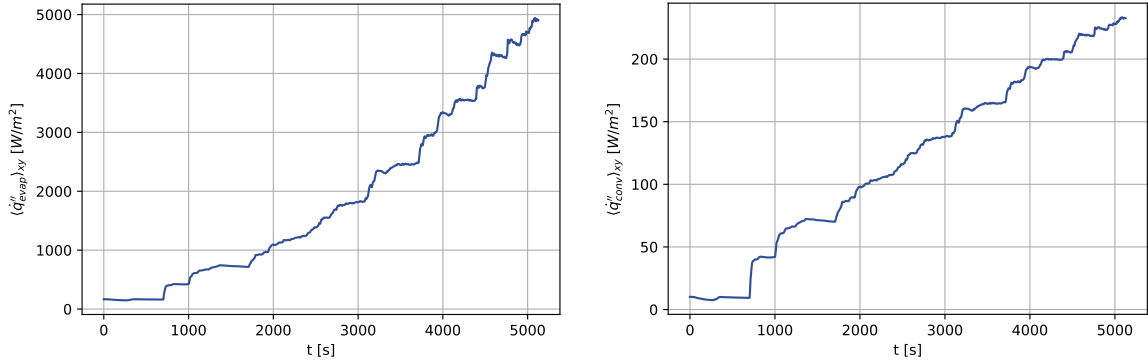


**Figure 3.7:** *Area-averaged free-surface temperature and area-averaged evaporative mass flux at the free surface as a function of time.*

In the first place, a steady increase in the area-averaged free-surface temperature is observed, reflecting the overall warming of the water over time. Moreover, during the first

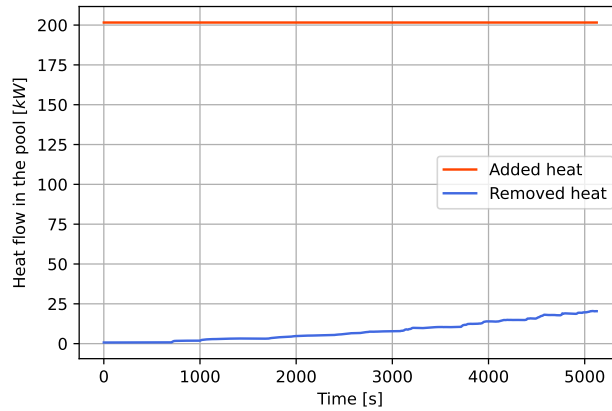
800 seconds, the temperature remains relatively stable before suddenly rising, which is consistent with the transient delay required for the thermal plumes to reach the free surface. The area-averaged evaporative mass flux follows a similar trend, as evaporation only depends on the free-surface temperature under given ambient conditions. The values obtained for the mass flux, ranging from 0.1 to 2 g/m<sup>2</sup>s, for free-surface temperatures between 303 K and 350 K, are consistent with those reported by Hay [10], confirming that the model has been correctly implemented.

We can also examine the evolution of the area-averaged evaporative and convective heat fluxes at the free surface, which together define the boundary condition imposed at this interface. These are presented in Figure 3.8.



**Figure 3.8:** Area-averaged evaporative and convective heat fluxes at the free-surface as a function of time.

The observations are similar to those made for the evaporative mass flux, as the heat fluxes also depend solely on the free-surface temperature under given ambient conditions. The values obtained are once again consistent with those reported by Hay [10]. However, examining these two fluxes separately provides limited insight. What truly matters is quantifying the total heat removed through the free surface in order to compare it with the heat produced by the assemblies. To do so, the two fluxes are simply added and multiplied by the free-surface area. The resulting quantity is shown in Figure 3.9.

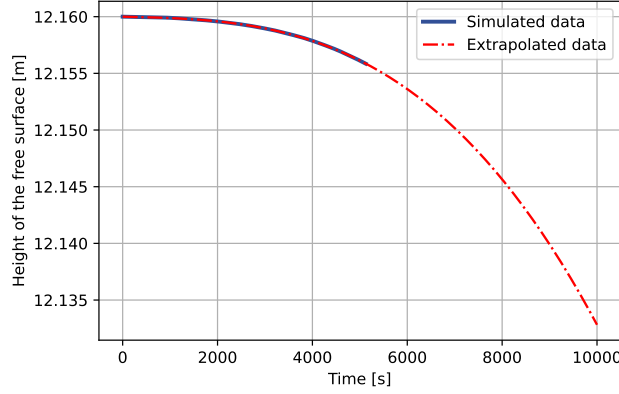


**Figure 3.9:** Heat loss through the free surface and heat generated by the assemblies.

The accumulation of heat in the pool, and consequently the rise in the bulk water temperature, is inevitable given that the heat removed through evaporation is, on average

over the duration of the simulation, ten times lower than the heat input. It is worth noting that, in this context, the heat generated by the assemblies does not simply refer to the thermal power  $P_a$  multiplied by the number of assemblies, but rather to the thermal power actually delivered to the fluid, which can be obtained by integrating  $q_f'''$  over the height of an assembly, as defined in equation (2.49). However, since  $\phi_a = 0.5$  in the two-dimensional geometry, these two quantities are equivalent.

To conclude the analysis of evaporation, we now focus on the descent of the free surface as water evaporates. As detailed in Section 2.3.4, at any given time, we distinguish between the height of the computational domain  $H_C$  and the height of the free surface  $H_S$ . However, in the present case, the total evaporated volume remains below 0.5% of the initial volume throughout the entire simulation. As a result, no remeshing has been performed. Consequently, presenting the evolution of  $H_C$  is of no interest. Instead, we extrapolate the expected evolution of the free-surface height  $H_S$ , as shown in Figure 3.10.



**Figure 3.10:** *Extrapolation of the height of the free surface from simulated data.*

The extrapolation was carried out by assuming a function of the form:

$$H_S(t) = -a^{bt} + ct + H_0 \quad (3.3)$$

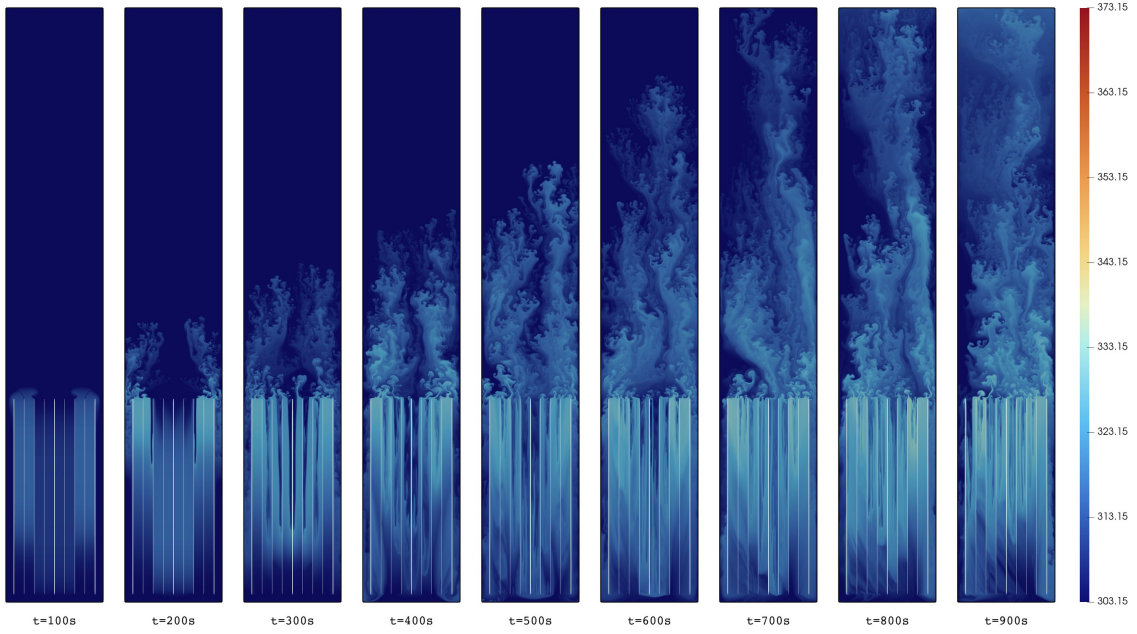
where  $H_0$  is the initial height of the free surface, denoted as  $H_{pool}$  throughout this document, and  $a$ ,  $b$  and  $c$  are the coefficients determined for the purpose of the extrapolation. Once the function (3.3) has been evaluated, it can be used to estimate the time at which the top of the assemblies becomes exposed to the ambient air. In the present case, this occurs after 19h, 22min and 36s. This value should however be taken with caution as it results from extrapolating the evaporation rate based on simulated data, that is, under the assumption that water remains entirely in the liquid phase. Consequently, it does not account for any boiling phenomena that may occur.

### 3.1.3 Comparison with the other cases

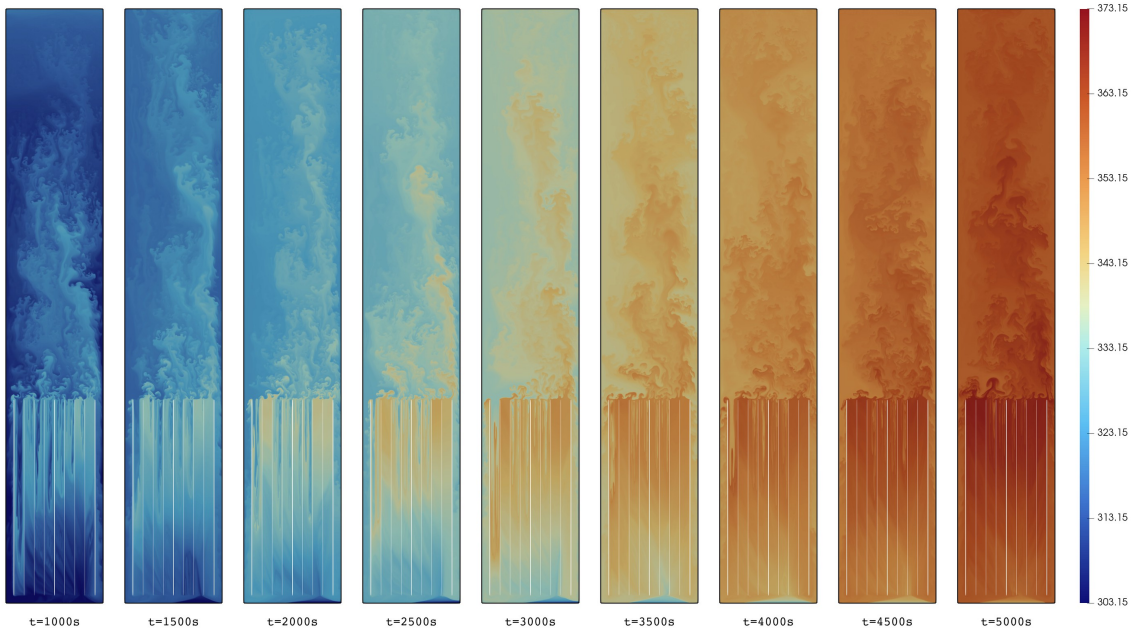
In what follows, we first highlight the main flow differences between cases, then compare the times for any point in the pool to reach 100°C and for the assemblies to be exposed to ambient air. Note that we will not present in details Cases 2 to 4 as we did for Case 1 in Section 3.1.2, as doing so would be very repetitive and add little value.

## Flow behaviour

Case 1, which represents the most basic heat distribution at the bottom of the pool, will be taken as the reference case in what follows. A first notable difference then appears in Case 2. In this configuration, the hot assemblies are placed at the edges of the rack, while the cold ones are at its centre. As we shall see, this distribution induces a particular flow behaviour within the pool. To observe this, we first examine the temperature field, as illustrated in Figures 3.11 and 3.12.



**Figure 3.11:** *Temperature field, in Kelvin, for  $t \in [100\text{ s}; 900\text{ s}]$  (Case 2).*

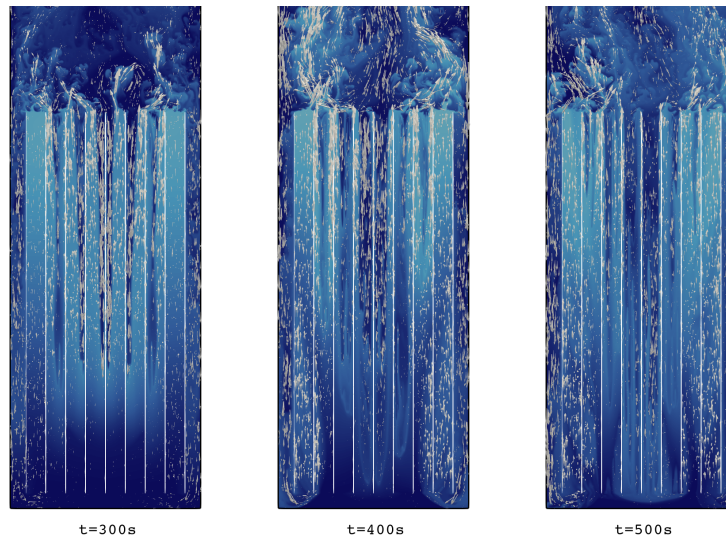


**Figure 3.12:** *Temperature field, in Kelvin, for  $t \in [1000\text{ s}; 5000\text{ s}]$  (Case 2).*

We observe that, in the early stages, thermal plumes rise from the hot assemblies. At this point, the formation of convection cells just above these hot assemblies induces a signif-

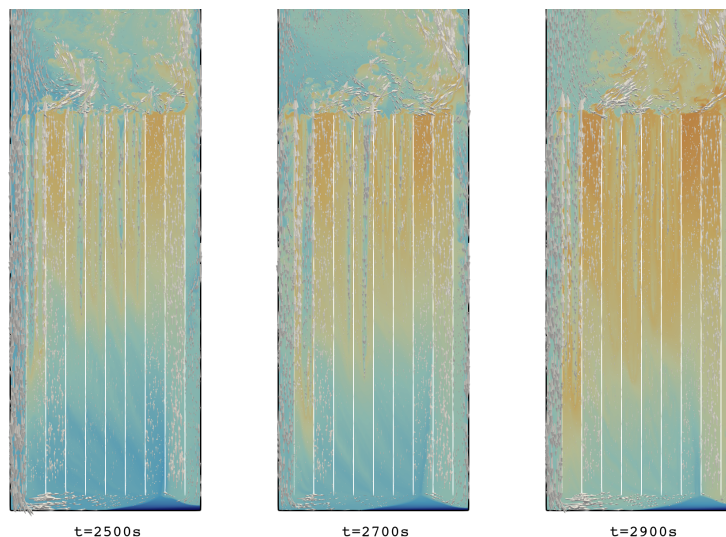


icant backflow within the central assemblies, effectively removing heat from them. This behaviour can be highlighted by superimposing the velocity vectors on the temperature field. This is illustrated in Figure 3.13.



**Figure 3.13:** *Zoom on the assemblies (Case 2). Velocity vectors are shown in white. The colour map used here is the same as the one in Figures 3.11 and 3.12.*

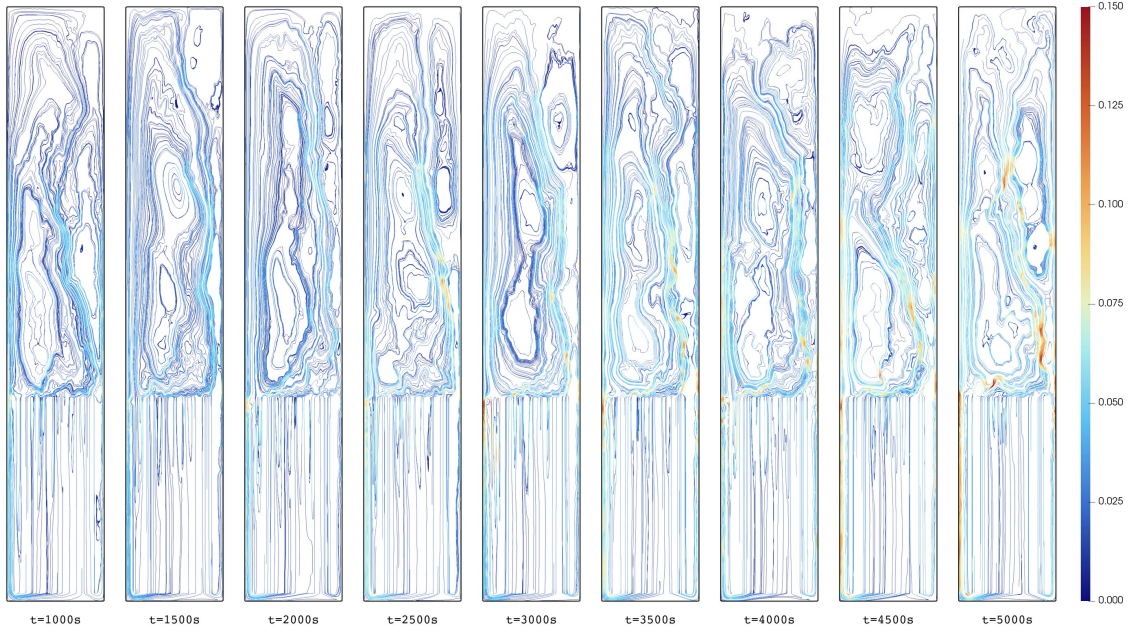
However, this effect tends to weaken rather quickly, though it does not vanish entirely. Indeed, Figure 3.12 shows that a backflow seems to persist throughout the entire simulated duration, although much less pronounced than in the initial moments. A similar observation can be made for the leftmost assembly. In fact, heat seems to be efficiently removed from this hot assembly. This can be explained by the development of a strong downward flow along the left side of the pool, as illustrated in Figure 3.14.



**Figure 3.14:** *Zoom on the assemblies (Case 2). Velocity vectors are shown in white. The colour map used here is the same as the one in Figures 3.11 and 3.12.*

This behaviour likely results from the fact that the thermal plumes tend to rise predominantly from the assemblies on the right side, rather than from both ends as initially observed. As the temperature rises, however, plumes start to reappear on the left side

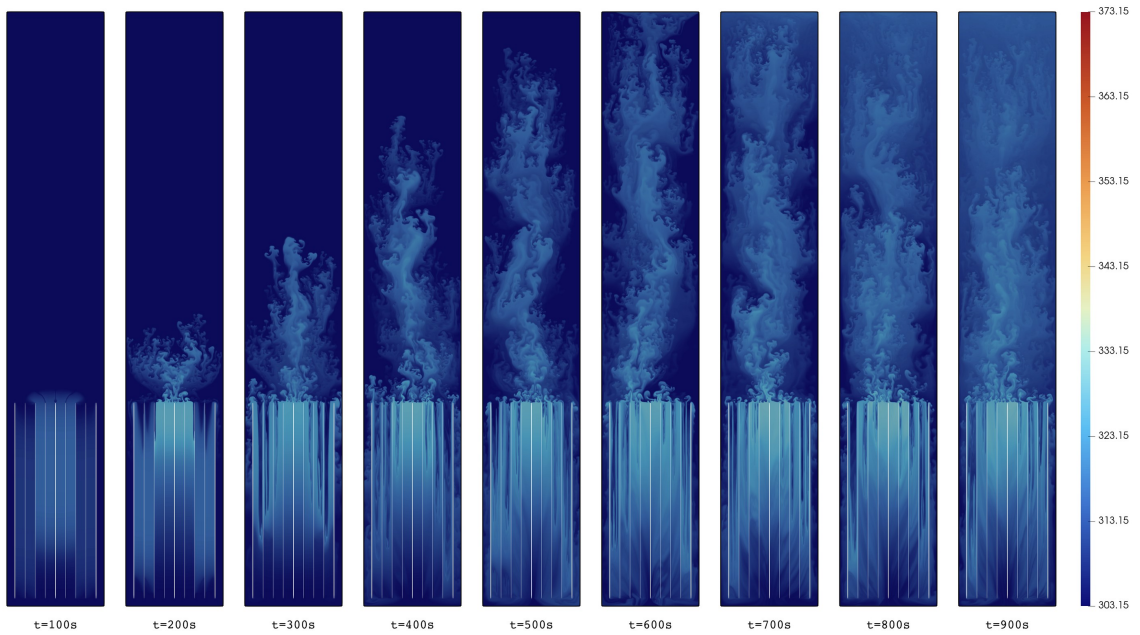
as well, and the backflow in the leftmost assembly disappears. This motion is also highlighted by the streamlines, as shown in Figure 3.15.



**Figure 3.15:** *Streamlines, coloured by velocity magnitude, in m/s, for  $t \in [1000\text{ s}; 5000\text{ s}]$  (Case 2).*

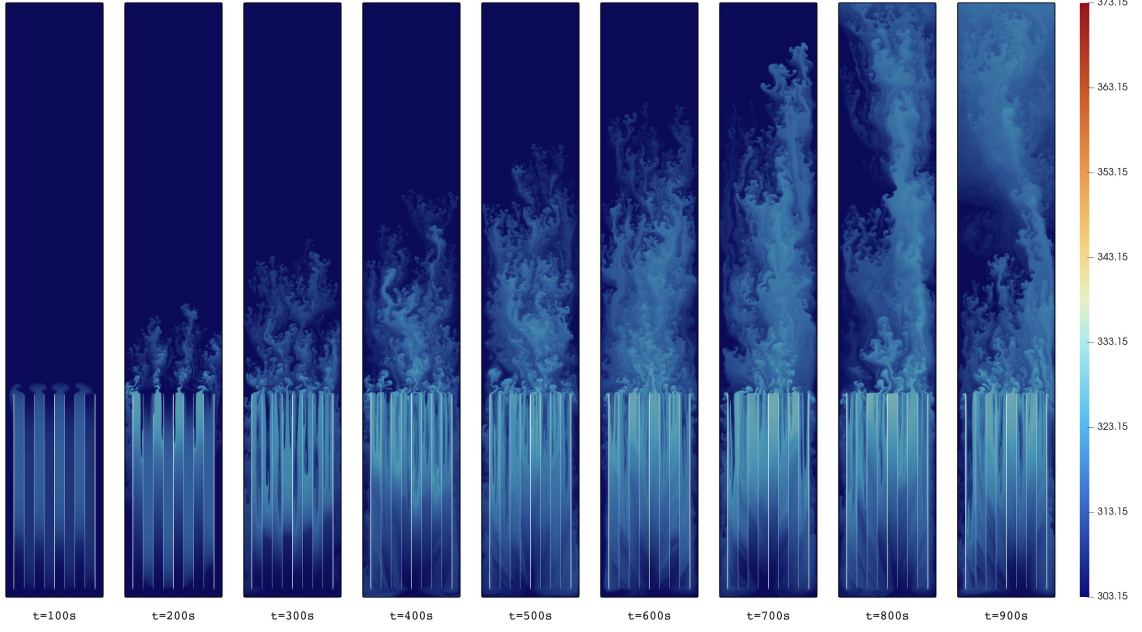
A predominant cell can then be observed on the left side of the pool, exhibiting a counter-clockwise flow. On the right side, the flow structures appear significantly more chaotic.

Regarding Case 3, the overall flow behaviour remains similar to that of Case 1. One notable difference is that the thermal plumes appear to reach the free surface after a shorter transient period, between 500 and 600 seconds, compared to around 700 seconds in Case 1, as shown in Figure 3.16.



**Figure 3.16:** *Temperature field, in Kelvin, for  $t \in [100\text{ s}; 900\text{ s}]$  (Case 3).*

Finally, regarding Case 4, the differences are minimal. The temperature field is shown in Figure 3.17 for completeness and to illustrate how the thermal plumes develop during the early stages. These plumes reach the free surface between 700 and 800 seconds, less than 100 seconds after those in Case 1.



**Figure 3.17:** *Temperature field, in Kelvin, for  $t \in [100\text{ s}; 900\text{ s}]$  (Case 4).*

### Evaporation analysis

Before comparing the evaporation rates between the different cases, the times required for the water to reach  $100^\circ\text{C}$  at any location in the pool are given in Table 3.1. Additionally, the variations with respect to Case 1, taken as the reference case, are also provided.

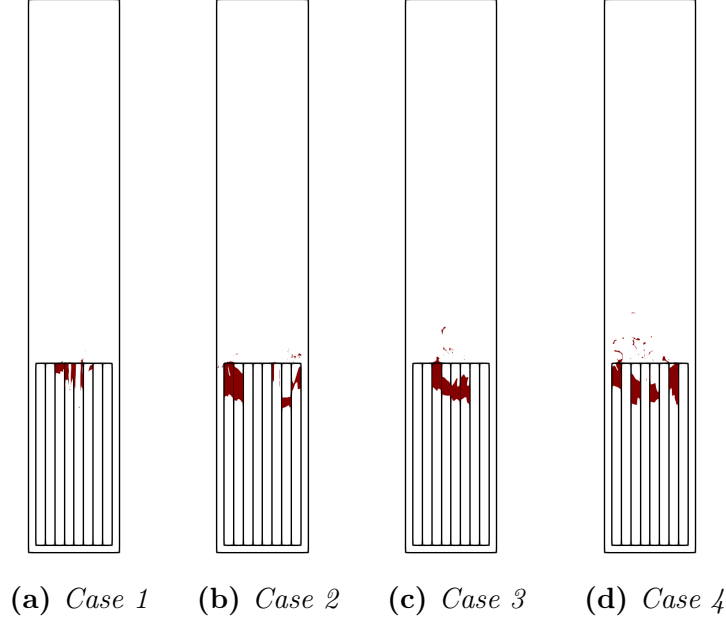
| Case 1 | Case 2 | Case 3 | Case 4 |
|--------|--------|--------|--------|
| 5299 s | 5318 s | 5389 s | 5386 s |
| /      | +0.36% | +1.70% | +1.64% |

**Table 3.1:** *Time, in seconds, for a point inside the pool to reach  $100^\circ\text{C}$ .*

We can immediately observe that these variations are very small, indicating that the heat distribution at the bottom of the pool has little influence on the time required for any location to reach  $100^\circ\text{C}$ .

It is important to clarify that, although we previously mentioned the boiling point of water at atmospheric pressure, we have avoided interpreting the  $100^\circ\text{C}$  threshold as the onset of boiling. Boiling occurs when the local temperature exceeds the local saturation temperature, which increases with pressure. As a result, at the bottom of the pool, the hydrostatic pressure raises the saturation point above  $100^\circ\text{C}$ . However, since Code\_Saturne is a purely single-phase solver, it does not account for boiling or any phase change. Therefore, the  $100^\circ\text{C}$  threshold is a numerical threshold only with no physical connection to boiling phenomena, which are beyond the scope of the present study.

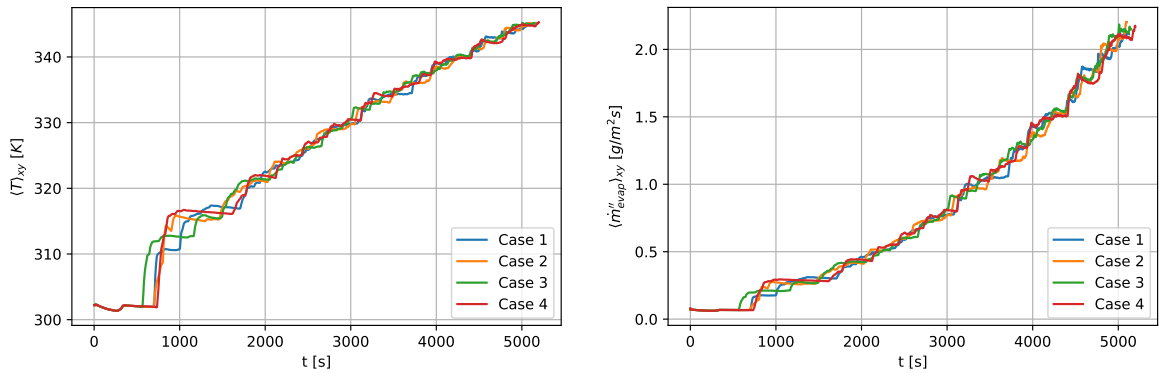
To locate these points within the pool, Figure 3.18 highlights the regions where the temperature lies between 99°C and 100°C, just before the end of the simulation.



**Figure 3.18:** Locations inside the pool that have reached 100°C.

Unsurprisingly, these hot regions are located at the upper part of the assemblies. In Case 1, with a uniform heat distribution, they are found on the central assemblies. In the other cases, they appear along the hot assemblies.

We can also compare the evolution of the area-averaged temperature at the free surface, as well as the area-averaged evaporative mass flux, across the different cases. These quantities are shown in Figure 3.19, where the data from Case 1 have been included to facilitate direct comparison.



**Figure 3.19:** Comparison of area-averaged free-surface temperature and area-averaged evaporative mass flux over time for the different cases.

Apart from initial differences during the first 1000 seconds, due to the varying times needed for the plumes to reach the free surface, the curves remain very similar. Nonetheless, even small differences accumulate over time and may lead to a faster overall descent of the free surface.



To conclude this section, we estimate the time required for the top of the assemblies to become exposed. To this end, we apply the same methodology as for Case 1 and use a fitting function of the form given in (3.3) to extrapolate the data obtained from our simulations. Table 3.2 then provides these times along with the relative variation compared to Case 1.

| Case 1        | Case 2        | Case 3       | Case 4        |
|---------------|---------------|--------------|---------------|
| 19h 22min 36s | 20h 27min 34s | 19h 2min 26s | 20h 11min 26s |
| /             | +5.59%        | -1.73%       | +4.2%         |

**Table 3.2:** *Time before exposition of the assemblies.*

We observe that in Case 2, although both leftmost assemblies heated significantly, as shown in Figure 3.18, this configuration still manages to delay the exposure of the assemblies by more than 5% compared to Case 1.

On a more general note, no clear correlation is observed between the time required for any location in the pool to reach 100°C and the time before the assemblies become exposed. Furthermore, we conclude that placing the hot assemblies at the ends of the rack and the cold ones in the centre, or alternating hot and cold assemblies, appears to delay the exposure of the assemblies. In contrast, placing the hot assemblies in the centre tends to accelerate the evaporation process and thus, the exposure of the assemblies.

### 3.1.4 Non-Oberbeck-Boussinesq flow

Until now, all simulations have been carried out under the Oberbeck-Boussinesq approximation. In this section, we compare the results obtained without invoking this assumption, focusing on the configuration of Case 4<sup>1</sup>. We recall that the governing system of equations solved in this case is given by (1.26)-(1.28).

To keep the main text concise, both the comparison of the flow behaviour and that of the evaporation rates between the two cases are presented in Appendix B.1. Regarding the time to reach 100°C, we find that without invoking the approximation, it is 5405 s, which is 0.36% higher than with the approximation (see Table 3.1). As for the time before exposure, we obtain 19h 27min and 24s, which is 3.63% lower than the case with the approximation (see Table 3.2).

To conclude, the results obtained without the Oberbeck-Boussinesq approximation differ from those invoking the approximation. However, the difference in the exposure time is only 3.6%, and no major change in the flow behaviour is observed, suggesting that the approximation still yields reasonably accurate results. Nevertheless, not relying on this approximation inevitably requires more computational resources. Details regarding numerical performance are given in the following section.

<sup>1</sup>The configuration of case 4 was somewhat arbitrarily selected for this comparison. Note that only this configuration was simulated without the Oberbeck-Boussinesq approximation as such simulations are significantly more computationally demanding and access to high-performance computing resources was limited due to heavy usage by other users.

### 3.1.5 Numerical performances

The computational resources used for this thesis were provided by the *Consortium des Équipements de Calcul Intensif* (CÉCI). In particular, the simulations were carried out on the HPC cluster *Lemaitre 4*, which features 40 nodes equipped with dual AMD EPYC 9534 64-core CPUs. Despite its substantial computational power, this cluster is shared and heavily used. To avoid long queue times, simulations were submitted in 4-hour blocks, using either 128 or 512 cores depending on availability. Hence, reporting the total simulation time does not allow for a meaningful comparison of computational cost. Instead, Table 3.3 provides the number of CPU core-hours used by each simulation.

| Case 1           | Case 2           | Case 3           | Case 4 (a)       | Case 4 (b)       |
|------------------|------------------|------------------|------------------|------------------|
| $\approx 3000$ h | $\approx 2800$ h | $\approx 2900$ h | $\approx 3000$ h | $\approx 4800$ h |

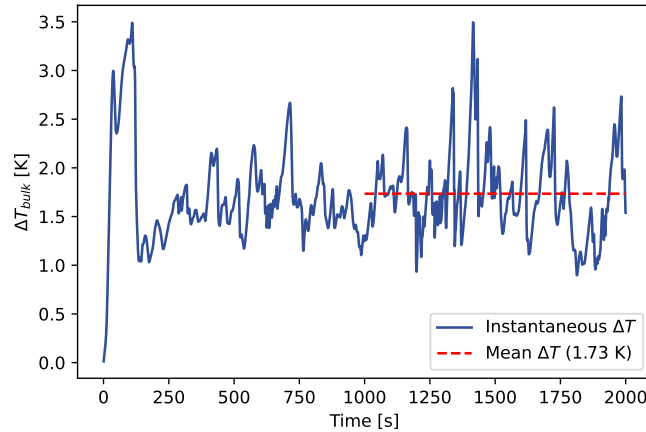
**Table 3.3:** Total CPU core-hours for each simulations. Case 4 (a) and 4 (b) refer to simulations performed with and without the Oberbeck-Boussinesq approximation, respectively.

We can therefore clearly observe that using the Oberbeck-Boussinesq approximation indeed leads to a significant reduction in computational time.

## 3.2 Three-dimensional geometry

### 3.2.1 Preliminary considerations

Similarly to the two-dimensional geometry, we begin by evaluating the Rayleigh number in the pure fluid region to characterize the flow regime. To do so, we perform a coarser simulation to estimate the temperature difference across the bulk. The results obtained are shown in Figure 3.20.

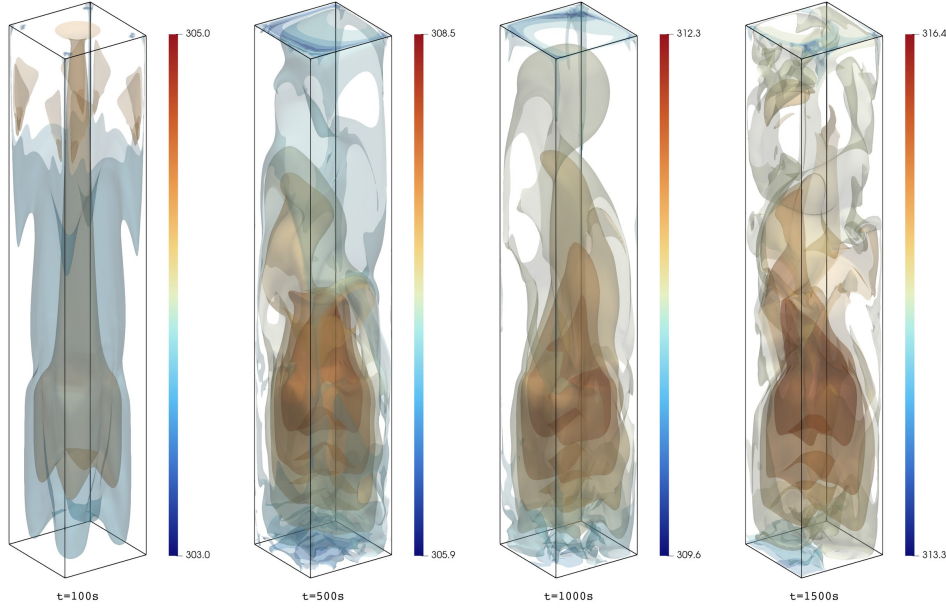


**Figure 3.20:** Instantaneous and time-averaged temperature difference across the bulk.

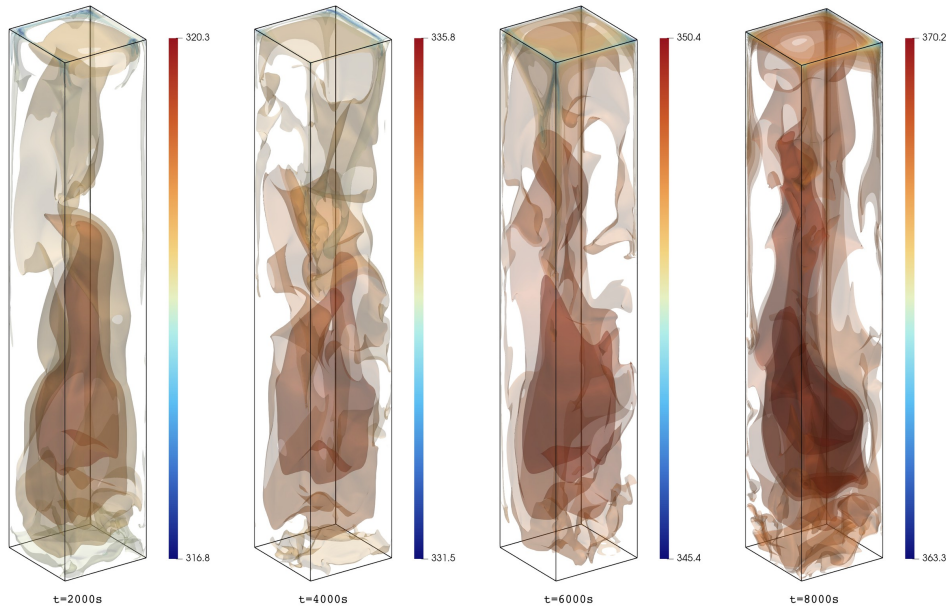
Here, the data suggests that the transient period before thermal plumes reach the free surface lasts less than 200 seconds. However, we still choose to average the temperature difference over the interval ranging from 1000 to 2000 seconds. The resulting mean temperature difference is 1.73K. Then, using (3.1) and (3.2) we get a Rayleigh number  $Ra_{bulk} \approx 2.3 \times 10^{10}$  and a reference velocity  $u_{0N} \approx 0.065$  m/s. Hence, we conclude that flow lies in the hard turbulent regime.

### 3.2.2 Numerical results - Case 1

Following the same approach as for the two-dimensional geometry, we begin by examining the temperature field. It is important to emphasize that, as demonstrated by the Rayleigh numbers calculated in Sections 3.1.1 and 3.2.1, these simulations are fundamentally different, and no direct comparison will be made in this section. Moreover, since the results are three-dimensional, temperature iso-surfaces are shown. As a result, the colour map varies from one time step to another. Figures 3.21 and 3.22 illustrate the evolution of the temperature field over the entire duration of the simulation.



**Figure 3.21:** *Temperature iso-surfaces, in Kelvin, for  $t \in [100s; 1500s]$ .*

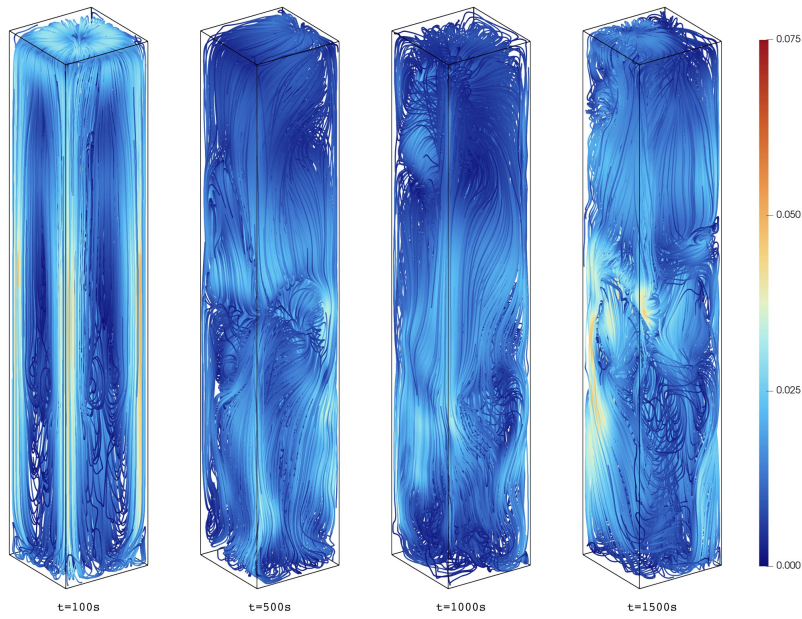


**Figure 3.22:** *Temperature iso-surfaces, in Kelvin, for  $t \in [2000s; 8000s]$ .*

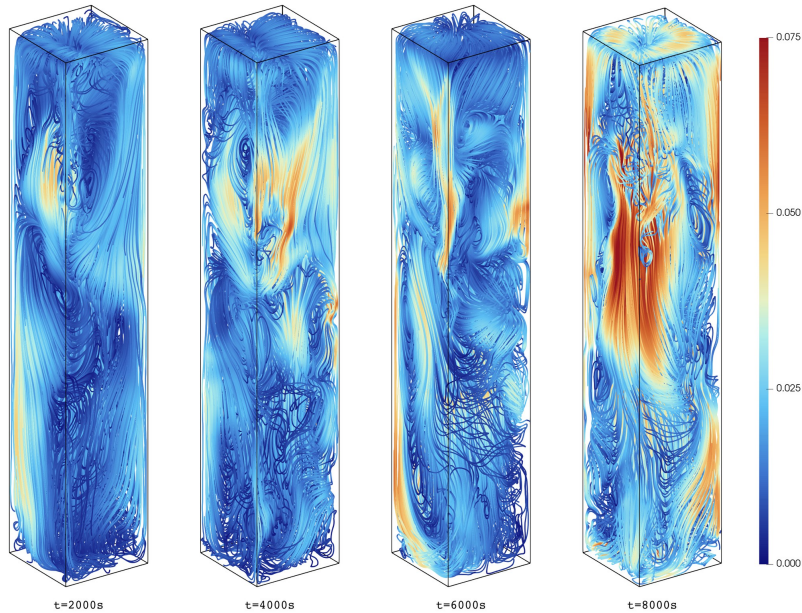
At the very early stages, we observe a thermal plume rising from the top of the assemblies toward the free surface. By the time the first 100 seconds have elapsed, this plume has

already reached the free surface. This is therefore consistent with the initial observation made in Figure 3.20, which suggested that the transient period was very short. Moreover, we see that the fluid temperature steadily increases over time. This is due to accumulation of heat inside the pool, and it will be further discussed later. In the present case, it takes 8427 seconds for any point within the pool to reach 100°C.

Streamlines can also be shown to provide a clearer understanding of the flow behaviour in the upper region of the pool, as illustrated in Figures 3.23 and 3.24.



**Figure 3.23:** *Streamlines, coloured by velocity magnitude, in m/s, for  $t \in [100\text{ s}; 1500\text{ s}]$ .*

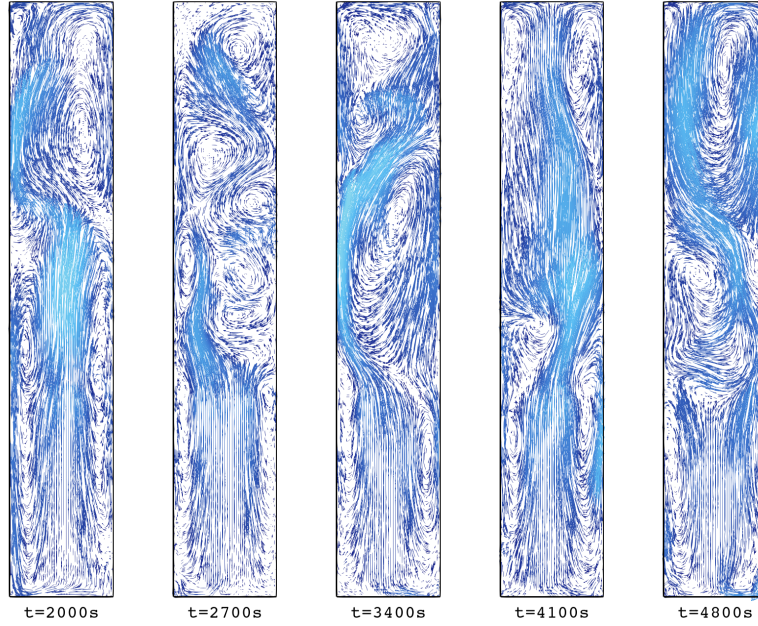


**Figure 3.24:** *Streamlines, coloured by velocity magnitude, in m/s, for  $t \in [2000\text{ s}; 8000\text{ s}]$ .*

These streamlines show that the fluid heated by the assemblies rises through the centre of the pool and then, upon reaching the free surface, the fluid cooled by evaporation



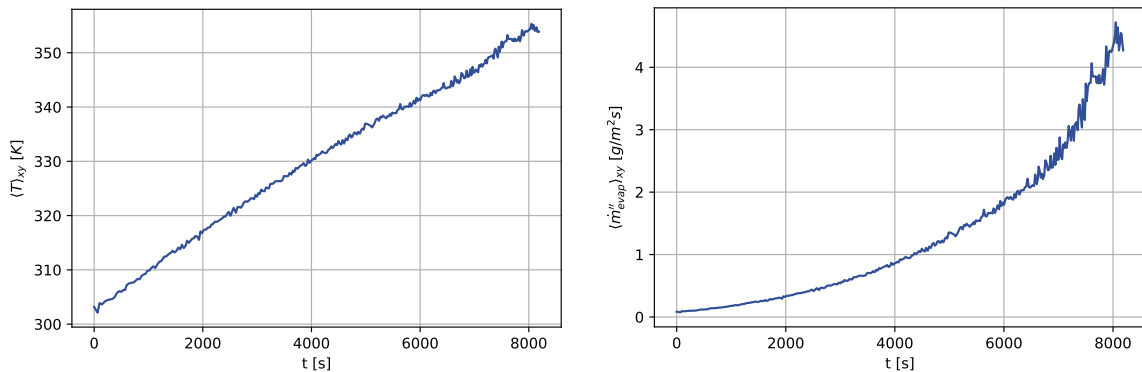
sinks along the sides of the pool. In addition, convection cells develop within the domain. While they are not particularly distinct in Figure 3.23, they become clearly apparent over the long term, as shown in Figure 3.24. To better understand the behaviour of these convection cells, a cross-section in the central  $x$ - $z$  plane is shown in Figure 3.25.



**Figure 3.25:** Velocity vectors in the central  $x$ - $z$  plane, coloured by velocity magnitude, in m/s. The colour map used here is the same as the one in Figures 3.23 and 3.24.

As we can see, the convection cells undergo significant reorganization over time. Still, a general trend can be identified. The cells tend to stack vertically along the domain height, and there are one or two of them across its width. The total number of cells, however, appears to vary randomly. As a final note, and with reference to Figure 3.24, the reference velocity of 0.065 m/s seems to be a fairly accurate estimate.

Now, regarding the evaporation at the free surface, the area-averaged free-surface temperature and the area-averaged evaporative mass flux are shown in Figure 3.26.

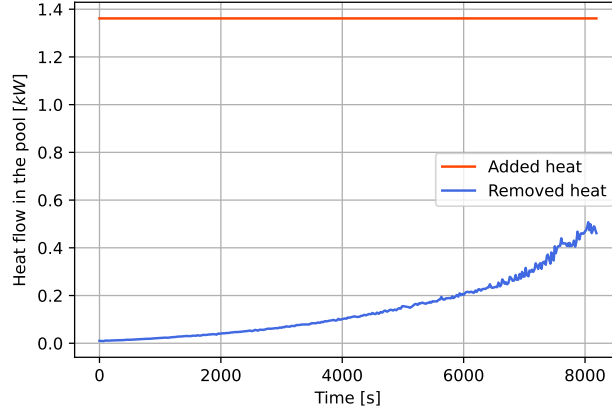


**Figure 3.26:** Area-averaged free-surface temperature and area-averaged evaporative mass flux at the free surface as a function of time.

We immediately observe a steady increase in the area-averaged free-surface temperature over time. This behaviour is consistent with the overall rise in temperature previously

noted in Figure 3.22. Moreover, the free-surface temperature begins to increase from the very first moments, as the thermal plume reaches the free surface very rapidly. Regarding the area-averaged evaporative mass flux, an exponential increase is observed. Furthermore, the values for the mass flux, ranging from 0.1 to 5 g/m<sup>2</sup>s over the temperature interval of 303 K to 355 K are consistent with those reported by Hay [10].

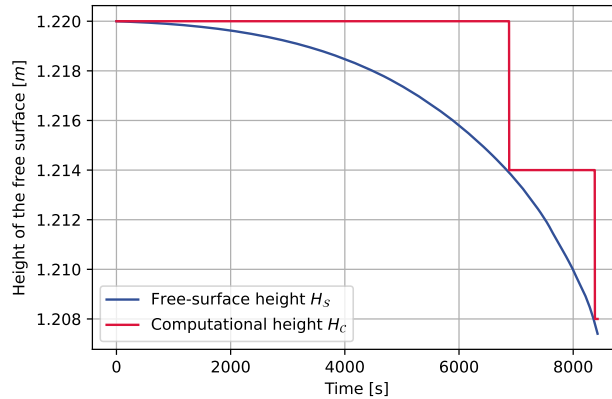
Regarding the heat flow inside the pool, Figure 3.27 shows both the heat removed through the free surface and the heat added to the fluid<sup>2</sup> by the assemblies.



**Figure 3.27:** *Heat loss through the free surface and heat generated by the assemblies.*

A global rise in the water temperature is clearly inevitable. Indeed, the heat added to the fluid by the assemblies is, on average over the course of the simulation, eight times higher than the amount of heat removed through the free surface.

Regarding the descent of the free surface, the evolution of the computational domain height  $H_c$  and the free-surface height  $H_s$  are shown in Figure 3.28.

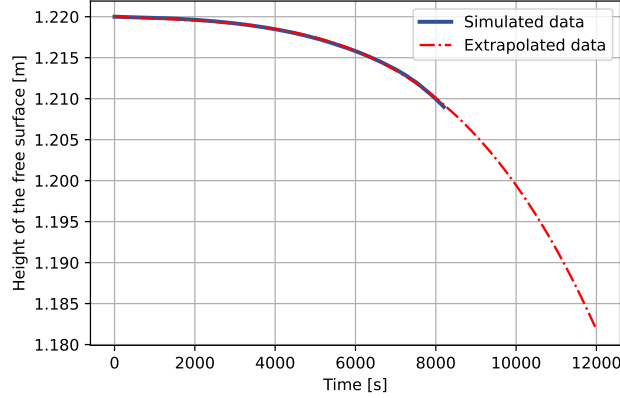


**Figure 3.28:** *Evolution of the computational domain height and the free-surface height.*

Concerning the height of the computational domain, it appears that two remeshing procedures were carried out during the simulation. As for the free-surface height, it follows a similar trend to that observed in the two-dimensional geometry. Therefore, the data

<sup>2</sup>We recall that the heat added to the fluid is defined as the integral over the height of the assemblies of the fluid volumetric heat source  $q_f'''$  defined in (2.49).

extrapolation will be performed using a function of the same form as the one given in (3.3). Figure 3.29 shows the resulting extrapolation.



**Figure 3.29:** *Extrapolation of the height of the free surface from simulated data.*

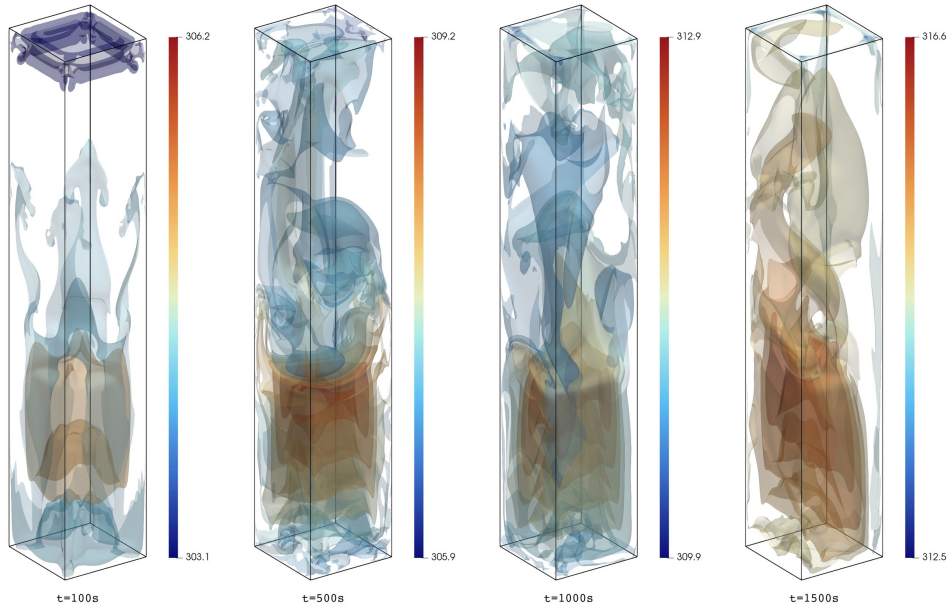
We observe that the function in (3.3) provides a good fit to the simulated data. Once evaluated, it indicates that the assemblies become exposed after 7h, 53min and 4s.

### 3.2.3 Comparison with the other cases

In the following, we briefly illustrate the differences in flow behaviour between the various cases and compare the times required for the assemblies to become uncovered.

#### Flow behaviour

Figure 3.30 illustrates the early stage evolution of the temperature field in Case 2.

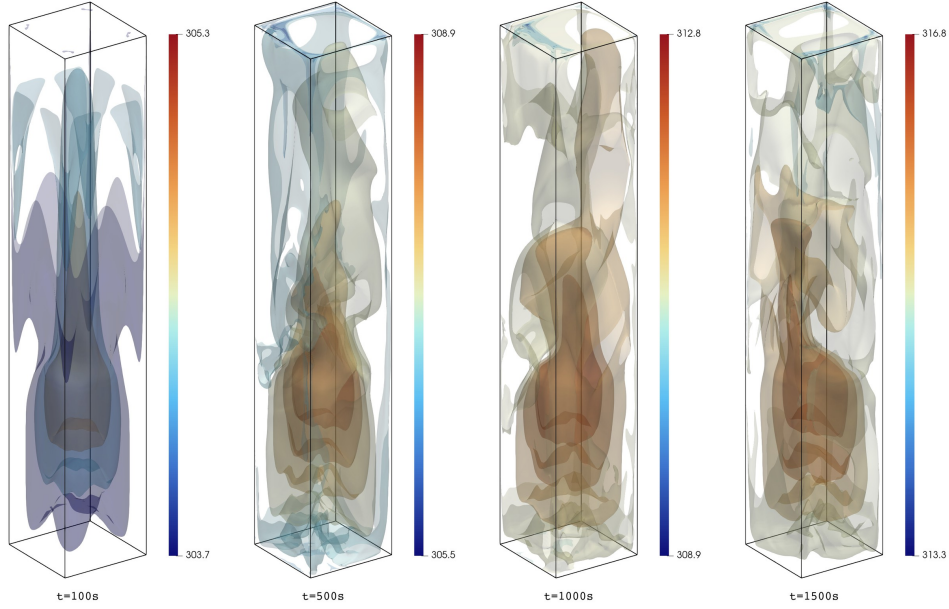


**Figure 3.30:** *Temperature iso-surfaces, in Kelvin, for  $t \in [100s; 1500s]$  (Case 2).*

In this case, the hot assemblies are located at the edges of the rack, which results in four thermal plumes rising from each corner of the pool. Interestingly, these plumes ascend

more slowly than in Case 1. The iso-surface at the top of the pool shows a slight decrease in temperature in that region due to evaporation.

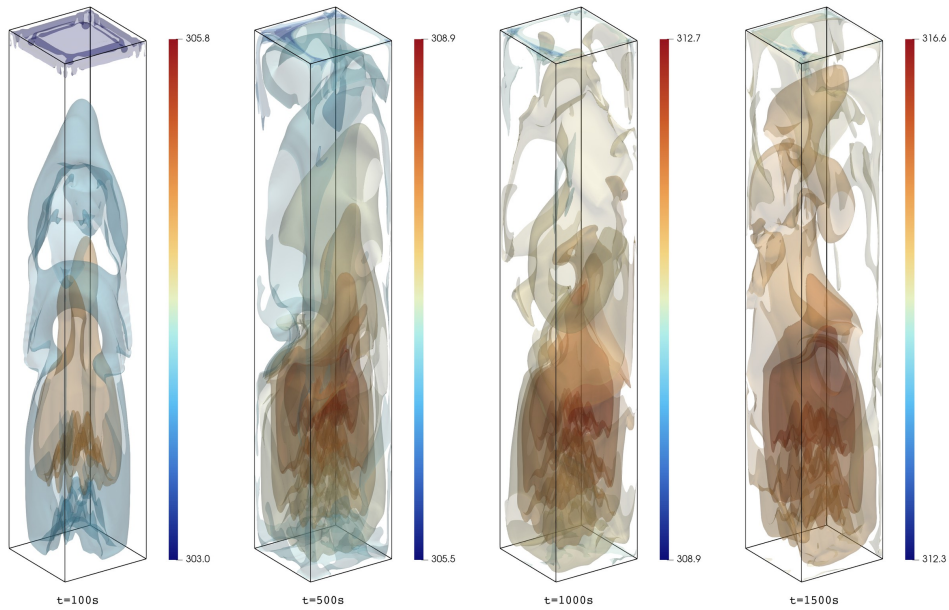
For Case 3, characterised by the hot assemblies located at the centre of the rack, the early stage evolution of the temperature field is shown in Figure 3.31.



**Figure 3.31:** *Temperature iso-surfaces, in Kelvin, for  $t \in [100s; 1500s]$  (Case 3).*

As expected, a thermal plume rises from the centre of the assemblies. However, this plume takes longer to reach the free-surface than in Case 1.

Finally, Figure 3.32 depicts the early temperature field evolution in Case 4.



**Figure 3.32:** *Temperature iso-surfaces, in Kelvin, for  $t \in [100s; 1500s]$  (Case 4).*

Here, the non-uniform heat distribution at the bottom of the pool gives the rising plume

a distinct shape, splitting it into two branches. Moreover, this configuration causes heat to remain trapped along a larger portion of the assemblies height.

Regarding the streamlines for the different cases, they appear very similar and provide no additional information. For this reason, they are not included in the present document.

### Evaporation analysis

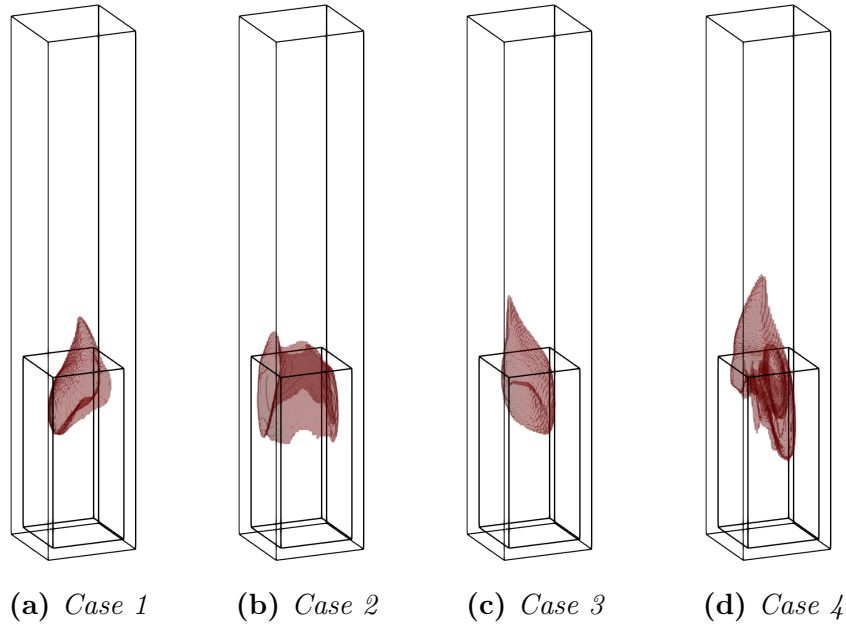
To begin, we compare the times required for the water to reach  $100^{\circ}\text{C}$  at any location in the pool. These times are provided in Table 3.4, along with their variations relative to Case 1, which serves as the reference.

| Case 1 | Case 2 | Case 3 | Case 4 |
|--------|--------|--------|--------|
| 8427 s | 8504 s | 8303 s | 8445 s |
| /      | +0.91% | -1.47% | +0.21% |

**Table 3.4:** *Time, in seconds, for a point inside the pool to reach  $100^{\circ}\text{C}$ .*

We observe that these variations are small, indicating a minor dependence on the heat distribution at the bottom of the pool.

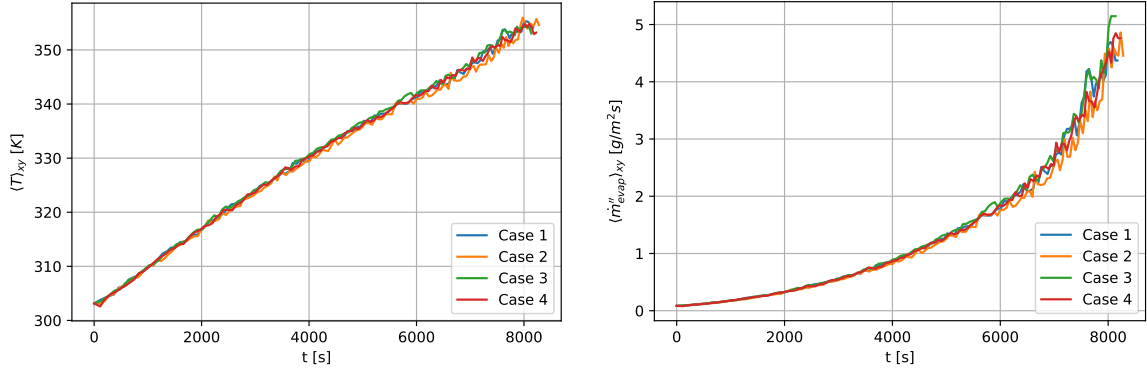
We now proceed to determine where these points are situated within the pool. To proceed, Figure 3.33 shows the locations where the temperature ranges between  $99^{\circ}\text{C}$  and  $100^{\circ}\text{C}$ , just before the end of the simulation.



**Figure 3.33:** *Locations inside the pool that have reached  $100^{\circ}\text{C}$ .*

We observe that, regardless of the case, these regions are located within the top of the central assemblies, as well as just above the assemblies, that is to say, in the fluid. This result is quite surprising, especially for Case 2, where the central assemblies are the cold ones but still reach  $100^{\circ}\text{C}$  first.

Regarding the evaporation rates, Figure 3.34 depicts the evolution of the area-averaged temperature at the free-surface, as well as the area-averaged evaporative mass flux, across the different cases.



**Figure 3.34:** Comparison of area-averaged free-surface temperature and area-averaged evaporative mass flux over time for the different cases.

Up to around 6000 seconds, no significant differences are observed. After that, Case 3 exhibits the highest evaporative mass flux, while Case 2 shows the lowest, with Cases 1 and 4 falling in between. Although the differences remain small, they accumulate over time, which can impact the overall descent of the free surface. The evolution of the computational domain height and free-surface height over time for the different cases is provided in Appendix C.1.

To conclude this section, we estimate the time required for the top of the assemblies to become exposed. To do so, we extrapolate the simulated data using the function given in (3.3). Table 3.5 provides these times along with the relative variation compared to Case 1.

| Case 1      | Case 2       | Case 3       | Case 4       |
|-------------|--------------|--------------|--------------|
| 7h 53min 4s | 8h 10min 14s | 7h 45min 58s | 7h 52min 58s |
| /           | +3.63%       | -1.50%       | -0.02%       |

**Table 3.5:** Time before exposition of the assemblies.

We conclude that placing the cold assemblies at the centre of the rack and the hot ones around them tends to delay the exposure of the assemblies. Alternating hot and cold assemblies does not appear to offer any advantage over a uniform distribution, but still performs better than placing the hot assemblies at the centre. Furthermore, there appears to be a slight correlation between the time required for any point in the pool to reach 100°C and the time until assembly exposure.

### 3.2.4 Non-Oberbeck-Boussinesq flow

In the previous sections, all simulations were performed under the Oberbeck-Boussinesq approximation. We will now compare the results with and without this approximation, focusing on the configuration of Case 4.



Without invoking the approximation, the time needed for a point to reach 100°C inside the pool is 8354 s, which is 1.08% less than with the approximation (see Table 3.4). For the time before exposure, we obtain 8h, 12min and 15s, which is 4.08% more than the case with the approximation (see Table 3.5). Regarding the flow behaviour, it is very similar. A comparison is provided in Appendix C.2.

Using the Oberbeck-Boussinesq approximation inevitably yields slightly different results, which are generally less accurate than those obtained without it. In this case, the difference in exposure time is about 4%. It would then be necessary to verify that the variations across the different cases scale similarly. However, the issue is that simulations without this approximation are significantly more costly, as will be highlighted in the following section.

### 3.2.5 Numerical performances

Similarly to the simulations for two-dimensional geometry, the simulations for the three-dimensional geometry were submitted in 4-hour blocks, using either 128 or 512 cores. Table 3.6 provides the number of CPU core-hours used by each simulation.

| Case 1           | Case 2           | Case 3           | Case 4 (a)       | Case 4 (b)       |
|------------------|------------------|------------------|------------------|------------------|
| $\approx 2400$ h | $\approx 2400$ h | $\approx 1900$ h | $\approx 1900$ h | $\approx 3000$ h |

**Table 3.6:** Total CPU core-hours for each simulations. Case 4 (a) and 4 (b) refer to simulations performed with and without the Oberbeck-Boussinesq approximation, respectively.

Surprisingly, Cases 3 and 4 (a) required less computational time than Cases 1 and 2. On the other hand, the simulation without the Oberbeck-Boussinesq approximation took considerably longer, as one could have expected.

## 3.3 Results discussion

Until now, we have considered the two-dimensional and three-dimensional geometries separately, as the Rayleigh number characterising the flow in the pure-fluid region differs significantly. However, some meaningful conclusions can still be drawn from the comparison of the various results.

Despite the fundamentally different flow behaviours, we observe that the heat distribution at the bottom of the pool has a non-negligible impact on the evaporation rates. Specifically, for a given total thermal power, it is advantageous to place the cold assemblies at the centre of the rack and the hot assemblies around them. To quantify this impact, this arrangement delays the exposure of the assemblies to ambient air by 6.94%<sup>3</sup> in the two-dimensional geometry and by 4.95%<sup>4</sup> in the three-dimensional case, compared to the opposite configuration where hot assemblies are placed at the centre and cold ones around them. A checkerboard-like distribution, in which hot and cold assemblies alternate, yields intermediate results. It performs better than placing hot assemblies at the centre but remains less effective than placing the cold ones at the centre.

<sup>3</sup>Relative variation between Case 2 and Case 3, deduced from the values in Table 3.2.

<sup>4</sup>Relative variation between Case 2 and Case 3, deduced from the values in Table 3.5.



Regarding the heat production of the assemblies, which, it should be recalled, is based on data provided by the ASNR as discussed in Section 2.6, it is significantly higher than the amount of heat that can be removed by evaporation at the free surface. As a result, the water temperature continuously rises until certain locations in the pool reach 100°C. Regardless of the case, or the geometry, these hotspots are mainly found inside the assemblies and particularly in their upper regions. As for the time required for these zones to appear, it varies little with the heat distribution but depends on the geometry. It can thus be assumed that the total heat load at the bottom of the pool, as well as the dimensions of the pool, are the primary factors influencing this time.

To conclude this section, we compare the general-flow behaviours observed in the two-dimensional and three-dimensional geometries, since detailed comparisons of the different cases within each geometry have already been carried out in their respective sections. In the two-dimensional geometry, we observed a highly turbulent flow with the formation of several fragmented thermal plumes. Moreover, these plumes took slightly more than ten minutes to reach the free surface. Regarding the flow pattern, two large convection cells were observed, extending from the top of the assemblies to the free surface and positioned side by side. In contrast, the three-dimensional geometry exhibited much more coherent and orderly convection cells, stacked vertically according to the height of the domain. Additionally, a single, well-defined thermal plume was observed in the early stages, taking only between one and two minutes to reach the free surface. These differences in flow behaviour can primarily be explained by the difference in scale used between the two geometries. Indeed, for a given fluid under fixed initial conditions, the Rayleigh number solely depends on the temperature difference across the domain under study and the cube of its height. Therefore, it is clear that the height of the domain, being the water column above the assemblies in the present study, plays a critical role. Since we considered a water column in the 3D geometry at a one-tenth scale of the one in the 2D geometry, we expect a difference of three orders of magnitude between the characteristic Rayleigh numbers of the flows in the two geometries. Next, there is the impact of the temperature difference, which is difficult to quantify *a priori*. In this study, we observed that the temperature difference between the two geometries varies by slightly less than one order of magnitude, which adds another order of magnitude of difference between the characteristic Rayleigh numbers. What should be retained is that, in reality, the water columns of spent fuel pools are several metres high, which results in a very large Rayleigh number and, consequently, intense turbulence. Using a reduced-scale geometry significantly lowers the level of turbulence, unless the temperature difference is artificially increased, which is not possible in the context of this study. As a result, the two-dimensional geometry yields a flow behaviour that is more representative of real conditions than the three-dimensional one. Nevertheless, as we have seen, the three-dimensional geometry still proves useful in highlighting certain phenomena, such as the impact of heat distribution at the bottom of the pool.

# Conclusion

In this Master’s thesis, we investigated the flow behaviour in spent fuel pools during the early stage of a loss-of-cooling accident, that is, when the fuel assemblies are still fully submerged in water and the only heat exchange with the environment takes place via evaporation through the free surface. Such a scenario occurred during the Fukushima disaster in Japan in 2011. To carry out this study, we used Code\_Saturne, an open-source software developed by EDF. Particular attention was given to the evaporation rates at the free surface, which allowed us to estimate the time before the assemblies become exposed to the ambient air. Two geometries were considered, namely a two-dimensional full-scale geometry and a three-dimensional one, in which the dimensions of the assemblies are more representative of reality, but at reduced scale. For each of these geometries, four different cases were analysed, each representing a distinct heat distribution at the bottom of the pool.

It was found that, for the same total thermal power at the bottom of the pool, placing the lower-power assemblies at the centre, surrounded by higher-power ones, increased the time before the assemblies become exposed to air, compared to the opposite configuration, where higher-power assemblies are located at the centre. On the other hand, a checkerboard-like power distribution, in which lower-power and higher-power assemblies alternate, yielded intermediate exposure times.

Regarding the flow behaviour, the two-dimensional geometry exhibited intense turbulence, characterized by chaotic convection cells. In contrast, the flow in the three-dimensional geometry showed much more coherent convective structures. These differences in flow regimes are a direct consequence of the reduced scale of the three-dimensional geometry. Nonetheless, the conclusions about the advantages of one heat distribution over another remain consistent across both geometries, although the differences are less pronounced in the three-dimensional geometry.

It is important to acknowledge the limitations of our study. To address these, we propose several improvements, which could further enhance the present work, arranged according to the order we deem most relevant.

- Regarding the volumetric heat source inside the assemblies, an additional equation should be solved within the fuel rods, and a heat transfer coefficient defined to accurately quantify the heat actually delivered by the rods to the fluid, as explicitly discussed in Section 2.6.
- Simulations should be performed without invoking the Oberbeck-Boussinesq approximation.

- A realistic model for head losses in the transverse direction of the rods of an assembly should be used instead of the simplistic approach developed in Section 2.5.
- A three-dimensional full-scale mesh should be developed to achieve a Rayleigh number in the pure fluid region that is more representative of reality. Additionally, this mesh should incorporate physical separations between each assembly.
- Boiling inside the assemblies should be considered. This phenomenon could either be modelled through additional heat transfer and phase change models or studied using a two-phase flow solver. For instance, Code\_Neptune, a multiphase-flow solver built on top of Code\_Saturne could be employed, although it is not open source.

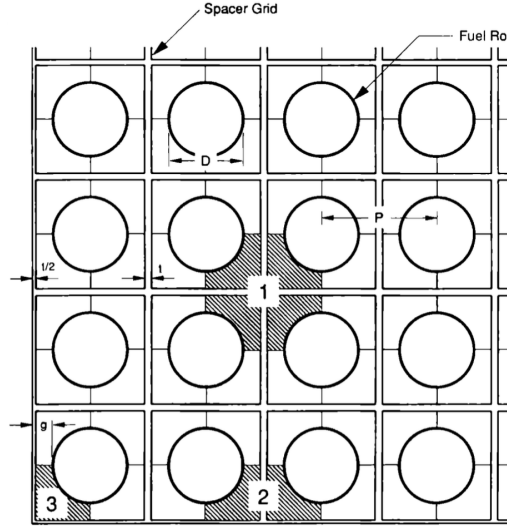
# Appendix A

## Head losses inside the assemblies

### A.1 Computation of the friction coefficients

The total pressure drop along a fuel assembly includes the friction pressure drop along the fuel rods and the form losses due to the presence of spacers. However, since spacers are not considered in the present work, form losses are not accounted for. In what follows, we will refer to the *bare rod bundles* as the set of fuel rods without spacers.

This section aims to determine the bare bundle friction coefficient  $C_{f,b}$ , both in the laminar and the turbulent regimes. To do so, we consider that each assembly can be divided into three subchannels, namely, the *interior* subchannel, the *edge* subchannel and the *corner* subchannel. Figure A.1 highlights the different subchannels.



**Figure A.1:** Cross-section of an assembly highlighting the different subchannels: interior subchannel (1), edge subchannel (2) and corner subchannel (3). This figure is taken from [58].

For each subchannel  $i$ , we compute a friction coefficient  $C_{f,i,\star}$ , defined as:

$$C_{f,i,\star} = a_{i,\star} + b_{i,\star}(F - 1) + c_{i,\star}(F - 1)^2 \quad (\text{A.1})$$

where  $F = P/D$  for the interior subchannel and  $F = W/D$  for the edge and corner subchannels, with  $D$  the diameter of a single fuel rod,  $P$  the distance between the centres

of two fuel rods, and  $W$  the sum of the radius of a fuel rod and the gap  $g$  between the walls and the closest fuel rod. The subscript " $\star$ " denotes either the laminar regime (" $L$ ") or the turbulent regime (" $T$ "). The coefficients  $a_{i,\star}$ ,  $b_{i,\star}$  and  $c_{i,\star}$  are listed in Tables A.1, A.2 and A.3.

Once the six friction factors have been determined<sup>1</sup>, the bare bundle friction factors for the laminar and turbulent regimes can be computed:

$$C_{f,b,\star} = D_{h,b} \left[ \sum_{i=1}^3 S_i \left( \frac{D_{h,i}}{D_{h,b}} \right)^{\frac{n}{2-n}} \left( \frac{C_{f,i,\star}}{D_{h,i}} \right)^{\frac{1}{n-2}} \right]^{n-2} \quad (\text{A.2})$$

with  $D_{h,b}$  the hydraulic diameter of the cross-section occupied by the fluid around the bare rod bundle,  $D_{h,i}$  the hydraulic diameter of the subchannel  $i$ ,  $S_i$  a geometric parameter relative to the subchannel  $i$ , defined in Section A.2, and  $n$  an exponent depending on the flow regime. For the laminar regime,  $n = 1$ , while for the turbulent regime,  $n = 0.18$ .

The friction factor for each flow regime is then computed as follows:

$$f_{D,\star} = \frac{C_{f,\star}}{Re_b^n} \quad (\text{A.3})$$

Regarding the transition regime, the friction factor is defined by weighting the laminar friction factor and turbulent friction factor by an intermittency factor  $\Psi$ , as discussed in Section 2.5.

Finally, the laminar to transition Reynolds number  $Re_{b,L}$  and the transition to turbulent Reynolds number  $Re_{b,T}$  are given by:

$$Re_{b,L} = 10^{[1.7(P/D)+0.78]} \quad (\text{A.4})$$

$$Re_{b,T} = 10^{[0.7(P/D)+3.30]} \quad (\text{A.5})$$

|                  | $a_1$  | $b_1$   | $c_1$    |
|------------------|--------|---------|----------|
| Laminar regime   | 35.55  | 263.7   | -190.2   |
| Turbulent regime | 0.1339 | 0.09059 | -0.09926 |

**Table A.1:** Coefficients in (A.1) for the interior subchannel (1).

|                  | $a_2$  | $b_2$   | $c_2$    |
|------------------|--------|---------|----------|
| Laminar regime   | 44.40  | 256.7   | -267.6   |
| Turbulent regime | 0.1430 | 0.04199 | -0.04428 |

**Table A.2:** Coefficients in (A.1) for the edge subchannel (2).

|                  | $a_3$  | $b_3$   | $c_3$    |
|------------------|--------|---------|----------|
| Laminar regime   | 58.83  | 160.7   | -203.5   |
| Turbulent regime | 0.1452 | 0.02681 | -0.03411 |

**Table A.3:** Coefficients in (A.1) for the corner subchannel (3).

---

<sup>1</sup>There is one friction factor for each subchannel, in each flow regime.

## A.2 Geometric parameters

For the arrangement of fuel rods shown in Figure A.1, the cross-sectional area  $A_i$  and wetted perimeter  $P_{w,i}$  of each subchannel are given by:

$$A_1 = P^2 - \pi(D^2/4) \quad P_{w,1} = \pi D \quad (\text{A.6})$$

$$A_2 = PW - \pi(D^2/8) \quad P_{w,2} = W + \pi(D/2) \quad (\text{A.7})$$

$$A_3 = W^2 - \pi(D^2/16) \quad P_{w,3} = 2W + \pi(D/4) \quad (\text{A.8})$$

Using these, the total cross-sectional area  $A_b$  and wetted perimeter  $P_{w,b}$  of the bare rod bundle can then be computed:

$$A_b = \sum_{i=1}^3 N_i A_i \quad \text{and} \quad P_{w,b} = \sum_{i=1}^3 N_i P_{w,i} \quad (\text{A.9})$$

with  $N_i$  the number of subchannels  $i$ . Consequently, we can compute the hydraulic diameter  $D_{h,i}$  and the geometric parameter  $S_i$  of each subchannel:

$$D_{h,i} = \frac{4A_i}{P_{w,i}} \quad \text{and} \quad S_i = \frac{N_i A_i}{A_b} \quad (\text{A.10})$$

Finally, the hydraulic diameter of the bare rod bundle  $D_{h,b}$  is given by:

$$D_{h,b} = \frac{4A_b}{P_{w,b}} \quad (\text{A.11})$$

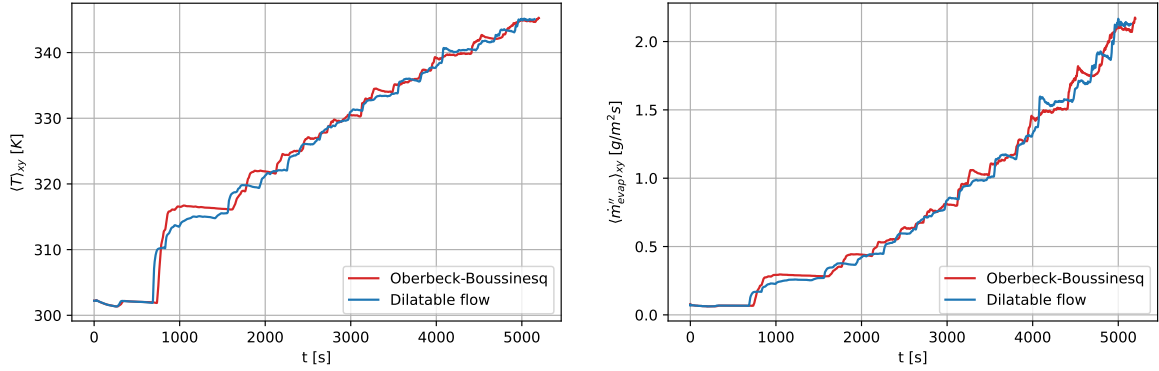
# Appendix B

## Additional figures for the two-dimensional geometry

### B.1 Non-Oberbeck-Boussinesq flow

This section aims to briefly present and compare the flow behaviour and evaporation rates for the configuration of Case 4, with and without the Oberbeck-Boussinesq approximation.

First, Figure B.1 shows the area-averaged free-surface temperature and the area-averaged evaporative mass flux for the case with and without the approximation.

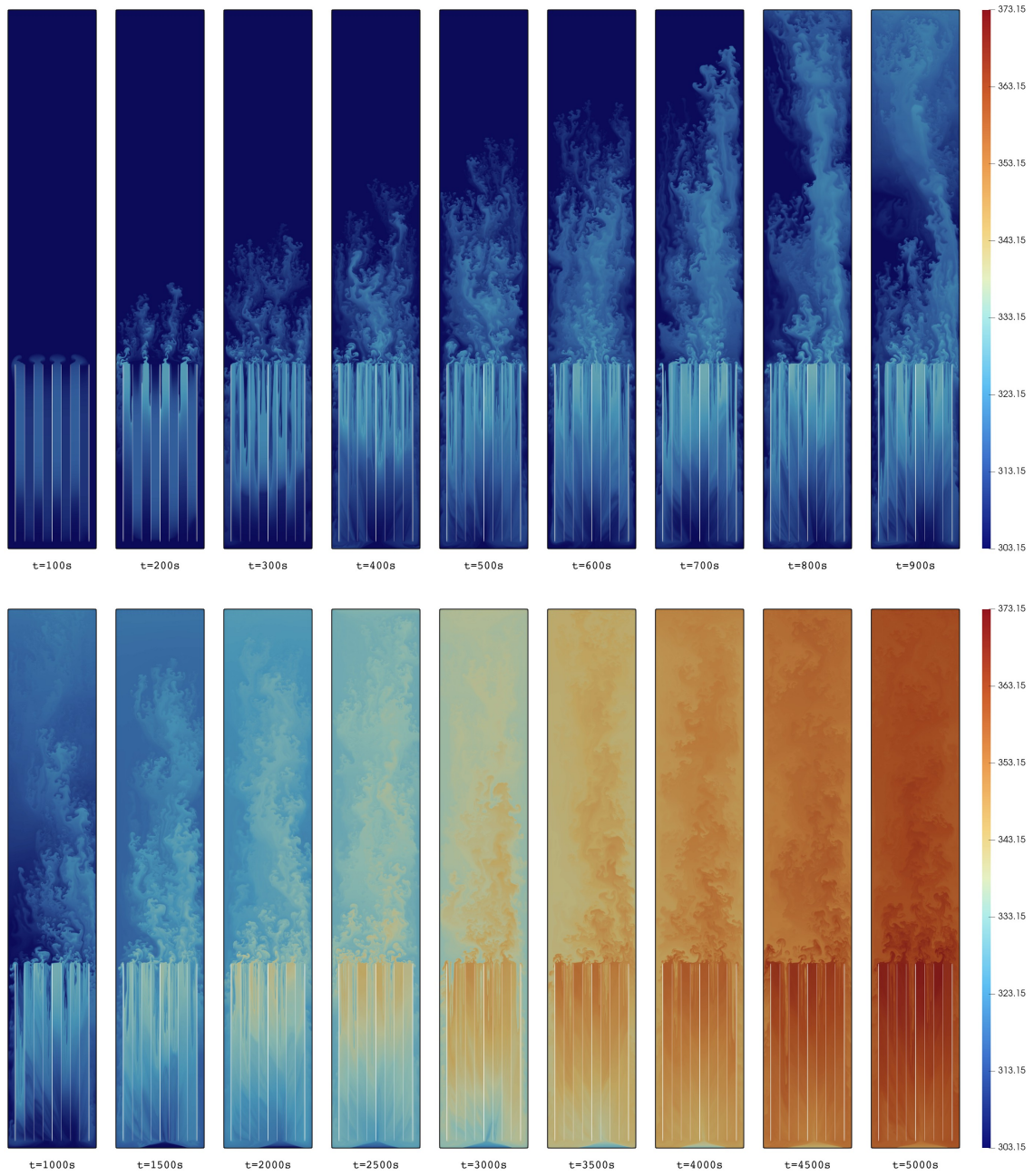


**Figure B.1:** Comparison of area-averaged free-surface temperature and area-averaged evaporative mass flux over time for Case 4 with and without the Oberbeck-Boussinesq approximation.

A slight overestimation by the Oberbeck-Boussinesq approximation is observed just after the transient period.

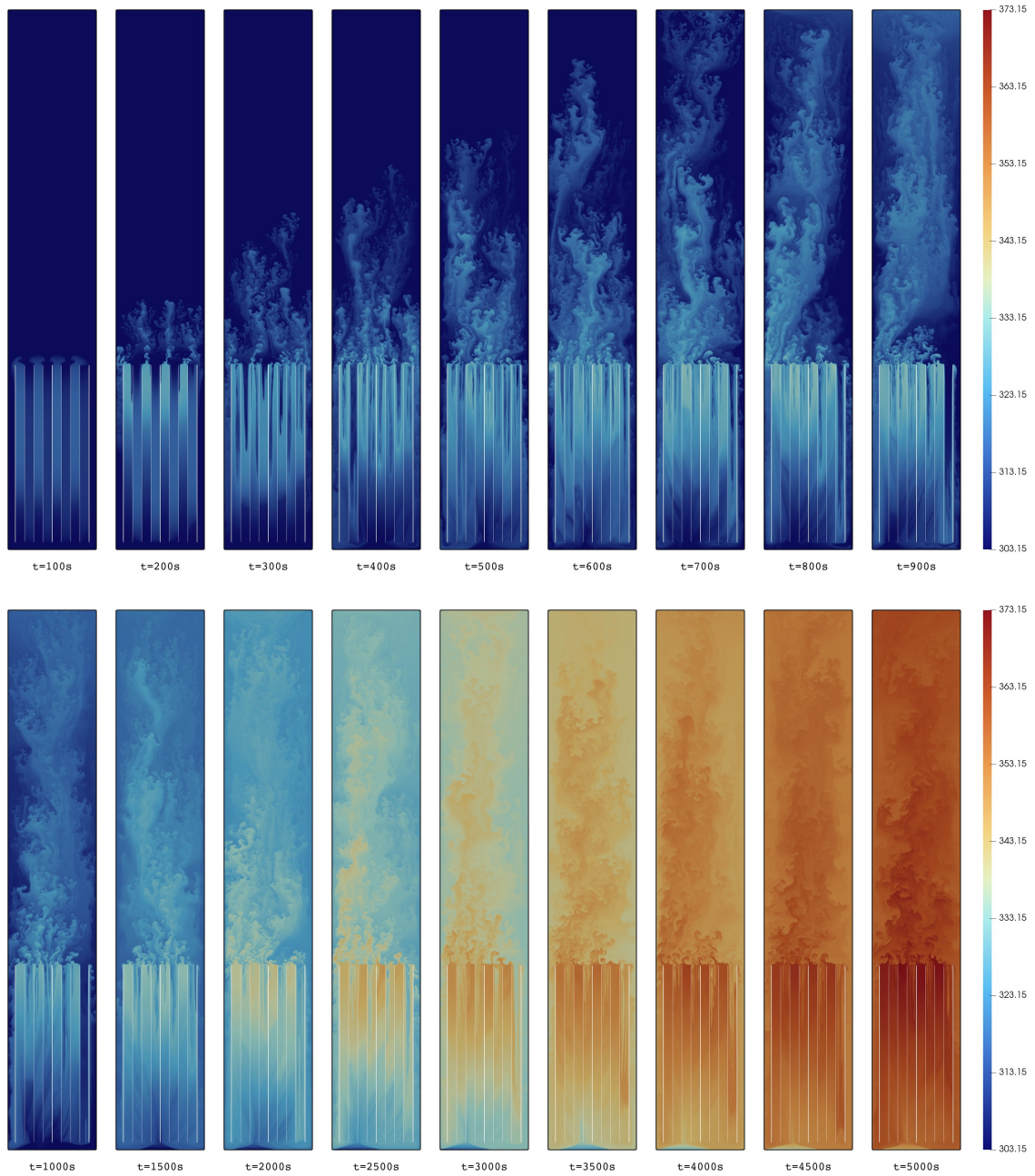


Figure B.2 presents the evolution of the temperature field throughout the simulation, under the Oberbeck-Boussinesq approximation.



**Figure B.2:** *Temperature field, in Kelvin, with the Oberbeck-Boussinesq approximation, for the configuration of Case 4.*

Figure B.3 depicts the evolution of the temperature field throughout the simulation, without invoking the Oberbeck-Boussinesq approximation.

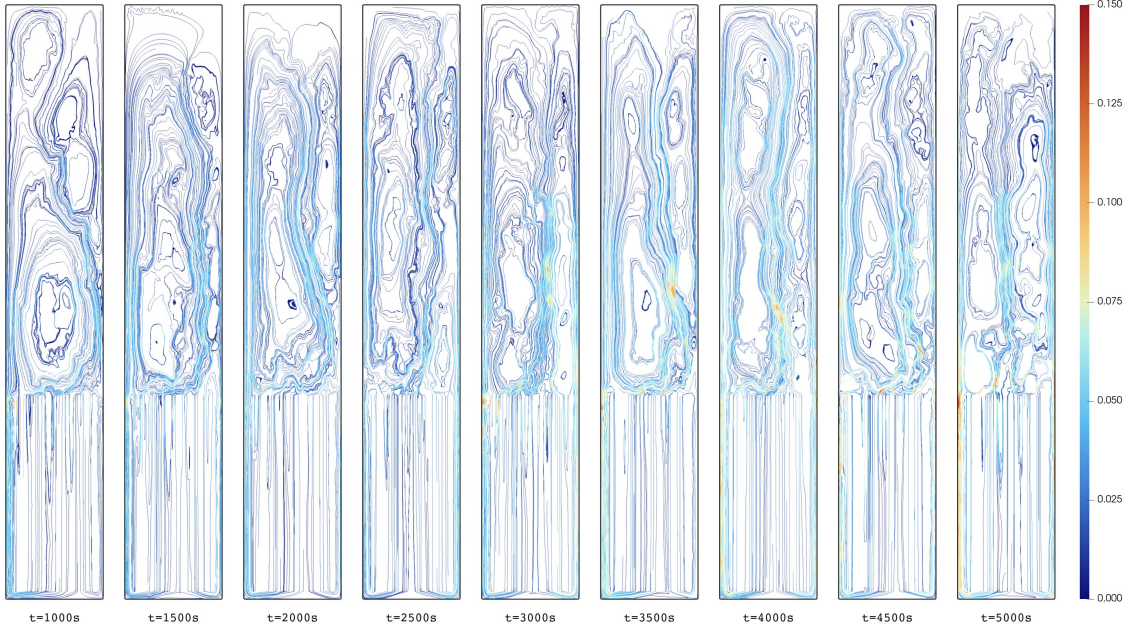


**Figure B.3:** *Temperature field, in Kelvin, without the Oberbeck-Boussinesq approximation, for the configuration of Case 4.*

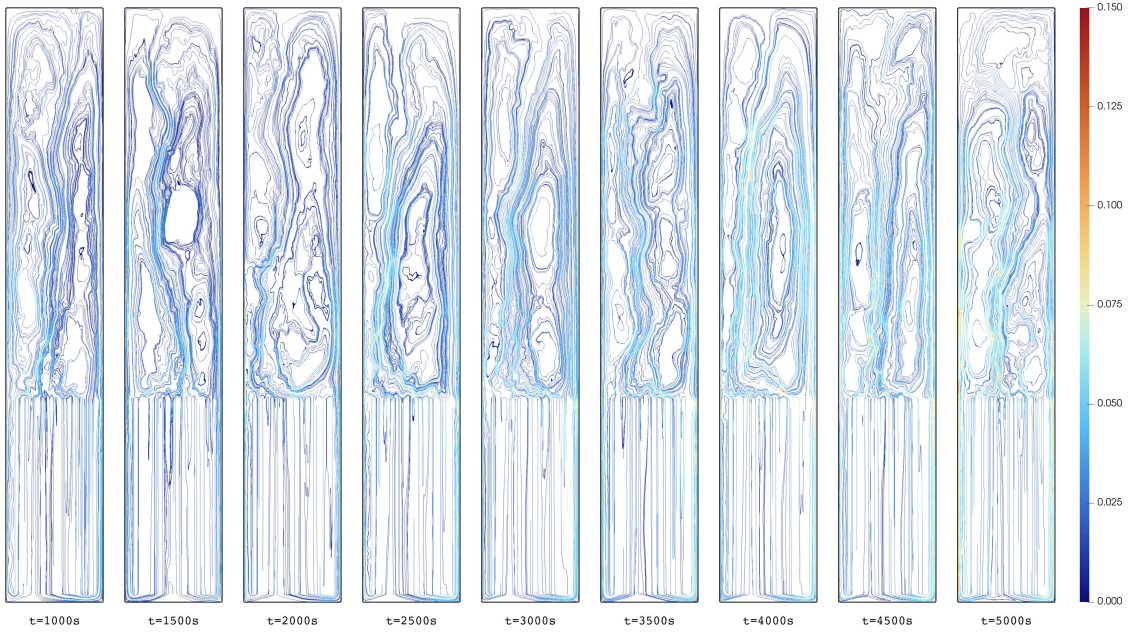
It is immediate that no major differences in the evolution of the temperature field are observed.



The evolution of the streamlines is shown in Figures B.4 and B.5, corresponding to the cases with and without the Oberbeck–Boussinesq approximation, respectively.



**Figure B.4:** *Streamlines, coloured by velocity magnitude, in m/s, with the Oberbeck–Boussinesq approximation, for the configuration of Case 4.*



**Figure B.5:** *Streamlines, coloured by velocity magnitude, in m/s, without the Oberbeck–Boussinesq approximation, for the configuration of Case 4.*

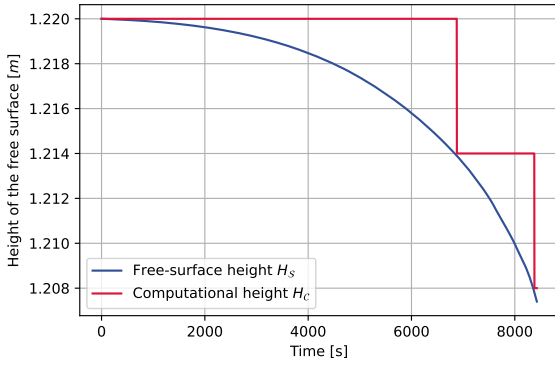
Here, we once again conclude that no major differences are observed. However, it is worth noting that in Figure B.4, which corresponds to the Oberbeck–Boussinesq approximation, the main convection cell is located on the left side of the domain, whereas in Figure B.5, without the approximation, it is located on the right side.

# Appendix C

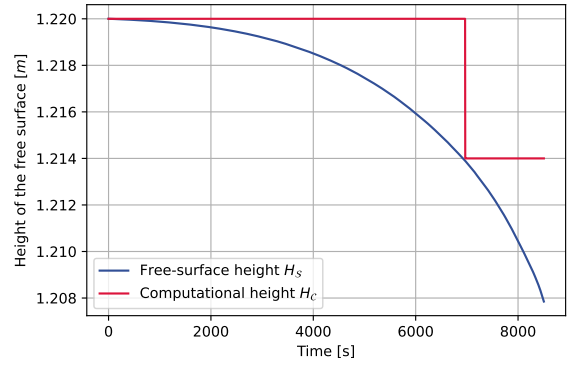
## Additional figures for the three-dimensional geometry

### C.1 Descent of the free-surface across the different cases

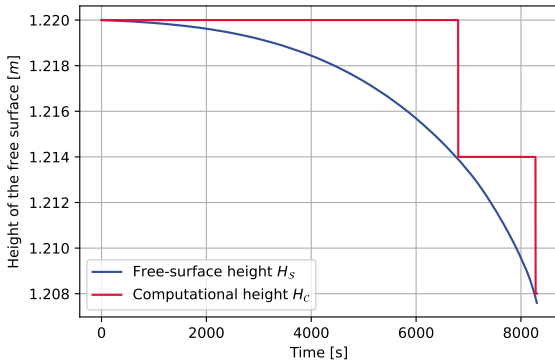
Figure C.1 shows the evolution of the computational domain height  $H_c$  and the free-surface height  $H_s$  for the different cases.



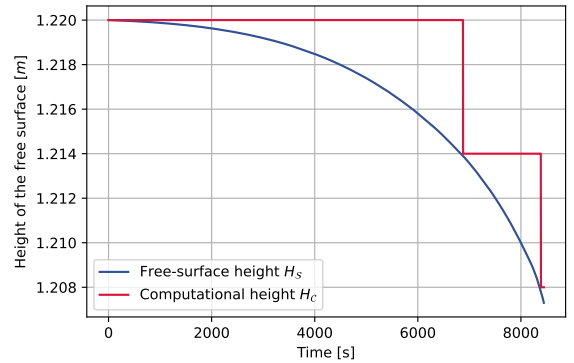
(a) Case 1



(b) Case 2



(c) Case 3



(d) Case 4

**Figure C.1:** Evolution of the computational domain height and the free-surface height for the different cases.

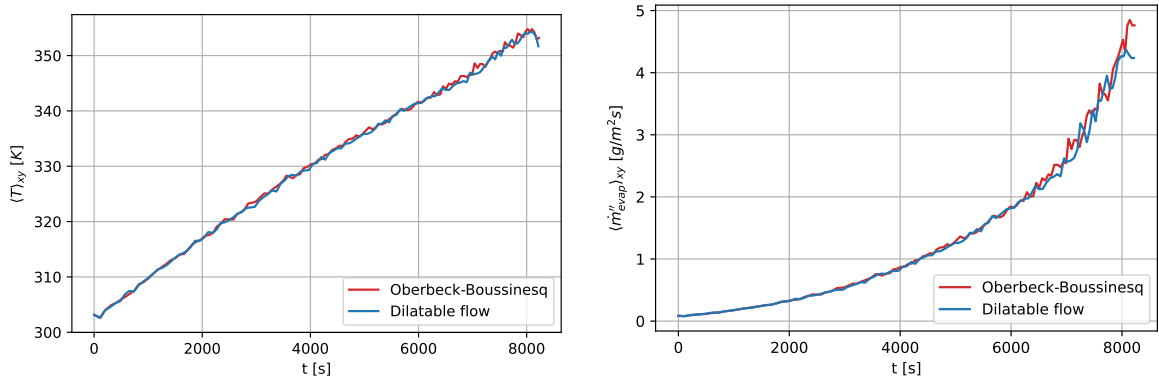
We observe that two remeshing procedures were carried out during the entire simulation

duration for each case, except for Case 2. This is consistent with the observations indicating that Case 2 exhibits the longest delay before the assemblies are exposed to the ambient air.

## C.2 Non-Oberbeck-Boussinesq flow

In this section, we briefly present and compare the flow behaviour and evaporation rates for the configuration of Case 4, with and without the Oberbeck-Boussinesq approximation.

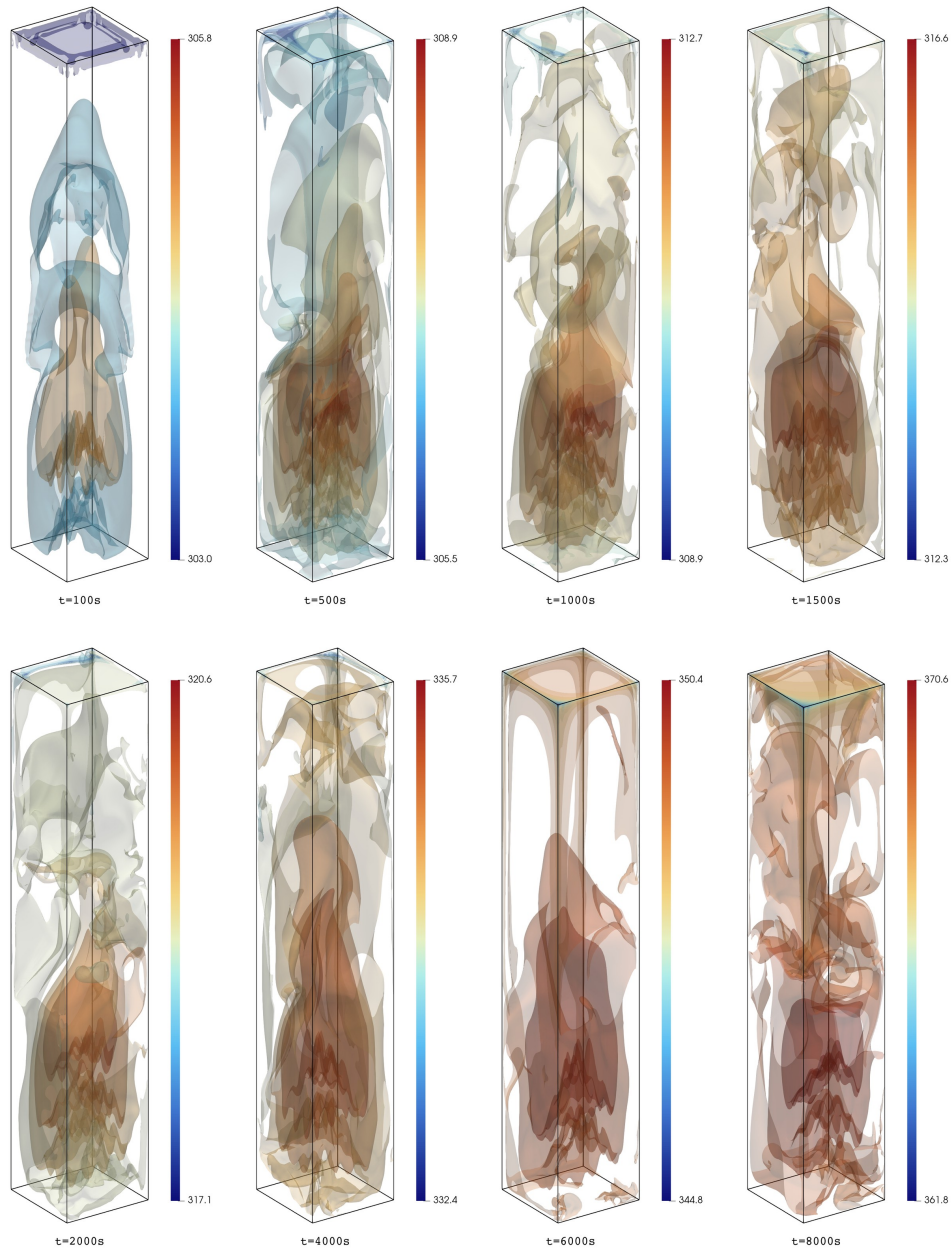
First, Figure C.2 shows the area-averaged free-surface temperature and the area-averaged evaporative mass flux for the case with and without the approximation.



**Figure C.2:** Comparison of area-averaged free-surface temperature and area-averaged evaporative mass flux over time for Case 4 with and without the Oberbeck-Boussinesq approximation.

Even though the area-averaged free-surface temperature tends to slightly decrease around the end of the simulation, the area-averaged evaporative mass flux increases, in particular in the case of the Oberbeck-Boussinesq approximation. This is likely caused by a hot spot in the case with the approximation, which can increase the area-averaged value, as the mass flux increases exponentially with temperature.

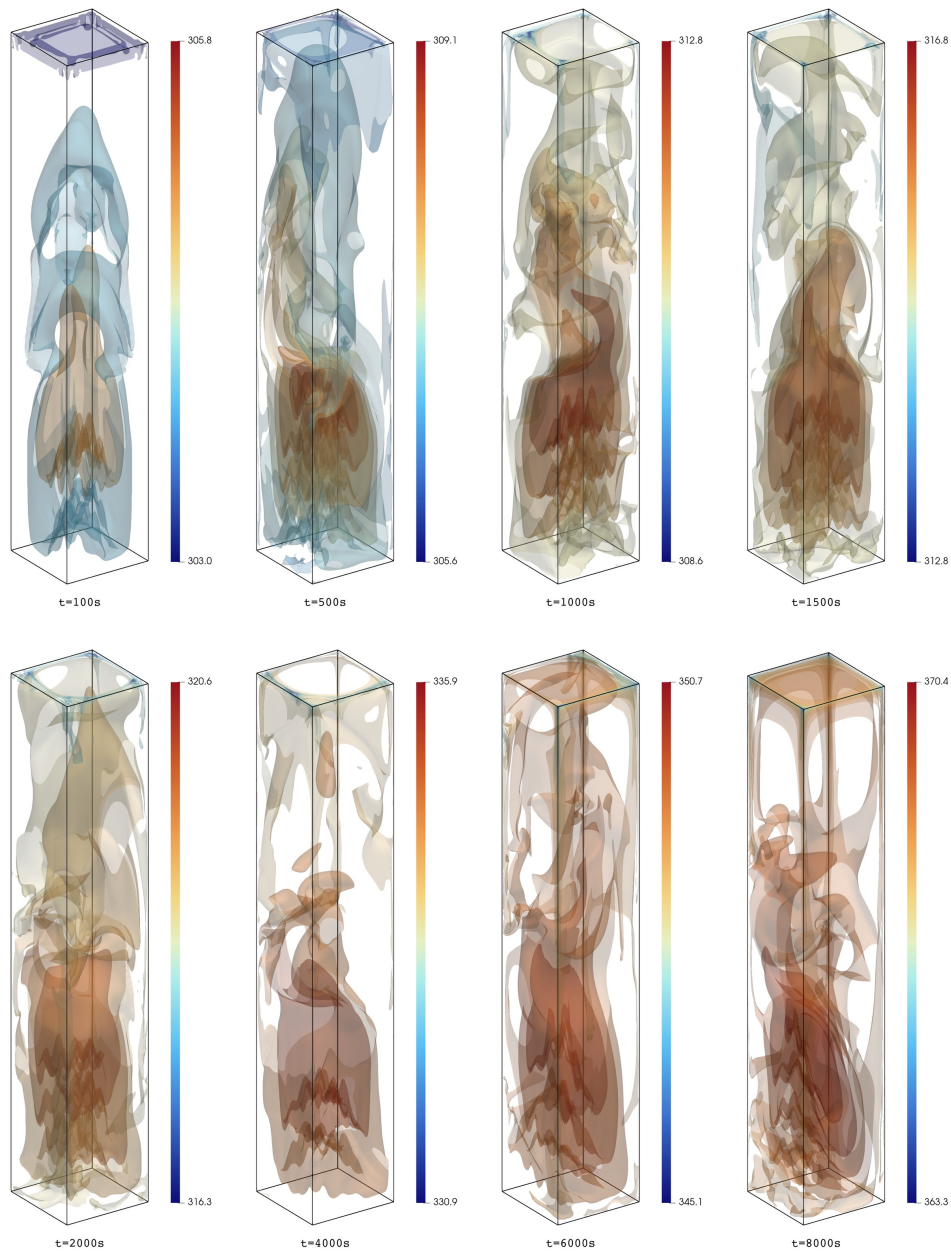
Figure C.3 provides the evolution of the temperature iso-surfaces throughout the simulation, with the Oberbeck-Boussinesq approximation.



**Figure C.3:** Temperature iso-surfaces, in Kelvin, with the Oberbeck-Boussinesq approximation, for the configuration of Case 4.



Figure C.4 shows the evolution of the temperature iso-surfaces throughout the simulation, without invoking the Oberbeck-Boussinesq approximation.

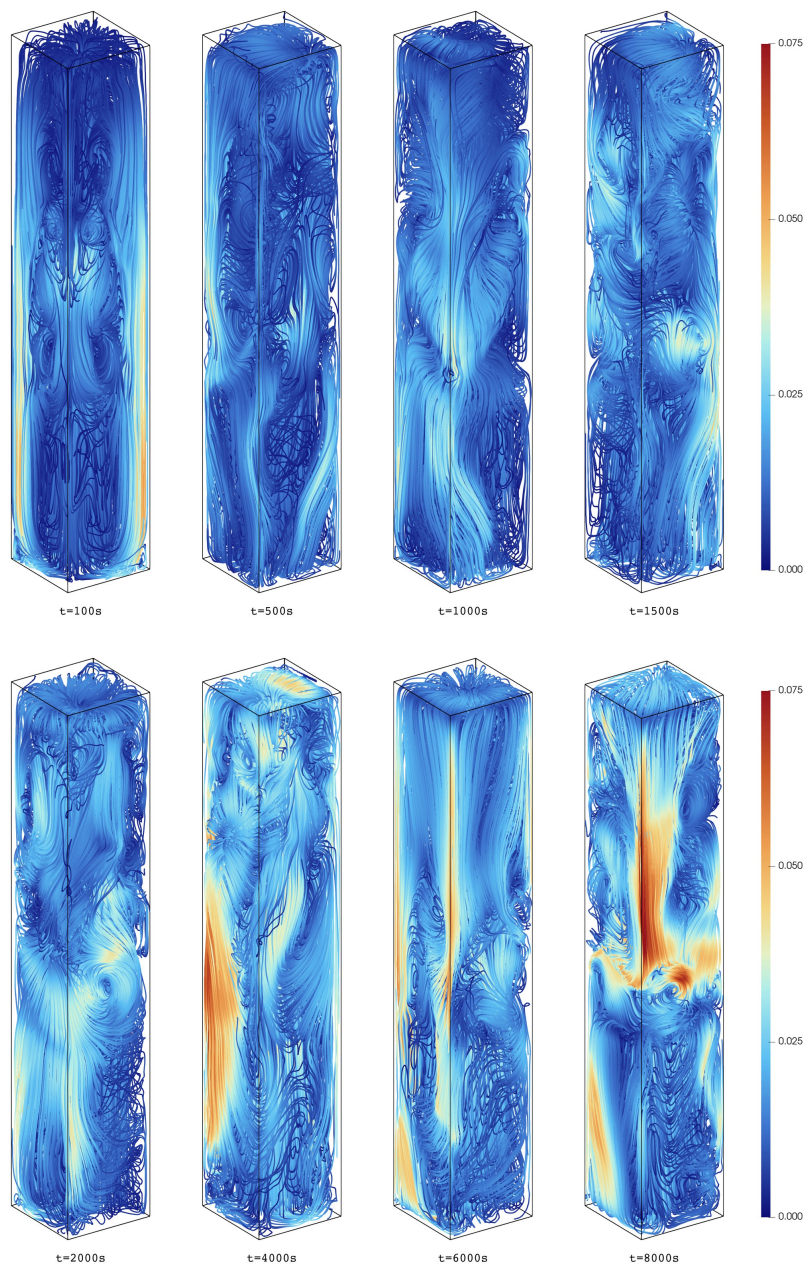


**Figure C.4:** *Temperature iso-surfaces, in Kelvin, without the Oberbeck-Boussinesq approximation, for the configuration of Case 4.*

Regarding the evolution of the temperature field inside the domain, no major differences are observed.

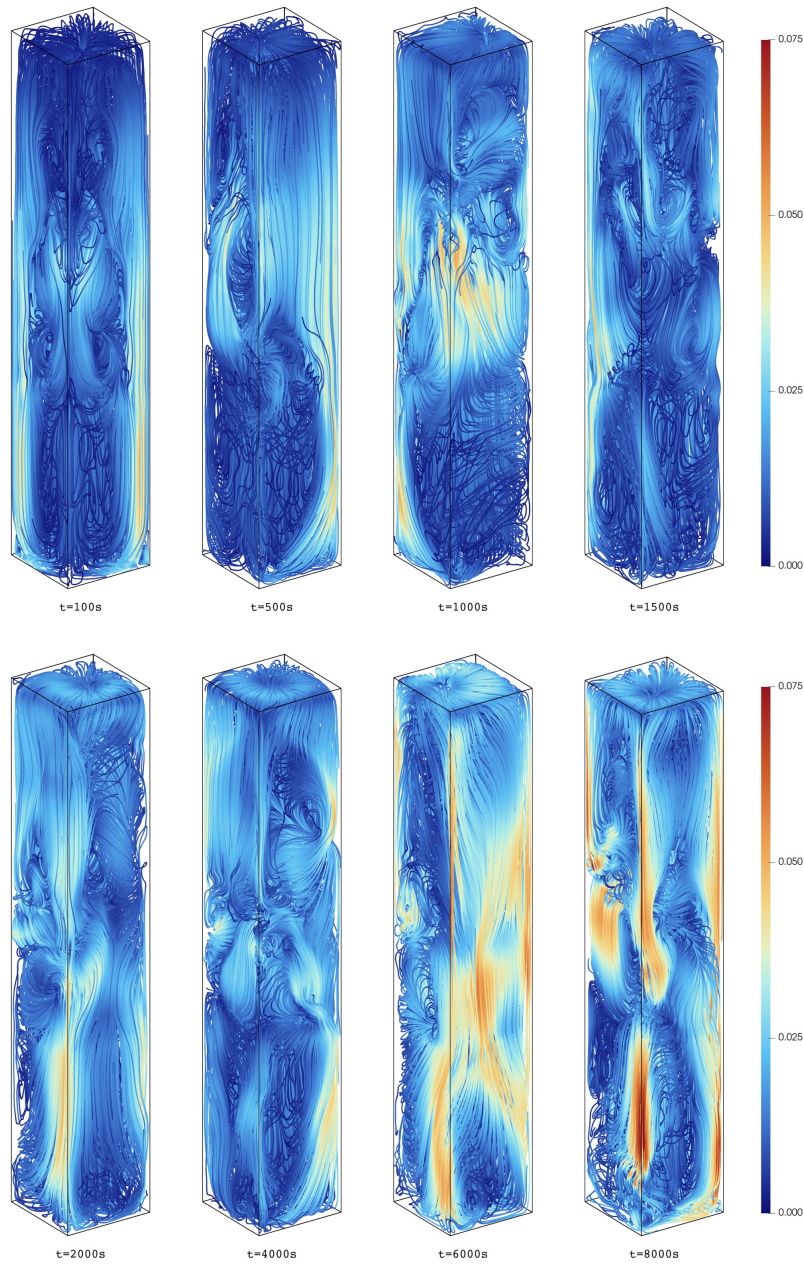


Figure C.5 depicts the evolution of the streamlines throughout the simulation, with the Oberbeck-Boussinesq approximation.



**Figure C.5:** *Streamlines, coloured by velocity magnitude, in m/s, with the Oberbeck-Boussinesq approximation, for the configuration of Case 4.*

Figure C.6 illustrates the evolution of the streamlines throughout the simulation, without invoking the Oberbeck-Boussinesq approximation.



**Figure C.6:** *Streamlines, coloured by velocity magnitude, in m/s, without the Oberbeck-Boussinesq approximation, for the configuration of Case 4.*

As can be seen, the convective behaviour of the flow exhibits the same overall characteristics in both the Oberbeck-Boussinesq and non-Oberbeck-Boussinesq cases.

# Bibliography

- [1] Le Monde, *En Belgique, le Parlement abroge la sortie du nucléaire, qui était inscrite dans une loi de 2003*, 15 May 2025, available on [https://www.lemonde.fr/international/article/2025/05/15/en-belgique-le-parlement-abroge-la-sortie-du-nucleaire-qui-etait-inscrite-dans-une-loi-de-2003\\_6606282\\_3210.html](https://www.lemonde.fr/international/article/2025/05/15/en-belgique-le-parlement-abroge-la-sortie-du-nucleaire-qui-etait-inscrite-dans-une-loi-de-2003_6606282_3210.html).
- [2] World Nuclear Association, *Nuclear Power in the World Today*, available on <https://world-nuclear.org/information-library/current-and-future-generation/nuclear-power-in-the-world-today>, accessed on 25 May 2025.
- [3] Gouvernement français, *Fiche thématique n°5: Le parc nucléaire existant*, available on [https://concertation-strategie-energie-climat.gouv.fr/fiche-thematique-ndeg5-le-parc-nucleaire-existant?utm\\_source=chatgpt.com](https://concertation-strategie-energie-climat.gouv.fr/fiche-thematique-ndeg5-le-parc-nucleaire-existant?utm_source=chatgpt.com), accessed on 25 April 2025.
- [4] World Nuclear Association, *How is used nuclear fuel managed?*, available on <https://world-nuclear.org/our-association/publications/technical-positions/how-is-used-nuclear-fuel-managed>, accessed on 25 May 2025.
- [5] J.-C. Nimal, *Sûreté et puissance résiduelle*, Autumn 2001, available on <https://www.cea.fr/multimedia/Documents/publications/clefs-cea/archives/fr/08-puissance.pdf>.
- [6] ANCCLI, *Livre Blanc VII de l'ANCCLI: Entreposage du combustible nucléaire: les piscines et leurs enjeux*, October 2021, available on <https://www.anccli.org/livres-blancs-3/>.
- [7] P. Coppolani, N. Hassenboelher, J. Joseph, J.-F. Petetrot, J.-P. Py, and J.-S. Zampa, *La chaudière des réacteurs à eau sous pression*, EDP Sciences, Les Ulis, 2004.
- [8] Orano, *La Hague*, available on <https://www.orano.group/en/nuclear-expertise/orano-s-sites-around-the-world/recycling-spent-fuel/la-hague/unique-expertise>, accessed on 19 May 2025.
- [9] D. Lohse and O. Shishkina, "Ultimate Rayleigh-Bénard turbulence", *Review of Modern Physics*, vol. 96, July-September 2024, 1-60.

- [10] W. Hay, *Evaporation-driven turbulent convection in water pools heated from below* (Thesis), UCLouvain, 2021.
- [11] N. Aksan, B. Smith, D. Araneo, B. Bousbia-Salah, A. L. Costa, F. D'Auria, A. Del Nevo, B. Neykov, N. Muellner, M. Marques, A. K. Nayak, D. Saha, and P. K. Vijayan, *Natural circulation in water cooled nuclear power plants*, IAEA, Vienna, November 2005.
- [12] M. A. Kohler and L. H. Parmele, "Generalized estimates of free-water evaporation", *Water Resources Research*, vol. 3, December 1967, 997-1005.
- [13] F. I. Morton, "Operational estimates of lake evaporation", *Journal of Hydrology*, vol. 66, October 1983, 77-100.
- [14] J. L. Duesterhaus, J. M. Ham, C. E. Owensby, and J. T. Murphy, "Water Balance of a Stock-Watering Pond in the Flint Hills of Kansas", *Rangeland Ecology & Management*, vol. 61, May 2008, 329-338.
- [15] ASNR, *Projet DENOPI, installations MIDI et ASPIC*, available on <https://www.irsn.fr/recherche/projet-denopi-installations-midi-aspic>, accessed on 15 May 2025.
- [16] V. Hamtiaux, *Numerical simulation of natural convection in mixed porous/pure-fluid domains* (Thesis), UCLouvain, 2024.
- [17] J. Carlier and M. V. Papalexandris, "Turbulent natural convection in an air–water system with evaporation across the free surface", *International Journal of Multi-phase Flow*, vol. 177, July 2024, 1-18.
- [18] G. Rubbers, *Numerical study of the cooling stability of spent nuclear fuels in storage and deactivation pools - Analysis of the natural convection in the pool* (Master Thesis), UCLouvain, 2023.
- [19] M. Magain, *Numerical study of the cooling stability of spent nuclear fuels in storage and deactivation pools* (Master Thesis), UCLouvain, 2024.
- [20] EDF, *code\_saturne 8.1 Theory Guide*, Chatou, 2023.
- [21] EDF, *code\_saturne version 7.0 practical user's guide*, Chatou, 2021.
- [22] G. Lebon, D. Jou, and J. Casas-Vázquez, *Understanding Non-equilibrium Thermodynamics*, Springer, Berlin, 2008.
- [23] M. V. Papalexandris, *Gas Dynamics*, Presses universitaires de Louvain, Louvain-la-Neuve, 2002.
- [24] R. B. Bird, W. E. Stewart, and E. N. Lightfoot, *Transport Phenomena*, 2nd ed., John Wiley & Sons, New York, NY, 2002.
- [25] EDF, *code\_saturne User's Forum*, available on <https://www.code-saturne.org/forum/viewtopic.php?p=14256&hilit=idilat#p14256>, accessed on 2 May 2025.
- [26] T. L. Bergman, A. S. Lavine, F. P. Incropera, and D. P. Dewitt, *Fundamentals of Heat and Mass Transfer*, 7th ed., John Wiley & Sons, Hoboken, NJ, 2011.



- [27] Y. A. Cengel, *Heat Transfer: A Practical Approach*, 2nd ed., McGraw-Hill, New York, NY, 2002.
- [28] D. C. Wilcox, *Turbulence Modeling for CFD*, 3rd ed., DCW Industries, La Cañada, CA, 2006.
- [29] J. G. Leishman, *Introduction to Aerospace Flight Vehicles*, self-published, 2025.
- [30] R. Manceau, *Codes de calcul industriels pour la simulation des écoulements turbulents, Turbulence* (Lecture notes), ENSMA, 2024, available on <https://inria.hal.science/hal03207435v8/file/CodesDeCalculIndustriels.pdf>.
- [31] J.-F. Wald, *Lois de paroi adaptatives pour un modèle de fermeture du second ordre dans un contexte industriel* (Thesis), Université de Pau et des Pays de l'Adour, 2016.
- [32] EDF, *code\_saturne User's Forum*, available on <https://www.code-saturne.org/forum/viewtopic.php?t=2753&hilit=URANS>, accessed on 1 May 2025.
- [33] G. Winckelmans and Y. Bartosiewicz, *Incompressible turbulent flows: Reynolds-averaged Navier-Stokes equations (RANS)*, *Fluid Mechanics II* (Lecture notes), UCLouvain, 2023.
- [34] B. Launder and D. B. Spalding, "The Numerical Computation of Turbulent Flow Computer Methods", *Computer Methods in Applied Mechanics and Engineering*, vol. 3, March 1974, 269-289.
- [35] D. C. Wilcox, "Formulation of the  $k - \omega$  Turbulence Model Revisited", *AIAA Journal*, vol. 46, November 2008, 2823-2838.
- [36] F. R. Menter, "Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications", *AIAA Journal*, vol. 32, August 1994, 1598-1605.
- [37] EDF, *code\_saturne Online Documentation*, available on [https://www.code-saturne.org/documentation/7.0/doxygen/src/group\\_\\_turbulence.html](https://www.code-saturne.org/documentation/7.0/doxygen/src/group__turbulence.html), accessed on 15 May 2025.
- [38] L. Prandtl, "Zur Turbulenten Strömung in Rohren und längs Platten", *Ergebnisse der Aerodynamischen Versuchsanstalt zu Göttingen*, vol. 4, 1930, 632-648.
- [39] T. V. Karman, "Mechanische Ähnlichkeit und Turbulenz", *Nachr. Ges. Wiss. Math-Phys Klasse Göttingen*, vol. 5, 1930, 58-76.
- [40] DNS Channel Data (2015), available on <https://turbulence.odn.utexas.edu>, accessed on 15 April 2025.
- [41] J. Bredberg, *On the Wall Boundary Condition for Turbulence Models*, *Department of Thermo and Fluid Dynamics* (Internal report), Chalmers University of Technology, 2000.
- [42] B. Castaing, G. Gunaratne, F. Heslot, L. Kadanoff, A. Libchaber, S. Thomae, X.-Z. Wu, S. Zaleski, and G. Zanetti, "Scaling of hard thermal turbulence in Rayleigh Bénard convection", *Journal of Fluid Mechanics*, vol. 204, July 1989, 1-30.

- [43] A. Bejan, *Convection Heat Transfer*, 4th ed., John Wiley & Sons, Hoboken, NJ, 2013.
- [44] B.-J. Ko, M.-H. Kim, and B.-J. Chung, "An experimental study on the transition criteria of open channel natural convection flows", *Journal of Mechanical Science and Technology*, vol. 26, April 2012, 1227-1234.
- [45] T. Fuchs, *New Technical Concepts of Spent Fuel Pool-Cooling and Implementation* (Conference paper), IAEA, Vienna, June 2016.
- [46] J.-C. Zerbib, G. Quétel, and A. Guillemette, *Les piscines de La Hague*, December 2011, available on <https://www.global-chance.org/IMG/pdf/JCZ-PiscinesLaHague-081211.pdf>.
- [47] K. M. Watson, "Thermodynamics of the Liquid State", *Industrial & Engineering Chemistry*, vol. 35, April 1943, 398-406.
- [48] W. Wagner and A. Pruss, "The IAPWS Formulation 1995 for the Thermodynamic Properties of Ordinary Water Substance for General and Scientific Use", *Journal of Physical and Chemical Reference Data*, vol. 31, June 2002, 387-535.
- [49] R. J. Goldstein, E. M. Sparrow, and D. C. Jones, "Natural Convection Mass Transfer Adjacent to Horizontal Plates", *International Journal of Heat and Mass Transfer*, vol. 16, May 1973, 1025-1035.
- [50] T. R. Marrero and E. A. Mason, "Gaseous Diffusion Coefficients", *Journal of Physical and Chemical Reference Data*, vol. 1, January 1972, 3-118.
- [51] J. H. Lienhard V and J. H. Lienhard IV, *A Heat Transfer Textbook*, 6th ed., Phlogiston Press, Cambridge, MA, 2024.
- [52] S.M. Bower and J.R. Saylor, "A study of the Sherwood-Rayleigh relation for water undergoing natural convection-driven evaporation", *International Journal of Heat and Mass Transfer*, vol. 52, June 2009, 3055-3063.
- [53] J. R. Lloyd and W. R. Moran, "Natural Convection Adjacent to Horizontal Surface of Various Planforms", *Journal of Heat Transfer*, vol. 96, November 1974, 443-447.
- [54] E.R.G. Eckert, H. Sakamoto, and T.W. Simon, "The heat/mass transfer analogy factor,  $Nu/Sh$ , for boundary layers on turbine blade profiles", *International Journal of Heat and Mass Transfer*, vol. 44, March 2001, 1223-1233.
- [55] I. H. Bell, J. Wronski, S. Quoilin, and J. V. Lemort, *Pure and Pseudo-pure Fluid Thermophysical Property Evaluation and the Open-Source Thermophysical Property Library CoolProp*, Industrial & Engineering Chemistry Research, vol. 53, January 2014, 2498-2508.
- [56] C. R. Wilke, "A Viscosity Equation for Gas Mixtures", *Journal of Chemical Physics*, vol. 18, April 1950, 517-519.
- [57] E. A. Mason and S. C. Saxena, "Approximate Formula for the Thermal Conductivity of Gas Mixtures", *Physics of Fluids*, vol. 1, September 1958, 361-369.

- [58] N. E. Todreas and M. S. Kazimi, *Nuclear Systems I - Thermal Hydraulic Fundamentals*, Taylor & Francis, Abingdon-on-Thames, 1993.
- [59] S.-K. Cheng and N. E. Todreas, "Hydrodynamic models and correlations for bare and wire-wrapped hexagonal rod bundles - Bundle friction factors, subchannel friction factors and mixing parameters", *Nuclear Engineering and Design*, vol. 92, April 1986, 227-251.
- [60] J. R. Lamarsh and A. J. Baratta, *Introduction to nuclear engineering*, 3rd ed., Prentice Hall, Upper Saddle River, NJ, 2001.
- [61] J. C. Neuber, "Evaluation of axial and horizontal burnup profiles", *Implementation of burnup credit in spent fuel management systems*, IAEA, Vienna, July 2000, 183-211.
- [62] S. Turek, L. Rivkind, J. Hron, and R. Glowinski, "Numerical analysis of a new time-stepping  $\theta$ -scheme for incompressible flow simulations", *Journal of Scientific Computing*, vol. 28, March 2006, 533-547.
- [63] H. K. Versteeg and W. Malalasekera, *An Introduction to Computational Fluid Dynamics*, 2nd ed., Pearson Education Limited, Harlow, 2007.
- [64] E. Isaacson and H. B. Keller, *Analysis of Numerical Methods*, Dover Publications, New-York, NY, 1966.
- [65] EDF, *Best Practice Guidelines 4 - Numerical parameters*, available on [https://www.code-saturne.org/cms/sites/default/files/file\\_attach/BPG/BPG-4-Numerical-Parameters.pdf](https://www.code-saturne.org/cms/sites/default/files/file_attach/BPG/BPG-4-Numerical-Parameters.pdf), accessed on 5 March 2025.
- [66] EDF, *code\_saturne Online Documentation*, available on [https://www.code-saturne.org/documentation/7.1/doxygen/src/cs\\_ug\\_parallel.html](https://www.code-saturne.org/documentation/7.1/doxygen/src/cs_ug_parallel.html), accessed on 7 May 2025.



UNIVERSITÉ CATHOLIQUE DE LOUVAIN

École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | [www.uclouvain.be/epl](http://www.uclouvain.be/epl)