

Modeling of Thermohydraulics in a Lead-Cooled Reactor System

A CFD Study Using OpenFOAM

August Møller & Emma Wesström



A Thesis for the Degree of Master of Science

University Supervisor: Christer Fureby

Company Supervisor: Fady Ishaq

Examiner: Jens Klingmann

June 5, 2026

ISRN LUTMDN/TMHP-26/5682-SE

ISSN 0282-1990

Department of Energy Sciences

Faculty of Engineering, LTH

Lund University

Acknowledgements

This work was conducted at Lund University in collaboration with Blykalla during the spring of 2026. The study aimed to develop and evaluate a simplified model of Blykalla's SEALER reactor concept using computational fluid dynamics. We would like to express our sincere gratitude to our supervisors Christer Fureby, Fady Ishaq, and Michel Chapuis for their continuous guidance and valuable insights throughout this project. We also gratefully acknowledge Blykalla for the opportunity to work on the SEALER reactor concept.

Finally, we would like to thank our families, friends, and partners for their continuous support and encouragement.

Lund, May 2026

August Møller and Emma Wesström

Abstract

This study investigates the thermal-hydraulics of a simplified model of Blykalla's Swedish Advanced Lead-cooled Reactor (SEALER) using computational fluid dynamics (CFD) in OpenFOAM. The objective is to develop and verify a system-level model for forced and natural convection. The reactor cooling circuit is represented using porous media and volumetric source terms. Simulations are conducted using Reynolds-averaged Navier-Stokes (RANS), employing both eddy-viscosity and Reynolds stress turbulence models. The model is initially validated for single-phase steady-state forced convection by comparison with reference data. Strong agreement is obtained with respect to mass flow rate, core temperature difference, and pressure drop, indicating a consistent implementation of the system components. A noticeable deviation of 18.6 % is observed in the steam generator pressure drop, which suggests sensitivity to the porous media implementation. The turbulence model has a negligible impact on system-level performance during forced convection. For natural convection, the results show a strong dependence on reactor geometry, where increased system height enhances circulation and thermal distribution. The choice of turbulence model has a more pronounced effect under natural convection conditions, where the Reynolds stress model predicts a lower mass flow rate and exhibits reduced numerical stability compared to the eddy-viscosity models. The influence of thermophysical modeling is assessed by comparing a temperature-dependent property model with the Boussinesq approximation. The Boussinesq model predicts a higher mass flow rate and a lower temperature rise across the core, consistent with its simplified treatment of density variations. Multiphase simulations using a volume of fluid (VOF) approach are conducted to assess the capability to predict buoyancy-driven flow in the presence of large density differences. The results indicate that the presence of a gas phase has a negligible impact on system performance and that the OpenFOAM treatment of buoyancy is sufficient for system-level analysis.

Contributions

This work was conducted collaboratively, where both authors contributed to development, evaluation, and analysis of the implemented OpenFOAM model. Although this work is divided into multiple chapters, the responsibilities were distributed so that both authors were involved in each part of the study.

This means that each section of this work was critically evaluated and agreed upon by both authors. This includes the theory, methodology, results, discussion and conclusions reported in this paper.

Use of AI

AI tools were used to a limited extent during the development of this thesis. ChatGPT was utilized as a supporting tool for language refinement and formatting consistency. It was also used to assist with debugging and improvement of the developed OpenFOAM environment and MATLAB code.

In all tasks supported by AI, the generated outputs were critically reviewed and verified by both authors to ensure technical correctness, academic integrity, and originality.

Nomenclature

Abbreviations

AR	Asymptotic Range
CFD	Computational Fluid Dynamics
CFL	Courant-Friedrichs-Lewy
DNS	Direct Numerical Simulation
GCI	Grid Convergence Index
LBE	Lead-Bismuth Eutectic
LES	Large Eddy Simulation
LFR	Lead-cooled Fast Reactor
LIC	Line Integral Convolution
LRR	Launder-Reece-Rodi
OECD	Organization for Economic Co-operation and Development
PISO	Pressure-Implicit with Splitting of Operators
RANS	Reynolds-Averaged Navier-Stokes
RSM	Reynolds Stress Model
RVACS	Reactor Vessel Auxiliary Cooling System
SEALER	Swedish Advanced Lead-cooled Reactor
SG	Steam Generator
SIMPLE	Semi-Implicit Method for Pressure-Linked Equations
SMR	Small Modular Reactor
SST	Shear Stress Transport
TVD	Total Variation Diminishing
VOF	Volume of Fluid

Latin symbols

A	Area	$[\text{m}^2]$
C	Inertial resistance coefficient	$[\text{m}^{-1}]$
C_{CFL}	Courant number safety factor	$[-]$
C_μ	Model constant in k - ε turbulence model	$[-]$
c_p	Specific heat capacity	$[\text{J kg}^{-1} \text{K}^{-1}]$
D	Viscous resistance coefficient	$[\text{m}^{-2}]$
d	Darcy coefficient	$[\text{m}^{-2}]$
e	Specific internal energy	$[\text{J kg}^{-1}]$
F	Mass flux	$[\text{kg m}^{-2} \text{s}^{-1}]$
F_{buoyancy}	Buoyancy force	$[\text{N m}^{-3}]$

F_σ	Volumetric surface tension force	$[\text{N m}^{-3}]$
f	Forchheimer coefficient	$[\text{m}^{-1}]$
$\mathbf{f}_p$	Momentum sink term	$[\text{N m}^{-3}]$
g	Gravitational acceleration	$[\text{m s}^{-2}]$
$\mathbf{g}$	Gravitational acceleration vector	$[\text{m s}^{-2}]$
h	Heat transfer coefficient	$[\text{W m}^{-2} \text{K}^{-1}]$
h	Specific enthalpy	$[\text{J kg}^{-1}]$
$\mathbf{h}$	Position vector (gravitational height)	$[\text{m}]$
k	Thermal conductivity	$[\text{W m}^{-1} \text{K}^{-1}]$
k	Turbulent kinetic energy	$[\text{m}^2 \text{s}^{-2}]$
M	Molar mass	$[\text{kg mol}^{-1}]$
$\dot{m}$	Mass flow rate	$[\text{kg s}^{-1}]$
N	Number of cells	$[-]$
$\mathbf{n}$	Outward unit normal vector	$[-]$
$\mathbf{P}_i$	Inertial (quadratic) resistance tensor	$[\text{kg m}^{-4}]$
$\mathbf{P}_v$	Viscous (linear) resistance tensor	$[\text{kg m}^{-3} \text{s}^{-1}]$
P/L	Pressure gradient	$[\text{Pa m}^{-1}]$
Pr	Prandtl number	$[-]$
p	Pressure	$[\text{Pa}]$
$p_{\min}$	Minimum pressure	$[\text{Pa}]$
p_{obs}	Observed order of accuracy	$[-]$
p_{rgh}	Modified pressure	$[\text{Pa}]$
Q	Heat transfer rate	$[\text{W}]$
Q_{calc}	Calculated heat transfer rate	$[\text{W}]$
Q_{filt}	Filtered heat transfer rate	$[\text{W}]$
R_{ij}	Reynolds stress tensor component	$[\text{m}^2 \text{s}^{-2}]$
r	Grid refinement ratio	$[-]$
S_E	Energy source term	$[\text{W m}^{-3}]$
$\mathbf{S}_M$	Momentum source term	$[\text{N m}^{-3}]$
S_ϕ	Source term for scalar transport	$[-]$
S_p	Implicit source term	$[-]$
S_u	Explicit source term	$[\text{W}]$
T	Temperature	$[\text{K}]$
T_0	Reference temperature	$[\text{K}]$
T_∞	Ambient air temperature	$[\text{K}]$
$\mathbf{u}$	Velocity vector	$[\text{m s}^{-1}]$

V	Volume	$[\text{m}^3]$
$\mathbf{v}_s$	Superficial velocity	$[\text{m s}^{-1}]$
y^+	Dimensionless wall distance	$[-]$
Δp	Pressure drop	$[\text{Pa}]$
ΔT	Temperature difference	$[\text{K}]$
Δt	Time step	$[\text{s}]$
Δx	Representative cell size	$[\text{m}]$

Greek symbols

α	Phase fraction (VOF)	$[-]$
α	Thermal diffusivity	$[\text{m}^2 \text{s}^{-1}]$
α_t	Turbulent thermal diffusivity	$[\text{m}^2 \text{s}^{-1}]$
β	Thermal expansion coefficient	$[\text{K}^{-1}]$
β	Relaxation/filter coefficient	$[-]$
ε	Turbulent dissipation rate	$[\text{m}^2 \text{s}^{-3}]$
Γ_ϕ	Generalized diffusion coefficient for ϕ	$[\text{kg m}^{-1} \text{s}^{-1}]$
μ	Dynamic viscosity	$[\text{Pa s}]$
μ_t	Turbulent dynamic viscosity	$[\text{Pa s}]$
ν	Kinematic viscosity	$[\text{m}^2 \text{s}^{-1}]$
ν_t	Turbulent kinematic viscosity	$[\text{m}^2 \text{s}^{-1}]$
ω	Specific dissipation rate	$[\text{s}^{-1}]$
ϕ	Transported scalar quantity	$[-]$
ϕ'	Fluctuation of scalar quantity	$[-]$
ρ	Density	$[\text{kg m}^{-3}]$
ρ_0	Reference density	$[\text{kg m}^{-3}]$
σ	Surface tension coefficient	$[\text{N m}^{-1}]$
τ	Viscous stress tensor	$[\text{Pa}]$

Contents

1	Introduction	1
1.1	Lead-cooled SMR concept	2
1.2	Scope and objectives	3
1.3	Delimitations	4
2	Background	5
2.1	Thermophysical properties of liquid metals	5
2.1.1	Specific features of liquid metals	5
2.1.2	Thermophysical properties of lead	5
2.2	Fundamentals of natural convection	6
2.3	The Boussinesq density approximation	7
2.4	CFD modeling framework	7
2.4.1	Governing equations	7
2.4.2	Discretization of convective terms	8
2.4.3	Discretization of diffusion terms and surface-normal gradients . . .	10
2.4.4	Interpolation and gradient schemes	11
2.4.5	Temporal discretization	11
2.4.6	Pressure-velocity coupling algorithms	12
2.4.7	Turbulence modeling theory	13
2.4.8	Multiphase modeling	15
2.5	OpenFOAM	17
2.5.1	Environment and structure	18
2.5.2	Solvers	19
2.5.3	Treatment of buoyancy forces	20
3	General methodology	20
3.1	Simplified reactor model	21
3.2	Post-processing and data evaluation	22
3.3	Simulation strategy	22
3.4	Boussinesq model	23
3.5	Temperature-dependent property model	23
3.6	Physics implementation	25
3.7	Conversion of parameters	26
3.8	Boundary conditions	28
3.9	Numerical setup	31

4	Single-phase steady-state forced convection	32
4.1	Forced convection setup	32
4.2	Mesh generation	32
4.3	Grid convergence analysis	33
4.4	Turbulence model comparison	35
4.5	Forced convection performance	37
4.6	Model validation and flow behavior	38
5	Single-phase steady-state natural convection	39
5.1	Natural convection setup	40
5.2	Mesh generation	40
5.3	Reactor height study	41
5.4	Near-wall resolution analysis	42
5.5	Grid convergence analysis	43
5.6	Turbulence model comparison	44
5.7	Boussinesq model performance	46
5.8	Governing physics and flow behavior	47
6	Multiphase steady-state natural convection	49
6.1	Multiphase natural convection setup	50
6.2	VOF model	50
6.3	Mesh generation	51
6.4	Boundary conditions	51
6.5	Numerical setup	52
6.6	Multiphase flow performance	52
6.7	Impact of the gas phase and flow behavior	53
7	Conclusion	54
8	Future work	56
	Appendix	62
A	MATLAB code for polynomial approximation	62
B	MATLAB code for mesh sensitivity study	63
C	Velocity fields for different turbulence models under forced convection	66
D	Velocity fields for different turbulence models under natural convection	68
E	Velocity fields for the Boussinesq model	72
F	Velocity fields from VOF simulations under natural convection	74

1 Introduction

Electricity is estimated to account for approximately 50 % of the total energy consumption by 2050 [1]. Thus, the development of low-emission energy sources is essential for achieving net-zero emissions. The transition towards a sustainable energy system has led to an increased interest in nuclear power as a stable and large-scale energy source with low greenhouse gas emissions. The role of nuclear energy in the future energy mix is further emphasized by the International Energy Agency, which identifies nuclear power as a key contributor to the transition towards a net-zero energy system. In addition, the majority of electricity is estimated to be sourced from solar and wind power by 2050, with nuclear energy expected to provide stable baseload generation, supporting grid reliability.

Although conventional nuclear power plants continue to operate worldwide, small modular reactors (SMRs) represent a significant shift in the nuclear industry [2]. SMRs provide compact, prefabricated designs and could enable reduced capital costs, which have historically posed challenges for the nuclear industry. In addition to electricity generation, SMRs enable applications such as district heating and deployment in decentralized energy systems, including microgrids and off-grid locations.

Regardless of the type of reactor, the design must be "inherently safe", meaning that the system can passively mitigate the consequences of failure scenarios, including loss of power or cooling [3]. Hence, a central objective of SMRs is the enhancement of passive safety systems. Through natural circulation driven by buoyancy forces, the system becomes less reliant on active safety components during failure scenarios, resulting in a simpler reactor design. Passive safety also reduces the risk of radioactive release into the environment [4].

Among the advanced reactor technologies being developed, the lead-cooled fast reactor (LFR) is recognized as a Generation IV reactor concept [5]. The LFR uses liquid lead or lead-bismuth eutectic (LBE) as the coolant, which provides several advantages. Lead is chemically inert, resulting in limited reactivity with air and water [6]. It is also a low-pressure coolant due to its high boiling point, allowing operation at near-atmospheric pressure. Additionally, lead provides inherent radiation shielding. These properties yield improved passive safety in case of failure scenarios.

Despite the advantages, LFRs present several challenges [6]. The high melting point of lead requires careful thermal management to prevent solidification. Furthermore, high operating temperatures and flow velocities promote corrosion of structural materials. The opacity of lead also complicates inspection and monitoring of internal components. In addition, the high density of lead results in increased mechanical loads, imposing strict requirements on structural integrity and seismic design.

1.1 Lead-cooled SMR concept

The Swedish Advanced Lead-cooled Reactor (SEALER), developed by the Swedish nuclear company Blykalla, is a SMR based on LFR technology, illustrated in Figure 1 [7]. It is designed for commercial electricity production and industrial heat applications. With an output of 55 MW (electric) and a fuel residence time of 25 years, the SEALER occupies a footprint of approximately 5 m × 5 m.



Figure 1: Conceptual illustration of the SEALER [7].

Liquid lead is used as a coolant and, due to its high boiling point of approximately 1700 °C, enables low-pressure operation [7]. Due to the relatively low power output and the thermophysical properties of liquid lead, decay heat can be passively removed through natural circulation. Moreover, the lead coolant is circulated through pipes made of aluminum-alloyed steel designed for corrosion resistance in liquid-lead environments.

The reactor is configured as a pool-type system, which means that the coolant circulates within a single vessel, as shown in Figure 2. The heat generated in the core is carried by the lead to the steam generator located in the upper part of the vessel. There, the thermal energy is transferred to the secondary steam cycle for electricity production. During standard operation, the coolant circulation is driven by primary pumps, providing forced convection through the core and the heat exchangers.

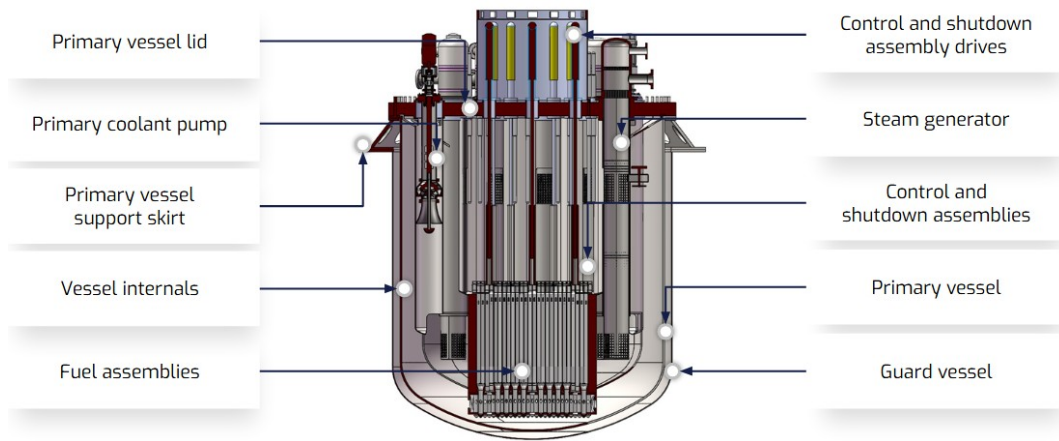


Figure 2: Cross-sectional illustration of the SEALER reactor vessel [8].

In the event of pump shutdown or loss of external power, the system is designed to utilize natural circulation [7]. This circulation is driven by density differences caused by temperature differences in the coolant. In addition to the steam generator, the system is equipped with a reactor vessel auxiliary cooling system (RVACS). This is a passive air-cooling system that removes heat from the system by transferring it to the air surrounding the vessel. This enables long-term decay heat removal without active cooling components.

Inside the cooling circuit, an argon cover gas is located above the liquid lead [9]. The compressibility of the gas enables thermal expansion and pressure changes in the coolant without imposing increased mechanical stresses on the vessel. In addition, the argon is used due to its chemical inertness. This helps prevent corrosion of the reactor vessel since the gas replaces air and water vapor in the free space above the lead.

1.2 Scope and objectives

Thermal-hydraulic analysis is essential for evaluating reactor performance and safety. Computational fluid dynamics (CFD) is a widely used tool for such analyses, enabling detailed investigation of flow behavior and heat transfer without the need for complex experimental setups.

The aim of this study is to develop a simplified CFD-based thermal-hydraulic model of the SEALER system in OpenFOAM based on a reference model in STAR CCM+ and to assess its predictive capability under both steady-state and transient operating conditions. The OpenFOAM setup is developed to provide a cost-effective alternative to other CFD software while allowing customization of the flow physics.

The work seeks to establish a single-phase steady-state model capable of reproducing key thermal-hydraulic parameters, including mass flow rates, pressure drops, and temperature distributions, through comparison with the reference STAR-CCM+ simulations.

Furthermore, the study aims to investigate the behavior and limitations of the implemented system during natural circulation by performing single-phase simulations. The single-phase simulations are then used as a reference to simulate multiphase flow where the argon gas cover is present. Due to limitations in multiphase-flow buoyancy treatment in STAR-CCM+, where a global reference density is used, buoyancy effects may not be fully captured when large density differences exist between phases. Therefore, the aim of the multiphase simulation is to evaluate how buoyancy treatment in OpenFOAM influences the system performance. In all simulations of this study, steady-state conditions are obtained either directly or through transient simulations run until a quasi-steady-state is reached.

1.3 Delimitations

To limit the scope of the study, several simplifications and limitations are introduced in the modeling approach. The reactor geometry is simplified by exploiting geometric symmetry, which allows a quarter of the system to be simulated. Further simplifications in the geometry include replacing complex internal structures such as fuel assemblies and steam generator tube bundles with porous media to account for their respective pressure drops. This is deemed sufficient, as the purpose of the study is to investigate system-level performance.

The flow is modeled using Reynolds-averaged Navier-Stokes (RANS) turbulence models. Higher accuracy approaches such as large eddy simulation (LES) and direct numerical simulation (DNS) are omitted due to their computational requirements.

The gas phase is neglected in single-phase simulations, as its influence on the bulk flow and heat transfer is assumed to be negligible under the considered conditions. Multiphase simulations are performed in selected cases to capture phase interactions during natural circulation, while sensitivity studies of the setup are limited to the single-phase simulations. Furthermore, exterior air flow is not explicitly modeled and is instead represented through boundary conditions at the system boundaries.

Neutronic feedback effects are not included in the model. Reactor power is instead integrated as a fixed volumetric heat source within the core region. Thus, thermal-hydraulic feedback on reactor power is beyond the scope of this study. Similarly, the steam generator cooling is simplified by representing it through a volumetric heat sink in the reactor geometry. Hence, the secondary heat carrier is not modeled in this setup.

The verification of the model is performed through comparison with reference CFD simulations rather than experimental measurements, since experimental data for the investigated reactor configuration are not available.

2 Background

2.1 Thermophysical properties of liquid metals

In this work, lead is used as the working fluid. Lead is considered a promising coolant for fast nuclear reactors due to its favorable thermophysical properties [9].

2.1.1 Specific features of liquid metals

Liquid metals, compared to other fluids such as air and water, have significantly higher thermal conductivity k and lower specific heat capacity c_p [9]. The kinematic viscosity ν of heavy liquid metals is often smaller than for air or water, leading to a larger Reynolds number for the same velocity. Combining these properties leads to the dimensionless Prandtl number,

$$Pr = \frac{c_p \mu}{k} = \frac{\nu}{\alpha} \quad (1)$$

where α is the thermal diffusivity and μ is the dynamic viscosity. The Prandtl number compares the transport of momentum, represented by the kinematic viscosity, to the transport of thermal energy, represented by the thermal diffusivity [9]. Pr does not depend on geometry or velocity, which means it is based only on the physical properties of the fluid. It is therefore an essential non-dimensional parameter in convective heat transfer problems.

Due to the very specific physical properties of liquid metals, the Prandtl number is more than two orders of magnitude lower compared to water and air, which are typically of the order $O(1)$ [10]. Liquid metals instead fall in the range $0.003 \leq Pr \leq 0.06$ [9].

A low Prandtl number implies that the thermal diffusivity dominates the momentum diffusivity ($\alpha \gg \nu$) [10]. As a result, the thermal boundary layer is significantly thicker than the viscous boundary layer. This implies that thermal energy diffuses much faster than momentum in liquid metals.

2.1.2 Thermophysical properties of lead

The thermophysical properties of liquid lead are strongly temperature-dependent. Empirical correlations for the main properties of liquid lead at atmospheric pressure are reported

in the Organization for Economic Co-operation and Development (OECD) Handbook on Lead-Bismuth Eutectic Alloy and Lead Properties [9]. The main correlations of liquid lead are summarized in Table 1.

Table 1: Correlations for thermophysical properties of liquid lead.

Property	Correlation	Temp. range	Est. error
ρ	$11441 - 1.2795 \cdot T$	601 K to 1900 K	1 %
c_p	$176.2 - 4.923 \cdot 10^{-2} \cdot T + 1.544 \cdot 10^{-5} \cdot T^2 - 1.524 \cdot 10^6 \cdot T^{-2}$	601 K to 1300 K	5 %
μ	$4.55 \cdot 10^{-4} \exp\left(\frac{1069}{T}\right)$	601 K to 1500 K	5 %
k	$9.2 + 0.011 \cdot T$	601 K to 1400 K	(15 %)

where T is the temperature in K. These correlations are valid in the temperature ranges specified in the OECD handbook and are summarized in Table 1. These ranges are representative of the operating conditions in liquid lead-cooled reactor systems. Unlike the other fluid properties, the thermal conductivity of lead has a varying temperature dependence in the available literature. Thus, a compromise between the available data is adopted in the handbook to represent technically pure lead under normal pressure. To emphasize this compromise on temperature dependency, brackets are placed around the estimated uncertainty in Table 1 [9].

2.2 Fundamentals of natural convection

Natural convection, also known as free convection, is a fundamental heat transfer mechanism in which fluid motion is induced by buoyancy forces resulting from temperature-dependent density variations [11]. In contrast to forced convection, where the flow is driven by an external source such as a fan or pump, natural convection is driven only by thermal gradients and gravitational effects, making it particularly important for passive cooling systems.

Natural convection arises when temperature differences within a fluid lead to density variations, which cause the warmer, less dense fluid to rise while the cooler, denser fluid moves downward to replace it [11]. This is referred to as a natural convection current, and the associated heat transfer is termed natural convection heat transfer. In a gravitational field, a fluid element with lower density surrounded by a denser fluid experiences an upward force called the buoyancy force, and its magnitude is equal to the weight of the fluid displaced by the body.

$$F_{\text{buoyancy}} = \rho_{\text{fluid}} g V_{\text{body}} \quad (2)$$

where ρ_{fluid} is the average density of the surrounding fluid, g is the gravitational acceleration, and V_{body} is the volume of the portion of the body immersed in the fluid [11].

The buoyancy force is opposed by viscous forces arising from shear stresses within the fluid and at solid boundaries [11]. As fluids move relative to one another, velocity gradients develop, which create shear stresses within the fluid that act to dissipate momentum and slow the flow. Under steady conditions, the resulting flow field is established when the driving buoyancy forces are balanced by the viscous resistance, which determines both the flow structure and the associated heat transfer characteristics.

In turbulent flow regimes, additional resistance arises from the formation of vortex structures, which enhance momentum dissipation by transferring kinetic energy to smaller scales where it is converted into heat, thus increasing the effective flow resistance [12].

2.3 The Boussinesq density approximation

An alternative approach to represent density variations and buoyancy effects in a fluid is the Boussinesq density approximation [13]. This approximation assumes that the density varies linearly with temperature, as described by Equation 3, where ρ denotes the density, ρ_0 the reference density, β the thermal expansion coefficient, T the local temperature, and T_0 the reference temperature.

$$\rho = \rho_0 (1 - \beta(T - T_0)) \quad (3)$$

This approximation introduces a reference density and temperature, and assumes that density variations are only significant in the buoyancy term of the momentum equation. This simplifies the system of equations introduced in Section 2.4.1 and is therefore more computationally efficient [13].

2.4 CFD modeling framework

This section introduces the CFD-related theoretical framework used to model the thermohydraulics of the reactor. The governing equations are first introduced. This is followed by a description of numerical discretization, including convective and diffusive terms, interpolation and gradient schemes, and temporal discretization. The pressure-velocity coupling strategy is then introduced. Finally, turbulence and multiphase modeling approaches are described.

2.4.1 Governing equations

The equations needed to model the thermohydraulics of the lead-cooled reactor are introduced in this section. First, Equation 4 represents the conservation of mass in which $\mathbf{u}$ represents the velocity vector [12]. This ensures conservation of mass within the control volume. During

steady-state conditions, the time derivative is zero as the solution is independent of time.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (4)$$

Following this, the Navier-Stokes equations are shown in Equations 5 and 6. These equations state that the rate of change of momentum (the temporal term and the convective term) is equal to the forces acting on the lead.

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \mathbf{S}_M \quad (5)$$

$$\boldsymbol{\tau} = \mu \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T - \frac{2}{3} (\nabla \cdot \mathbf{u}) \mathbf{I} \right). \quad (6)$$

These forces are represented on the right-hand side of the equations, where the first term represents pressure forces, the second term represents viscous forces (the diffusion term) and the third term represents the body forces [12]. One of the relevant body forces in natural circulation is buoyancy, which is expressed in Equation 7 for the temperature-dependent density. The other body force is pressure loss due to the porous media which is expressed in Equation 8 [14]. Here, the force is dependent on the viscous and inertial pressure loss coefficients D and C .

$$\mathbf{S}_M^g = \rho(T) \mathbf{g} \quad (7)$$

$$\mathbf{S}_M^{por} = -\mu(T) \mathbf{D} \mathbf{u} - \frac{1}{2} \rho(T) \mathbf{C} |\mathbf{u}| \mathbf{u} \quad (8)$$

In addition, Equation 9 applies the first law of thermodynamics [12]. This means that the rate of change of internal energy is equal to the work of pressure forces, heat conduction (diffusion), viscous dissipation, and volumetric heat sources which are introduced in the same order in Equation 9. Here, e represents specific internal energy.

$$\frac{\partial(\rho e)}{\partial t} + \nabla \cdot (\rho e \mathbf{u}) = -p \nabla \cdot \mathbf{u} + \nabla \cdot (k \nabla T) + \boldsymbol{\tau} : \nabla \mathbf{u} + S_E \quad (9)$$

2.4.2 Discretization of convective terms

In problems where fluid flow is present, transport is governed by the combined effects of convection and diffusion processes [12]. The steady convection-diffusion equation for a scalar quantity ϕ can be written as

$$\nabla \cdot (\rho \mathbf{u} \phi) = \nabla \cdot (\Gamma_\phi \nabla \phi) + S_\phi \quad (10)$$

where $\rho \mathbf{u}$ represents the mass flux, Γ_ϕ the diffusion coefficient, and S_ϕ a source term [12]. By integrating over a control volume and applying Gauss' divergence theorem in Equation

11 where $\mathbf{a}$ is an arbitrary vector quantity and $\mathbf{n}$ is the surface-normal vector, the transport equation is transformed into a balance of fluxes across the control volume which yields Equation 12.

$$\int_{CV} \nabla \cdot \mathbf{a} dV = \int_A \mathbf{n} \cdot \mathbf{a} dA \quad (11)$$

$$\int_A \mathbf{n} \cdot (\rho \phi \mathbf{u}) dA = \int_A \mathbf{n} \cdot (\Gamma_\phi \nabla \phi) dA + \int_{CV} S_\phi dV \quad (12)$$

The left-hand side gives the net convective flux, while the right-hand side contains the net diffusive flux and the generation or destruction of the property ϕ within the control volume [12]. This shows that the convective flux depends on the values of the transported variable ϕ at the faces of the control volume. Since these values are not directly known, they must be approximated using interpolation schemes. Unlike diffusion, which acts in all directions along gradients, convection transports information primarily in the flow direction. This introduces stability constraints and makes the choice of discretization schemes critical.

To approximate the face values of ϕ , different discretization schemes can be applied. These schemes differ in their accuracy and stability characteristics.

Central differencing (linear) scheme The central differencing scheme approximates the face value as the average of the neighboring cell values [12]. For a one-dimensional control volume, this can be written as

$$\phi_e = \frac{\phi_P + \phi_E}{2} \quad (13)$$

$$\phi_w = \frac{\phi_W + \phi_P}{2} \quad (14)$$

where ϕ_P denotes the value at the current control volume, and ϕ_E and ϕ_W denote the values at the neighboring control volumes to the east and west, respectively. The subscripts e and w refer to the values at the east and west cell faces.

This scheme is second-order accurate, unbounded and does not introduce artificial diffusion in the same way as the upwind scheme [12]. Unbounded means that the interpolated face value can attain values outside the range of its neighboring cells. However, the scheme does not account for the direction of the flow as both neighboring nodes influence the face value equally. In convection-dominated flows, this may lead to non-physical oscillations and unstable solutions. In OpenFOAM, the *Gauss linear* scheme represents this discretization [15].

Upwind scheme To address this limitation, the first-order upwind differencing scheme accounts for the flow direction by taking the value of ϕ at a control volume face from the upstream cell [12]. For a one-dimensional control volume, this can be expressed as

$$\phi_e = \begin{cases} \phi_P, & \text{if } F_e > 0 \\ \phi_E, & \text{if } F_e < 0 \end{cases} \quad (15)$$

$$\phi_w = \begin{cases} \phi_W, & \text{if } F_w > 0 \\ \phi_P, & \text{if } F_w < 0 \end{cases} \quad (16)$$

where F is the mass flow rate per unit area. This ensures that the convected quantity is transported in the correct physical direction. The upwind scheme is therefore stable but introduces numerical diffusion, which can smear steep gradients. In OpenFOAM, the upwind scheme is implemented with the *Gauss upwind* scheme [16].

High-resolution schemes The limitations of higher-order schemes, such as the occurrence of artificial oscillations and non-physical undershoots and overshoots, have led to the development of high-resolution schemes [12]. These schemes are often formulated within the framework of total variation diminishing (TVD) methods and are designed to combine the accuracy of higher-order discretizations with the stability and boundedness of upwind schemes. This is achieved by introducing limiter functions that reduce the higher-order contribution in regions with strong gradients. An example is the limited linear scheme, which blends upwind and central differencing to improve accuracy while maintaining stability. In OpenFOAM, this corresponds to the *limitedLinear* scheme used for the discretization of convective terms [17].

Since continuity is never exactly satisfied during convergence of a steady-state solution, there will always be some small continuity error during convergence of a steady-state solution. Therefore, the convective term can employ the *bounded* formulation in OpenFOAM [17]. This adds a correction to the transport equation to compensate continuity imbalances. This acts as a convergence aid to stay bounded as the simulation approaches a stable solution.

2.4.3 Discretization of diffusion terms and surface-normal gradients

In the governing equations, the diffusion of momentum and energy is represented by the Laplacian operator $\nabla \cdot (\Gamma_\phi \nabla \phi)$ where ϕ is the transported quantity, as seen in Equation 10 [18]. In accordance with Gauss's theorem, highlighted in Equation 11, the diffusion term is dependent on the surface-normal gradient. In OpenFOAM, Laplacian terms are commonly discretized using the second-order unbounded *Gauss linear* scheme, which corresponds to

central differencing on orthogonal meshes employing Gauss’s theorem. A fully orthogonal mesh means that, for any two neighboring cell centers, a line can be drawn between them that is parallel to the normal vector of the cell face.

However, with complex geometries, non-orthogonality is unavoidable. This means that discretizations of surface-normal gradient terms and diffusion terms can be modified using limiters and correction procedures, which make the terms less sensitive to non-orthogonality [18].

In OpenFOAM, the *corrected* scheme can be added to the discretization of these terms, which maintains second-order accuracy when non-orthogonality is present [18]. It does this by adding a non-orthogonal contribution to the original orthogonal discretization to compensate for geometric misalignment. Conversely, an *uncorrected* scheme means that no non-orthogonal correction is performed.

Moreover, OpenFOAM provides *limited* schemes for diffusion and surface-normal gradient terms, which improves stability while still accounting for non-orthogonality [18]. By defining a scalar value between 0 and 1 for the limiter, only a fraction of the non-orthogonal correction is applied in each iteration.

2.4.4 Interpolation and gradient schemes

In addition to the discretization of convective and diffusive terms, interpolation and gradient schemes are required to fully discretize the governing equations. Interpolation schemes are used to evaluate field variables at cell faces based on cell-centered values. Standard interpolation schemes include the unbounded second-order *linear* scheme in OpenFOAM [19].

Gradient schemes determine the discretization of the gradient operator $\nabla\phi$, which is required for terms such as the pressure gradient in the momentum equation [18]. In OpenFOAM, gradients are typically computed using the *Gauss linear* scheme, meaning that a Gauss-based formulation combined with linear interpolation is used [20].

To ensure numerical stability in regions with steep physical gradients, limiter-based schemes can be applied [21]. For example, the *cellMDLimited* scheme introduces a multi-directional limiter that constrains the reconstructed gradients to maintain boundedness while preserving second-order accuracy in smooth regions.

2.4.5 Temporal discretization

For transient simulations, the governing equations retain the time derivative which must be discretized in order to advance the solution in time. Three common time-discretization approaches are the explicit, implicit, and Crank-Nicolson schemes [12].

In an explicit scheme, the solution at the new time step is computed solely from the previous time step [22]. In turn, the method is computationally inexpensive for each iteration. However, the scheme is conditionally stable and must satisfy stability constraints associated with convective and diffusive transport. The convective stability requirement is expressed through the Courant-Friedrichs-Lewy (CFL) condition as seen in Equation 17 for a one-dimensional case. Here, u denotes the characteristic velocity, Δt is the time step, Δx is the spatial step, and C_{CFL} is the safety factor. The expression limits the fluid particle from having a larger spatial step than the grid.

$$\Delta t \leq C_{\text{CFL}} \frac{\Delta x}{u} \quad (17)$$

Additionally, diffusive processes such as heat conduction must satisfy Equation 18, which represents the limit for the one-dimensional case [12]. The condition means that the time step must be small enough for the diffusion to be resolved.

$$\Delta t \leq \frac{\rho c_p \Delta x^2}{2k} \quad (18)$$

In contrast, the implicit scheme evaluates the governing equations using unknown values in the new time step, which requires solving a system of equations. Thus, a system of equations is solved at each time step, which increases computational cost. Nevertheless, the implicit method is unconditionally stable, which is useful if large time steps are of interest [12]. In OpenFOAM, this temporal discretization is implemented using the *Euler* scheme [18].

Unlike the explicit and implicit schemes, the Crank-Nicolson scheme is of second-order accuracy [12]. It combines the contributions of the previous and new time steps. This scheme is unconditionally stable, but large time steps can cause numerical oscillations or loss of boundedness.

In OpenFOAM, transient solvers can adopt an adjustable time step by specifying the maximum allowable Courant number limit [23]. This limits the current time step to satisfy the Courant number condition.

2.4.6 Pressure-velocity coupling algorithms

To ensure consistency between the momentum and continuity equations, pressure-velocity coupling algorithms are required [12]. This is because the momentum equation depends on the pressure field and is in turn adjusted to enforce mass conservation.

One common algorithm is the Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) [12]. It is an iterative method in which the pressure field is guessed in each iteration. Following this, the discretized momentum equations are solved using the pressure field which yields the velocity field. With this, the pressure correction is calculated, which gives the corrected

pressure and velocities. The final step is to solve the remaining discretized transport equations and determine if convergence is reached by evaluating if continuity is satisfied. If this is not the case, the corrected pressure and velocity field is used as new guesses as the previous steps are repeated. There are multiple versions of this algorithm, one of which is called SIMPLEC. This particular model omits less significant terms in the correction equation.

Another algorithm is the Pressure Implicit with Splitting of Operators (PISO) [12]. It involves one predictor step in addition to two corrector steps and can be interpreted as an extension of SIMPLE with a further corrector step to enhance it. The second velocity and pressure correction occurs in the same time step, resulting in a more accurate solution without multiple outer iterations.

PIMPLE is a combination of the PISO and SIMPLE algorithms which incorporates outer iterations and inner pressure-velocity corrections [24]. This improves stability and convergence in transient flows with larger possible time steps compared to PISO.

2.4.7 Turbulence modeling theory

Laminar flow is completely predicted by the Navier-Stokes equations. However, most flows are turbulent and therefore require turbulence modeling [12]. The three primary turbulence modeling methods are: DNS, LES, and RANS. Due to the computational requirements of DNS and LES, RANS is the most commonly used turbulence modeling framework for industrial applications. The resulting averaged equations in compressible flow are seen in Equations 19 and 20, and the averaged transport equation is seen in Equation 21. In these equations, an overbar represents a time-averaged scalar or vector, and the tilde indicates a density-weighted or Favre-averaged variable, while an apostrophe denotes its fluctuation. The variable ϕ represents an arbitrary scalar or vector quantity, while ϕ' denotes its fluctuating component.

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_j} (\bar{\rho} \tilde{u}_j) = 0 \quad (19)$$

$$\frac{\partial (\bar{\rho} \tilde{u}_i)}{\partial t} + \frac{\partial}{\partial x_j} (\bar{\rho} \tilde{u}_j \tilde{u}_i) = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \left(\mu \frac{\partial \tilde{u}_i}{\partial x_j} \right) - \frac{\partial}{\partial x_j} (\overline{\rho u'_i u'_j}) + S_{M_i} \quad (20)$$

$$\frac{\partial (\bar{\rho} \tilde{\phi})}{\partial t} + \frac{\partial}{\partial x_j} (\bar{\rho} \tilde{u}_j \tilde{\phi}) = \frac{\partial}{\partial x_j} \left(\Gamma_\phi \frac{\partial \tilde{\phi}}{\partial x_j} \right) - \frac{\partial}{\partial x_j} (\overline{\rho u'_j \phi'}) + S_\phi \quad (21)$$

As can be seen in the equations above, additional terms appear with the fluctuating variables which are referred to as Reynolds stresses [12]. These stresses represent the mean effect of turbulence on the flow and result in 6 additional unknown variables. Boussinesq hypothesized that the Reynolds stresses are proportional to the mean deformation rates. This linear relation, in turn, depends on the turbulent dynamic viscosity or the eddy viscosity μ_t . Hence, a

turbulence closure model is needed to solve for the turbulent viscosity. Some of the most common approaches are k - ε , k - ω and k - ω Shear-Stress Transport (SST) closure models.

Firstly, the k - ε model expresses the turbulent viscosity as seen in Equation 22, where C_μ is a dimensionless constant, k is the turbulent kinetic energy, and ε is the rate of dissipation of turbulent kinetic energy [12]. This introduces two additional transport equations, one for the kinetic energy in Equation 23 and one for the dissipation rate in Equation 24. In these equations, there are five adjustable constants: C_μ , σ_k , σ_ε , $C_{1\varepsilon}$, and $C_{2\varepsilon}$ that are determined based on comprehensive data fitting for a range of turbulent flows. Additionally, the term S_{ij} represents the mean rate of strain tensor. The advantages of this model are its numerical stability and its ability to predict fully developed turbulent flow while being computationally inexpensive. Near walls, it has a weaker performance, in addition to having poor predictions of flow separation.

$$\mu_t = \bar{\rho} C_\mu \frac{k^2}{\varepsilon} \quad (22)$$

$$\frac{\partial(\bar{\rho}k)}{\partial t} + \frac{\partial(\bar{\rho}\tilde{u}_j k)}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\frac{\mu_t}{\sigma_k} \frac{\partial k}{\partial x_j} \right] + 2\mu_t S_{ij} S_{ij} - \bar{\rho}\varepsilon \quad (23)$$

$$\frac{\partial(\bar{\rho}\varepsilon)}{\partial t} + \frac{\partial(\bar{\rho}\tilde{u}_j \varepsilon)}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\frac{\mu_t}{\sigma_\varepsilon} \frac{\partial \varepsilon}{\partial x_j} \right] + C_{1\varepsilon} \frac{\varepsilon}{k} 2\mu_t S_{ij} S_{ij} - C_{2\varepsilon} \bar{\rho} \frac{\varepsilon^2}{k} \quad (24)$$

Similarly to the k - ε model, the k - ω model introduces an additional transport equation for k [12]. This closure model expresses the turbulent viscosity in accordance with Equation 25. This means that the second transport equation describes the specific dissipation rate ω . These transport equations are shown in Equations 26, 27, and 28. Similarly to the previous turbulence transport equations, these include the adapted model constants σ_k , σ_ω , γ_1 , β_1 , β^* . In addition they include the term P_k which represents the rate of production of turbulent kinetic energy.

$$\mu_t = \bar{\rho} \frac{k}{\omega} \quad (25)$$

$$\frac{\partial(\bar{\rho}k)}{\partial t} + \frac{\partial(\bar{\rho}\tilde{u}_j k)}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\left(\mu + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] + P_k - \beta^* \bar{\rho} k \omega \quad (26)$$

$$P_k = 2\mu_t S_{ij} S_{ij} - \frac{2}{3} \bar{\rho} k \frac{\partial \tilde{u}_j}{\partial x_j} \quad (27)$$

$$\frac{\partial(\bar{\rho}\omega)}{\partial t} + \frac{\partial(\bar{\rho}\tilde{u}_j \omega)}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\left(\mu + \frac{\mu_t}{\sigma_\omega} \right) \frac{\partial \omega}{\partial x_j} \right] + \gamma_1 \left(2\bar{\rho} S_{ij} S_{ij} - \frac{2}{3} \bar{\rho} \omega \frac{\partial \tilde{u}_j}{\partial x_j} \right) - \beta_1 \bar{\rho} \omega^2 \quad (28)$$

The main advantage of this model is its near-wall accuracy as it does not rely on wall functions [12]. As a result, it performs well in predicting boundary layer flows and heat

transfer across boundaries. Compared to the $k-\varepsilon$ model, it predicts adverse pressure gradients with greater precision. Inversely, its performance in free stream flow is worse than that of the $k-\varepsilon$ model.

The $k-\omega$ SST model combines the advantages of $k-\varepsilon$ and $k-\omega$ by transforming the $k-\varepsilon$ model into a $k-\omega$ model in the near-wall region [12]. In addition, it uses blending functions to transition between the two models. This allows for a more accurate prediction of the flow. One of the main problems with these three models is that they are incapable of capturing the turbulent energy production stemming from anisotropy of the normal stresses. Moreover, they cannot incorporate the effects on the turbulence of the extra strains and body forces.

A modeling framework that resolves the issue of anisotropy is the Reynolds stress equation model (RSM) [12]. A widely used closure model is the Launder, Reece and Rodi (LRR). This model does not have issues with anisotropy and body forces as it accounts for directional effects of the Reynolds stress field. This is achieved by solving the transport equations for each of the Reynolds stresses R_{ij} defined in Equation 29. The transport equation for the Reynold stresses is summarized in Equation 30. In this, the operator $\frac{D}{Dt}$ denotes the material derivative which represents the local temporal derivative plus the convective transport. The remaining terms in the equation, from left to right include the rate of production P_{ij} , the transport by diffusion D_{ij} , the rate of dissipation ε_{ij} , transport due to turbulent pressure-strain interaction Π_{ij} , and the transport caused by rotation Ω_{ij} . Due to the complexity of the RSM model, only the general transport equation is presented here. The full formulation of the transport equation for the Reynold stresses can be found in Versteeg and Malalaesekera [12].

$$R_{ij} = -\frac{\tau_{ij}}{\rho} = \overline{u'_i u'_j} \quad (29)$$

$$\frac{DR_{ij}}{Dt} = \frac{\partial R_{ij}}{\partial t} + \frac{\partial(\tilde{u}_k R_{ij})}{\partial x_k} = P_{ij} + D_{ij} - \varepsilon_{ij} + \Pi_{ij} + \Omega_{ij} \quad (30)$$

By solving the additional transport equations for the Reynolds stresses, the stresses are not based on the Boussinesq hypothesis [12]. This makes the model more suitable for flows involving strong anisotropy. Nevertheless, the increased fidelity comes at the cost of increased computational expense due to the six additional transport equations for the Reynolds stresses. Additionally, the model uses a dissipation rate equation similar to that used in the $k-\varepsilon$ model, which implies that the inaccuracies in the dissipation model are similar for both models.

2.4.8 Multiphase modeling

In order to extend the governing equations to two-phase flow, the volume of fluid (VOF) method can be introduced [25]. To describe the interface between the phases, a scalar phase fraction, α , is introduced, representing the volume fraction of phase 1. Consequently, the

volume fraction of phase 2 is given by $1-\alpha$. Therefore, the continuity equation for each of the two phases can be constructed accordingly in Equations 31 and 32.

$$\frac{\partial(\alpha\rho_1)}{\partial t} + \nabla \cdot (\alpha\rho_1\mathbf{u}_1) = 0 \quad (31)$$

$$\frac{\partial((1-\alpha)\rho_2)}{\partial t} + \nabla \cdot ((1-\alpha)\rho_2\mathbf{u}_2) = 0 \quad (32)$$

Furthermore, a momentum equation is formulated for each phase in Equations 33 and 34. The left-hand side of the equation represents the transient and convective transport of momentum. The right-hand side contains pressure, viscous stresses, gravity, and interfacial forces.

$$\begin{aligned} & \frac{\partial(\alpha\rho_1\mathbf{u}_1)}{\partial t} + \nabla \cdot (\alpha\rho_1\mathbf{u}_1 \otimes \mathbf{u}_1) \\ &= -\alpha\nabla p_1 + \nabla \cdot \left[\alpha\mu_1 \left(\nabla\mathbf{u}_1 + (\nabla\mathbf{u}_1)^T - \frac{2}{3}(\nabla \cdot \mathbf{u}_1)\mathbf{I} \right) \right] \\ & \quad + \alpha\rho_1\mathbf{g} + \mathbf{F}_\sigma \end{aligned} \quad (33)$$

$$\begin{aligned} & \frac{\partial((1-\alpha)\rho_2\mathbf{u}_2)}{\partial t} + \nabla \cdot ((1-\alpha)\rho_2\mathbf{u}_2 \otimes \mathbf{u}_2) \\ &= -(1-\alpha)\nabla p_2 + \nabla \cdot \left[(1-\alpha)\mu_2 \left(\nabla\mathbf{u}_2 + (\nabla\mathbf{u}_2)^T - \frac{2}{3}(\nabla \cdot \mathbf{u}_2)\mathbf{I} \right) \right] \\ & \quad + (1-\alpha)\rho_2\mathbf{g} + \mathbf{F}_\sigma \end{aligned} \quad (34)$$

The term $\mathbf{F}_\sigma$ represents the interfacial surface tension force. For liquid lead, the surface tension is known to be temperature-dependent and can be estimated using empirical correlations [9]. A commonly used expression is given by

$$\sigma(T) = (525.9 - 0.113 \cdot T) \cdot 10^{-3} \quad (35)$$

where the surface tension σ is given in Nm^{-1} and T in K. The correlation is valid over a temperature range relevant for molten lead and shows good agreement with the experimental data [9].

The movement of the interface is then predicted by introducing a transport equation for the phase fraction α .

$$\frac{\partial\alpha}{\partial t} + \nabla \cdot (\alpha\mathbf{u}) = 0 \quad (36)$$

The phase equations mentioned above describe a general two-fluid flow. In the regular VOF approach, both fluids are assumed to use a common velocity and pressure field as seen in

Equations 37 and 38.

$$\mathbf{u}_1 = \mathbf{u}_2 = \mathbf{u} \quad (37)$$

$$p_1 = p_2 = p \quad (38)$$

With this assumption, the governing equations are solved using one set of flow variables with the material properties having local variations depending on the phase fraction, which is seen in Equations 39, 40, and 41.

$$\rho(\alpha, T) = \alpha\rho_1(T) + (1 - \alpha)\rho_2(T) \quad (39)$$

$$\mu(\alpha, T) = \alpha\mu_1(T) + (1 - \alpha)\mu_2(T) \quad (40)$$

$$k(\alpha, T) = \alpha k_1(T) + (1 - \alpha)k_2(T) \quad (41)$$

Assuming that the velocity and pressure fields are shared between the fluids, the phase-wise equations can be combined into a single set of equations for the mixture. Therefore, the equations introduced in Section 2.4.1 retain the same structure, but with locally varying properties depending on the phase fraction as seen in Equations 42, 43, 44, and 45.

$$\frac{\partial \rho(\alpha, T)}{\partial t} + \nabla \cdot (\rho(\alpha, T)\mathbf{u}) = 0 \quad (42)$$

$$\frac{\partial (\rho(\alpha, T)\mathbf{u})}{\partial t} + \nabla \cdot (\rho(\alpha, T)\mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot \left[\mu(\alpha, T) \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T - \frac{2}{3}(\nabla \cdot \mathbf{u})\mathbf{I} \right) \right] + \mathbf{S}_M + \mathbf{F}_\sigma \quad (43)$$

$$\frac{\partial (\rho(\alpha, T)e)}{\partial t} + \nabla \cdot (\rho(\alpha, T)e\mathbf{u}) = -p\nabla \cdot \mathbf{u} + \nabla \cdot (k(\alpha, T)\nabla T) + \boldsymbol{\tau} : \nabla \mathbf{u} + S_E \quad (44)$$

$$\boldsymbol{\tau} = \mu(\alpha, T) \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T - \frac{2}{3}(\nabla \cdot \mathbf{u})\mathbf{I} \right) \quad (45)$$

2.5 OpenFOAM

OpenFOAM is an open-source CFD toolbox that provides a flexible framework for solving a wide range of fluid flow, heat transfer, and multiphase problems. It is developed and distributed through two main branches: OpenFOAM Foundation and OpenCFD Ltd., both of which originate from the same code base but are independently maintained. The OpenFOAM Foundation develops and distributes OpenFOAM as an open-source project [26]. In contrast, OpenCFD Ltd. develops the software independently and provides regular updates [27].

In this chapter, the OpenFOAM environment, structure, solvers and buoyancy treatment is introduced.

2.5.1 Environment and structure

OpenFOAM simulations are organized using a case directory structure consisting of *0*, *constant* and *system* as illustrated in Figure 3. These directories may vary slightly in content depending on the solver, as they rely on slightly different or additional files.

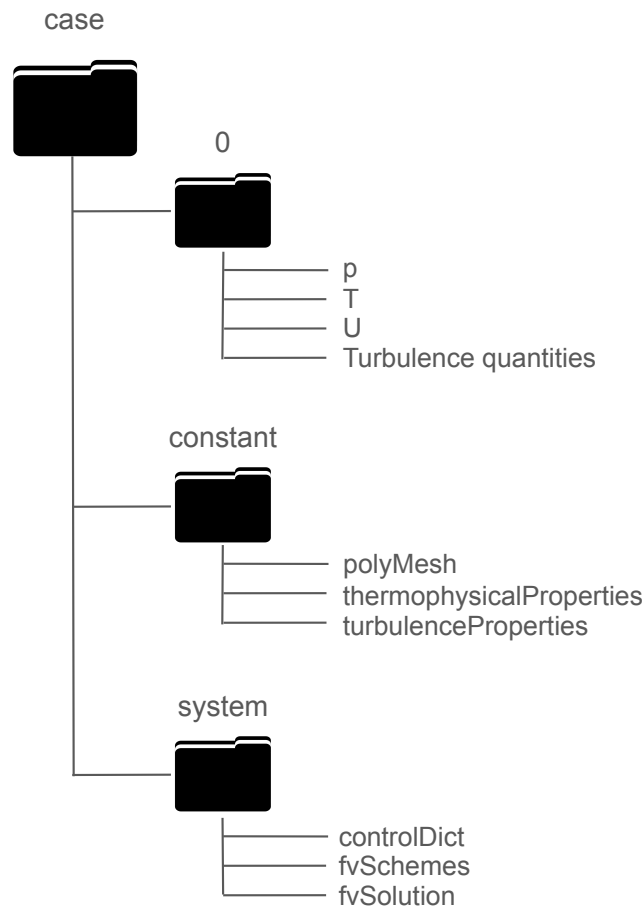


Figure 3: General structure of an OpenFOAM case directory.

0 The *0*-directory contains the initial and boundary conditions for all solved fields.

constant The *constant*-directory includes the mesh description, located in the *polyMesh* subdirectory, as well as physical and transport properties of the fluid.

system The *system*-directory contains settings related to the numerical solution procedure, including control parameters in *controlDict*, discretization schemes in *fvSchemes*, and solver settings in *fvSolution*.

In addition to the standard case structure, OpenFOAM provides flexibility through the use of additional dictionaries depending on the simulation requirements. For instance, *fvOptions* found in the *constant* directory is used to define the source terms.

2.5.2 Solvers

OpenFOAM provides a range of solvers adapted to different physical models and flow regimes. For simulations involving buoyancy-driven flows and multiphase systems, solvers such as *buoyantSimpleFoam*, *buoyantBoussinesqSimpleFoam*, and *compressibleInterFoam* can be used to model reactor cooling circuits as will be elaborated on in this section.

The *buoyantSimpleFoam* solver is a steady-state solver tailored for single-phase buoyancy-driven turbulent flows that involve heat transfer [28]. It is a variable density and pressure-based solver that utilizes the SIMPLE algorithm. The model accounts for density variations through its thermophysical model that can be adapted based on assumptions. In addition, it supports porosity modeling and integration of source terms. Moreover, the fields it produces consist of velocity, temperature, pressure, and dynamic pressure. The energy equation in this solver uses enthalpy instead of internal energy.

An alternative solver for buoyancy-driven flows is *buoyantBoussinesqSimpleFoam*, which applies the Boussinesq approximation (see Section 2.3) [29]. In this approach, density variations are neglected everywhere except for the buoyancy term, where it is assumed to be linearly dependent on temperature. This simplifies the governing equations and is suitable for flows with small temperature differences, where density variations remain limited.

The solver *compressibleInterFoam* is designed for two immiscible fluids with variable density and non-isothermal conditions and uses the VOF phase-fraction approach [30]. In contrast to *buoyantSimpleFoam*, this is a transient solver that utilizes the PIMPLE algorithm for pressure-velocity coupling. It shares several features with the single-phase solver in which it accounts for buoyancy effects, enables porosity modeling, and considers density variations through its configurable thermophysical model. The model solves the energy equation with temperature instead of internal energy.

As mentioned above, the thermophysical model of the fluid is adaptable in the *thermophysicalProperties*-directory [31]. Here, the equation of state used in the solver can be selected based on the application. One such model is *icoPolynomial*, which enables user-defined temperature-dependent relations, although the relations are required to be in the form of standard polynomials [32].

2.5.3 Treatment of buoyancy forces

Buoyancy forces can be treated differently depending on the numerical formulation of the governing equations. In variable-density flow, STAR-CCM+ accounts for buoyancy using Equation 46, where the force depends on the deviation from a global reference density [33].

$$\mathbf{F}_{\text{Buoyancy}} = (\rho - \rho_{\text{ref}})\mathbf{g} \quad (46)$$

This formulation may introduce inaccuracies in VOF simulations involving two immiscible fluids with large density differences. Since a single reference density is used, one of the phases may experience an unrealistic buoyancy force, leading to nonphysical flow behavior.

In OpenFOAM, solvers such as *buoyantSimpleFoam* and *compressibleInterFoam* instead solve for a modified pressure, p_{rgh} , defined as

$$p_{\text{rgh}} = p - \rho(\mathbf{g} \cdot \mathbf{h}) \quad (47)$$

where $\mathbf{g} \cdot \mathbf{h}$ represents the gravitational potential [34, 35].

Substituting this formulation into the momentum equation yields a buoyancy force that depends on local density gradients rather than a global reference density, as expressed in Equation 48.

$$\mathbf{F}_{\text{Buoyancy}} = -(\mathbf{g} \cdot \mathbf{h})\nabla\rho \quad (48)$$

This formulation is particularly advantageous for multiphase flows with large density differences, as it avoids the need for a global reference density and improves the physical consistency of buoyancy-driven flow simulations.

3 General methodology

This study is divided into four parts, where this section presents the general methodology applied throughout the work. The following chapters include the methodology, results, and discussion of single-phase steady-state forced convection, single-phase steady-state natural convection, and multiphase natural convection. Intermediate analyses are presented where necessary to motivate the modeling choices adopted in subsequent investigations.

Single-phase simulations are used to investigate the thermal-hydraulic behavior of the system under forced and natural convection. The model is subsequently extended to multiphase simulations using a VOF approach to represent the liquid-gas interface. All simulations in this study are performed using OpenFOAM v2312.

3.1 Simplified reactor model

The simplified reactor model represents a quarter of the complete reactor in which the vertical walls represent symmetry planes. The outer wall of the cold pool represents the RVACS system. Depending on the modeling approach, a free surface may be present at the top of the reactor. In multiphase simulations, this surface corresponds to the interface between the liquid lead and the argon cover gas, whereas in the single-phase simulations it is treated as a solid boundary. The simplified reactor model used in the simulations is illustrated in Figures 4 and 5.

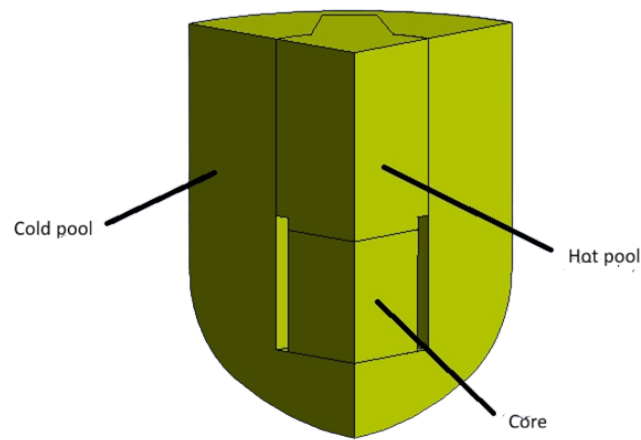


Figure 4: Front view of the simplified reactor.

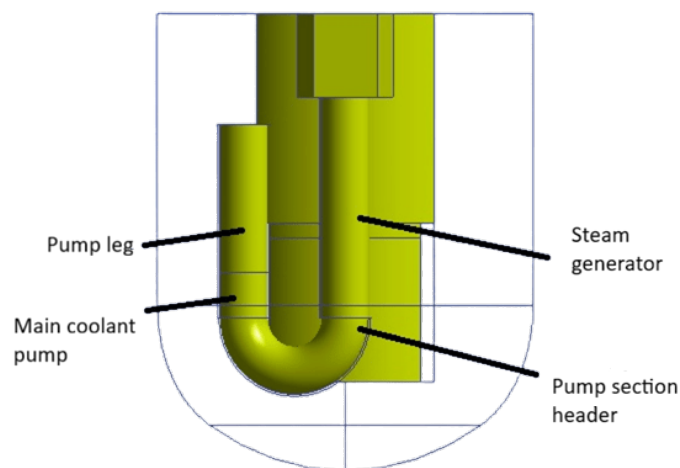


Figure 5: Rear view of the simplified reactor without the cold pool.

The liquid lead flows in a closed circuit without any prescribed inlet or outlet in the system. The coolant is heated in the reactor core before flowing through the hot pool and into the steam generator (SG). The flow is then redirected through the pump section header into the main coolant pump, after which it exits through the pump leg into the cold pool.

For single-phase natural convection simulations, additional reactor geometries are considered to investigate the influence of reactor height on flow behavior.

3.2 Post-processing and data evaluation

The flow properties evaluated in this study are obtained through post-processing of the numerical solution fields using OpenFOAM field functions applied to predefined face zones in the model. These zones are defined at the inlet and exit of each component.

The mass flow rate is calculated by summing the flux across the core and steam generator cross-sections. Depending on the solver, either ϕ or $\rho\phi$ is used for this purpose. In the constant density solver, ϕ represents the volumetric flux and is therefore scaled with a reference density, whereas in the variable-density solvers, ϕ and $\rho\phi$ respectively represent the mass flux. The pressure drop across each component is determined from the difference in the cross-sectional average of the modified pressure field p_{rgh} between the inlet and the outlet. Similarly, the temperature is evaluated as an area-averaged quantity across the corresponding cross-section.

3.3 Simulation strategy

The simulations are performed stepwise to ensure consistent and reliable development of the numerical model. First, single-phase steady-state forced convection simulations are conducted to verify the physics implementation with reference data from STAR-CCM+.

The verified setup is then applied to single-phase steady-state natural convection to analyze the governing flow behavior under buoyancy-driven conditions. Since no reference data are available for this case, the results are evaluated based on observed trends in global performance parameters and flow characteristics.

To assess the influence of thermophysical property modeling, two different fluid models are considered in the single-phase natural convection setup. The first model applies the Boussinesq hypothesis, where density variations are only accounted for in the buoyancy term. The second model uses temperature-dependent thermophysical properties derived from polynomial correlations presented in Section 2.1.2.

Finally, the model is extended to multiphase using a VOF approach to represent the liquid-gas interface. Although the multiphase solver is transient, the results are evaluated under quasi-steady-state conditions to enable comparison with the single-phase simulations. By comparing to the single-phase equivalent, the treatment of buoyancy in the OpenFOAM solvers can be evaluated.

3.4 Boussinesq model

In the Boussinesq model, the Boussinesq density approximation is adopted by using the *buoyantBoussinesqSimpleFoam* solver. In this study, the constant thermophysical properties of lead are evaluated at a temperature of 1273 K, using the correlations presented in Table 1. This temperature is used consistently throughout the study, including the implementation of porous media coefficients, to ensure a uniform property basis in all simulations. The resulting properties used for the solver are summarized in Table 2.

Table 2: Reference thermophysical properties of liquid lead used in Boussinesq model

Property	Value
ρ	$9.81 \cdot 10^3 \text{ kg m}^{-3}$
c_p	$138 \text{ J kg}^{-1} \text{ K}^{-1}$
μ	$1.06 \cdot 10^{-3} \text{ Pa s}$
k	$23.2 \text{ W m}^{-1} \text{ K}^{-1}$
Pr	0.00625
β	$1.30 \cdot 10^{-4} \text{ K}^{-1}$

3.5 Temperature-dependent property model

In alignment with the reference study, temperature-dependent relations are used to model the properties of the lead. The assumed relations of the fluid properties are stated in Table 1. To incorporate the temperature-dependent equations into OpenFOAM, the equation of state is set to *icoPolynomial*. This model requires standard polynomial expressions.

In addition, the molar mass of lead is specified in *thermophysicalProperties* to ensure consistency with the equation of state formulation. The molar mass is set to the standard value in the periodic table, $molWeight = 207.3 \text{ g mol}^{-1}$.

Since the temperature-dependent properties c_p and μ are not in standard polynomial form, they are implemented in OpenFOAM using polynomial approximations of correlations from the literature. The approximations are obtained using the least-squares polynomial regression

in MATLAB with *polyfit* and *polyval* functions. Regressions are performed over the valid temperature ranges for c_p and μ , as presented in Table 1.

The precision of the polynomial approximations and their relative errors are shown in Figure 6. The maximum relative deviations from the original correlations are 0.0024 % and 0.49 % for c_p and μ , respectively. These deviations remain within the estimated uncertainty of the correlations reported in the LBE handbook. The MATLAB code used for the polynomial fitting is provided in the Appendix A.

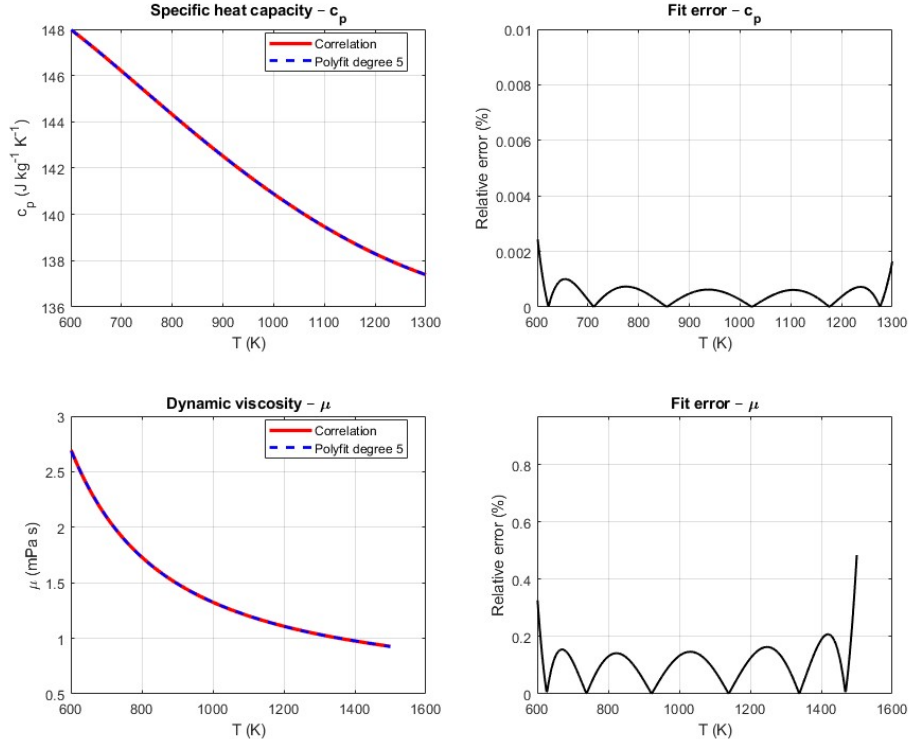


Figure 6: Polynomial approximation of specific heat capacity and dynamic viscosity.

Based on the polynomial regression, the resulting expressions for the specific heat capacity and dynamic viscosity are given as follows:

$$c_p(T) = 134.8 + 0.1046T - 2.383 \cdot 10^{-4}T^2 + 2.201 \cdot 10^{-7}T^3 - 9.820 \cdot 10^{-11}T^4 + 1.780 \cdot 10^{-14}T^5 \quad (49)$$

$$\mu(T) = 2.335 \cdot 10^{-2} - 8.435 \cdot 10^{-5}T + 1.349 \cdot 10^{-7}T^2 - 1.112 \cdot 10^{-10}T^3 + 4.644 \cdot 10^{-14}T^4 - 7.791 \cdot 10^{-18}T^5 \quad (50)$$

where T is the temperature in Kelvin.

3.6 Physics implementation

The physical behavior of the system is represented through simplified models of the core, steam generator, pump, and RVACS system. These components are implemented through source terms and coded functions located in the *constant/fvOptions* directory of the OpenFOAM case. This also requires defining the cell regions that each of the components occupy in the mesh, enforcing that each source term is strictly applied in relevant components.

Firstly, the reactor core and the steam generator act as a heat source and a heat sink, respectively. To represent this, volumetric heat transfer functions are implemented in these components. The core heat transfer to the coolant is implemented using the OpenFOAM *scalarSemiImplicitSource* function, which introduces a volumetric heat source into the energy equation [36].

Similarly to the core, the steam generator is modeled as a volumetric heat sink. Due to the dynamic coupling between the steam generator and the RVACS, a coded function is implemented to capture their coupled cooling behavior. Initially, RVACS is represented as a boundary heat flux, as described by Equation 51 where Q denotes the heat transfer, A is the boundary surface area, and T_{lead} is the local lead temperature. The convective heat transfer coefficient h and ambient air temperature T_{∞} are defined depending on the simulated operating condition.

$$Q = -A \cdot h \cdot (T_{\text{lead}} - T_{\infty}) \quad (51)$$

Consequently, the heat transfer rate to the air is dynamically dependent on the varying lead temperature. To maintain the overall energy balance, the steam generator removes the remaining heat from the system using a *scalarCodedSource*, as described by Equation 52 [37]. As a result, the cooling rate varies in both the steam generator and at the air boundary surrounding the vessel.

$$Q_{\text{sg}} = Q_{\text{core}} - Q_{\text{air}} \quad (52)$$

The coded function works by probing the real-time heat generation in the core, which is constant in this case, and the varying heat loss through the walls. Based on this, it calculates the necessary heat extraction required from the steam generator. A low-pass filter with the value $\beta = 0.005$ is also implemented to dampen numerical fluctuations, ensuring that the solver converges smoothly even with large temperature gradients. The filtered heat source Q_{filt}^n at the current time step n is calculated using the following recursive relation:

$$Q_{\text{filt}}^n = (1 - \beta) Q_{\text{filt}}^{n-1} + \beta Q_{\text{calc}}^n \quad (53)$$

where Q_{calc}^n is the raw calculated value from the energy balance at the current iteration. Using a small β , the solution prioritizes historical values, effectively filtering out high-frequency numerical noise that could otherwise lead to divergence in the coupled solver.

In addition, the reactor core and the steam generator are represented using porous media to account for the hydraulic resistance generated by the internal structures of these components. In the OpenFOAM model, this resistance is implemented using *explicitPorositySource*, which introduces a momentum sink in the momentum equation. Porous resistance is modeled using the Darcy-Forchheimer formulation, where pressure losses are represented by viscous and inertial resistance terms [38].

The pump is represented using the *vectorSemiImplicitSource* function in OpenFOAM, which introduces a momentum source corresponding to a pressure gradient per unit length [36].

To validate the setup, the mass flow rate and pressure drop in the core and the steam generator are compared with the reference simulation in which the reference study obtained the flow properties shown in Table 3.

Table 3: Component properties obtained from a reference simulation performed in STAR-CCM+.

Property	Value
$\dot{m}$	800 kg s^{-1}
Δp_{core}	55 kPa
Δp_{SG}	35 kPa
ΔT_{core}	150 K

3.7 Conversion of parameters

The parameters used in the OpenFOAM model are derived from the reference STAR-CCM+ simulation. Since the two solvers implement source terms and porous resistance slightly differently, the parameters must be converted before they can be used in the OpenFOAM formulation. The data provided from the STAR-CCM+ model are summarized in Table 4. Note that the core power corresponds to a quarter of the total power.

Table 4: Component properties from STAR-CCM+ model.

Property	Value
Q_{core}	17.5 MW
$\text{Core}_{\text{linear}}$	$6.4 \cdot 10^4 \text{ kg m}^{-3} \text{ s}^{-1}$
$\text{Core}_{\text{nonlinear}}$	$6.4 \cdot 10^6 \text{ kg m}^4$
$\text{SG}_{\text{linear}}$	$1.15 \cdot 10^3 \text{ kg m}^{-3} \text{ s}^{-1}$
$\text{SG}_{\text{nonlinear}}$	$1.15 \cdot 10^5 \text{ kg m}^4$
P/L	$2.0 \cdot 10^5 \text{ Pa m}^{-1}$

The core power represents the thermal energy source in the core, while the linear and nonlinear coefficients of the core and the steam generator correspond to the viscous and inertial resistance terms in the porous media model. The quantity P/L represents the pressure gradient generated by the pump, expressed as the pressure increase per unit length. In this model, this term acts as a momentum source that drives the flow through the system.

The fluid properties used for parameter conversion correspond to the reference thermophysical properties defined in Section 3.4 and summarized in Table 2.

In the Boussinesq formulation, the energy equation is written in terms of temperature rather than energy. Consequently, the heat source must be scaled by the product of the reference density and the specific heat capacity, $\rho \cdot c_p$, to obtain the corresponding temperature source term.

For the temperature-dependent formulation, the core power is implemented directly as a source term in the energy equation. The total power of the core is specified as $S_u = 17.5 \text{ MW}$ while the implicit coefficient is set to $S_p = 0$, corresponding to a purely explicit heat source in the core region.

The reactor core and the steam generator are further represented using a porous media model in both simulations. In STAR-CCM+, the effect of the porous region is introduced as a momentum sink term, expressed as:

$$\mathbf{f}_p = -(\mathbf{P}_v \mathbf{v}_s + \mathbf{P}_i |\mathbf{v}_s| \mathbf{v}_s) \quad (54)$$

where $\mathbf{P}_v$ is the viscous (linear) resistance tensor and $\mathbf{P}_i$ is the inertial (quadratic) resistance tensor [33]. The superficial velocity is denoted by $\mathbf{v}_s$. This source term represents the momentum loss induced by the porous structure.

In OpenFOAM, the porous media model is implemented using the Darcy-Forchheimer formulation, where the momentum sink is expressed as

$$S = -\mu d \mathbf{u} - \frac{1}{2} \rho f |\mathbf{u}| \mathbf{u} \quad (55)$$

where d is the Darcy coefficient representing viscous resistance and f is the Forchheimer coefficient representing inertial resistance [38]. Since internal structures are not modeled in the geometry, the velocity $\mathbf{u}$ is the same as the superficial velocity $\mathbf{v}_s$.

To obtain equivalent coefficients in OpenFOAM, the viscous resistance tensor $\mathbf{P}_v$ of the STAR-CCM+ model is divided by the dynamic viscosity of the fluid to obtain the Darcy coefficient d . The inertial resistance tensor $\mathbf{P}_i$ is divided by the fluid density and multiplied by two to obtain the Forchheimer coefficient f . This ensures that the viscous and inertial contributions to the pressure drop in the OpenFOAM formulation reproduce the same hydraulic resistance as in the reference STAR-CCM+ model. The Darcy and Forchheimer

coefficients have a x-, y-, and z-component. The reference STAR-CCM+ model employs the flow resistance in the flow-wise direction, which is why the converted coefficients are employed only in the y-direction. To remain consistent with the STAR-CCM+ simulation, the other components are set to sufficiently large values to impose a unidirectional flow.

Additionally, the momentum equation in the Boussinesq model is expressed per unit mass. As a result, the pressure gradient obtained from the STAR-CCM+ model must be divided by a reference density before being applied as a source term.

For the temperature-dependent model, the governing equations are expressed per unit volume and the density is already included in the formulation. Therefore, the pressure gradient from the STAR-CCM+ model can be applied directly as the pump source term without additional scaling. The resulting parameters used in the OpenFOAM simulations are summarized in Tables 5 and 6.

Table 5: Resulting component properties for *buoyantBoussinesqSimpleFoam*.

Property	Value
Q_{core}	$13.0 \text{ K m}^3 \text{ s}^{-1}$
$\text{Core}_{\text{linear}}$	$5.06 \cdot 10^7 \text{ m}^{-2}$
$\text{Core}_{\text{nonlinear}}$	$1.30 \cdot 10^3 \text{ m}^{-1}$
$\text{SG}_{\text{linear}}$	$9.10 \cdot 10^5 \text{ m}^{-2}$
$\text{SG}_{\text{nonlinear}}$	23.4 m^{-1}
P/L	20.4 m s^{-2}

Table 6: Resulting component properties for *buoyantSimpleFoam*.

Property	Value
Q_{core}	17.5 MW
$\text{Core}_{\text{linear}}$	$5.06 \cdot 10^7 \text{ m}^{-2}$
$\text{Core}_{\text{nonlinear}}$	$1.30 \cdot 10^3 \text{ m}^{-1}$
$\text{SG}_{\text{linear}}$	$9.10 \cdot 10^5 \text{ m}^{-2}$
$\text{SG}_{\text{nonlinear}}$	23.4 m^{-1}
P/L	$2.0 \cdot 10^5 \text{ Pa m}^{-1}$

3.8 Boundary conditions

The boundary conditions in the setup are integrated into OpenFOAM in the *0*-directory. They are selected to ensure physical consistency with the reactor model while maintaining numerical robustness. Furthermore, the side (0° and 90°) boundaries are treated as cyclicAMI

interfaces, enabling an accurate representation of the full reactor geometry. The symbol * represents a collection of boundaries in the OpenFOAM setup that are treated with identical boundary conditions. For example, there are two different free surfaces, one for the hot pool and one for the cold pool, which are summarized with the collective name *free_surface**.

Depending on the modeling approach, certain boundary conditions are modified. In particular, the single-phase free-surface boundary is replaced with a liquid-gas interface in multiphase simulations.

Moreover, the boundary conditions presented in this section are for the $k-\omega$ SST model. To clarify, all boundary conditions used are based on the official OpenFOAM documentation [39].

Velocity Solid walls are assigned the condition *noSlip*, which enforces zero velocity on the wall surface [12]. The free surface regions are modeled using a *slip* condition. This ensures a zero normal velocity while allowing tangential motion without shear stress. This allows for a simplified representation of the free surface in single-phase simulations. The summarized boundary conditions are shown in Table 7.

Table 7: Boundary conditions for the velocity field $\mathbf{u}$.

Boundary	Type	Specification
walls.*	noSlip	$\mathbf{u} = 0$
free_surface*	slip	$\mathbf{u} \cdot \mathbf{n} = 0$
air_wall	noSlip	$\mathbf{u} = 0$

Temperature All solid walls and free-surface regions use the *zeroGradient* boundary condition, corresponding to zero heat flux through these boundaries. The exception is the *air_wall* patch, which is modeled using *externalWallHeatFluxTemperature* according to Equation 51 to represent the RVACS system. This enables heat removal to the surrounding air without explicitly modeling the external flow domain. The ambient air temperature and convective heat transfer coefficients are assumed constant. Table 8 summarizes the applied temperature boundary conditions.

Table 8: Boundary conditions for temperature T .

Boundary	Type	Specification
free_surface*	zeroGradient	$\nabla T \cdot \mathbf{n} = 0$
air_wall	externalWallHeatFluxTemperature	$h = 15 \text{ W m}^{-2} \text{ K}^{-1}, T_{\infty} = 288 \text{ K}$
walls.*	zeroGradient	$\nabla T \cdot \mathbf{n} = 0$

Modified pressure A *fixedFluxPressure* boundary condition is applied to all non-periodic boundaries as seen in Table 9. This ensures that the pressure gradient at solid boundaries remains consistent with the mass flux obtained from the momentum equation.

Table 9: Boundary conditions for dynamic pressure p_{rgh} .

Boundary	Type	Specification
walls.*	fixedFluxPressure	0
free_surface*	fixedFluxPressure	0
air_wall	fixedFluxPressure	0

Static pressure The static pressure field is defined as *calculated*, which implies that it is derived from p_{rgh} and the hydrostatic contribution. It is therefore not independently solved but reconstructed during the solution process. In Table 10, p_{internal} means that each cell along the boundary uses the static pressure value of their adjacent cells inside the mesh domain.

Table 10: Boundary conditions for static pressure p .

Boundary	Type	Specification
walls.*	calculated	$p = p_{\text{internal}}$
free_surface*	calculated	$p = p_{\text{internal}}$
air_wall	calculated	$p = p_{\text{internal}}$

Turbulent boundary conditions Turbulent boundary conditions are implemented using standard OpenFOAM wall functions to ensure numerical stability and consistency with the turbulence models. For the turbulent kinematic viscosity, the *nutUSpaldingWallFunction* boundary condition is applied. This wall function provides a continuous velocity profile across a wide range of y^+ values, which improves the robustness of the near-wall treatment [40].

An explicit analysis of the resulting y^+ values is not performed in this study, as such investigation is considered outside the scope of the present work. Instead, the influence of the near-wall resolution is assessed through the wall refinement study presented in Section 5.4, where the first layer thickness at the exterior wall is systematically varied to evaluate its impact on the natural circulation flow.

At the free surface, the turbulent quantities are assigned *zeroGradient* boundary conditions in order to minimize the influence of the free surface on the flow.

The complete set of turbulent boundary conditions used in this study is summarized in Table 11.

Table 11: Boundary conditions for turbulence quantities, α_t , ω , ν_t and k

Field	Boundary	Type
α_t	walls.*	alphatWallFunction
–	free_surface	zeroGradient
–	air_wall	alphatWallFunction
ω	walls.*	omegaWallFunction
–	free_surface	zeroGradient
–	air_wall	omegaWallFunction
ν_t	walls.*	nutUSpaldingWallFunction
–	free_surface	zeroGradient
–	air_wall	nutUSpaldingWallFunction
k	walls.*	kqRWallFunction
–	free_surface	zeroGradient
–	air_wall	kqRWallFunction

3.9 Numerical setup

The numerical setup is specified in *fvSchemes* and *fvSolution* within the *system*-directory. The selected numerical schemes are determined based on the ability to achieve convergence on the coarsest mesh for the mesh sensitivity study for the forced convection setup. This results in a numerical setup that prioritizes robustness over spatial accuracy, which is appropriate since this study focuses on bulk flow quantities rather than solving local flow behavior. This numerical setup is used throughout the study, while some adjustments are made and highlighted in the following chapters where it is found necessary. The SIMPLEC algorithm is utilized in all single-phase simulations.

For the discretization of the convective terms, the first-order *bounded Gauss upwind* scheme is applied for the momentum and energy convection terms to ensure numerical stability and boundedness. The convective turbulence quantities are discretized using the limited *bounded Gauss linear* scheme to achieve higher order accuracy while maintaining stability.

The diffusion terms are discretized using a *Gauss linear* limited scheme to balance numerical stability and spatial accuracy. Gradients are evaluated using the *Gauss linear* scheme to maintain spatial accuracy. To balance accuracy in the complex geometry while mitigating errors associated with mesh non-orthogonality, limited surface normal gradients are used in combination with *Gauss linear* limited scheme for the diffusion terms. The *linear* scheme is

employed for interpolation as it achieves higher accuracy and exhibits minimal impact on stability as observed in these particular simulations.

Unless explicitly stated in the following chapters, standard under-relaxation values are used in the range of 0.1 – 0.6 where alterations within this range show negligible impact on the predicted results.

4 Single-phase steady-state forced convection

The purpose of the steady-state forced convection case is to verify the implemented physics by comparing global hydraulic performance with the STAR-CCM+ model. These evaluated quantities include the mass flow rate and the pressure drop in the core and the steam generator. This is because the flow is expected to be primarily governed by the pump and porous media implementation. To assess the numerical setup, a grid convergence analysis and a turbulence model comparison are conducted.

4.1 Forced convection setup

The forced convection simulations represent normal operating conditions in which the coolant circulation is driven by the primary pump. The pump source term and the steam generator cooling are therefore active throughout the simulations.

The RVACS boundary is modeled using a convective heat transfer coefficient of $h = 15 \text{ W m}^{-2} \text{ K}^{-1}$ together with an ambient air temperature of $T_{\infty} = 288 \text{ K}$, consistent with the STAR-CCM+ reference setup.

4.2 Mesh generation

Since the quantities of interest are bulk flow characteristics rather than detailed wall heat transfer, near-wall resolution is not prioritized in this case. Furthermore, the system behavior is expected to be largely governed by pressure losses in porous media regions, since these represent the dominant hydraulic resistance in the system.

With this in mind, three inflation layers for all solid boundaries are applied, with a first layer thickness of 0.01 m, a growth rate of 1.3, and a cell size of 0.05 m. The resulting mesh is shown in Figure 7.

All meshes used throughout this study are generated using Ansys Fluent. This particular mesh does not include any local refinements in order to simplify the mesh sensitivity analysis.



Figure 7: Forced convection mesh with cell size 0.05 m

4.3 Grid convergence analysis

Initially, a mesh sensitivity analysis is conducted on the temperature-dependent property model to assess the discretization uncertainty of the numerical solution. All numerical schemes are kept unchanged throughout the mesh study to ensure that any observed differences between mesh levels are solely attributed to the variation in mesh resolution.

For the forced convection case, a set of eight progressively refined meshes are generated. The resulting mass flow rate and pressure drop are evaluated as functions of the total cell count to assess the sensitivity of the solution to mesh refinement. To further quantify grid convergence, the Grid Convergence Index (GCI) method proposed by Celik et al. [41] is applied.

Three meshes are selected from the initial set, denoted Grid 1, Grid 2, and Grid 3. The corresponding mesh properties are summarized in Table 12.

Table 12: Mesh properties used for the GCI analysis.

Mesh	Number of cells (N)	Total volume (V_{tot})	Cell size
Grid 1 (Fine)	1 776 496	29.8 m ³	0.034 m
Grid 2 (Medium)	763 574	29.8 m ³	0.050 m
Grid 3 (Coarse)	243 685	29.7 m ³	0.100 m

The representative cell size is defined as

$$\Delta x = \left(\frac{V_{\text{tot}}}{N} \right)^{1/3} \quad (56)$$

where V_{tot} is the total volume of the domain and N is the number of cells. Based on this, the grid refinement ratios are defined as

$$r_{21} = \frac{\Delta x_2}{\Delta x_1}, \quad r_{32} = \frac{\Delta x_3}{\Delta x_2} \quad (57)$$

The differences between successive grid solutions are expressed as

$$\varepsilon_{21} = \phi_2 - \phi_1 \quad (58)$$

$$\varepsilon_{32} = \phi_3 - \phi_2 \quad (59)$$

where ϕ represents the evaluated quantities, in this case the mass flow rate or the pressure drop. Monotonic convergence is identified when these differences retain the same sign, while a change in sign indicates oscillatory convergence behavior.

The apparent order of accuracy p_{obs} is obtained iteratively, and the discretization error is estimated using the GCI formulation. The GCI values of the fine and medium grid are computed using a safety factor of 1.25, as recommended in the literature [41].

To verify that the solution lies within the asymptotic range (AR) of convergence, the parameter AR is evaluated [42]. The values of AR approaching unity indicate that the solution is within the asymptotic convergence region and that further grid refinement is expected to yield only marginal improvements in the accuracy of the solution while significantly increasing the computational cost. The complete MATLAB implementation of the GCI procedure is provided in Appendix B for transparency.

The grid convergence behavior is first assessed by examining the variations of the main flow parameters with mesh resolution. The predicted mass flow rate and pressure drop are plotted as functions of the total number of cells, as shown in Figure 8. A clear reduction in variation with increasing mesh resolution is observed, indicating convergence of the solution.

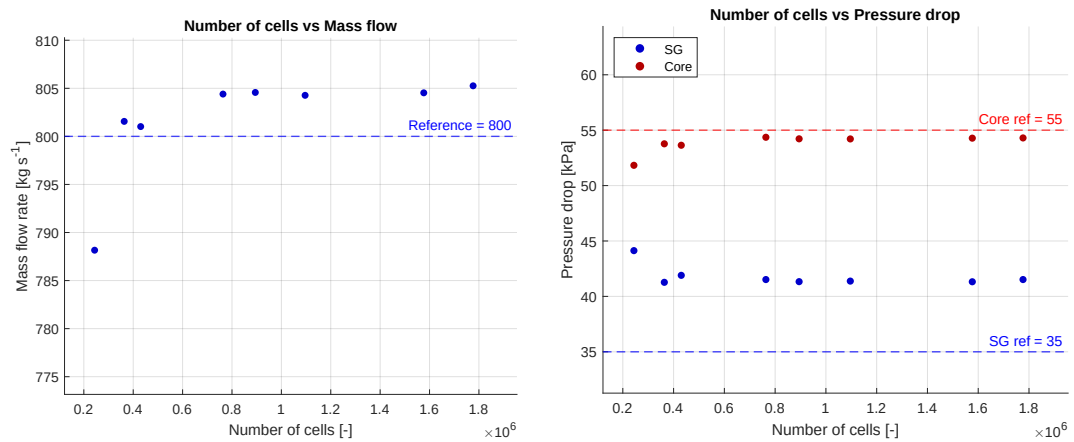


Figure 8: Mesh sensitivity study of the predicted pressure drop and mass flow rate.

To quantify the convergence of the grid, the GCI method is applied. The results of the analysis for the mass flow rate and the pressure drop in the core and the steam generator are presented in Table 13. The mass flow rate and the steam generator pressure drop exhibit monotonic convergence with mesh refinement, while the core pressure drop shows a slight oscillatory behavior, but still converges toward a consistent solution.

Table 13: Results of the grid convergence analysis.

Property	$\dot{m}$	Δp_{core}	Δp_{SG}
r_{21}	1.3251	1.3251	1.3251
r_{32}	1.4632	1.4632	1.4632
p_{obs}	7.54	9.93	23.74
GCI^{21}	0.018 %	0.0090 %	$1.2 \cdot 10^{-6}$ %
GCI^{32}	0.15 %	0.14 %	$9.3 \cdot 10^{-4}$ %
AR	1.001	0.925	1.000

The observed order of accuracy is higher than the nominal second-order spatial accuracy of the numerical discretization scheme. In the present case, the difference between the medium and fine grids is extremely small for several evaluated quantities, resulting in a very small estimated error between these grid levels and consequently an overestimation of the apparent order of accuracy. In general, the convergence behavior indicates a consistent spatial convergence of the numerical solution.

In conjunction, the calculated GCI values are below 1 % for all the evaluated quantities, indicating low discretization uncertainty. The asymptotic range parameter AR is close to unity for the mass flow rate and the pressure drop in the steam generator, suggesting that these quantities are within the asymptotic convergence range. For the core pressure drop, the value is slightly below unity, indicating that the asymptotic range may not be fully reached. However, the discretization uncertainties remain small for all quantities. Additionally, the residuals remained stable for all meshes.

Based on these results, the cell size of the medium mesh is selected for subsequent simulations as it provides a suitable balance between computational cost and accuracy. This corresponds to the fourth mesh from the left side in Figure 8.

4.4 Turbulence model comparison

To assess the influence of turbulence closure on the predicted reactor performance, three RANS turbulence models are evaluated: the $k-\omega$ SST model, the standard $k-\varepsilon$ model and the LRR model.

The turbulence modeling required modification with respect to the original numerical schemes and solution settings. Firstly, the *bounded Gauss upwind* scheme is adopted for the LRR and $k-\varepsilon$ model for the convective turbulence terms, since they did not yield stable solutions with the *bounded Gauss linear* limited scheme. The additional schemes for the Reynolds stresses utilized the *bounded Gauss linear* scheme to properly capture the anisotropy of flow. Moreover, to combat divergence issues for the LRR model, a significantly reduced under-relaxation factor of 0.001 is applied to the turbulence properties ε and the Reynolds stress tensor R to avoid numerical instability.

Figure 9 presents the percentage deviation from the reference values for the predicted mass flow rate and pressure drop across the reactor core and the steam generator for the different turbulence models.

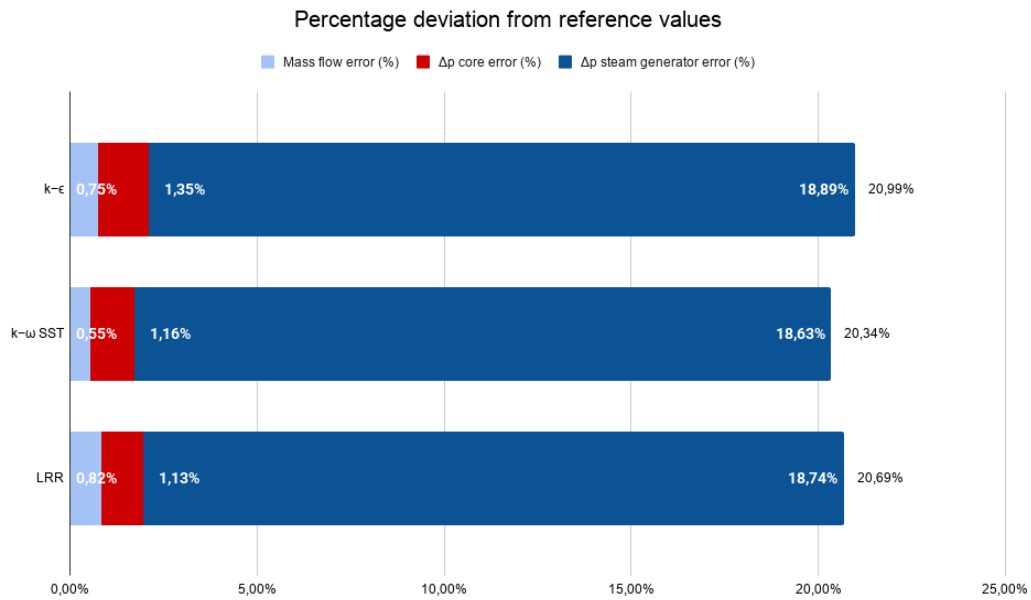


Figure 9: Comparison of predicted results using different turbulence models.

All turbulence models produce similar results. The deviation in the predicted mass flow rate is small for all models, remaining below 1%. The predicted pressure drop across the reactor core also shows only minor differences between the models.

In contrast, the pressure drop across the steam generator exhibits a significantly larger deviation from the reference value. The differences between the $k-\varepsilon$, $k-\omega$ SST and LRR models are very small, indicating that the choice of turbulence model has only a limited influence on the predicted hydraulic behavior of the system in this case.

In addition to the quantitative comparison, the flow fields are visually inspected in ParaView using Surface Line Integral Convolution (LIC), as shown in Figure 10 (see Appendix F for details on Surface LIC). Additional flow visualizations are provided in Appendix C.

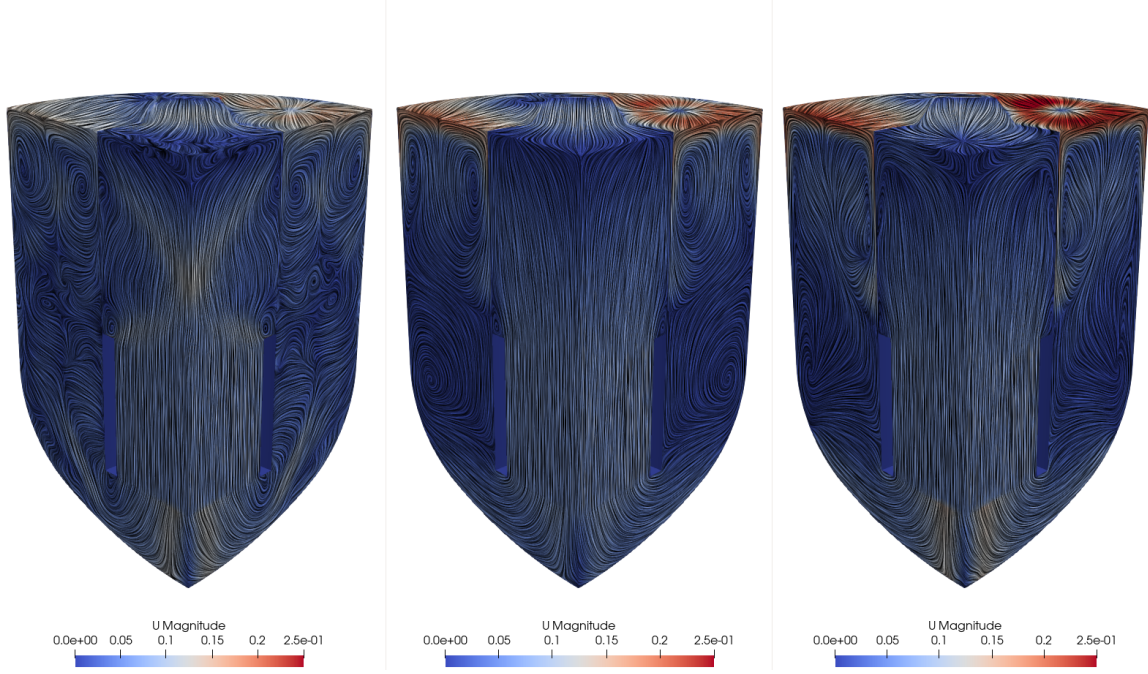


Figure 10: Velocity field for different turbulence models. From left to right: LRR, $k-\epsilon$, and $k-\omega$ SST.

Noticeable differences in the velocity fields are observed between the turbulence models, indicating variations in the predicted local flow structures. The LRR shows more complex flow structures, which is consistent with its ability to resolve anisotropic turbulence effects. Despite these local differences, the impact on global quantities remains limited.

Therefore, the $k-\omega$ SST model is selected as the reference model in the natural convection setup due to its robustness and reliable performance for internal flows. The LRR exhibited oscillatory behavior during the simulations while requiring significant under-relaxation, indicating reduced numerical stability for the present setup.

4.5 Forced convection performance

With the mesh sensitivity study and turbulence closure model analysis conducted, the concluding results are presented in Table 14.

Table 14: Component properties for the selected mesh and turbulence model

Property	Value
$\dot{m}_{\text{core/SG}}$	804 kg s^{-1}
Δp_{core}	54.4 kPa
Δp_{SG}	41.5 kPa
ΔT_{core}	148 K

The velocity field is illustrated earlier in Figure 10 in addition to cross-sectional views found in Appendix F.

4.6 Model validation and flow behavior

The steady-state forced convection simulations serve to validate the implemented OpenFOAM model against the STAR-CCM+ reference data. The predicted mass flow rate, temperature difference, and pressure drop in the reactor core are in good agreement with the reference values, with a deviation below 1 %. This indicates that the porous media and momentum source terms are correctly implemented to reproduce the system hydraulics.

In contrast, a large deviation is observed for the pressure drop across the steam generator. Since the deviation is consistent across all turbulence models, it is unlikely that the turbulence model is the root cause. One possible source of uncertainty is the polynomial approximation used for the temperature-dependent heat capacity and dynamic viscosity. However, the maximum fitting errors are small, suggesting that the polynomial approximation is unlikely to explain the observed difference.

The implemented model and the reference STAR-CCM+ model integrate porous media representations of the steam generator to model the pressure drop in this component, indicating that the discrepancy does not originate from the modeling approach itself. Furthermore, the agreement in pressure drop observed for the reactor core suggests that the conversion of coefficients is unlikely to be the primary source of the deviation.

The deviation in the steam generator therefore indicates that the pressure drop in this component is more sensitive to the prescribed porous media coefficients. Since the mass flow through the core and steam generator is identical, whereas the steam generator has a significantly smaller flow cross-section, the mean velocity in this component is higher in accordance with the continuity equation. Hence, uncertainties in the conversion of coefficients may be amplified in the steam generator compared to the reactor core. This effect is likely more pronounced in the nonlinear component since the pressure loss scales with the square of the velocity.

The discrepancy may also be influenced by the choice of reference temperature used to evaluate the fluid properties in parameter conversion. The selected temperature does not represent the temperature range observed in the simulations, which may introduce deviations in the resulting Darcy-Forchheimer coefficients, since both viscosity and density directly affect these parameters.

Furthermore, the limited influence of the turbulence model on bulk flow parameters indicates that the forced convection scenario is mainly governed by the fixed pressure gradient, as evidenced by the streamlined flow in the core and tube sections. Moreover, the models exhibit differences in local flow structures, where the LRR exhibits a greater ability to capture smaller vortices and flow separation around sharp geometries. However, all turbulence models are capable of reproducing large-scale turbulence in key regions such as the main pump leg exit and the cold pool with similar characteristics.

Due to the flow being dominated by the momentum source and porous media, this suggests that lower-order closure models, such as the $k-\varepsilon$ and $k-\omega$ SST, are sufficient to predict system-level performance in the forced convection case. Although the LRR model provides a more detailed representation of turbulence, this additional fidelity does not translate into improved predictions of the bulk flow behavior. Furthermore, the LRR model was found to be numerically sensitive, since it required significant under-relaxation and conservative schemes to achieve convergence. Combined with its higher computational cost, this results in significantly longer simulation times compared to the turbulent viscosity models.

The mesh sensitivity analysis further supports the reliability of the results. The predicted bulk flow parameters exhibit a clear convergence with the mesh refinement, indicating a grid-independent solution. Additionally, the low discretization uncertainty determined in the grid convergence index implies that numerical errors have a negligible impact on the obtained values. Consequently, the selected mesh refinement provides sufficient accuracy without excessive computational cost.

5 Single-phase steady-state natural convection

The purpose of this section is to evaluate the thermal-hydraulic behavior of the system during natural convection conditions. Since reference data is not available for this case, a qualitative analysis is conducted together with an evaluation of trends in bulk flow parameters to determine the influence of reactor height, near-wall resolution, mesh refinement, and turbulence closure model.

The primary quantities used to evaluate the natural circulation performance are the mass flow rate and the difference in core temperature. Although these global metrics do not capture

local thermal effects, they provide a robust measure of system-level cooling performance, which is the main focus of this part of the study.

5.1 Natural convection setup

The natural convection simulations represent decay heat removal conditions in which the primary pump is deactivated and the coolant circulation is driven solely by buoyancy forces. To represent residual heat conditions, the core power is reduced to 2% of the nominal operating power. Furthermore, the steam generator cooling is deactivated, while the RVACS boundary remains active to remove heat from the reactor vessel.

To achieve realistic residual heat removal, the convective heat transfer coefficient is increased to $h = 25 \text{ W m}^{-2} \text{ K}^{-1}$, while the ambient air temperature remains $T_\infty = 288 \text{ K}$.

As mentioned in the previous chapter, this investigation uses the $k-\omega$ SST model as the reference turbulence model. Therefore, conclusions regarding reactor height, mesh sensitivity, and near-wall resolution are based on this model only.

5.2 Mesh generation

For the steady-state natural convection case, buoyancy-driven flow and wall heat transfer play a significant role in the flow development. Since the outer wall is responsible for heat removal through air cooling, additional refinement is applied in this region. The exterior wall uses a first layer thickness of 0.008 m, while the remaining walls use a thickness of 0.025 m, which helps to maintain consistent mesh quality throughout the domain. A close-up view of the generated mesh is shown in Figure 11.

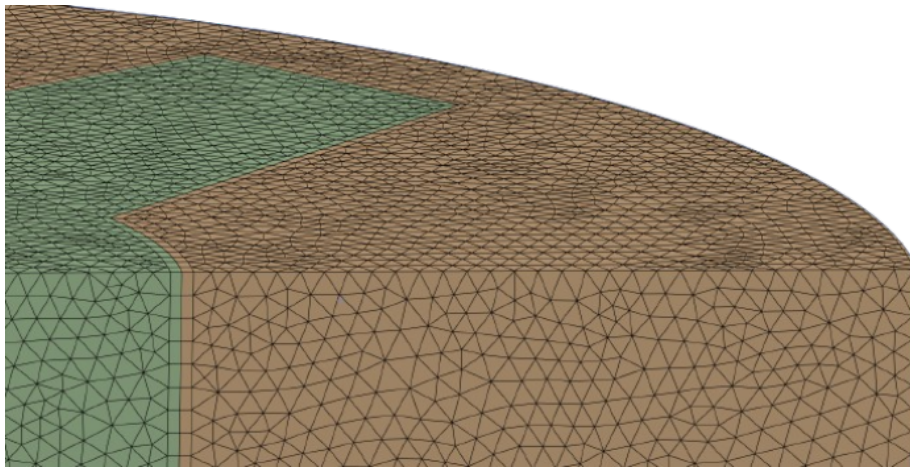


Figure 11: Close-up view of the natural convection mesh illustrating the different first layer thicknesses.

5.3 Reactor height study

To investigate the influence of reactor height on flow behavior under natural convection conditions, three geometrical configurations are considered. In addition to the base model, two extended configurations are introduced, with the total reactor height increased by 1 m and 2 m, respectively.

Critical components, including the core, the steam generator, the pump, and the pump section header, remain unchanged. The components directly affected by the change in height include the pump leg, the hot pool, and the cold pool regions which are extruded proportionally with the new reactor height.

The same numerical setup, including mesh resolution, boundary conditions, and turbulence model, is applied consistently across all three configurations.

The models are compared on the basis of the resulting flow structure and temperature distribution, with particular focus on the development of the flow in the pump leg and tube section.

The influence of reactor height on flow behavior is illustrated by comparing the three configurations. As shown in Figure 12, the base model exhibits local flow disturbances caused by vortices in the pump section header.

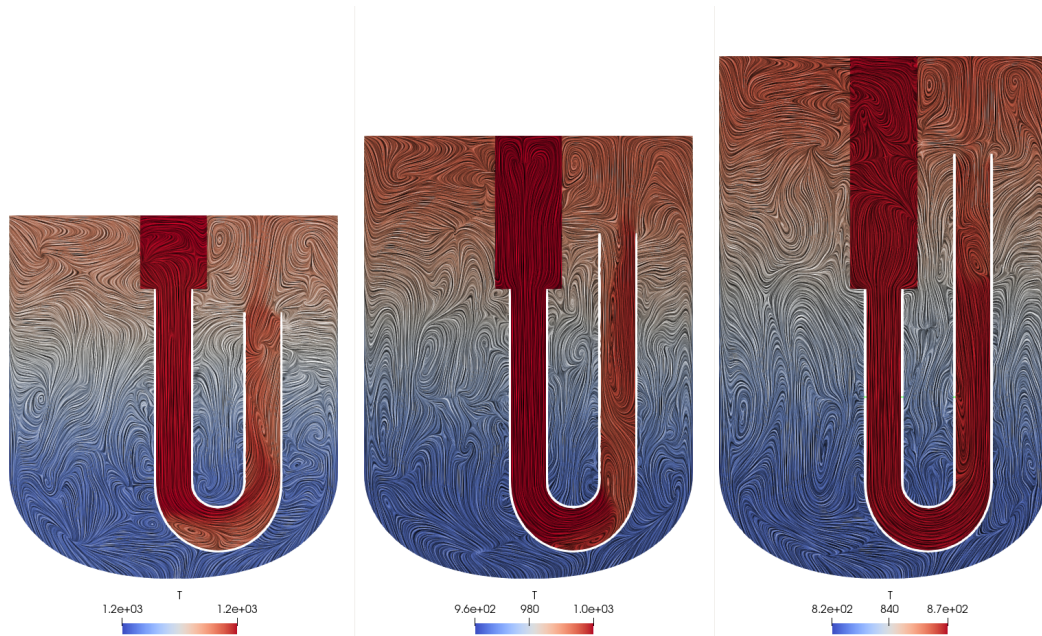


Figure 12: Vertical cross-section of temperature distribution and flow field at three different reactor heights.

As the reactor height increases, these disturbances are shifted downstream, resulting in smoother and more developed flow. Consequently, the temperature distribution in the tube section becomes more uniform.

Table 15 presents the corresponding flow quantities for the different reactor configurations. Increasing the reactor height leads to a higher mass flow rate, a more uniform temperature distribution, and a reduced temperature difference across the core.

Table 15: Comparison of flow and thermal quantities for different reactor heights.

Model	$\dot{m}$	ΔT_{core}
Base	46.5 kg s ⁻¹	54.5 K
1 m	52.7 kg s ⁻¹	46.6 K
2 m	54.1 kg s ⁻¹	44.8 K

These results show that the natural convection in the present setup is strongly influenced by the reactor height. Increasing the height improves the flow development and enhances the transport of thermal energy.

Consequently, the tallest configuration is used in the following investigations, as it provides improved flow conditions for heat removal from the core.

5.4 Near-wall resolution analysis

The near-wall mesh configuration described in the mesh generation section is further evaluated by comparing different mesh setups with varying first layer thicknesses, 0.008 m and 0.016 m, as well as the effect of using a single layer compared to multiple layers on the exterior wall. The mesh configurations are presented in Table 16.

Table 16: Overview of mesh configurations used in the near-wall resolution study.

Mesh	First layer thickness	Growth rate
0.008 m, 1 layer	0.008 m	1.2
0.008 m, 8 layers	0.008 m	1.2
0.016 m, 1 layer	0.016 m	1.2
0.016 m, 6 layers	0.016 m	1.2

The near-wall resolution of the exterior wall is evaluated by comparing the mass flow rate and the temperature difference across the core, as presented in Table 17.

Table 17: Overview of results from the near-wall resolution study.

Mesh	$\dot{m}$	ΔT_{core}
0.008 m, 1 layer	54.2 kg s ⁻¹	44.5 K
0.008 m, 8 layers	54.1 kg s ⁻¹	44.8 K
0.016 m, 1 layer	58.8 kg s ⁻¹	41.3 K
0.016 m, 6 layers	57.4 kg s ⁻¹	41.2 K

The results show that the number of boundary layers has a negligible impact on the predicted global quantities. Only minor differences in mass flow rate and temperature difference across the core are observed between configurations with one and multiple layers for the same first layer thickness.

In contrast, the thickness of the first layer has a more noticeable influence on the solution. The coarser near-wall resolution, 0.016 m, results in higher mass flow rates and lower temperature differences across the core, indicating that the near-wall gradients are not sufficiently resolved.

Based on these observations, the configuration with a first layer thickness of 0.008 m and a single layer is selected. This configuration improves near-wall resolution while avoiding unnecessary mesh complexity and computational cost that multiple inflation layers would introduce. Further refinement of the first layer thickness is omitted as it is outside the scope of this study.

5.5 Grid convergence analysis

Based on the selected near-wall configuration, a grid convergence analysis is performed to assess whether the previously selected mesh resolution is sufficient to capture the flow physics or if further refinement of the cell size is required.

In the natural convection case, three meshes are compared. The coarse mesh uses the same cell size as the mesh selected from the forced convection scenario while two further refined meshes are compared. The mesh properties are presented in Table 18.

Table 18: Mesh properties for the meshes used in natural convection grid analysis.

Mesh	Number of cells (N)	Cell size
Grid 1 (Fine)	1 957 796	0.0425 m
Grid 2 (Medium)	1 710 250	0.0450 m
Grid 3 (Coarse)	1 402 312	0.0500 m

Similarly to the mesh study conducted in forced convection, the meshes compared utilize unchanged numerical schemes to isolate potential variations to mesh refinements.

The meshes are compared based on the mass flow rate and the core temperature difference and the results are presented in Table 19.

Table 19: Mesh refinement results

Mesh	$\dot{m}$	ΔT_{core}
Grid 1 (Fine)	56.5 kg s ⁻¹	42.7 K
Grid 2 (Medium)	57.5 kg s ⁻¹	42.4 K
Grid 3 (Coarse)	54.2 kg s ⁻¹	44.5 K

Since the deviations in the mass flow rate and core temperature are deemed negligible, the coarse mesh is selected as the mesh for the remaining parts of the study. This eliminates additional computational costs associated with a further decreased cell size.

5.6 Turbulence model comparison

To evaluate the influence of turbulence modeling in the natural convection regime, the same set of RANS models is considered: the k - ω SST model, the k - ε model and the LRR.

Beyond bulk flow performance metrics, the turbulence models are analyzed based on velocity fields to determine their ability to capture key turbulence structures.

The numerical schemes and solution settings are kept unchanged in this turbulence model comparison with a few modifications required for numerical stability. In particular, the LRR model exhibited an increased numerical sensitivity and therefore required additional stabilization measures.

To improve stability, *uncorrected* Laplacian and surface-normal gradient schemes are employed together with the *Gauss linear* limited scheme for gradients and Reynolds stress divergence terms. Furthermore, reduced under-relaxation factors of 0.1 are applied for u , p_{rgh} , h , and ρ , while values of 0.03 and 0.01 are used for ε and R , respectively.

Similarly to the mesh refinement analysis, the mass flow rate and the temperature difference across the core are compared for the three models and are shown in Table 20. The eddy-viscosity models yield similar results, while the LRR model differs noticeably in both bulk flow parameters. In addition, the LRR simulations require substantially higher computational effort and exhibit reduced numerical robustness.

Due to this discrepancy, as well as the increased computational cost of the LRR model, the $k-\omega$ SST model is selected for the remaining parts of the study. This choice also maintains consistency with the forced convection simulations presented earlier in the work.

Table 20: Turbulence model comparison.

Turbulence model	$\dot{m}$	ΔT_{core}
$k-\omega$ SST	54.2 kg s^{-1}	44.5 K
$k-\varepsilon$	57.5 kg s^{-1}	42.0 K
LRR	45.0 kg s^{-1}	50.8 K

An overview of the turbulence captured by the different models is shown in Figure 13. Additional illustrations are provided in Appendix D. In the same appendix, illustrations of the Reynolds shear stress component fields are presented in Figure 23.

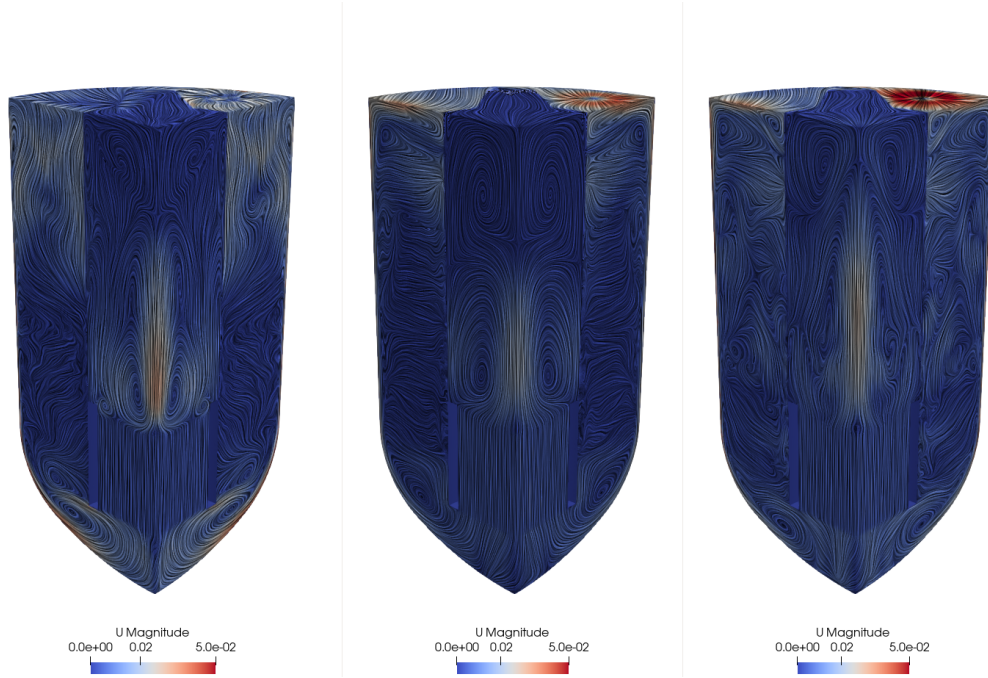


Figure 13: Velocity field for different turbulence models. From left to right: LRR, $k-\varepsilon$, and $k-\omega$ SST.

It is observed that all models are capable of capturing large-scale vortices in key areas like the core inlet and outlet, as well as the pump leg exit. The three models additionally predict different degrees of turbulence structures in the pump leg. Moreover, the LRR shows the ability to capture additional turbulence in the pump section header in addition to anisotropic effects in the steam generator. The steam generator also exhibits a less uniform velocity profile in the LRR case.

5.7 Boussinesq model performance

The selected reactor height, mesh refinement, and turbulence model are subsequently used to evaluate the Boussinesq model and compare its behavior with the previously established temperature-dependent model. The bulk flow parameters are summarized in Table 21.

The comparison is performed to assess whether the simplified density treatment used in the Boussinesq approximation is sufficient to capture the dominant buoyancy-driven flow behavior in the system.

Table 21: Performance metrics for the Boussinesq simulation under steady-state natural convection.

Model	$\dot{m}$	ΔT_{core}
Boussinesq	60.8 kg s^{-1}	40.4 K

An overview of the velocity field is presented in Figure 14. Additional cross-sectional views are provided in Appendix E.



Figure 14: Velocity field in Boussinesq simulation under steady-state natural convection conditions.

5.8 Governing physics and flow behavior

Natural convection simulations show a strong dependence on reactor height. As the height increases, the predicted mass flow rate increases, while the temperature difference across the core is reduced. In addition, the flow becomes more stable, exhibits fewer local disturbances and a more uniform temperature distribution. This trend can be explained by buoyancy-driven forces being the dominant mechanism governing the flow. Increasing the reactor height enhances the cumulative buoyancy effect as a result of larger density differences over the flow path, resulting in stronger circulation and a higher mass flow rate due to the increased hydrostatic pressure difference induced by buoyancy.

Furthermore, the extended flow path due to the extruded geometry enables a more fully developed flow. Consequently, the influence of local recirculation is reduced. As a result, the vortices observed in the pump section header for the shorter reactors are not present in the taller reactors. This yields an improved thermal distribution in the system, which is essential for decay heat removal.

The decrease in temperature difference across the core is a product of multiple factors. The primary reason is that the volumetric heat source is constant, which implies that the temperature difference across the core must decrease when the mass flow rate increases. This is because the core heat transfer is proportional to the mass flow rate and temperature difference. In addition, the increased circulation observed in the taller configurations improves the heat transport throughout the system, which distributes the thermal energy more evenly as highlighted earlier. This may in turn also contribute to the lower temperature difference across the core.

Even though flow development is improved with height, all models show some recirculation in the pipe section. Nevertheless, the thermal energy is transferred out into the cold pool, although to different extents. This behavior is consistent with the low Prandtl number of liquid lead, where thermal diffusion dominates momentum diffusion, which means that heat can spread efficiently even in the presence of local flow disturbances.

Apart from the geometry, the mesh showed negligible sensitivity to refinement of the cell size and the number of prismatic inflation layers along the exterior wall. In contrast, altering the first layer thickness of the prismatic layer on the exterior wall had a noticeable impact on the mass flow rate and the core temperature difference.

A possible explanation is the temperature gradient at the wall, which depends on the distance between the wall and the center of the first cell. Refinement of the first layer thickness therefore affects the resolution of the wall temperature gradient and consequently the predicted heat transfer. Since the buoyancy forces are directly influenced by the heat transfer at the wall, this may in turn affect the mass flow rate. In addition, refinement of the near-wall mesh may also

influence the predicted wall shear stresses and associated pressure losses, which can further affect the natural circulation flow rate. The observed trend in mass flow rate is consistent with the energy balance of the system, as a reduced mass flow rate requires a larger temperature rise to maintain the same thermal power.

Unlike the forced convection case, the choice of turbulence model has a more pronounced effect on the predicted results. Although the three models predict general recirculation in some parts of the geometry such as the entrance and exit of the core, the model predictions differ significantly with respect to detailed turbulence structures and bulk flow quantities.

The LRR predicts a lower mass flow rate and a higher temperature difference across the core compared to the other closure models. The reduced mass flow rate predicted by the LRR model can be attributed to its ability to capture anisotropic turbulence effects. An analysis of the Reynolds stress tensor field reveals that the stress components are relatively uniform throughout the domain, while significant spatial variations are present in the steam generator. Moreover, high velocities are observed in the steam generator in the LRR model. This indicates that turbulence in the steam generator is anisotropic, leading to momentum transport that is not fully aligned with the primary flow direction. As a result, the effective momentum transport in the flow direction is reduced, leading to a lower predicted mass flow rate. Since the porous resistance is imposed along a prescribed flow direction, this suggests an interaction between anisotropic turbulence and directionally dependent porous media modeling, which may contribute to the observed deviation.

The LRR model also exhibited numerical sensitivity that was more pronounced than in the forced convection case. Consequently, the simulation required substantial under-relaxation and conservative numerical settings to achieve convergence, while showing noticeable sensitivity to non-orthogonality corrections. This highlights the model's limiting applicability in the current case setup while also compromising some of the gain from the increased fidelity of the model. Despite this, it cannot be concluded that the LRR model predicts less or more accurate flow since no reference data is available and because other RSM models are not included in this study to confirm the deviating flow characteristics stemming from anisotropic turbulence effects.

In contrast, the $k-\varepsilon$ and $k-\omega$ SST models produce consistent and stable results with comparatively small differences in the predicted bulk flow performance. This similarity likely stems from the fact that both models rely on the eddy-viscosity assumption, treating turbulence as isotropic and resulting in comparable flow predictions.

Although the bulk flow parameters are similar, the models predict noticeably different turbulence structures in the system. In particular, the $k-\varepsilon$ model has less developed flow in the main pump leg section of the pipe compared to the $k-\omega$ SST model. Even though the $k-\omega$ SST exhibits a more coherent flow structure in this component, it cannot be concluded from these

results that this is the most accurate flow prediction, as $k-\varepsilon$ may also represent physically plausible flow features. However, the differences between these models are consistent with the known tendency for the $k-\varepsilon$ model to perform less reliably in flows with strong curvature and recirculation.

Despite having different abilities to model near-wall flow behavior, the eddy-viscosity models show similar system-level performance. This indicates that near-wall effects have a negligible influence on the bulk flow in the present case. Given their numerical robustness and lower computational cost, these models show greater applicability for system-level analysis of natural convection in this simplified reactor geometry. This also suggests that dominant flow features are governed by large-scale buoyancy rather than detailed turbulence structures.

It is further observed that the Boussinesq model predicts a higher mass flow rate and a lower temperature difference across the core. This over-predicted mass flow rate is expected and aligns well with the theory introduced in this study. As the reference temperature used in the model is outside the observed temperature range of the system, the buoyancy force is larger than what is predicted by the temperature-dependent property model, since the local temperature difference is the driving factor behind the buoyancy force. The increased mass flow in turn yields a lower temperature increase across the core due to the constant core heat transfer as previously discussed.

Similar flow development is seen in the tube section and at the tube exit of the Boussinesq model, while turbulence structures in the remaining regions of the domain show noticeable differences. As the mass flow rate in this model is greater, it is not possible to isolate these discrepancies to the treatment of density. The higher mass flow rate in the model may cause a change in the momentum distribution in the flow, which leads to altered vortex structures and altered recirculation zones in the open pools.

6 Multiphase steady-state natural convection

This section extends the single-phase natural convection case by introducing a gas phase above the liquid lead free surface. The objective is to evaluate how the gas region influences the predicted flow behavior and thermal performance under natural convection conditions. Since the multiphase solver is transient, the results are evaluated under quasi-steady-state conditions. As the primary goal is to assess the feasibility of implementing multiphase flow, mesh sensitivity analysis and turbulence model comparison are omitted in this chapter.

6.1 Multiphase natural convection setup

The multiphase simulations are based on the previously established single-phase natural convection setup. To represent the free surface more realistically, a cover gas region consisting of argon is introduced above the liquid lead.

The model is implemented using the *compressibleInterFoam* solver, which resolves the interface between the liquid lead and the cover gas using a VOF approach.

By using the *compressibleInterFoam* solver, temperature is implicitly used as the primary variable for energy. For this reason, the core heating requires units of K s^{-1} , which means that the power input (previously measured in Watts) is scaled with the reference specific heat capacity for the multiphase model. This yields a heat injection rate of 2544 K s^{-1} .

6.2 VOF model

In the multiphase case, the thermophysical properties are defined separately for each phase through dedicated files, namely *thermophysicalProperties.liquid* and *thermophysicalProperties.gas*, in addition to a general *thermophysicalProperties* file. This structure enables the liquid and gas phases to be assigned independent equations of state and transport properties.

The general *thermophysicalProperties* file defines the global multiphase properties used by the solver. These include the minimum pressure $p_{\min}$ and the surface tension coefficient σ . In this work, a constant value of $p_{\min} = 10000 \text{ Pa}$ and $\sigma = 0.42759 \text{ N m}^{-1}$ are applied. The surface tension coefficient is obtained from the correlation given in Equation 35.

The liquid phase uses the same thermophysical property definitions as those applied in the single-phase simulations. However, in the multiphase formulation these properties are explicitly assigned to the liquid phase through *thermophysicalProperties.liquid*. This ensures consistency between the single-phase and multiphase cases.

The argon gas phase is defined in *thermophysicalProperties.gas* and is modeled as a perfect gas, following the equation of state presented in Equation 60 [43].

$$\rho = \frac{p}{RT} \quad (60)$$

The thermophysical properties of argon are assumed constant and are evaluated at a representative temperature of 873 K. The specific heat capacity is obtained from temperature-dependent data provided by NIST [44], while the dynamic viscosity and thermal conductivity are taken from experimental data reported by Kestin et al. [45]. The tabulated values are converted to SI units where necessary and the Prandtl number is calculated from the corresponding transport properties. The resulting properties are summarized in Table 22.

Table 22: Thermophysical properties of argon used in the multiphase simulations.

Property	Value
M	39.9 g mol^{-1}
c_p	$520 \text{ J kg}^{-1} \text{ K}^{-1}$
μ	$5.06 \cdot 10^{-5} \text{ Pa s}$
Pr	0.664

6.3 Mesh generation

The mesh used for the VOF simulation uses the cell size and wall refinement that are deemed sufficient in the single-phase natural convection evaluation.

To mitigate numerical instability, a further vertical refinement of the liquid-gas interface is introduced to improve the stability of the VOF calculations. In this refined region, a cell size of 0.002 m is used, as seen in Figure 15.

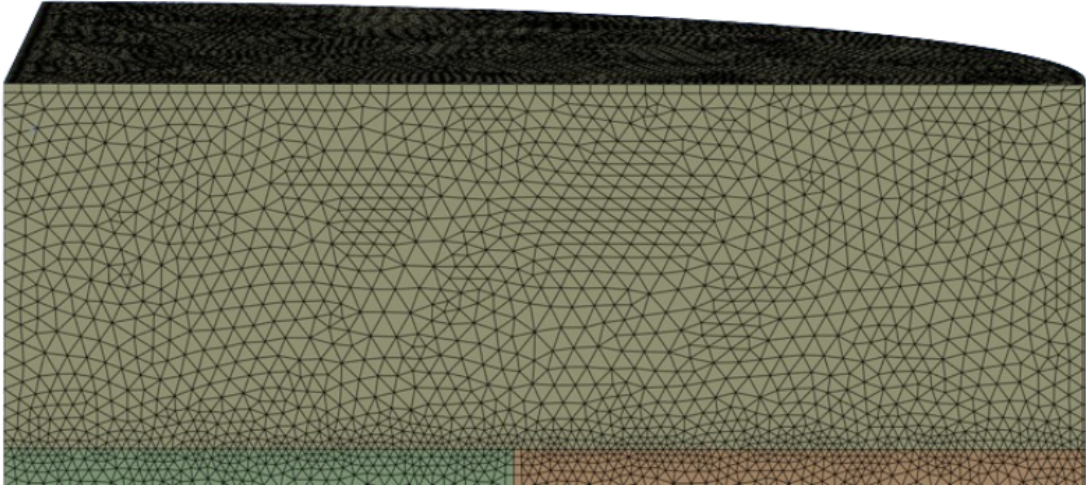


Figure 15: VOF mesh with interface refinement.

6.4 Boundary conditions

Boundary conditions are kept unchanged with a few exceptions. The first difference is that the patch name *free_surface* is removed from the model, since this boundary is now an interface between the lead and the argon. This means that no boundary conditions are specified here.

Moreover, an additional boundary condition is needed for the scalar phase fraction α . These conditions are summarized in Table 23 below.

Table 23: Boundary conditions for the phase fraction α .

Boundary	Type	Specification
walls.*	zeroGradient	$\nabla \alpha \cdot \mathbf{n} = 0$
air_wall	zeroGradient	$\nabla \alpha \cdot \mathbf{n} = 0$

6.5 Numerical setup

The numerical setup for the multiphase case utilizes the PIMPLE-algorithm, with the time derivative discretized using the first-order implicit *Euler* scheme.

Similarly to single-phase simulations, robustness is prioritized. Hence, the first-order *Gauss upwind* scheme is applied to the convective divergence terms, including the transport of momentum, energy, turbulence, and phase fraction. The viscous stress term found in the momentum equation is discretized using a *Gauss linear* scheme. The *cellMDLimited Gauss linear* scheme is used for the gradients to improve stability due to sensitivity to mesh quality.

To maintain stable transient behavior, a variable timestep is employed throughout the simulations. The timestep is automatically adjusted to satisfy both the global Courant number and the phase-fraction Courant number constraints. In the present study, these limits are set to 0.3 and 0.1, respectively, in order to ensure stable interface transport and avoid excessive numerical oscillations near the liquid-gas interface.

6.6 Multiphase flow performance

The performance of the multiphase simulations under steady-state natural convection is summarized in Table 24. The results are evaluated in terms of the mass flow rate and the temperature difference across the reactor core, which are used as indicators of the overall heat removal capability of the system.

Table 24: Performance metrics for the VOF simulations under steady-state natural convection.

Model	$\dot{m}$	ΔT_{core}
Multiphase	52.5 kg s ⁻¹	45.5 K

The velocity field obtained from the multiphase steady-state simulation is illustrated in Figure 16. Additional velocity field visualizations, including vertical and horizontal cross-sections, are provided in Appendix F.



Figure 16: Velocity field in the VOF simulation under steady-state natural convection conditions.

6.7 Impact of the gas phase and flow behavior

The multiphase simulation of the reactor system during natural convection shows system-level bulk flow performance similar to the single-phase setup. This indicates that the introduction of the gas pocket above the lead has a negligible influence on the overall system under the investigated conditions. This also suggests that OpenFOAM, using the *compressibleInterFoam* solver, is capable of capturing the key features of natural convection-driven flow involving two fluids with a large density difference.

Although the bulk flow parameters are similar, the velocity fields reveal different characteristics in the flow. This is particularly evident in critical sections such as the pipe, where the multiphase simulation predicts a more uniform velocity profile than the equivalent single-phase simulation. These differences are expected for several reasons. Firstly, the single-phase simulations represent a steady-state solution with a simplified slip-wall boundary condition. This means that the solid boundary can contribute to the backflow observed in the single-phase flow. In contrast, the transient multiphase case replaces the free surface boundary with a deformable liquid-gas interface which generates less recirculation in the tube section. Furthermore, a comparison is made between steady-state and transient simulations approaching quasi-steady-state conditions. Therefore, part of the observed differences may

originate from the transient nature of the simulations rather than the multiphase formulation itself. A transient single-phase simulation could help isolate these effects more clearly. This means that the transient simulation captures temporal variations, whereas the steady-state simulation represents a time-independent flow field.

Additionally, the VOF simulation uses the first-order upwind scheme for the phase-fraction transport, since this study prioritizes robustness over interface sharpness. This may introduce numerical diffusion in the interface between the fluids, which may affect local flow structures.

Despite the potential numerical diffusion caused by the discretization of the phase-fraction, the simulation predicts strong localized velocity gradients near the lead-gas interface. The behavior is expected due to the large density difference between the fluids. This is because the pressure and shear forces exerted on the low density gas by the liquid lead result in locally amplified accelerations in the gas phase, which yields high velocities at the interface.

However, these localized velocities are only present at the interface, which is consistent with the relatively large momentum diffusivity of the gas. The efficient redistribution of momentum in the gas phase causes the localized velocity gradients to rapidly dissipate away from the interface.

7 Conclusion

This study demonstrates that the implemented OpenFOAM model is capable of accurately reproducing the thermohydraulic performance of the system under forced convection conditions. The predicted mass flow rate, temperature difference, and pressure drop across the reactor core show strong agreement with reference data, indicating a correct implementation of porous media, momentum sources, and volumetric energy sources. However, a noticeable deviation remains in the predicted pressure drop across the steam generator, which introduces uncertainty in the predictive accuracy of the model for this particular component.

For the forced convection case, the flow is primarily controlled by the prescribed momentum source and the porous media. Despite differences in local flow structures, the turbulence model has a negligible effect on the predicted bulk flow parameters. This suggests that lower-order closure models are sufficient for predicting system-level performance under forced convection conditions. Moreover, the LRR model was found to be more numerically sensitive than the other closure models, as it required significant under-relaxation.

In contrast, the natural convection simulations reveal a significant dependence on geometric configuration and modeling assumptions. The flow is driven by buoyancy forces, which are strongly coupled to the reactor height and near-wall resolution. The results indicate that an increase in the reactor height is necessary to achieve a sufficient mass flow rate and thermal

distribution. In addition, the buoyancy-driven flow is sensitive to the thickness of the first layer of the exterior wall.

The natural convection case also demonstrates sensitivity to the turbulence modeling. The LRR model predicts a lower mass flow rate and higher temperature gradients compared to the two-equation models. Although this can be explained by its ability to predict anisotropic features of the turbulence, the model was found to be numerically sensitive and computationally demanding.

Unlike the LRR, the $k-\varepsilon$ and $k-\omega$ SST models provide consistent and stable predictions with significantly lower computational cost, making them more suitable for this application. Despite their consistent performance, their predictions cannot be considered more accurate than the LRR model for natural convection flow, since no experimental data are available.

The mesh sensitivity analysis for the forced convection setup established grid-independence, thereby strengthening the validity of the results. Additional mesh refinements in the natural circulation setup further confirmed that a finer mesh is not required, despite the change in flow regime. This indicates that the results are largely insensitive to mesh resolution, regardless of flow regime.

The Boussinesq model produced results consistent with the expected behavior of the Boussinesq approximation for natural convection flows. This suggests that the density approximation is sufficient for single-phase simulations of the reactor setup. While the local flow structure may vary due to an increased flow rate, the reduced numerical complexity of the model does not negatively compromise system-level performance. However, the over-predicted flow rate indicates that the model is sensitive to the selected reference temperature.

Finally, multiphase simulations demonstrate that the presence of a gas phase has a negligible impact on system-level performance during natural convection. The predicted mass flow rate and temperature difference across the core are consistent with the single-phase simulation, further indicating that the simplified free-surface representation is sufficient for system-level analysis. The results also show that the OpenFOAM solvers used in this study are capable of capturing buoyancy effects without causing nonphysical flow behavior.

Nevertheless, differences in local velocity fields are observed between the single-phase and multiphase simulations. These differences are expected due to the fundamental differences between the two modeling approaches, primarily arising from the treatment of the free surface and the use of a transient solver in the multiphase setup.

Overall, the results of this work demonstrate that the simplified CFD system model implemented in OpenFOAM is capable of reproducing the dominant thermohydraulic behavior of the lead-cooled reactor system. Furthermore, the findings may support the future development of passive safety systems and simplified reactor simulations in OpenFOAM.

8 Future work

Despite the demonstrated capability of the simplified reactor system in OpenFOAM, several aspects remain to be further investigated to improve physical fidelity and predictive capability.

One important area is the porous media representation of the steam generator, since the predicted pressure drop showed a consistent deviation from the reference simulations. Further investigations of the Darcy-Forchheimer coefficients and their sensitivity to the selected reference temperature could improve the accuracy of the model.

Additional work is also encouraged regarding turbulence modeling during natural convection. Since the LRR model predicted different flow behavior compared to the eddy-viscosity models, higher-fidelity turbulence models or alternative Reynolds Stress Models could provide further insight into anisotropic turbulence effects in the reactor system.

Moreover, the natural convection simulations demonstrated sensitivity to the near-wall treatment of the exterior wall. Further near-wall refinement studies could therefore improve the prediction of buoyancy-driven flow and heat transfer.

Further work is encouraged to investigate the sensitivity of the Boussinesq model to the selected reference temperature and its influence on buoyancy-driven flow. In addition, the applicability of the Boussinesq approximation in multiphase flow simulations could also be investigated.

The multiphase setup could also be further developed to better isolate the effects of the liquid-gas interface and transient flow behavior. This could include additional transient single-phase simulations or further investigations of interface discretization and mesh refinement near the free surface.

Another important extension would be transient simulations of pump shutdown scenarios to investigate the transition from forced to natural circulation. This would provide additional insight into the transient-hydraulic behavior and passive safety performance of the reactor system.

Finally, future work could focus on extending the simplified reactor model toward more realistic reactor geometries and operating conditions to further assess the applicability of OpenFOAM for passive safety analysis of lead-cooled reactor systems.

References

- [1] International Energy Agency. Net zero by 2050: A roadmap for the global energy sector. Technical report, International Energy Agency (IEA), October 2021. https://iea.blob.core.windows.net/assets/7ebafc81-74ed-412b-9c60-5cc32c8396e4/NetZeroBy2050-ARoadmapfortheGlobalEnergySector-SummaryforPolicyMakers_CORR.pdf, Revised version, October 2021 (4th revision), accessed 2026-04-23.
- [2] Jonas Kristiansen Nøland, Martin N. Hjelmeland, Christian Hartmann, Lina Bertling Tjernberg, and Magnus Korpås. Overview of small modular and advanced nuclear reactors and their role in the energy transition. *IEEE Transactions on Energy Conversion*, 2024. <https://kth.diva-portal.org/smash/get/diva2:1944371/FULLTEXT01.pdf>, accessed 2026-04-23.
- [3] World Nuclear Association. Small modular reactors, 2025. <https://www.world-nuclear.org/information-library/nuclear-power-reactors/small-modular-reactors/small-modular-reactors>, accessed 2026-02-25.
- [4] International Atomic Energy Agency. What are small modular reactors (smrs)?, 2023. <https://www.iaea.org/newscenter/news/what-are-small-modular-reactors-smrs>, accessed 2026-02-25.
- [5] Generation IV International Forum. Lead-cooled fast reactor – 2020 annual report. https://www.gen-4.org/gif/upload/docs/application/pdf/2021-05/lfr_gif_2020_annual_report.pdf, 2021. Accessed: 2026-02-25.
- [6] Generation IV International Forum. Lead fast reactors (lfr), 2020. <https://www.gen-4.org/generation-iv-criteria-and-technologies/lead-fast-reactors-lfr>, accessed 2026-04-20.
- [7] Blykalla AB. Technology, 2026. <https://www.blykalla.com/technology>, accessed 2026-03-04.
- [8] Jakob Oskarsson. Blykalla: Building advanced lead reactors. Presentation for Big Science Sweden, August 2025. <https://www.bigsciencesweden.se/media/bytozhed/blykalla-presentation-jakob-oskarsson.pdf>, accessed 2026-04-21.
- [9] OECD Nuclear Energy Agency, editor. *Handbook on Lead-bismuth Eutectic Alloy and Lead Properties, Materials Compatibility, Thermal-hydraulics and Technologies*. OECD Publishing, 2015.
- [10] Luciano Cinotti, Craig F. Smith, Carlo Artioli, Giacomo Grasso, and Giovanni Corsini. Lead-cooled fast reactor (lfr) design: Safety, neutronics, thermal hydraulics, structural

- mechanics, fuel, core, and plant design. In Dan Gabriel Cacuci, editor, *Handbook of Nuclear Engineering*, pages 2750–2840. Springer, 2010.
- [11] Yunus A. Çengel and Afshin J. Ghajar. Natural convection. In *Heat and Mass Transfer: Fundamentals and Applications*, pages 533–597. McGraw-Hill Education, 5 edition, 2015.
 - [12] H. K. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Pearson Education Limited, 2 edition, 2007.
 - [13] Antonio Barletta, Michele Celli, and D. Andrew S. Rees. On the use and misuse of the oberbeck–boussinesq approximation. *Physics*, 5(1):298–309, 2023.
 - [14] Matthias Vanderhaegen, Jan Vierendeels, and Baudouin Arien. Cfd analysis of the myrrha primary cooling system. *Nuclear Engineering and Design*, 241:775–784, 2011.
 - [15] OpenFOAM Foundation. Linear divergence scheme, 2023. <https://doc.openfoam.com/2312/tools/processing/numerics/schemes/divergence/rtm/linear/>, accessed 2026-05-06.
 - [16] OpenFOAM Foundation. Upwind divergence scheme, 2023. <https://doc.openfoam.com/2312/tools/processing/numerics/schemes/divergence/rtm/upwind/>, accessed 2026-05-06.
 - [17] OpenFOAM Foundation. Divergence schemes, 2023. <https://doc.openfoam.com/2312/tools/processing/numerics/schemes/divergence/>, accessed 2026-05-06.
 - [18] openCFD Ltd. Openfoam v2312 user guide: 6.2 numerical schemes, 2023. <https://www.openfoam.com/documentation/user-guide/6-solving/6.2-numerical-schemes>, accessed 2026-04-29.
 - [19] OpenCFD Ltd. Openfoam v2306 documentation: Interpolation schemes, 2023. <https://doc.openfoam.com/2306/tools/processing/numerics/schemes/interpolation/>, accessed 2026-04-29.
 - [20] OpenCFD Ltd. Openfoam v2312 documentation: Gradient schemes, 2023. <https://doc.openfoam.com/2312/tools/processing/numerics/schemes/gradient/>, accessed 2026-04-29.
 - [21] OpenCFD Ltd. Openfoam v2306 documentation: cellmdlimited gradient scheme, 2023. <https://doc.openfoam.com/2306/tools/processing/numerics/schemes/gradient/rtm/cellMDLimited/>, accessed 2026-04-29.
 - [22] Nickolay Y. Gnedin, Vadim A. Semenov, and Andrey V. Kravtsov. Enforcing the courant–friedrichs–lewy condition in explicitly conservative local time stepping schemes. *Journal of Computational Physics*, 359:230–240, 2018.

- [23] OpenCFD Ltd. Openfoam user guide: Time and data input/output control, 2023. <https://www.openfoam.com/documentation/user-guide/6-solving/6.1-time-and-data-inputoutput-control>, accessed 2026-05-06.
- [24] Christopher J. Greenshields and Henry G. Weller. The pimple algorithm, 2022. CFD Direct Notes on CFD: General Principles, <https://doc.cfd.direct/notes/cfd-general-principles/the-pimple-algorithm>, accessed 2026-03-18.
- [25] Mohammad Javad Ketabdar. Free surface flow simulation using vof method. In Ricardo Lopez-Ruiz, editor, *Numerical Simulation: From Brain Imaging to Turbulent Flows*, chapter 15. IntechOpen, London, 2015.
- [26] OpenFOAM Foundation. Openfoam: The open source cfd toolbox, 2026. <https://openfoam.org>, accessed 2026-04-16.
- [27] OpenCFD Ltd. Openfoam: Open source cfd software, 2026. <https://www.openfoam.com>, accessed 2026-04-16.
- [28] OpenCFD Ltd. buoyantsimplefoam.c source code (openfoam v2312). https://api.openfoam.com/2312/buoyantSimpleFoam_8C_source.html, 2023. Accessed: 2026-04-09.
- [29] OpenCFD Ltd. buoyantboussinesqsimplefoam.c source code (openfoam v2312). https://api.openfoam.com/2312/buoyantBoussinesqSimpleFoam_8C.html, 2023. Accessed: 2026-04-028.
- [30] OpenCFD Ltd. compressibleinterfoam.c source code (openfoam v2312), 2023. https://api.openfoam.com/2312/compressibleInterFoam_8C_source.html, accessed 2026-04-09.
- [31] OpenCFD Ltd. Openfoam v2306 documentation: Thermophysical models. <https://doc.openfoam.com/2306/tools/processing/models/thermophysical/>, Accessed: 2026-03-31, 2023.
- [32] OpenCFD Ltd. icopolynomial: Incompressible polynomial equation of state, 2023. <https://doc.openfoam.com/2306/tools/processing/models/thermophysical/equation-of-state/rtm/icoPolynomial/>, accessed 2026-04-23.
- [33] Siemens Digital Industries Software. *STAR-CCM+ User Guide*. Siemens, 2025. Version 2510.
- [34] OpenCFD Ltd. Openfoam v2312 source code: compressibleinterfoam pressure equation (peqn.h), 2023. https://api.openfoam.com/2312/multiphase_2compressibleInterFoam_2pEqn_8H_source.html, accessed 2026-04-23.

- [35] OpenCFD Ltd. Openfoam v2312 source code: buoyantsimplefoam pressure equation (peq.n.h), 2023. https://api.openfoam.com/2312/heatTransfer_2buoyantSimpleFoam_2pEqn_8H_source.html, accessed 2026-04-23.
- [36] OpenFOAM Foundation. semiimplicitsource fvoption, 2023. <https://doc.openfoam.com/2312/tools/processing/numerics/fvoptions/sources/rtm/semiImplicit/>, accessed 2026-05-05.
- [37] OpenFOAM Foundation. codedsource fvoption, 2023. <https://doc.openfoam.com/2306/tools/processing/numerics/fvoptions/sources/rtm/coded/>, accessed 2026-05-05.
- [38] OpenCFD Ltd. Openfoam v2312 api: Darcyforchheimer class reference, 2023. https://api.openfoam.com/2312/classFoam_1_1porosityModels_1_1DarcyForchheimer.html#details, accessed 2026-03-04.
- [39] OpenCFD Ltd. Openfoam v2312 user guide: A.4 standard boundary conditions, 2023. www.openfoam.com/documentation/user-guide/a-reference/a.4-standard-boundary-conditions, accessed 2026-03-30.
- [40] Fangqing Liu. A study of porous media modeling in OpenFOAM, 2016. Project report, Chalmers University of Technology, course: High Performance Computing for Computational Fluid Dynamics, https://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2016/FangqingLiu/openfoamFinal.pdf, accessed 2026-04-21.
- [41] Ismail B. Celik, Urmila Ghia, Patrick J. Roache, and Christopher J. Freitas. Procedure for estimation and reporting of uncertainty due to discretization in cfd applications. *Journal of Fluids Engineering*, 130(7):078001, July 2008.
- [42] Thomas-Raphael Teschner. How to manage uncertainty in cfd: The grid convergence index, n.d. <https://cfd.university/blog/how-to-manage-uncertainty-in-cfd-the-grid-convergence-index/>, accessed 2026-02-26.
- [43] OpenCFD Ltd. Openfoam v2306 documentation: perfectgas equation of state. <https://doc.openfoam.com/2306/tools/processing/models/thermophysical/equation-of-state/rtm/perfectGas/>, Accessed: 2026-05-20, 2023.
- [44] National Institute of Standards and Technology (NIST). Nist chemistry webbook: Argon (cas 7440-37-1), 2026. <https://webbook.nist.gov/cgi/cbook.cgi?ID=C7440371>, accessed 2026-04-15.
- [45] J. Kestin, S. T. Ro, and W. A. Wakeham. Viscosity of the noble gases in the temperature range 25–700°c. *Journal of Chemical Physics*, 57(8):4119–4124, 1972.

- [46] Brian Cabral and Leith Casey Leedom. Imaging vector fields using line integral convolution. *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 263–270, 1993. https://cgl.ethz.ch/teaching/scivis_common/Literature/CabralLeedom93.pdf, accessed 2026-04-28.

Appendix

A MATLAB code for polynomial approximation

```
1 %% Polynomial Fit for cp and mu
2 clear; close all; clc;
3
4 T_cp = linspace(601, 1300, 500); % Valid range cp: 601-1300 K
5 T_visc = linspace(601, 1500, 500); % Valid range mu: 601-1500 K
6
7 %% cp
8 % cp = 176.2 - 4.923e-2*T + 1.544e-5*T^2 - 1.524e6*T^-2 [Jkg^-1K^-1]
9 cp_original = 176.2 - 4.923e-2 .* T_cp + 1.544e-5 .* T_cp.^2 - 1.524e6 .* T_cp.^(-2)
10 ;
11 degree_cp = 5;
12 warning('off', 'MATLAB:polyfit:RepeatedPointsOrRescale');
13 p_cp = polyfit(T_cp, cp_original, degree_cp);
14 cp_fit = polyval(p_cp, T_cp);
15 c_cp = flip(p_cp);
16 err_cp = abs(cp_fit - cp_original) ./ cp_original * 100;
17
18 %% Dynamic Viscosity
19 % Pb: mu = 4.55e-4 * exp(1069/T) [Pas]
20 mu_original = 4.55e-4 * exp(1069 ./ T_visc);
21
22 degree_visc = 5;
23 p_visc = polyfit(T_visc, mu_original, degree_visc);
24 warning('on', 'MATLAB:polyfit:RepeatedPointsOrRescale');
25 mu_fit = polyval(p_visc, T_visc);
26 c_visc = flip(p_visc);
27 err_visc = abs(mu_fit - mu_original) ./ mu_original * 100;
28
29 %%
30 fprintf('cp_(degree_%d), valid_601-1300_K, max_error: %.5f%%\n', degree_cp, max(
31     err_cp));
32 fprintf('cpCoeffs<8>_cp'); fprintf('%0.6e', c_cp); fprintf('0_0);\n\n');
33 fprintf('mu_(degree_%d), valid_601-1500_K, max_error: %.5f%%\n', degree_visc, max(
34     err_visc));
35 fprintf('muCoeffs<8>_visc'); fprintf('%0.6e', c_visc); fprintf('0_0);\n\n');
36
37 %% Plot
38 figure('Position', [100 100 1000 800]);
39 sgtitle('Lead(Pb) Thermophysical property fits', 'FontSize', 14, 'FontWeight', '
40     bold');
```

```

41 plot(T_cp, cp_original, 'r-', 'LineWidth', 2.5, 'DisplayName', 'Correlation');
42 hold on;
43 plot(T_cp, cp_fit, 'b--', 'LineWidth', 2, 'DisplayName', sprintf('Polyfit_degree_%d',
    , degree_cp));
44 xlabel('T(K)'); ylabel('c_p(J/kg^{1}K^{1})');
45 title('Specific_heat_capacity_c_p');
46 legend('Location', 'best'); grid on;
47
48 % cp error
49 subplot(2,2,2);
50 plot(T_cp, err_cp, 'k-', 'LineWidth', 1.5);
51 xlabel('T(K)'); ylabel('Relative_error(%)');
52 title('Fit_error_c_p');
53 yline(5, 'r--', 'LineWidth', 1.5, 'Label', 'Table_uncertainty_5%');
54 grid on;
55 ylim([0, max(max(err_cp)*2, 0.01)]);
56
57 % Viscosity curves
58 subplot(2,2,3);
59 plot(T_visc, mu_original*1e3, 'r-', 'LineWidth', 2.5, 'DisplayName', 'Correlation');
60 hold on;
61 plot(T_visc, mu_fit*1e3, 'b--', 'LineWidth', 2, 'DisplayName', sprintf('Polyfit_
    degree_%d', degree_visc));
62 xlabel('T(K)'); ylabel('\mu(mPa.s)');
63 title('Dynamic_viscosity_\mu');
64 legend('Location', 'best'); grid on;
65
66 % Viscosity: error
67 subplot(2,2,4);
68 plot(T_visc, err_visc, 'k-', 'LineWidth', 1.5);
69 xlabel('T(K)'); ylabel('Relative_error(%)');
70 title('Fit_error_\mu');
71 yline(5, 'r--', 'LineWidth', 1.5, 'Label', 'Table_uncertainty_5%');
72 grid on;
73 ylim([0, max(max(err_visc)*2, 0.01)]);

```

B MATLAB code for mesh sensitivity study

```

1 V3 = 29.7921; V2 = 29.7989; V1 = 29.8;
2
3 n3 = 243685; n2 = 763574; n1 = 1776496;
4
5 phi3_m = 788.16; phi2_m = 804.40; phi1_m = 805.26;
6
7 phi3_core = 51.83; phi2_core = 54.36; phi1_core = 54.30;
8
9 phi3_sg = 44.13; phi2_sg = 41.5245; phi1_sg = 41.5242;

```

```

10
11 h3 = (V3/n3)^(1/3);
12 h2 = (V2/n2)^(1/3);
13 h1 = (V1/n1)^(1/3);
14
15 r32 = h3/h2;
16 r21 = h2/h1;
17
18 eps32_m = phi3_m - phi2_m;
19 eps21_m = phi2_m - phi1_m;
20
21 eps32_core = phi3_core - phi2_core;
22 eps21_core = phi2_core - phi1_core;
23
24 eps32_sg = phi3_sg - phi2_sg;
25 eps21_sg = phi2_sg - phi1_sg;
26
27 s_m = 1 * sign(eps32_m/eps21_m);
28 s_core = 1 * sign(eps32_core/eps21_core);
29 s_sg = 1 * sign(eps32_sg/eps21_sg);
30
31 % Startgissning (frsta termen i Eq 3)
32 % MASS FLOW
33 p_m = abs(log(abs(eps32_m/eps21_m))) / log(r21);
34
35 for k = 1:50
36     q = log((r21^p_m - s_m)/(r32^p_m - s_m));
37     p_new = abs(log(abs(eps32_m/eps21_m)) + q) / log(r21);
38
39     if abs(p_new - p_m) < 1e-10
40         break
41     end
42
43     p_m = p_new;
44 end
45
46 % CORE
47 p_core = abs(log(abs(eps32_core/eps21_core))) / log(r21);
48
49 for k = 1:50
50     q = log((r21^p_core - s_core)/(r32^p_core - s_core));
51     p_new = abs(log(abs(eps32_core/eps21_core)) + q) / log(r21);
52
53     if abs(p_new - p_core) < 1e-10
54         break
55     end
56
57     p_core = p_new;
58 end

```

```

59
60 % SG
61 p_sg = abs(log(abs(eps32_sg/eps21_sg))) / log(r21);
62
63 for k = 1:50
64     q = log((r21^p_sg - s_sg)/(r32^p_sg - s_sg));
65     p_new = abs(log(abs(eps32_sg/eps21_sg)) + q) / log(r21);
66
67     if abs(p_new - p_sg) < 1e-10
68         break
69     end
70
71     p_sg = p_new;
72 end
73
74 Fs = 1.25;
75 %Rkna Richardson, mass
76 phi_ext21_m = (r21^p_m * phi1_m - phi2_m) / (r21^p_m - 1);
77 ea21_m = abs((phi1_m - phi2_m)/phi1_m);
78 eext21_m = abs((phi_ext21_m - phi1_m) / phi_ext21_m);
79 GCI21_m = Fs * ea21_m / (r21^p_m - 1)
80
81 phi_ext32_m = (r32^p_m * phi2_m - phi3_m) / (r32^p_m - 1);
82 ea32_m = abs((phi2_m - phi3_m)/phi2_m);
83 eext32_m = abs((phi_ext32_m - phi2_m) / phi_ext32_m);
84 GCI32_m = Fs * ea32_m / (r32^p_m - 1)
85
86 asymp_m = GCI32_m / (GCI21_m * r21^p_m);
87
88 %Rkna Richardson, core
89 phi_ext21_core = (r21^p_core * phi1_core - phi2_core) / (r21^p_core - 1);
90 ea21_core = abs((phi1_core - phi2_core)/phi1_core);
91 eext21_core = abs((phi_ext21_core - phi1_core) / phi_ext21_core);
92 GCI21_core = Fs * ea21_core / (r21^p_core - 1)
93
94 phi_ext32_core = (r32^p_core * phi2_core - phi3_core) / (r32^p_core - 1);
95 ea32_core = abs((phi2_core - phi3_core)/phi2_core);
96 eext32_core = abs((phi_ext32_core - phi2_core) / phi_ext32_core);
97 GCI32_core = Fs * ea32_core / (r32^p_core - 1)
98
99 asymp_core = GCI32_core / (GCI21_core * r21^p_core);
100
101 %Rkna Richardson, sg
102 phi_ext21_sg = (r21^p_sg * phi1_sg - phi2_sg) / (r21^p_sg - 1);
103 ea21_sg = abs((phi1_sg - phi2_sg)/phi1_sg);
104 eext21_sg = abs((phi_ext21_sg - phi1_sg) / phi_ext21_sg);
105 GCI21_sg = Fs * ea21_sg / (r21^p_sg - 1)
106
107 phi_ext32_sg = (r32^p_sg * phi2_sg - phi3_sg) / (r32^p_sg - 1);

```

```

108 ea32_sg = abs((phi2_sg - phi3_sg)/phi2_sg);
109 eext32_sg = abs((phi_ext32_sg - phi2_sg) / phi_ext32_sg);
110 GCI32_sg = Fs * ea32_sg / (r32^p_sg - 1)
111
112 asymp_sg = GCI32_sg / (GCI21_sg * r21^p_sg);

```

C Velocity fields for different turbulence models under forced convection

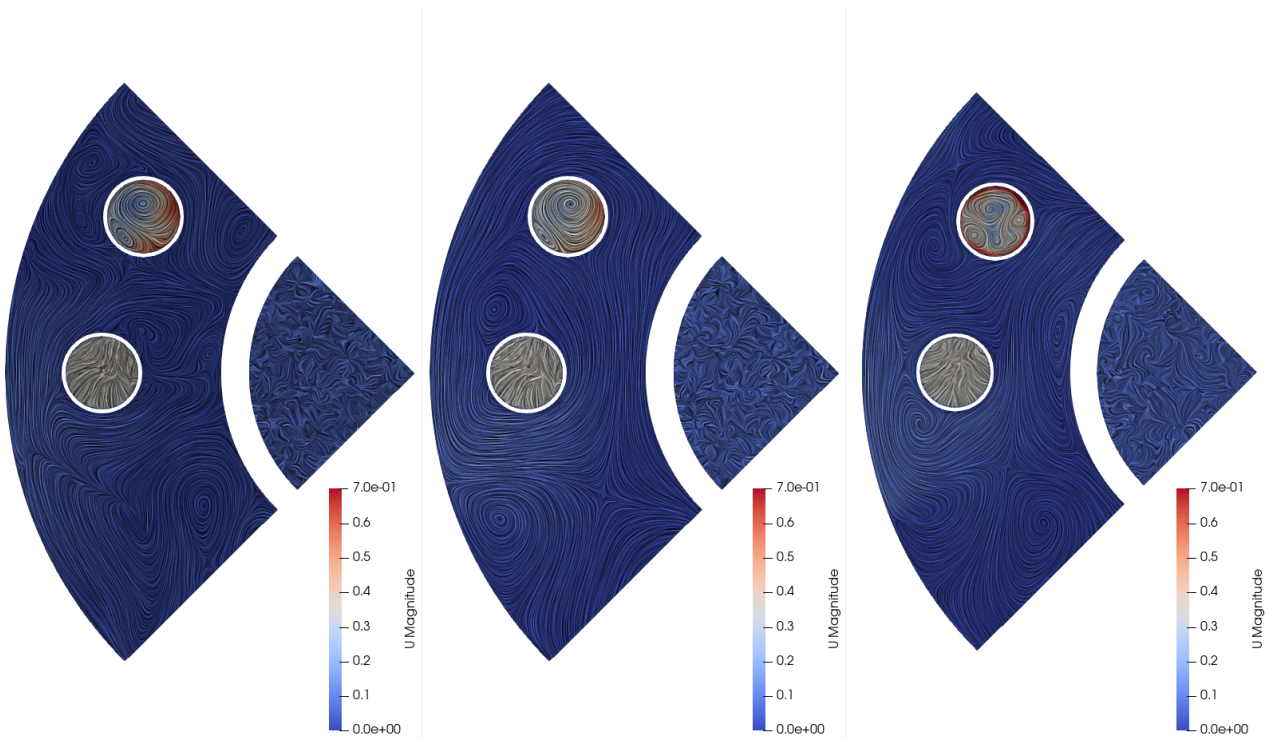


Figure 17: Velocity field in a horizontal cross-section for different turbulence models. From left to right: LRR, $k-\epsilon$, and $k-\omega$ SST.

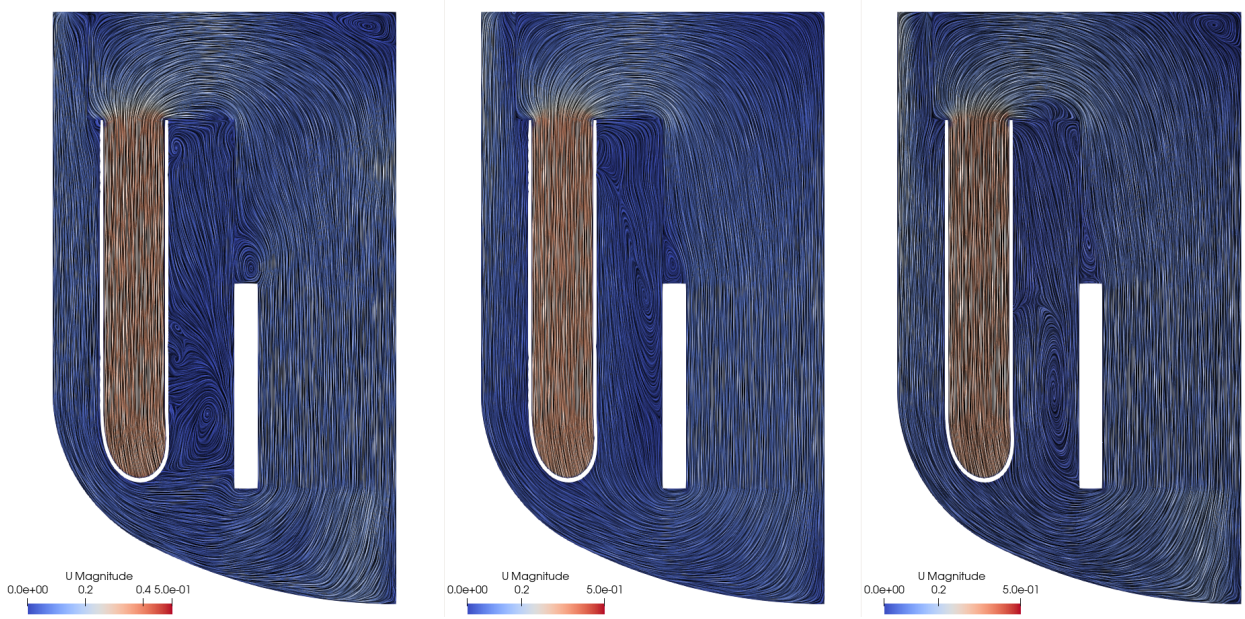


Figure 18: Velocity field in a vertical cross-section for different turbulence models. From left to right: LRR, $k-\epsilon$, and $k-\omega$ SST.

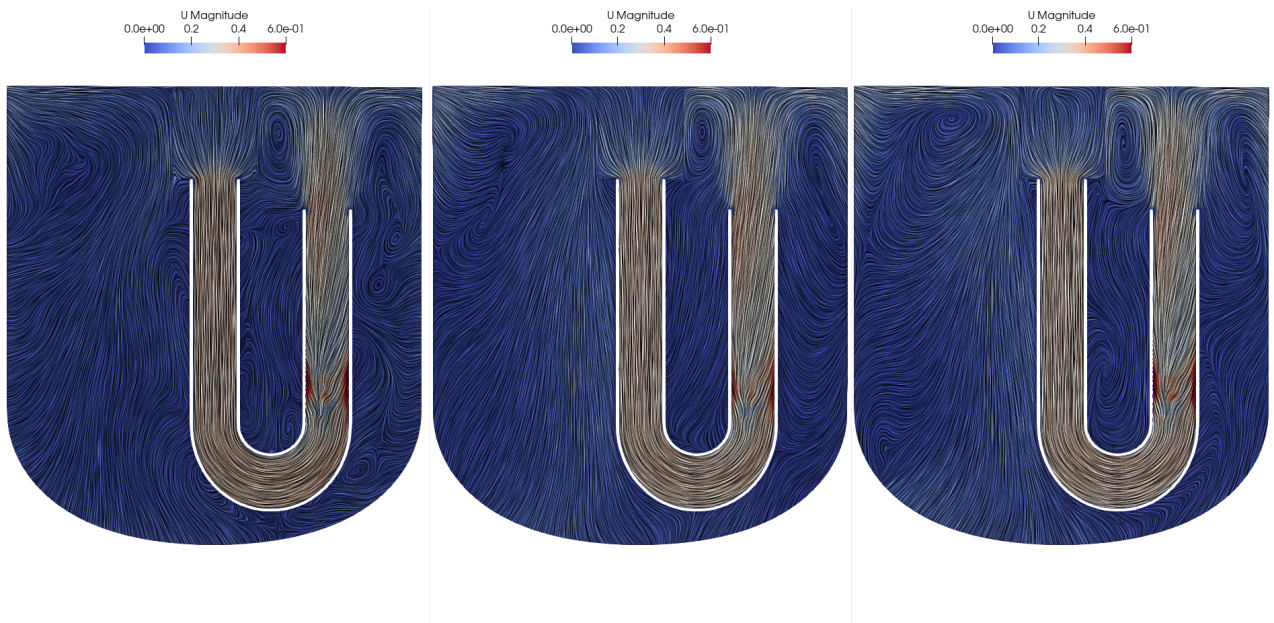


Figure 19: Velocity field in a vertical cross-section for different turbulence models. From left to right: LRR, $k-\epsilon$, and $k-\omega$ SST.

D Velocity fields for different turbulence models under natural convection

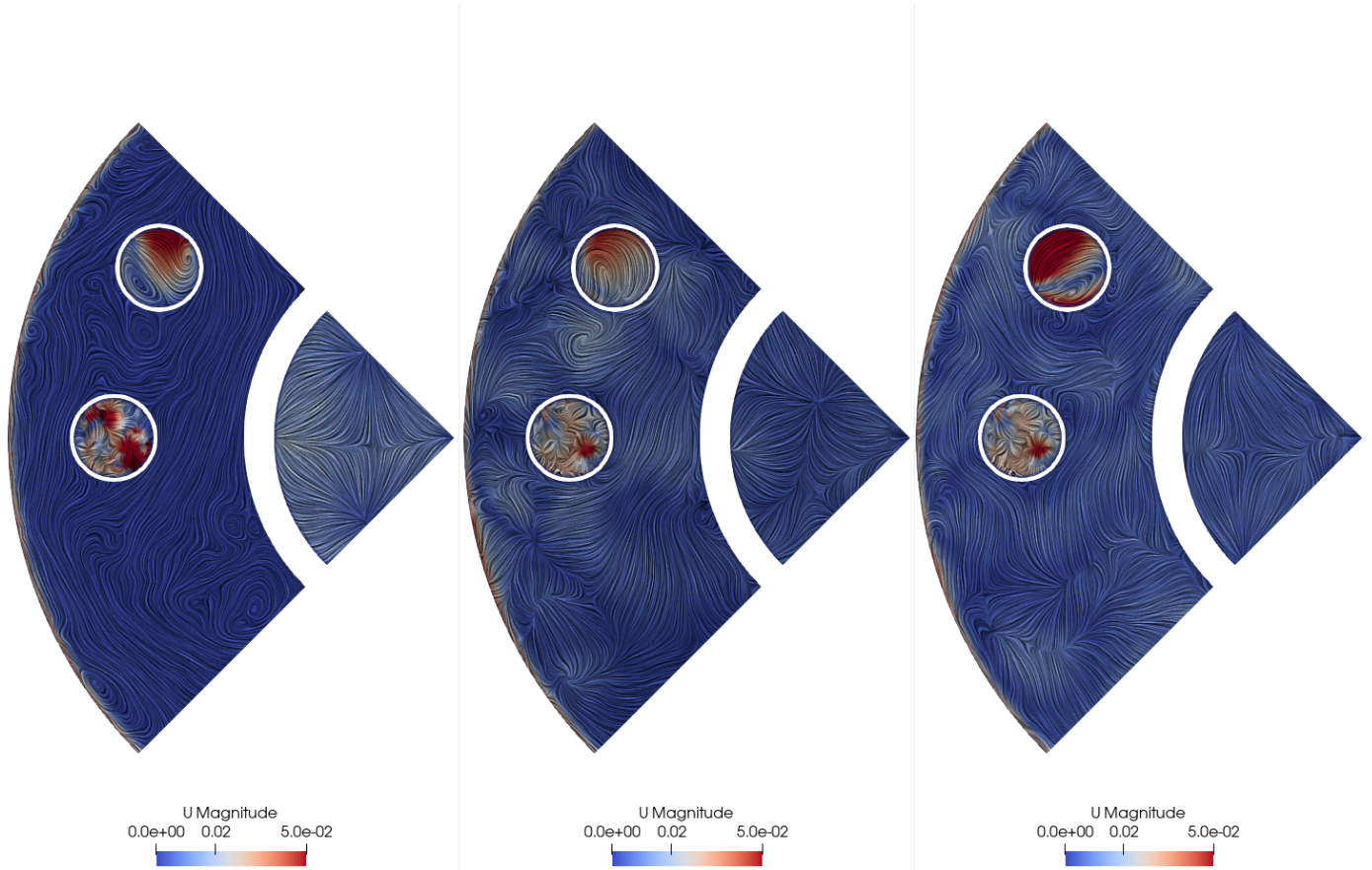


Figure 20: Velocity field in a horizontal cross-section for different turbulence models. From left to right: LRR, $k-\epsilon$, and $k-\omega$ SST.

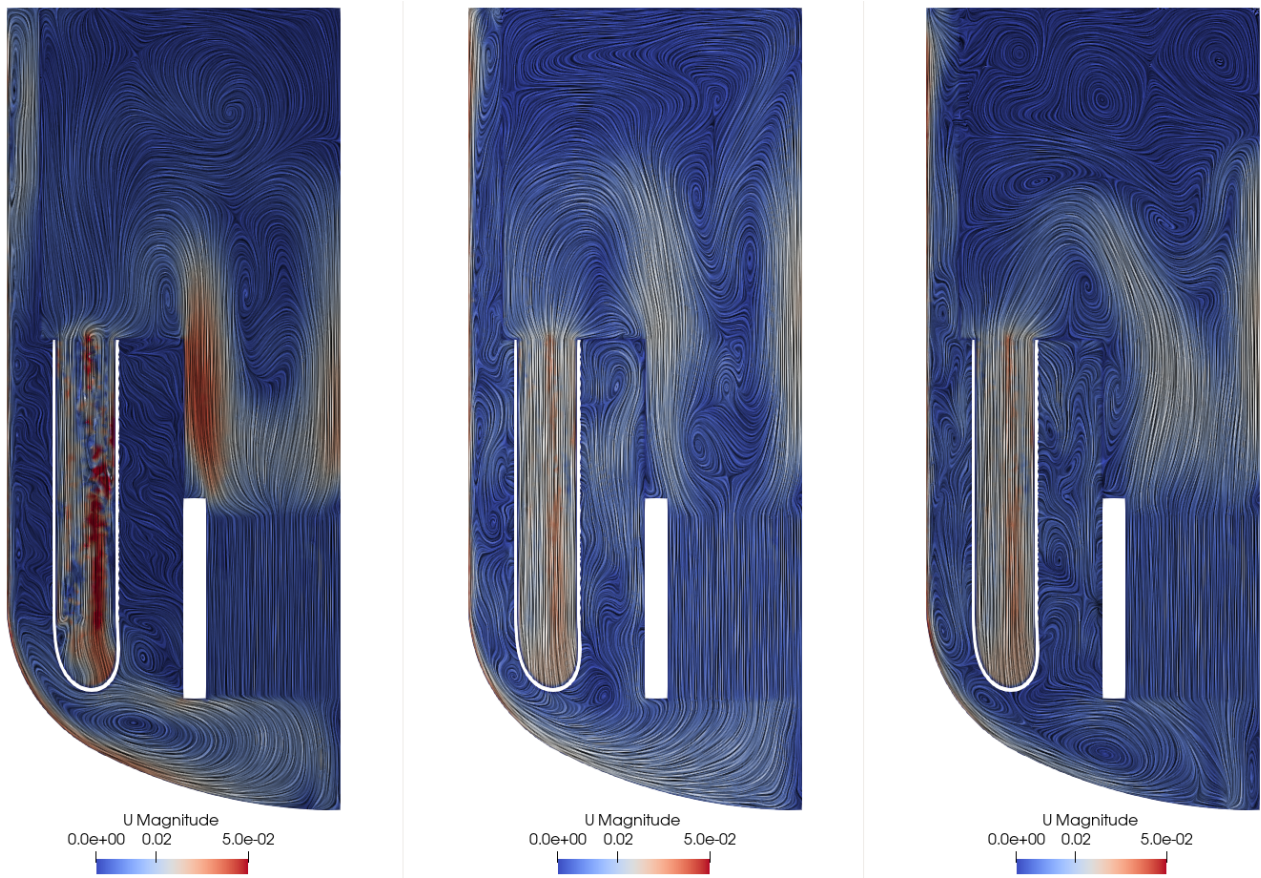


Figure 21: Velocity field in a vertical cross-section for different turbulence models. From left to right: LRR, $k-\epsilon$, and $k-\omega$ SST.

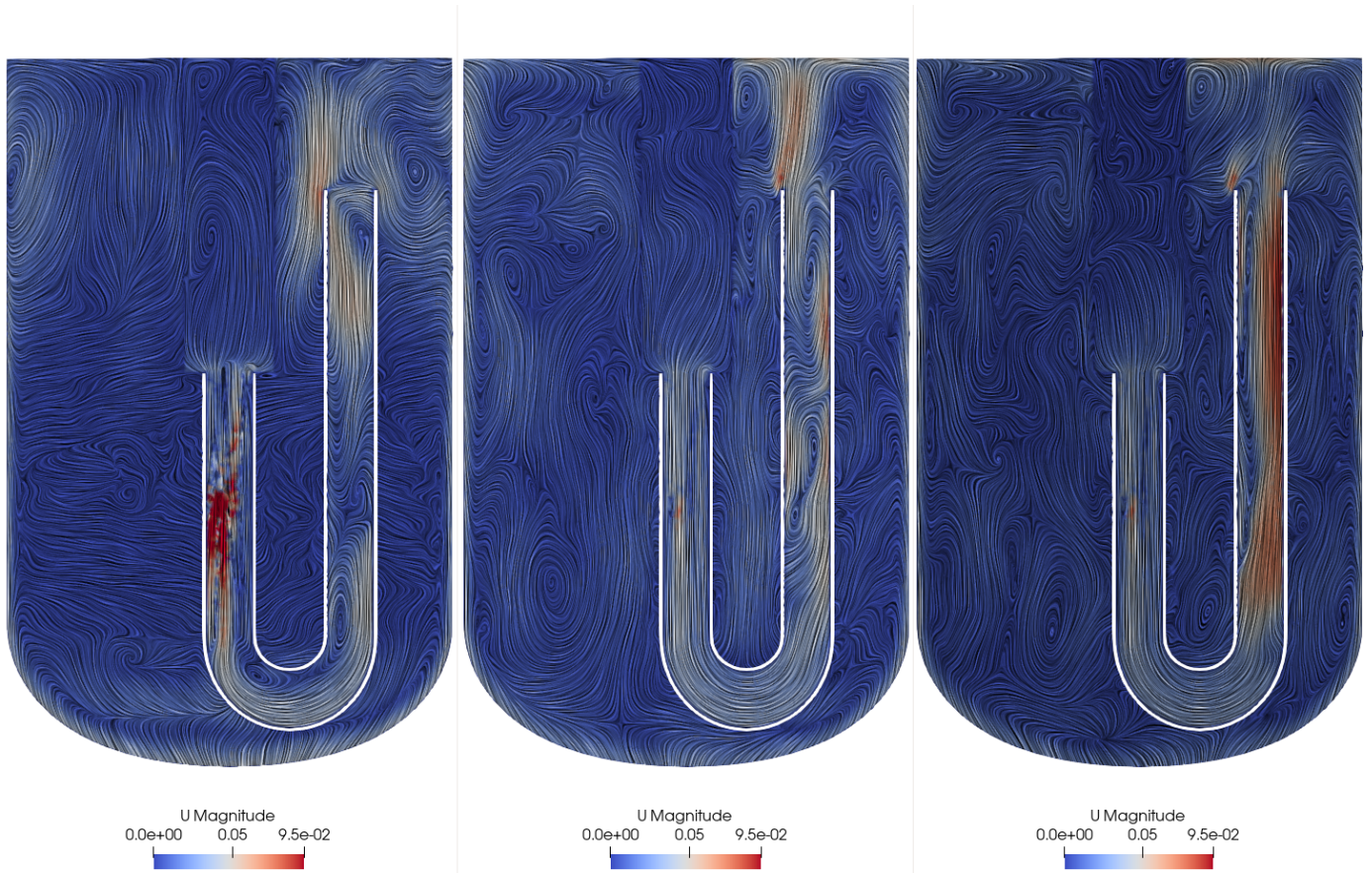


Figure 22: Velocity field in a vertical cross-section for different turbulence models. From left to right: LRR, $k-\epsilon$, and $k-\omega$ SST.

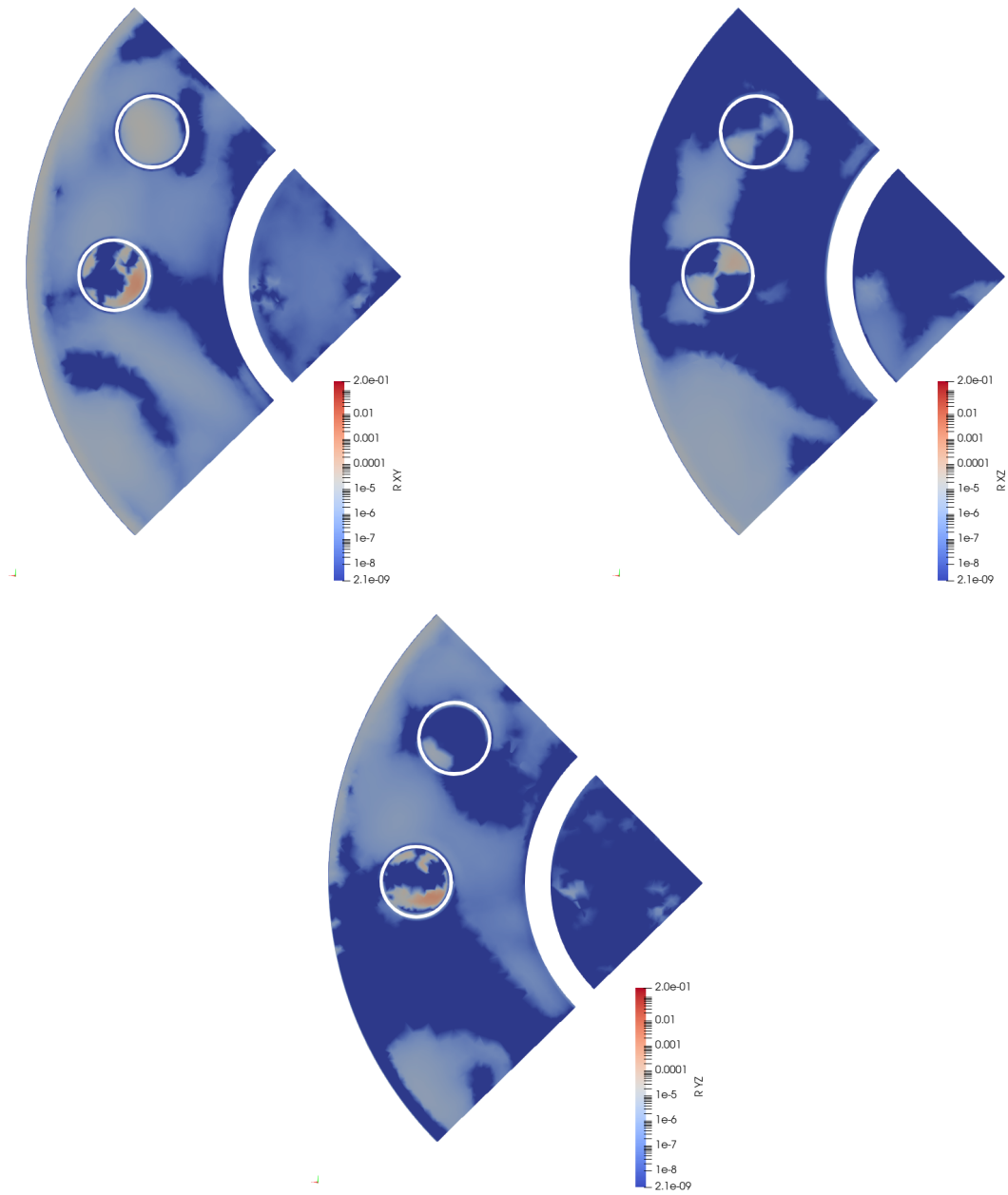


Figure 23: Shear components of the Reynolds stress tensor in horizontal cross-section in the LRR model.

E Velocity fields for the Boussinesq model

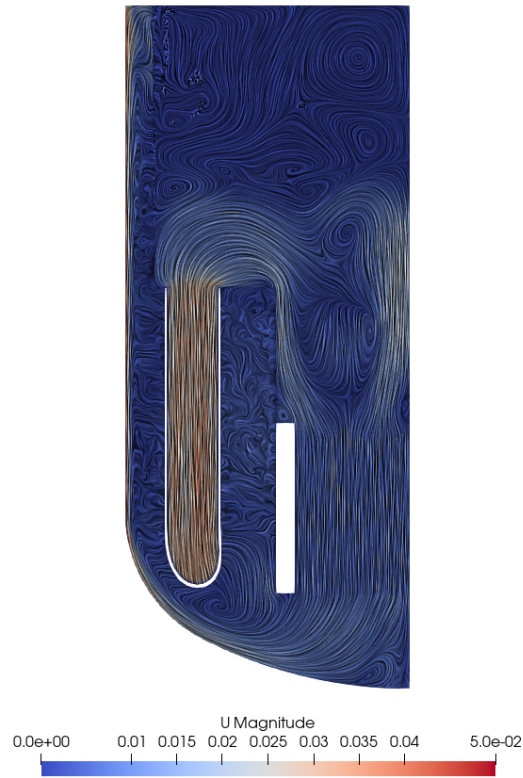


Figure 24: Velocity field in a vertical cross-section for the Boussinesq model.



Figure 25: Velocity field in a vertical cross-section for the Boussinesq model.

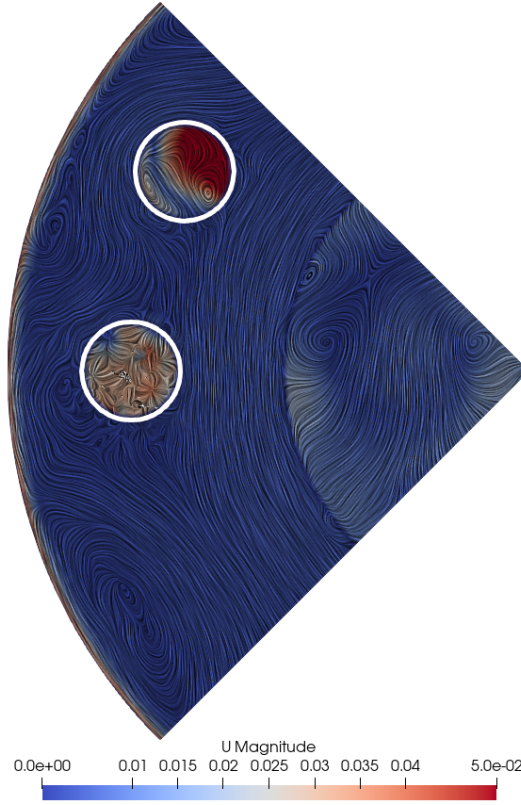


Figure 26: Velocity field in a horizontal cross-section for the Boussinesq model.

F Velocity fields from VOF simulations under natural convection

Surface LIC is a visualization technique used to represent vector fields by convolving noise along streamlines. This produces a texture that highlights the local flow direction and structure within the field. [46]

The dotted appearance observed in some regions of the horizontal cross-section is not a numerical issue, but a consequence of how the method works. Surface LIC only represents the velocity components tangent to the slice plane. For a slice in the XZ-plane, only U_x and U_z contribute:

$$U_{xz} = \sqrt{U_x^2 + U_z^2} \quad (61)$$

If the flow is primarily in the Y-direction, the velocity is mostly normal to the slice. In that case, the in-plane velocity becomes very small, and the LIC algorithm cannot form continuous lines, resulting in a dotted pattern.

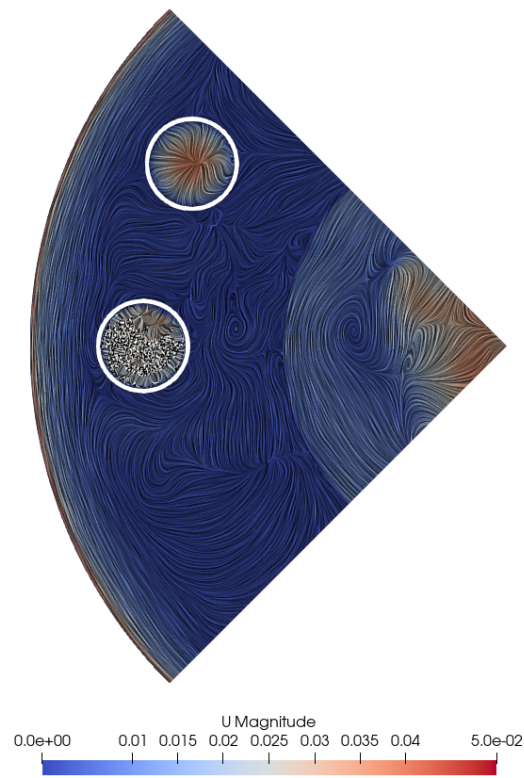


Figure 27: Velocity field in a horizontal cross-section in the VOF simulation under steady-state natural convection conditions.

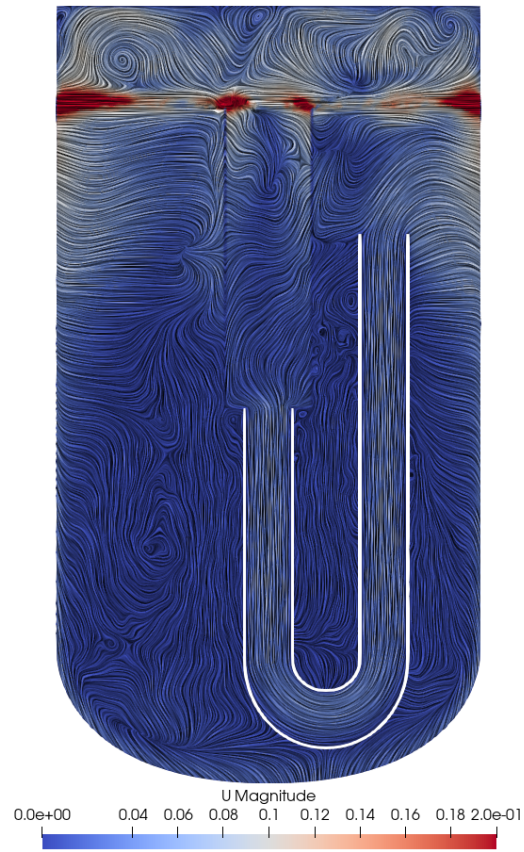


Figure 28: Velocity field in a vertical cross-section in the VOF simulation under steady-state natural convection conditions.

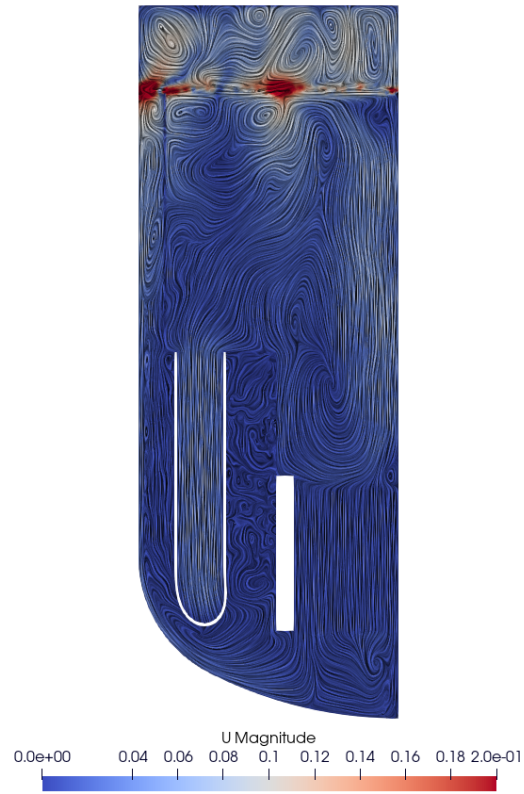


Figure 29: Velocity field in a vertical cross-section in the VOF simulation under steady-state natural convection conditions.