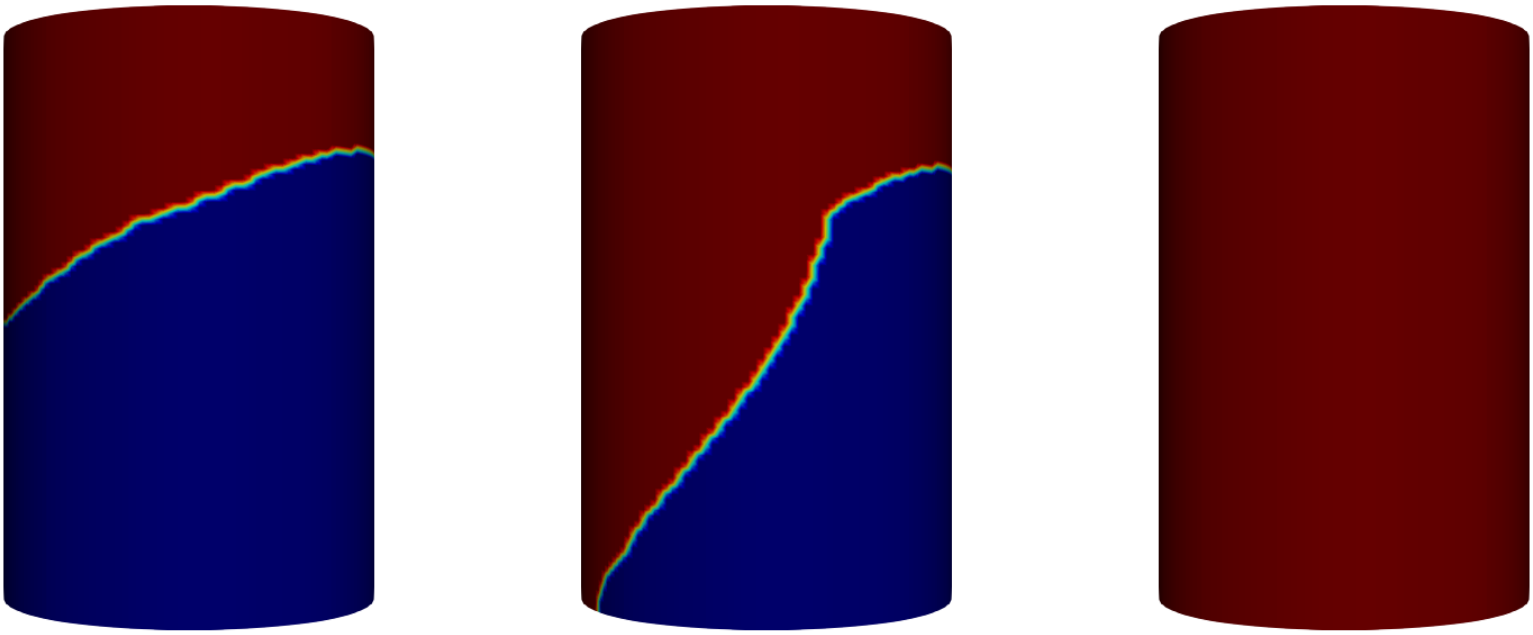
Investigating the Effect of Freeze Plug Inclination Angle on Melting Behaviour with the Linearised Enthalpy Method

C.Z. van Dijk



TU Delft

Investigating the Effect of Freeze Plug Inclination Angle on Melting Behaviour with the Linearised Enthalpy Method

by

C.Z. van Dijk

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Wednesday 25 January 2023 at 13:30 CET.

Student number:	4570324
Project duration:	28 February 2022 – 18 January 2023
Daily supervisor:	ir. B.J. Kaaks
Thesis committee:	dr.ir. D. Lathouwers (supervisor)
	dr.ir. M. Rohde
	dr. S.R. de Roode

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Abstract

The Molten Salt Fast Reactor (MSFR) is a next generation nuclear reactor. The MSFR may play an important role in combatting global warming whilst ensuring energy safety considering fossil fuels' finiteness. A key safety mechanism for the MSFR is the freeze plug, a valve underneath the reactor core designed to melt during reactor malfunction, letting the fuel salt drain into an emergency draining system.

In this thesis, the effect of the inclination angle of the freeze plug on its melting behaviour was investigated. This was done via the Linearised Enthalpy Method for phase change in *OpenFOAM 8*, a Finite Volume Method software package. First, two benchmark cases - one of inclined gallium melting and one of a heater copper fin in air - were performed. These showed reasonable agreement with established results.

A cylindrical freeze plug consisting of LiF-ThF_4 salt was defined, with copper and hastelloy annuli around it to simulate the cooling and the draining pipe, respectively. Steady-state simulations were performed for inclination angles of $\theta = 0°, 15°, 30°, 45°, 60°$, and $90°$. Via simulations on meshes with 40,500 and 1.1 million cells, it was shown that the freeze plug did not form at $\theta = 60°$ and $90°$. For the other four inclination angles, the resulting freeze plug was found to be mesh convergent via a third mesh of 3.6 million cells.

Then, a transient simulation mimicking reactor malfunction was performed for the inclination angles $0° \leq \theta \leq 45°$. It was found that the start and end opening times - the times where the melt front first and last reached the freeze plug's bottom - were reduced by 74% and 52%, respectively for $\theta = 45°$ with respect to $\theta = 0°$. The other two θ's also showed lower opening times compared to $\theta = 0°$. The inclination angle thus was found to have a significant effect on the freeze plug's melting behaviour, shortening the opening times.

Physical improvements for the model include the incorporation of molten salt leakage after start opening time, and settling. Computational improvements are also possible, such as the refinement of the mesh near the salt boundary or adaptive mesh refinement near the solid-liquid boundary. Symmetry can be exploited to speed up results.

Acknowledgements

This thesis would not have been possible without the help and support of other people. Firstly, I would like to thank Bouke. He often took the term ‘daily supervisor’ very literally, considering the frequency of the useful small meetings he and I had during the most hectic time period of my project. I would also like to express my gratitude to Danny, whose feedback and advice during the project was very useful. A note of gratitude also goes to Martin, who adopted me into his group for the Monday morning meetings which turned out to be very useful.

The next group of people I would like to mention are my fellow master students of the NMRP and MP&T groups. Throughout the duration of the projects students came and left, but Daniel, Daphne, Emma, Erik, Floris, Luuk, Maas, Mees, Nick, Pascal, Ruben, Sylvie, Thomas, and Thorben all left a positive influence on my motivation. This was especially apparent during the summer period, where walking alone over the campus turned out to be less fun than with other people, and nobody came up with the idea to drink orange juice with chocolate milk.

Of course, people outside the reactor institute exist, some of which helped me with my project as well. Firstly Ilse, who supported me from both Leiden and Stockholm and multiple times helped me through some hard moments. With my former housemates Chris and Lars I could always talk about my and their theses, but also play a game of The Crew or enjoy some culinary feats during dinner time. More thanks go to my friends: Bram, Jeroen, Julian, Kees, Matthias, Rick, Richard, Robin, Thorben (once again), Willem.

I would also like to thank my family: Mamusia, Papa, Natalia, Piotr, Oma, and Babusia. A special note of thanks goes to my Opa, who always supported me in my educational and other endeavours, but who passed away in September. Opa, thank you for helping me become the person I am today.

Delft, January 2023

Nomenclature

Field values		
ℓ	Liquid fraction	[−]
$\boldsymbol{u}$	Velocity	$\mathrm{m\,s^{-1}}$
ρ	Mass density	$\mathrm{kg\,m^{-3}}$
$\underline{\underline{\tau}}$	Stress tensor	Pa
φ'''	Heat flux	$\mathrm{W\,m^{-2}}$
H	Volumetric enthalpy	$\mathrm{J\,m^{-3}}$
p	Pressure	Pa
T	Temperature	K
Thermodynamic material constants		
α	Heat diffusivity	$\mathrm{m^2\,s^{-1}}$
β	Thermal expansion coefficient	$\mathrm{K^{-1}}$
λ	Heat conductivity	$\mathrm{W\,m^{-1}\,K^{-1}}$
μ	Dynamic viscosity	Pa s
ν	Kinematic viscosity	$\mathrm{m^2\,s^{-1}}$
ρ_0	Reference mass density	$\mathrm{kg\,m^{-3}}$
c_p	Specific heat capacity	$\mathrm{J\,kg^{-1}\,K^{-1}}$
L	Specific latent heat of fusion	$\mathrm{J\,kg^{-1}}$
T_0	Reference temperature	K
Physical domain parameters		
θ	Inclination angle	[−]
h_{min}	Minimum freeze plug height	m
l	Length	m
w	Width	m
Dimensionless numbers		
Co	Courant number	[−]
Nu	Nusselt number	[−]
Pe	Péclet number	[−]
Pr	Prandtl number	[−]
Ra	Rayleigh number	[−]
Re	Reynolds number	[−]
CFD-related symbols		

Δt	Time step length	s
Δt_0	Previous time step length	s
Δx	Cell size	[−]
ϵ	Constant used for calculation of Darcy source term	[−]
γ	Weighing factor for temporal discretisation	[−]
$\boldsymbol{A}$	Darcy source term	kg m^{-2} s^{-2}
$\boldsymbol{H}$	Non-diagonal Navier-Stokes coefficient matrix	kg m^{-2} s^{-2}
$\underline{\underline{A}}$	Diagonal Navier-Stokes coefficient matrix	kg m^{-3} s^{-1}
$\underline{\underline{M}}$	Overall Navier-Stokes coefficient matrix	kg m^{-3} s^{-1}
ξ	Limiter	[−]
D	Darcy constant	kg m^{-3} s^{-1}
f	Weighing factor for weighted average	[−]
N	Amount	[−]
r	Over- or underrelaxation parameter	[−]
V	Volume	m^3
x	x coordinate	m

Other symbols

$\boldsymbol{g}$	Gravity	m s^{-2}
A	Area	m^2
h	Heat transfer coefficient	W m^{-2} K^{-1}
l_c	Characteristic length	m
P	Perimeter	m

Subscripts

$(\dots)_l$	Liquid
$(\dots)_m$	Melt or melting
$(\dots)_s$	Solid
$(\dots)_{app}$	Apparent
$(\dots)_{cell}$	Cell
$(\dots)_{conv}$	(Due to) convection
$(\dots)_c$	Cold
$(\dots)_{end}$	Ending
$(\dots)_h$	Hot
$(\dots)_{init}$	Initial
$(\dots)_{max}$	Maximum
$(\dots)_{min}$	Minimum
$(\dots)_{open}$	Opening
$(\dots)_{sens}$	Sensible

$(\ldots)_{start}$ Starting

$(\ldots)_{top}$ At the top

Superscripts

$(\ldots)^{o}$ Old, previous

$(\ldots)^{oo}$ Second old, one before previous

Contents

1 Introduction

1.1. The need for renewable energy

It was a pleasure to burn - with these words, Bradbury (1953) opened his famous novel *Fahrenheit 451*. Seventy years after the book's publication, in the non-fictional world, it is not books we burn. Rather, fossil fuels such as oil, natural gas, and coal are burnt to meet the ever increasing global energy consumption.

However, despite it being a pleasant and easy way to generate energy, burning fossil fuels is unsustainable: they are finite, and burning them releases carbon dioxide which raises the temperature of the Earth's atmosphere through enhancing the greenhouse effect. The Intergovernmental Panel on Climate Change (IPCC) - an organisation of the United Nations - has set a limit of 1.5 °C warming with respect to pre-industrial revolution levels (Intergovernmental Panel on Climate Change, nd), to minimise the already-present effects such as rising sea levels, floods, and heat waves.

The IPCC goal is mainly achieved by using clean energy sources such as solar, wind and hydroelectric power. However, they have disadvantages, such as the limited reliability, as they are dependent on external factors (the Sun, wind or water flow). On the other hand, nuclear energy is a reliable energy source, as a nuclear power plant can run for years uninterrupted without much greenhouse gas emission. This makes it a good supplement to or alternative for the previously mentioned renewable energy sources.

In a 2018 report, the IPCC considered 90 pathways to achieve the goal of 1.5 °C warming. On average, global nuclear power would have to generate 1160 GW of electricity in 2050 (Intergovernmental Panel on Climate Change, 2018), almost thrice as much as the 394 GW of nuclear power in 2020 (Organisation for Economic Co-operation and Development, 2021).

1.2. Nuclear energy

Nuclear energy technology has been around since the 1930s, when Enrico Fermi discovered that neutrons were able to cause nuclear fission. This led to the first nuclear chain reaction in 1942, and the first nuclear power plant ten years later. Since then, many 'traditional' light water reactors (LWRs) have been built (World Nuclear Association, 2020). However, they have two major disadvantages. Firstly, uranium is a finite energy source, with enough being present at Earth for 90 years of present demand (World Nuclear Association, 2022c). Furthermore, the public is anxious about meltdowns, such as in Chernobyl in 1986 (World Nuclear Association, 2022a) and Fukushima in 2011 (World Nuclear Association, 2022b).

In 2001, the Generation IV Forum (GIF) was formed to find a solution for these concerns. The GIF is a collaboration between fourteen countries and the EU. GIF's goal is to select a nuclear reactor design which is better than the LWR in terms of sustainability, economics, safety and reliability, and proliferation resistance and physical protection (GEN IV International Forum, 2022). The GIF has selected six nuclear reactor designs that could potentially fulfil these criteria, one of which is the Molten Salt Reactor (MSR).

1.3. The Molten Salt Fast Reactor

1.3.1. The Molten Salt Reactor Experiment

In the 1960s, the first MSR was developed in the Oak Ridge National Laboratory (ORNL) in the United States (Oak Ridge National Laboratory, 2016). For the Molten Salt Reactor Experiment (MSRE), a 7.5 MW_{th} reactor was constructed in 1964, it went critical in 1965 and operated until 1969. In that year, the MSRE was stopped due to lack of government funding, which resulted from the focus on other technologies such as the LWR.

The MSRE reactor consisted of a core with molten LiF-BeF_2-ZrF_4-UF_4 inside; about 1 mole-% was UF_4. The salt mixture acted as both the fuel and the primary coolant. A secondary coolant was LiF-BeF_2. A graphite core had the role of moderator. The reactor operated at 650 °C.

1.3.2. Working of the Molten Salt Fast Reactor

After 30 years of inactivity, at the beginning of the 21st century interest in the MSR arose once again after the GIF chose it as one of the possible next-generation nuclear reactor designs. One of the subtypes of the MSR is the Molten Salt Fast Reactor (MSFR). This MSR consists of a core with molten salt inside. The salt is a mixture of different ions, including the nuclear fuel, LiF-ThF_4-UF_4. The MSFR is a fast-spectrum breeder reactor[1] (SAMOSAFER, nda), allowing the reduction of reprocessing requirements and a better reactor breeding ratio.

A schematic overview of the MSFR is given in Figure 1.1. In the reactor core (labelled 'Reactor' in the figure), fission of the molten salt takes place, generating heat. The molten salt is pumped through the heat exchangers, transferring part of its heat to water in a secondary loop. The water vapourises and the resulting steam powers the turbine. This in turn powers the generator, resulting in electrical power.

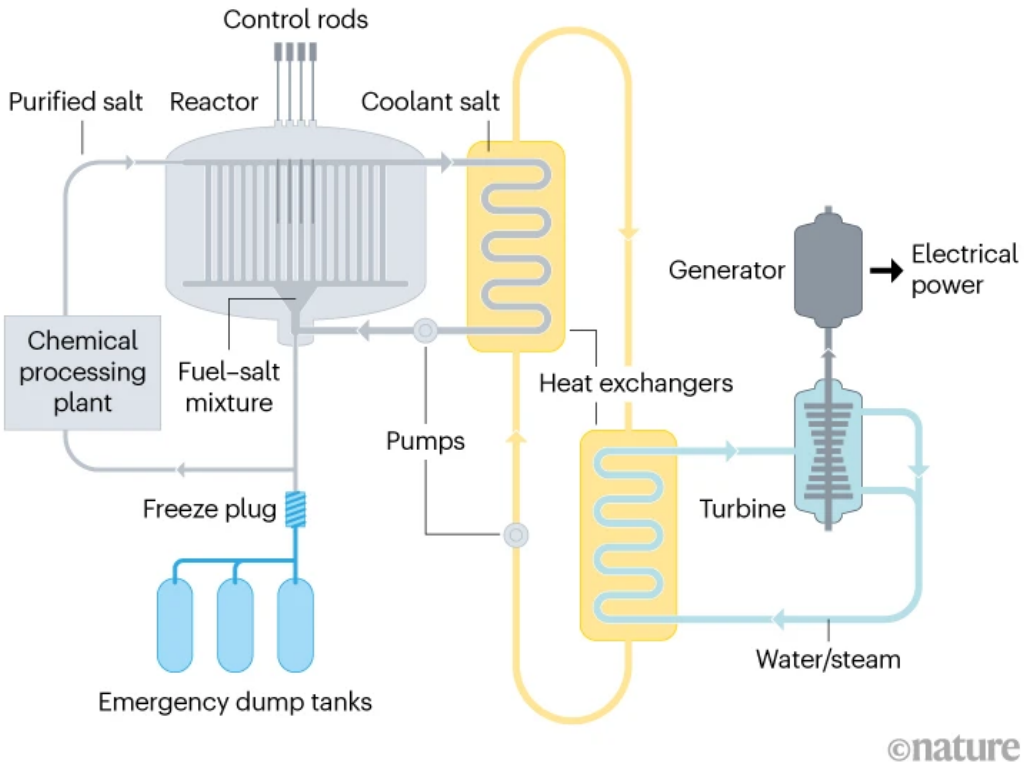


Figure 1.1: A schematic overview of the Molten Salt Fast Reactor (Nature, 2021).

SAMOSAFER (Severe Accident MOdeling and Safety Assessment for fluid-fuel Energy Reactors) is a project whose goal is to develop an MSFR. The project brings together several European and one

[1] A breeder reactor is a reactor that produces more fissile material than it consumes.

Canadian companies and research institutes (SAMOSAFER, ndb) and is a continuation of the SAMOFAR project. Other projects aiming to develop an MSR is the Russian MOSART (MOlten Salt Actinide Recycler & Transformer) or ZhSR-S project (Ignatiev et al., 2012) and the TMSR-LF1 (Thorium Molten Salt Reactor - Liquid Fuel 1) project by Shangai Institute of Applied Physics (Dai, 2017).

1.3.3. Advantages of the MSR and MSFR

MSRs have multiple safety advantages compared to the conventional Light Water Reactor to reduce the possibility and gravity of accidents. Firstly, the high operating temperatures make them more efficient in generating electricity and make them suitable for other means than electricity generation. The low pressure decreases the chance of structural damage of the reactor, increasing the safety. MSRs also generate less high-level waste and can operate using various fuel cycles, allowing for the extension of fuel resources (International Atomic Energy Agency, nd). An additional advantage of the MSFR is the fuel's strongly negative temperature feedback, stabilising the nuclear reaction. Finally, at the fast neutron spectrum the capture cross sections of the fission products are much lower than in the thermal one (World Nuclear Association, 2023). This increases the energy yield of the fuel and allows the MSFR to continue for a longer time before the fuel is sent for batch processing to remove the fission products or the fuel is disposed of.

1.4. The freeze plug

1.4.1. Concept

Besides the previously mentioned safety mechanisms, the freeze plug has been developed. The freeze plug, also called freeze valve (Chisholm et al., 2020) or cold plug (Giraud et al., 2019), consists of a certain material underneath the reactor core. The material must have its melting point at higher temperature than the operating temperature of the reactor; often, the same material as the molten salt is chosen. Before starting up the reactor, the freeze plug's material is liquefied and poured into a pipe underneath the reactor core. A mechanical valve prevents the freeze plug's liquid material from draining. The freeze plug's material is then cooled until it becomes solid. Finally, the mechanical valve is removed, with the freeze plug being immobilised by the adhesive forces between the freeze plug and the wall. Once the freeze plug is formed, the molten salt is placed into the reactor core such that operation can commence.

The freeze plug is actively cooled below its melting point throughout the operation of the MSFR. However, in case of an accident it is possible that the cooling system turns off and the cooling stops. As the reactor operates at a higher temperature than the freeze plug's melting point, part of the freeze plug then melts and the freeze plug loses contact with the wall. The freeze plug then falls into the emergency draining system due to the pressure of the molten salt above it. As a result, the reactor core drains as the molten salt falls into the dump tanks as well. These are designed to force subcriticality in the reaction of the molten salt, eventually stopping the nuclear chain reaction. By draining, the temperature of the molten salt is prevented from causing any structural damage to the reactor core during the anomaly. The passive melting of the freeze plug thus acts as a fail-safe to avert grave accidents or the damaging of the MSFR by the molten salt's temperature.

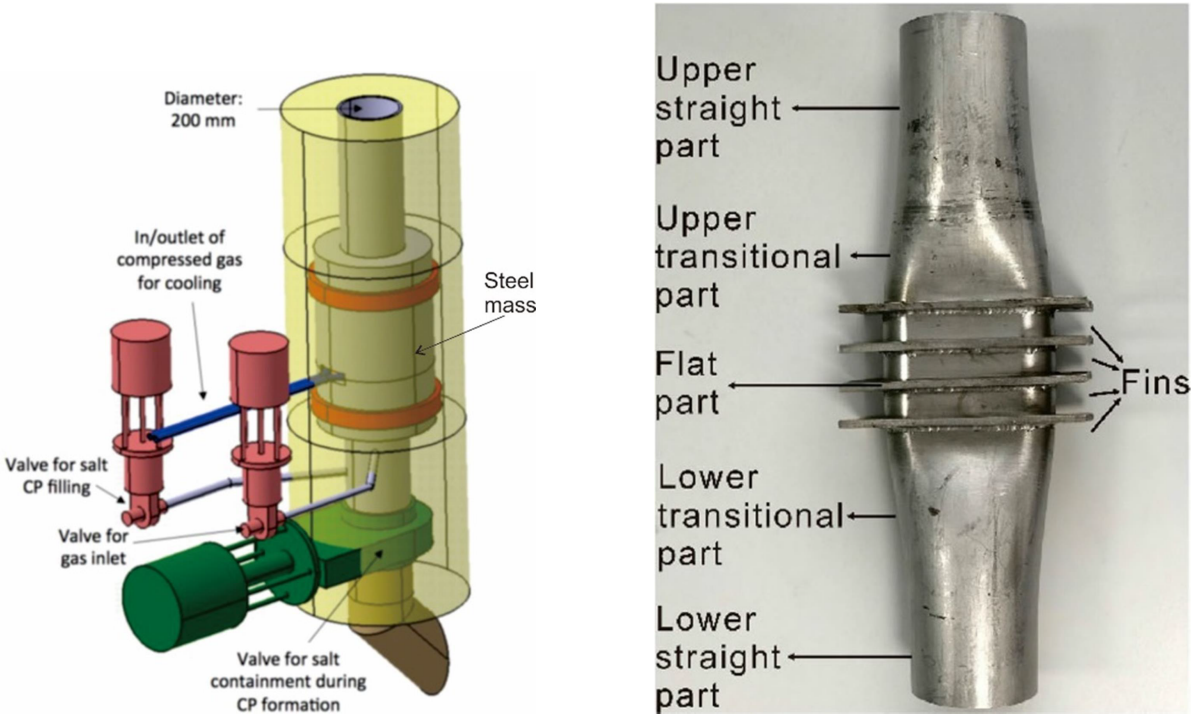
1.4.2. Requirements

The freeze plug has two main requirements (Chisholm et al., 2020, Shafer, 2018). On the one hand, in case of an incident the freeze plug must melt before the structural materials of the reactor core reach a temperature of 1473 K (Brovchenko et al., 2013). Estimates on how long this takes vary from 480 s (Brovchenko et al., 2013) to 1600 s (Ghetta et al., 2017, Shafer, 2018). As determining the required time is outside the scope of this thesis, the intermediate value of $t = 1000$ s is chosen as the working point in this thesis. Conversely, the freeze plug must not melt due to thermal fluctuations during normal operation of the MSFR to avoid it from shutting down without necessity. These contradictory criteria pose a vast design challenge which is to be solved using numerical and analytical methods, such as the ones described previously in this chapter.

1.4.3. Designs

Two main freeze plug designs exist (Chisholm et al., 2020). The first one is the MSFR design, developed by the European SAMOFAR and SAMOSAFER projects. The MSFR freeze plug is a vertically-oriented cylinder of salt located in the draining pipe of the MSFR. The cooling system is implemented in the form of a copper annulus surrounding the draining pipe. The copper is cooled by argon gas flowing through separate pipes. Around the freeze plug, a steel mass can be located to act as an energy storage device to accelerate melting in case of reactor malfunction (Giraud et al., 2019). This freeze plug design is shown in Figure 1.2a.

The other main freeze plug design is the TMSR (Thorium Molten Salt Reactor) design, a mostly Chinese project led by Shanghai Institute of Applied Physics. Jiang et al. (2020) developed a freeze plug in the form of a flattened cylinder. It includes metallic fins to speed up the melting. A schematic design is given in Figure 1.2b.



(a) Figure taken from Giraud et al. (2019)

(b) Figure taken from Jiang et al. (2020)

Figure 1.2: The MSFR (a) and TMSR (b) freeze plugs designs.

In this thesis, the MSFR design has been chosen as a basis instead of the TMSR design, for three main reasons. Foremost, the MSFR design is simpler and therefore easier to implement in a numerical model. The freeze plug itself can be modelled as a cylinder, while the other aspects such as steel mass and cooling system can be implemented through e.g. boundary conditions or source terms in the simulation. Contrarily, the fins and flattened out part of the TMSR freeze plug must be implemented in the geometry, which is more difficult to achieve. The MSFR design's cylindrical shape also ensures only one way to incline the freeze plug exists, contrary to the TMSR's two ways. This reduces the degrees of freedom that must be investigated in the thesis. Besides this, as the names imply, the MSFR freeze plug design is more suitable for operation in the MSFR due to e.g. material composition and freeze plug dimensions. Therefore using the MSFR design is expected to have more relevance for the SAMOSAFER project. It also gives more data from previous research to compare results with.

1.4.4. Previous research

As the freeze plug is an important safety mechanism of the MSFR, various research has been conducted about its melting. Giraud et al. (2019) have developed a freeze plug design via the SAMOFAR project. They made an experimental design of the freeze plug in the FFFER (Forced Fluoride Flow for Experimental Research). The FFFER facility was a LiF-NaF-KF eutectic salt forced convection loop.

The goals of this experiment were studying the liquid/gas separation in a salt cleaning process and acquiring experience in designing high-temperature salt experiments. This experience was then used in the SWATH (Salt at Wall: Thermal excHanges) experiment. They showed a freeze plug is feasible, both with experiments and simulations, which agreed reasonably with each other. However, they recommended more numerical research on solidification. Note that the results from this paper cannot be directly translated to the ones in this thesis, because a different geometry and salt were used.

Shafer (2018) numerically investigated various parameters of the SWATH freeze plug design via the COMSOL software package. She showed that using one freeze plug instead of seven speeds up melting time. She also investigated the optimal parameters for cooling temperature and freeze plug height. Shafer took into consideration the conjugate heat transfer via the piping and cooling system of the freeze plug. Reus (2021) developed a model for freeze-plug melting for the open source Finite Volume Method software *OpenFOAM*, and achieved reasonably similar results to Shafer (2018). Aji (2020) showed that melting time heavily decreases with increasing inclination angle of the freeze plug. However, Aji's geometry and materials were more consistent with the TMSR instead of the MSFR, so this result cannot be directly extrapolated to the MSFR.

1.5. Project goals

The goal of this Thesis was to computationally investigate the effect of inclination angle on the melting behaviour of the MSFR freeze plug. Two benchmark cases have been analysed to ensure the solver's validity; one to validate the phase change, the other to investigate the working of conjugate heat transfer. The following questions need to be answered:

- How does the solver perform the benchmark cases with respect to previous results?
- What effect does the inclination angle have on the formation of a freeze plug for normal MSFR operation?
- What effect does the inclination angle have on the melting behaviour of the MSFR freeze plug?

2

Theory

Solid-liquid phase change processes play a role in many engineering purposes, such as thermal energy storage (Cheng et al., 2022) and temperature regulation (Guo et al., 2022). Investigating them is challenging due to several reasons. Firstly, the boundary between the two phases - the phase change interface - may displace in time and can have a complex shape. Also, it constitutes a strong discontinuity in enthalpy, but a weak one in temperature. Finally, in case of flow, there is a constant mutual interaction between the flow field and the phase change interface.

Only few phase change problems have a known analytical solution (Krishnan et al., 2022), for instance the one-dimensional Stefan problem first described by and eponymously named after Stefan (1891). To solve other problems, a numerical solution is required.

In this chapter, the relevant theoretical background for understanding the numerical modelling of solid-liquid phase change is explained. Sections 2.1 to 2.4 give the physical background, while 2.5 to 2.8 explain the numerical methods to solve phase change problems.

2.1. Heat transfer

The starting point for solving problems with solid-liquid phase change is the energy equation, which can be written as (Fermigier, 2017):

$$\frac{DH}{Dt} = \nabla \cdot (\lambda \nabla T) + c_p T \frac{D\rho}{Dt} + q''' \ , \tag{2.1}$$

where H is the volumetric enthalpy, λ the material's heat conductivity, T its local temperature, c_p the specific heat capacity, and q''' the volumetric energy source or sink term. $\frac{DH}{Dt} \equiv \frac{\partial H}{\partial t} + \boldsymbol{u} \cdot \nabla H$ is the material derivative of H, with an equivalent formulation for $\frac{D\rho}{Dt}$.

For an incompressible material with constant c_p, and if $q''' = 0$ everywhere, Equation 2.1 can then be rewritten as[1]:

$$\frac{\partial T}{\partial t} + \boldsymbol{u} \nabla T = \alpha \nabla^2 T \ . \tag{2.2}$$

with $\alpha = \frac{\lambda}{\rho c_p}$ as the heat diffusivity. ρ is the fluid's density and c_p its specific heat capacity; ρc_p is the volumetric heat capacity. In a solid, $\boldsymbol{u} = \boldsymbol{0}$, so Equation 2.1 can be further simplified as

$$\frac{\partial T}{\partial t} = \alpha \nabla^2 T \ . \tag{2.3}$$

2.2. Equations of motion

For a fluid, the analogue of Newton's second law is the Navier-Stokes equation[2] (NSE). For an incompressible fluid, it reads as follows (van den Akker and Mudde, 2014):

$$\frac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}\boldsymbol{u}) = -\nabla p + \nabla \cdot \underline{\underline{\tau}} + \rho \boldsymbol{g} \ , \tag{2.4}$$

[1] See Fermigier (2017) for a derivation

[2] The plural 'Navier-Stokes Equations' is also found in literature; in that case, the three spatial components are treated separately.

The terms in the NSE signify the following:

- $\frac{\partial \rho \boldsymbol{u}}{\partial t}$: The local acceleration in case of unsteady flow (i.e. flow changing in time).
- $\nabla \cdot (\rho \boldsymbol{u}\boldsymbol{u})$: The convective acceleration in case of non-uniform flow (i.e. flow changing along a streamline).
- ∇p: The pressure gradient.
- $\nabla \cdot \underline{\underline{\tau}}$: The divergence of the stress tensor. For a Newtonian fluid, $\underline{\underline{\tau}}$ is given by (Sonin, 2001):

$$\underline{\underline{\tau}} = \mu \left(\left(\nabla \boldsymbol{u} + \nabla (\boldsymbol{u})^T \right) - \frac{2}{3} (\nabla \cdot \boldsymbol{u}) \underline{\underline{I}} \right) , \tag{2.5}$$

  where $\underline{\underline{I}}$ is the unity tensor.

- $\rho \boldsymbol{g}$: The gravity source term. In this thesis, the Boussinesq approximation can be used. This neglects all density differences except when the density is multiplied by $\boldsymbol{g}$ and simplifies the NSE without strongly affecting the solution of the NSE. The Boussinesq approximation is valid when the temperature differences, vertical scale, and Mach number[3] of the flow are small, and shock waves are not considered (Kundu et al., 2016), all of which apply for the flow in this thesis. The Boussinesq approximation approximates ρ via:

$$\rho = \rho_{ref} (1 - \beta (T - T_{ref})) , \tag{2.6}$$

  where ρ_{ref} is the density at the reference temperature T_{ref} and β is the material's thermal expansion coefficient.

Another equation governing fluid dynamics is the mass continuity equation (Sonin, 2001):

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0 , \tag{2.7}$$

which for an incompressible fluid, such as a liquid, reduces to

$$\nabla \cdot \boldsymbol{u} = 0 . \tag{2.8}$$

2.3. Dimensionless numbers

Convection can be divided into two types: forced and natural convection. Forced convection arises from pressure differences due to sources external to the fluid, e.g. pumps, fans or suction devices. Conversely, natural convection (also called free convection) originates from the fluid itself; specifically a density gradient due to a nonuniform temperature field. For both types of convection, the Reynolds number Re can be defined to compare the relative effect of momentum-diffusive and convective time scales (van den Akker and Mudde, 2014):

$$Re = \frac{ul}{\nu} = \frac{\text{momentum diffusion time scale}}{\text{convection time scale}} , \tag{2.9}$$

where l is the length of the fluid's container and ν is the fluid's kinematic viscosity. The Reynolds number is an important indicator of turbulence: if approximately $Re < 2300$, the flow is laminar, while for $Re > 2900$ the flow is turbulent (for intermediate Reynolds numbers the fluid is in the transition régime between turbulent and laminar flow).

The Péclet number Pe is the thermal analogue of the Reynolds number (Fermigier, 2017):

$$Pe = \frac{ul}{\alpha} = \frac{\text{thermal diffusion time scale}}{\text{convection time scale}} . \tag{2.10}$$

[3] $Ma = \frac{\text{maximum flow speed}}{\text{speed of sound in the medium}}$

If $Pe \gg 1$, diffusion takes place in much longer timescales than convection and can be ignored. Conversely, if $Pe \ll 1$, convection can be neglected.

In case the Boussinesq approximation (Equation 2.6) is valid, the Rayleigh number Ra can be calculated. This is a special type of Péclet number valid only for natural convection (Squires and Quake, 2005). Near a vertical wall Ra is defined as (Kenjeres, 2021):

$$Ra = \frac{\beta \Delta T l^3 g}{\nu \alpha} = \frac{\text{diffusion time scale}}{\text{natural convection time scale}} . \tag{2.11}$$

The Prandtl number (Pr) is a material property which compares the momentum and thermal diffusivity. It is calculated by (van den Akker and Mudde, 2014):

$$Pr = \frac{\nu}{\alpha} = \frac{\text{thermal diffusion time scale}}{\text{momentum diffusion time scale}} . \tag{2.12}$$

Materials with a low Prandtl number, e.g. molten metals, diffuse heat faster than momentum. As a result, the thermal boundary layer is thicker than the momentum one.

The Nusselt number (Nu) can be used to investigate the relative effect of convective and diffusive heat transfer. As diffusive heat transfer at a certain position and time is independent of the flow field, the Nusselt number is also a measure for the total heat transfer. It is given by (van den Akker and Mudde, 2014):

$$Nu = \frac{h l_c}{\lambda} = \frac{\text{diffusive heat transfer time scale}}{\text{convective heat transfer time scale}} , \tag{2.13}$$

where l_c is a characteristic length and h the heat transfer coefficient.

2.4. Inclination

2.4.1. Inclined horizontal wall

The inclination angle of the geometry of a liquid has a strong influence on the role of the convection in the phase change process. We introduce θ, which is the angle between a wall of a cavity and the gravity acceleration vector, i.e. $\theta = 0$ corresponds to an upright wall. This is visualised in Figure 2.1.

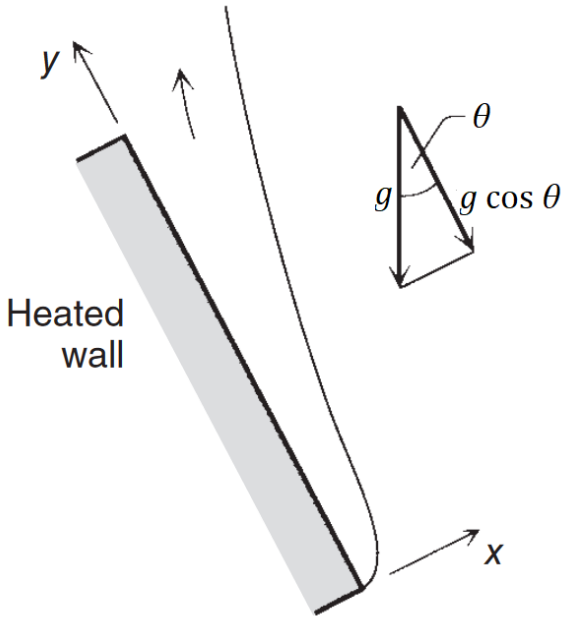


Figure 2.1: The definition of θ for an inclined heated wall. The figure is adapted from Bejan (2013).

For $-60° < \theta < 60°$, the Rayleigh number is simply calculated via the gravity component parallel to the wall, so Equation 2.11 is adapted as follows (Bejan, 2013):

$$Ra = \frac{\beta \Delta T l^3 g \cos\theta}{\nu \alpha} \,. \tag{2.14}$$

This equation is valid only if the flow near the wall is laminar. In the turbulent régime, results correlate better with g instead of $g\cos\theta$ (Vliet, 1969), so Equation 2.11 is used instead. For fluids with $0.7 \leq Pr \leq 6.9$ (the water-air range), the Nusselt number can be calculated as follows (Vliet and Ross, 1975):

$$\overline{Nu} = \begin{cases} 0.55 Ra^{1/5} & \text{laminar flow;} \\ 0.17 Ra^{1/4} & \text{turbulent flow.} \end{cases} \tag{2.15}$$

For laminar flow, the Rayleigh number (and as a result the Nusselt number) is lower in an inclined medium than in an upright one. However, the transition between laminar and turbulent flow has been observed to happen at a much lower Rayleigh number at inclined plates: in experiments using water, the transition régime between laminar and turbulent flow has been found to be between $5 \cdot 10^{12} < Ra < 10^{14}$ for an upright medium, and $6 \cdot 10^7 < Ra < 6 \cdot 10^9$ for one inclined by 60° (Bejan, 2013). For the Ra values where the flow régime (turbulent or laminar) depends on the inclination angle, turbulent flow has a higher Nusselt number than laminar one, according to Equation 2.15.

2.4.2. Vertical wall

For $|\theta| \geq 60°$, a wall cannot be described as an inclined horizontal wall anymore. In the most extreme case, when $\theta = \pm 90°$, the characteristic length l_c can be calculated (Bejan, 2013):

$$l_c = \frac{A}{P} \,, \tag{2.16}$$

where A is the plane surface and P its perimeter. The characteristic Rayleigh number Ra_{l_c} is then calculated by Equation 2.11, except it is based on the characteristic length l_c instead of the cavity height l. The characteristic Nusselt number is calculated as follows for a hot surface facing upward or a cold surface facing down (which cause equivalent fluid behaviour)

$$\overline{Nu_{l_c}} = \begin{cases} 0.54 Ra_{l_c}^{1/4} & \text{if } 10^4 < Ra_{l_c} < 10^7 \,; \\ 0.15 Ra_{l_c}^{1/3} & \text{if } 10^7 < Ra_{l_c} < 10^9 \,. \end{cases} \tag{2.17}$$

For a hot surface facing downward or a cold surface facing upward Nu_{l_c} is given by:

$$\overline{Nu_{l_c}} = 0.27 Ra_{l_c}^{1/4} \quad \text{if } 10^5 < Ra_{l_c} < 10^{10} \,. \tag{2.18}$$

2.4.3. Experimental results

Inclining a domain has a big effect on its thermophysical behaviour, including phase change. Zennouhi et al. (2017) simulated four cases of inclined melting of gallium: one with melting from a side, one with melting from the bottom and two intermediate cases. The total melting time was 30% shorter for the bottom-heated case compared to the side-heated one. Besides the different Rayleigh and Nusselt numbers, this was primarily due to the different number and size of Bénard cells in the convection-dominated regime. Korti and Guellil (2020) experimentally melted paraffin in a square cavity from the bottom, side and in one intermediate case. Compared to the sideways-heated case, the total melting times were 48% and 56% shorter for the intermediate and bottom-heated case. For this reason, an inclined freeze plug is expected to have a shorter melting time than an upright one, benefitting the design of the MSFR.

2.5. Solving the equations of motion

2.5.1. Numerical methods for partial differential equations

Finding a general analytical solution of the equations of motion is a famous unsolved problem in modern mathematics (Clay Mathematics Institute, 2022). Except for a few simple cases such as Poisseuille

flow (van den Akker and Mudde, 2014), the equations of motion can only be solved numerically. The branch of physics dedicated to numerically solving problems including fluid flow is Computational Fluid Dynamics (CFD). The three most common numerical methods for discretising partial differential equations (PDEs) are the Finite Element Method (FEM), Finite Difference Method (FDM) and the Finite Volume Method (FVM).

In the Finite Volume Method, the geometry of the domain is discretised into numerous subdomains called cells or control volumes, as can be seen in Figure 2.2. These cells can have any convex shape. Every physical point of the physical domain gets transformed to exactly one cell. During every time step of the simulation, the equations governing the physics of the domain are solved in each cell (Chen, 2021). The Finite Element Method splits up the physical domain into several interconnected elements such as triangles or quadrilaterals. The physical equations are solved on each element node, rather than a volume, and these solutions are used to approximate the solution for the entire PDE (van Kan et al., 2014). The Finite Difference Method consists of approximating the derivatives in the PDE by finite differences, which are calculated using the values of the function at nearby points. The solutions to the PDE are then obtained by solving a system of algebraic equations that results from the finite difference approximations (Frey, 2017).

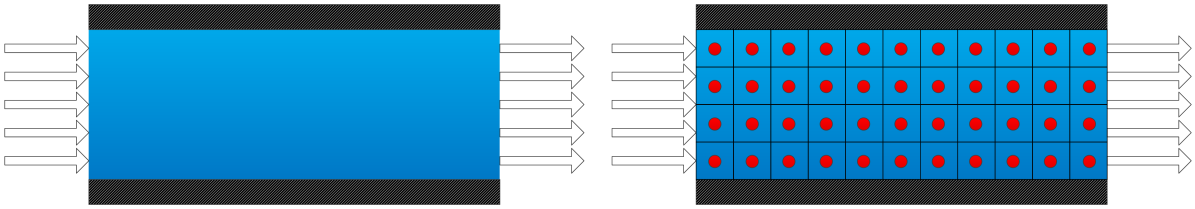


Figure 2.2: A 2D domain in which channel flow takes place (left) and its decomposition in cells (right). The cell centres are indicated with red dots. The figure is taken from Chilvers (2014).

An advantage of the FVM is the fact it is relatively intuitive and easy-to-implement method. As it is based on conservation laws, it locally conserves field variables such as energy, pressure, and velocity. As a result, it is often used in problems involving fluid flow and heat and mass transfer (Cadence PCB solutions, nd). Splitting up of the domain in control volumes makes the FVM a suitable method for complex domains or boundary conditions in comparison with the FDM (Sjodin, 2016). It also makes it a suitable method for parallel computing (Delis and Mathioudakis, 2009), resulting in a higher simulation speed in computationally heavy simluations.

A disadvantage of the FVM is its poor suitability for applications other than computational fluid dynamics; for e.g. electrodynamics or astrophysics, the FEM or FDM are often more useful (Sjodin, 2016). This is due to the fact derivative schemes higher than second order are difficult to implement (Sjodin, 2016). The FVM is also worse in handling sharp discontinuitites or gradients than the FDM, as it is built on averaging, smoothing out sharp features.

In this thesis, the FVM has been chosen as the method of choice because of its suitability for computational fluid dynamics and parallel computing. Simultaneously, its disadvantages are not as applicable considering the geometry is relatively simple and no steep gradients or discontinuities in pressure, temperature, or velocity had been expected to be present.

2.5.2. Finite Volume Method

In the FVM, for cuboid control volumes, their faces are often denoted with lowercase letters using the four cardinal directions (n, e, s, w) and f and b (for front and back). Neighbouring cells are indicated with uppercase letters, and the cell itself with P. This is shown in Figure 2.3.

The fields inside a cell are integrated over its volume[4] and divided by the volume, yielding the average value. This value is 'stored' at the cell centre. This for example allows the discretisation of the steady-

[4] In case of a 3D simulation. In case of a 2D simulation the integration takes place over the cell area, in a 1D simulation over its length.

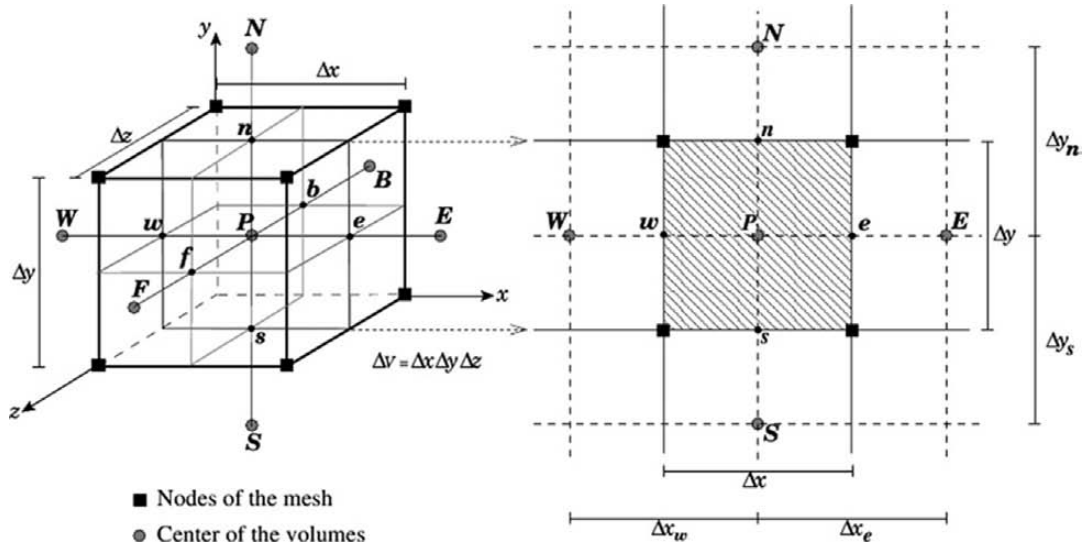


Figure 2.3: A 3D (left) and 2D (right) FVM control cell with faces indicated with lowercase compass notation and neighbouring cells indicated by uppercase compass notation. The figure is taken from de la Cruz and Monsivais (2014).

state diffusion equation for a certain scalar φ:

$$\underbrace{\nabla \cdot (\rho \boldsymbol{u} \varphi)}_{\text{convection}} = \underbrace{\nabla \cdot (\Gamma \nabla \varphi)}_{\text{diffusion}} + \underbrace{S}_{\text{source}} \ , \tag{2.19}$$

where Γ is a viscous term. Equation 2.19 is integrated over each control volume as follows:

$$\int_V \nabla \cdot (\rho \boldsymbol{u} \varphi) dV = \int_V \nabla \cdot (\Gamma \nabla \varphi) dV + \int_V S dV \ , \tag{2.20}$$

and Gauss's divergence theorem is applied for the convection and diffusion:

$$\int_V \nabla \cdot (\rho \boldsymbol{u} \varphi) dV = \oint_S \rho \boldsymbol{u} \varphi dS = \sum_i \int_{f_i} \rho \boldsymbol{u} \varphi dS \ , \tag{2.21}$$

where f_i indicate the different faces. An equivalent formulation is found for the diffusive term.

As an illustration, for one dimension, Equations 2.19 to 2.21 together form the following equation:

$$\rho \boldsymbol{u} \varphi|_e - \rho \boldsymbol{u} \varphi|_w = \Gamma \nabla \varphi|_e - \Gamma \nabla \varphi|_w + \int_w^e S dx \ . \tag{2.22}$$

The values at the cell faces $\varphi|_w$ and $\varphi|_e$, as well as their derivatives, are calculated via differencing schemes. Differencing schemes can be divided into first-order differencing schemes, which only use first-order terms in a Taylor approximation, and second-order differencing schemes, which also use the second order terms. Higher-order schemes also exist but are not discussed in this thesis. First-order schemes yield less physical results but are more numerically stable than second-order schemes for convection-dominated flow (Wimshurst, 2018b).

A straightforward way to calculate the value $\varphi|_e$ is via a weighted average[5]:

$$\varphi|_e = f \varphi|_P + (1 - f) \varphi|_E \ . \tag{2.23}$$

[5]The calculation of $\varphi|_w$ and in multidimensional cases $\varphi|_n$, $\varphi|_s$, $\varphi|_f$ and $\varphi|_b$ works analogously to $\varphi|_e$.

One second-order differencing scheme is the central differencing scheme, which takes the average of the cells next to the face:

$$f = \frac{1}{2} \,. \tag{2.24}$$

Another differencing scheme scales the weighing factor f with the distance between the face and the neighbouring cell centres (Wimshurst, 2018b):

$$f = \frac{x_E - x_e}{x_E - x_P} \,. \tag{2.25}$$

In case of an equidistant mesh $f = 1/2$, and the scheme is the same as central differencing. This is a straightforward differencing scheme, but it has been shown to cause unphysical oscillatory behaviour (Wimshurst, 2018b). An alternative scheme is the (first-order) upwind scheme, which takes into account the direction of the flow:

$$f = \begin{cases} 0 & \text{if } u_x|_e < 0 \,; \\ 1 & \text{if } u_x|_e \geq 0 \,. \end{cases} \tag{2.26}$$

A drawback of the upwind scheme is its proneness to numerical diffusion. This is diffusion caused not by physical but numerical sources, such as the convection differencing scheme and the temporal discretisation (which is discussed later in this paragraph) (Jasak, 1996). Numerical diffusion can be reduced by increasing the mesh resolution or by using a second or third-order differencing scheme. In one-dimensional flow, aligning the mesh with the flow direction removes numerical diffusion.

The linear upwind scheme is a second-order scheme based on the upwind scheme used primarily for convection. It calculates $\varphi|_e$ via (Wimshurst, 2018b)

$$\varphi|_e = \begin{cases} \varphi|_E + \nabla\varphi|_E(x_E - x_P) & \text{if } u_x|_e < 0 \,; \\ \varphi|_P + \nabla\varphi|_P(x_E - x_P) & \text{if } u_x|_e \geq 0 \,. \end{cases} \tag{2.27}$$

The gradient $\nabla\varphi$ can be found in several ways, for instance by (Wolf Dynamics, nd):

$$\nabla\varphi|_P = \frac{\varphi|_e - \varphi|_w}{x_e - x_w} \,, \tag{2.28}$$

and similarly for $\nabla\varphi|_E$.

The linear upwind scheme is prone to oscillations, as a steep gradient can make $\varphi|_e$ bigger or smaller than both $\varphi|_P, \varphi|_E$, which is almost always unphysical. This error can be prevented by introducing a limiter for $\nabla\varphi$:

$$\varphi|_e = \begin{cases} \varphi_E + \frac{\xi}{2}\nabla\varphi|_E(x_E - x_P) & \text{if } u_j\big|_e < 0 \,; \\ \varphi_P + \frac{\xi}{2}\nabla\varphi|_P(x_E - x_P) & \text{if } u_j\big|_e \geq 0 \,, \end{cases} \tag{2.29}$$

where $0 \leq \xi \leq 1$. ξ reduces the influence of the gradient. Note that if $\xi = 0$, the limited linear upwind scheme reduces to the upwind scheme. Conversely, $\xi = 1$ the scheme falls back to central differencing.

2.5.3. Temporal discretisation

Besides spatial derivatives, temporal derivatives also have to be accounted for in the FVM. A popular choice is the family of backward differentiation formulae (BDF) (Iserles, 1996):

$$\frac{\partial\varphi}{\partial t} \approx \frac{1}{\Delta t}\sum_{j=0}^{M} \gamma_j \varphi^{n+1-j} \,, \tag{2.30}$$

where M indicates the order of the BDF scheme. $M = 1$ and $M = 2$ are the most popular choices. Δt is the time step length. The superscript $n + 1 - j$ indicates the enthalpy value at different time steps. γ_j are weighing factors, given for BDF1 (also known as implicit Euler) by:

$$\begin{pmatrix} \gamma_0 \\ \gamma_1 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \tag{2.31}$$

and BDF2 by

$$\begin{pmatrix} \gamma_0 \\ \gamma_1 \\ \gamma_2 \end{pmatrix} = \begin{pmatrix} 1 + \dfrac{\Delta t}{\Delta t + \Delta t_0} \\ -\dfrac{\Delta t + \Delta t_0}{\Delta t_0} \\ \dfrac{\Delta t^2}{(\Delta t + \Delta t_0)\Delta t_0} \end{pmatrix}. \tag{2.32}$$

Here Δt and Δt_0 are the lengths of the current and previous timesteps, respecively. As always the case for time derivatives in BDF schemes, $\sum_i \gamma_i = 0$; also, if $\Delta t = \Delta t_0$ in BDF2, then $(\gamma_0, \gamma_1, \gamma_2) = (3/2, -2, 1/2)$.

2.5.4. The Courant number

In the FVM, for every cell the Courant number can be calculated as follows:

$$Co = \frac{u\Delta t}{\Delta x} = \frac{\text{particle distance travelled in one time step}}{\text{cell size}}. \tag{2.33}$$

Simulations with a high Co run fast but can yield unphysical results or become unstable. With the algorithms used in this thesis, $Co \leq 1$ must be enforced to ensure physical behaviour[6].

2.6. PIMPLE

In the algorithm used in this thesis, at every time step, first the equations of motion are solved. One algorithm to achieve this is the PIMPLE algorithm (Greenshields, 2022, Wimshurst, 2018a). It is a combination of the SIMPLE (Semi-Implicit Method for Pressure Linked Equations) and the PISO (Pressure Implicit with Splitting of Operators) algorithms.

The PIMPLE algorithm works as follows: first, the p and $\boldsymbol{u}$ fields are initialised. The NSE (Equation 2.4) is solved to yield a predictor for $\boldsymbol{u}$. The continuity criterion is used to solve the pressure field. From the pressure field, then the velocity field is solved. This process is repeated until convergence.

A more thorough description of the PIMPLE algorithm is given in Appendix A.

2.7. Melting and solidification

In this thesis, solid-liquid phase change is the key physical phenomenon. The relevant phase change material (PCM) is locally solid or liquid. The interface between the solid and liquid areas, the phase change interface, is sharp for pure chemical substances and eutectic mixtures. Non-eutectic mixtures have a phase change interface with a certain thickness, the mushy zone[7].

Under constant pressure, part of the enthalpy of a material is proportional to its temperature. This enthalpy is called the sensible enthalpy of the material. However, phase change requires energy in the form of latent heat L to overcome the interatomic or interionic binding in the solid.

The total volumetric enthalpy as function of temperature is equal to

$$H(T, \ell) = \begin{cases} c_{p,s}\rho_s T & T < T_m \quad \text{(solid)}; \\ c_{p,s}\rho_s T_m + \ell\rho_s L & T = T_m \quad \text{(undergoing phase change)}; \\ c_{p,s}\rho_s T_m + \rho_s L + c_{p,l}\rho_l (T - T_m) & T > T_m \quad \text{(liquid)}. \end{cases} \tag{2.34}$$

ℓ indicates the liquid fraction, which is 0 in a solid and 1 in a liquid. In computational methods, a PCM can be experiencing local phase change which is indicated with a value of $0 < \ell < 1$. T_m is the PCM's melting temperature.

The relation of Equation 2.34 is shown in Figure 2.4. As can be seen, a liquid-solid interface constitutes a strong discontinuity in enthalpy, but a weak one in temperature[8].

[6] In principle higher values can also be possible, but it is not known *a priori* whether they will guarantee the validity of the simulations.

[7] Explaining the difference between a eutectic and non-eutectic mixture is outside the scope of this thesis. The only relevant property is the different phase change behaviour: eutectics have a sharp phase change interface, non-eutectics have a mushy zone.

[8] I.e. the value of the enthalpy is discontinuous, while for the temperature only the gradient is.

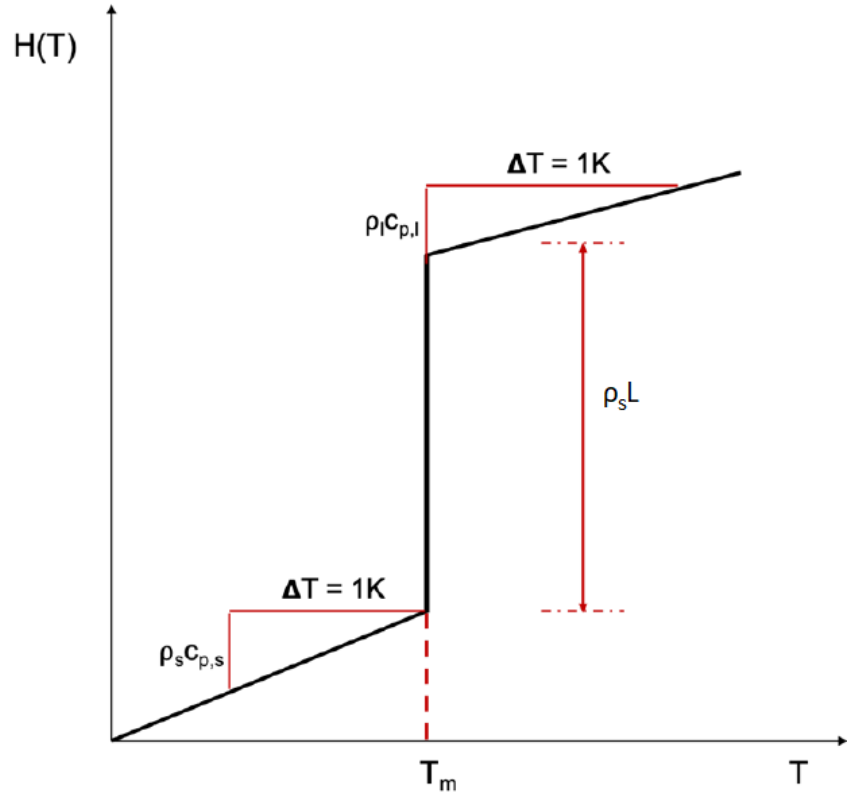


Figure 2.4: The enthalpy-temperature relation. When fully solid or fully liquid, enthalpy increases linearly with temperature. However, to overcome phase change, additional energy (indicated with $\rho_s L$) must be added. The figure is adapted from Kaaks et al. (2022).

A PCM's thermophysical properties such as λ and c_p are assumed to be constant (i.e. not temperature-dependent) within a certain phase. The thermophysical properties during phase change can be evaluated in several ways: by assigning either the solid or liquid values (i.e. $\varphi = \varphi_s$ or $\varphi = \varphi_l$ for any thermophysical property φ), by a non-weighted average (i.e. $\varphi = \frac{1}{2}(\varphi_s + \varphi_l)$) or a weighted average (i.e. $\varphi = (1-\ell)\varphi_s + \ell\varphi_l$). In practice, the difference between the different methods is not very relevant as only a small fraction of the computational domain is undergoing phase change at a given time.

2.8. Phase change models

2.8.1. Types of models

A plethora of phase change models exist, and many ways to distinguish them do as well. One such way is the categorisation between fixed-domain and moving-domain models. Fixed-domain models solve the phase change problem on a fixed grid. Moving-domain models require a new grid to be made during the simulation, e.g. refining it near the phase-change boundary or where the flow is higher. This allows for a more accurate simulation, but comes at the expense of computational power and is more complex to implement. The moving domain approach has problems simulating phase change problems with multiple fronts or fronts (dis)appearing over the course of the simulation (Lacroix and Voller, 1990), and many methods are not valid generally, but for a limited amount of problems. Because of these reasons, the fixed-domain approach has been used in this thesis.

Fixed-domain models can further be divided into implicit and explicit interface tracking models. Implicit models find the phase change interface directly from domain values such as temperature or enthalpy, while explicit models use a separate calculation to find the phase change interface. Implicit interface tracking is easier to implement and more robust than explicit one. Also, problems for explicit interface tracking arise in the definition of the phase change boundary and the implementation of the coupling boundary conditions between the two domains. Therefore, implicit tracking has been used in this thesis.

Fixed-domain models can also be categorised into microscale and macroscale models. Microscale models simulate phase change on small physical scales such as electronics (Tang et al., 2011). For bigger domains such as the one used in this thesis, macroscale models are most suitable.

Many fixed-domain, implicit, macroscale methods have been developed to evaluate the temperature field. These include the Source Based Method, Apparent Heat Capacity Method and the Linearised Enthalpy Method.

2.8.2. Source Based Method

In the Source Based Method (SBM), the enthalpy jump, and therefore the phase change, is often assumed to be spread out over a temperature interval instead of being present only at T_m. Selected liquid fraction-temperature relations for an example PCM are given in Figure 2.5.

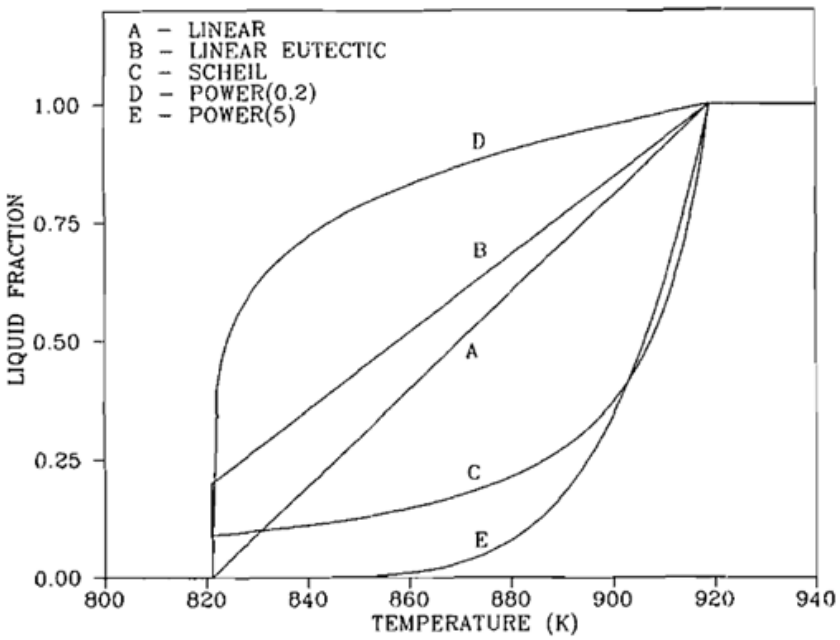


Figure 2.5: Possible relations between liquid fraction and temperature of an example PCM in the source-based method. The figure is taken from Voller and Swaminathan (1991).

The main step is the splitting of the enthalpy in its latent and sensible components to mimic the effect of the latent heat peak:

$$H = \rho c_p T + \ell L \ . \tag{2.35}$$

Assuming only heat transfer by diffusion, this expression is substituted into Equation 2.1, yielding

$$\rho c_p \frac{\partial T}{\partial t} = \nabla \cdot (\lambda \nabla T) - L \frac{d\ell}{dt} \ . \tag{2.36}$$

In a numerical iterative scheme, this equation is discretised. The iterative process consists starts with setting $\ell_p^0 = \ell_p^{old}$, i.e. on each node p the initial liquid fraction is set to that of the previous timestep (Voller and Swaminathan, 1991). Then, the temperature field, Equation 2.36 is solved. The liquid fraction is then updated via

$$\ell_p^{m+1} = \ell_p^m + r\Delta \tag{2.37}$$

with

$$\Delta = \frac{\rho c_p}{L}(T_p^{m+1} - T_p^m) \ . \tag{2.38}$$

And finally ℓ is corrected to prevent unphysical values, via

$$\begin{aligned} &\text{set } \ell^{m+1} = 0 \text{ if } \ell^{m+1} < 0 \ ; \\ &\text{set } \ell^{m+1} = 1 \text{ if } \ell^{m+1} > 1 \ . \end{aligned} \tag{2.39}$$

These steps (except setting $\ell_p^0 = \ell_p^{old}$) are followed until the desired convergence has been reached. If desired, the enthalpy can be calculated during iteration or after convergence via Equation 2.35. The iterative scheme of the SBM is shown in Figure 2.6.

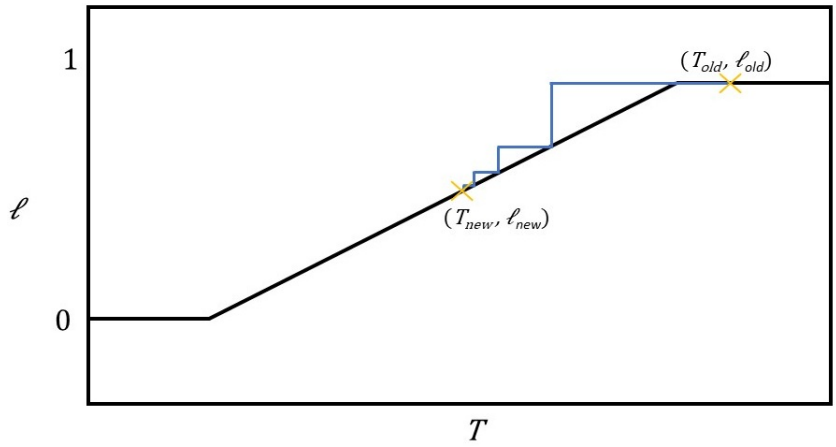


Figure 2.6: The iterative scheme of the SBM for a single node. The iteration starts at $(T_m^0, \ell_m^0) = (T_m^{old}, \ell_m^{old})$ (step 1). As the temperature field is solved, the temperature is updated (step 2). Then the liquid fraction is updated by forcing (T_p^1, ℓ_m^1) back on the T, ℓ curve (step 3). Step 4 is skipped as $0 \leq \ell \leq 1$. Steps 2 and 3 (and 4) are then reiterated until conversion at $(T_p^{new}, \ell_p^{new})$. The figure is based on Swaminathan and Voller (1993) and Reus (2021).

The SBM is a robust phase method but it takes long compared to other methods. For example, two numerical simulations performed by Faden et al. (2019) with the SBM needed 4 and 5.5 times as much time to finish as equivalent simulations using the Linearised Enthalpy Method discussed later in this Paragraph.

2.8.3. Apparent Heat Capacity Method

As is the case in the SBM, in the Apparent Heat Capacity Method (AHCM), the change of the enthalpy and liquid fraction are again smeared out over a mushy zone with a width of $2\Delta T$ when the PCM is undergoing phase change. A possible shape of the liquid fraction-temperature relation is linear, as is shown in Figure 2.7. Note the parallels with graph A of Figure 2.5. The liquid fraction is then given by (Voller and Swaminathan, 1991):

$$\ell = \begin{cases} 0 & T \leq T_m - \Delta T \quad \text{(solid)}; \\ \dfrac{T - T_m + \Delta T}{2\Delta T} & T_m - \Delta T < T < T_m + \Delta T \quad \text{(mushy zone)}; \\ 1 & T \geq T_m + \Delta T \quad \text{(liquid)}. \end{cases} \tag{2.40}$$

The apparent heat capacity, $c_{p,app}$ is then introduced. This is given by (Yao and Chait, 1993):

$$c_{p,app} = \begin{cases} c_{p,s} & T \leq T_m - \Delta T \quad \text{(solid)}; \\ \dfrac{1}{\rho}\left(\ell\rho_l c_{p,l} + (1-\ell)\rho_s c_{p,s}\right) + L\dfrac{d\xi_m}{dT} & T_m - \Delta T < T < T_m + \Delta T \quad \text{(mushy zone)}; \\ c_{p,l} & T \geq T_m + \Delta T \quad \text{(liquid)}. \end{cases} \tag{2.41}$$

where $\xi_m \equiv \dfrac{1}{2}\dfrac{\ell\rho_l - (1-\ell)\rho_s}{\ell\rho_l + (1-\ell)\rho_s}$. As $c_{p,app} = \dfrac{\partial H}{\partial T}$, this expression can be used to solve the energy equation, Equation 2.1.

The AHCM is a straightforward phase change model. However, the introduction of the mushy zone has no legitimate physical justification for pure materials and eutectics. Moreover, the fact the AHCM is not energy-conservative presents a problem for the size of the mushy zone: if it is too small, it may be skipped in a phase change problem, violating conservation of energy. Conversely, if the mushy zone is

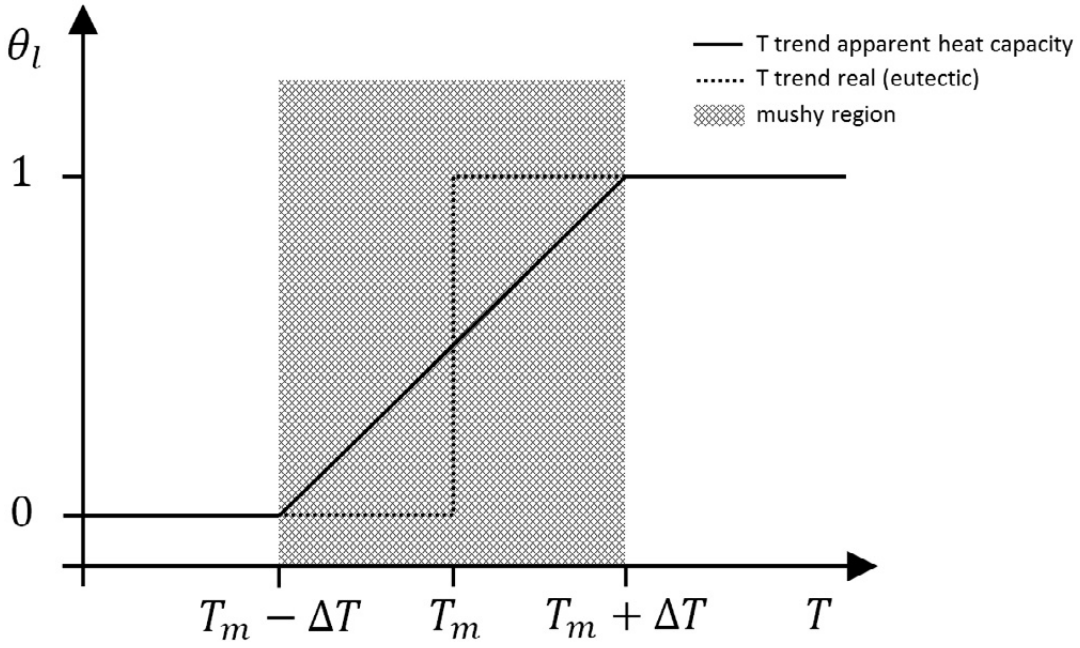


Figure 2.7: The liquid fraction (indicated with θ_l) as function of temperature. In reality, the liquid fraction behaves as a step function around the melting temperature. In the AHCM, the liquid fraction is a linearly increasing function in the mushy zone between $T = T_m \pm \Delta T$. Figure taken from Tiberga et al. (2019).

too big, it may cause accelerated phase change as it could cover the entire solid phase (Shafer, 2018), while using smaller time steps to prevent the phase change from being skipped is computationally expensive.

2.8.4. Linearised Enthalpy Method

The Linearised Enthalpy Method (LEM) was first proposed by Swaminathan and Voller (1993) as the Optimum Method. In the LEM, the semi-discretised energy equation is obtained. The first step is discretising the enthalpy derivative via a backward Euler formula (see Equation 2.30). In this thesis, the BDF2 formula was chosen. The value of $\frac{\partial H}{\partial t}$ is used in the semi-discretised energy equation (Faden et al., 2019):

$$\frac{\gamma_0 H^k + \gamma_1 H^o + \gamma_2 H^{oo}}{\Delta t} + \nabla \cdot (\boldsymbol{u} H^{k+1}) = \lambda \nabla^2 T^{k+1} \ , \tag{2.42}$$

where k and $k+1$ are iteration steps. Also, the substitutions $H^{n+1} = H^{k+1}$, $H^n = H^o$, and $H^{n-1} = H^{oo}$ have been made. The latter two indicate the value of H at the previous (old) time step and the one before, respectively. The values of γ_i have been calculated from Equation 2.32.

Instead of the total volumetric enthalpy H, only the sensible enthalpy H_{sens} is used in the convection term, following König-Haagen et al. (2020) and Kaaks and Lathouwers (nd). H^{k+1} can then be expressed in T^{k+1} for the convection term via Faden et al. (2019):

$$H^{k+1}_{conv} = H^{k+1}_{sens} = \rho c_p T^{k+1} \ . \tag{2.43}$$

The temperature derivative of the enthalpy is approximated via a first order Taylor expansion:

$$H^{k+1} = H^k + \frac{\partial H}{\partial T}(T^{k+1,*} - T^k) \ . \tag{2.44}$$

Here, the asterisk (*) indicates the use of a predictor.

This expression is then plugged into Equation 2.42, yielding

$$\frac{\gamma_0 H^k + \gamma_1 H^o + \gamma_2 H^{oo}}{\Delta t} + \frac{\partial H}{\partial T}\frac{T^{k+1,*} - T^k}{\Delta t} + \nabla \cdot \left(\boldsymbol{u} \left(H^k_{sens} + \frac{\partial H_{sens}}{\partial T}(T^{k+1,*} - T^k) \right) \right) = \lambda \nabla^2 T^{k+1,*} \ . \quad (2.45)$$

Since $T^{k+1,*} - T^k \to 0$ upon convergence, the exact formulation for $\frac{\partial H}{\partial T}$ is not very important. The following expression has been chosen (Nedjar, 2002):

$$\frac{\partial H}{\partial T} = \frac{r}{2}(\rho_l c_{p,l} + \rho_s c_{p,s}) \ , \quad (2.46)$$

where r is an over- or underrelaxation parameter to improve the convergence speed of the algorithm.

The iterative scheme for solving the problem consists of the following steps:

1. Solve Equation 2.45 with a matrix solver. This is possible, as it is linear in $T^{k+1,*}$.
2. Calculate H^{k+1} from the new temperature field via Equation 2.44.
3. Force the temperature back on the T-H curve by rewriting Equation 2.34 as follows (Faden et al., 2019):

$$T(H) = \begin{cases} \dfrac{H}{c_{p,s}\rho_s} & H < H_s \ ; \\ T_m & H_s \leq H \leq H_l \ ; \\ T_m + \dfrac{H - H_l}{c_{p,l}\rho_l} & H > H_l \ , \end{cases} \quad (2.47)$$

   where H_s is the volumetric enthalpy of the solid PCM at the melting point. Likewise, H_l is the volumetric enthalpy of the liquid PCM at the melting point. This step is necessary to correct for cells skipping the phase change.

These steps are repeated until the convergence criterion is reached.

The liquid fraction ℓ of the PCM can be calculated via

$$\ell(H) = \begin{cases} 0 & H < H_s \ ; \\ \dfrac{H - H_s}{\rho_s L} & H_s \leq H \leq H_l \ ; \\ 1 & H > H_l \ . \end{cases} \quad (2.48)$$

With respect to the SBM, the advantage of the LEM is its quick convergence. An advantage with respect to the AHCM are the LEM's inherent energy-conservativeness and the fact it does not depend on a mushy zone, which is unphysical for pure materials and eutectic salts. In this thesis, the LEM was chosen as the phase change model.

2.9. Darcy source term

The equations of motion (Equations 2.4 and 2.7) are valid only for fluids. However, in a phase change problem also solids are present. Because of this, the Navier-Stokes equation is adapted:

$$\frac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}\boldsymbol{u}) = -\nabla p + \nabla \cdot \underline{\underline{\tau}} + \rho \boldsymbol{g} + \boldsymbol{A} \ . \quad (2.49)$$

$\boldsymbol{A}$ is the Darcy source term. Its purpose is to force $\boldsymbol{u} = \boldsymbol{0}$ if the PCM is locally solid, while not having an effect on the NSE when the PCM is locally liquid. $\boldsymbol{A}$ is given by (Brent et al., 1988):

$$\boldsymbol{A} = -D\frac{(1-\ell)^2}{\ell^3 + \varepsilon}\boldsymbol{u} \ . \quad (2.50)$$

D represents the Darcy constant and ε is a small constant preventing errors when $\ell = 0$. Following Reus (2021), in this work the values $D = 10^{10}$ kg m^{-3} s^{-1} and $\varepsilon = 10^{-3}$ have been chosen.

Other methods to fulfil the goal of the Darcy source term exist, e.g. the switch-off technique, which abruptly switches off the velocity when the PCM is solid (Voller et al., 1987). The switch-off technique leads to big gradients, producing instabilities, oscillations, and convergence issues (König-Haagen et al., 2017). Another method is the variable viscosity method, which progressively increases the fluid's viscosity to very high values in its solid state (Gartling, 1978). The variable viscosity method speeds up phase change due to numerical diffusion on the solid-liquid boundary (Voller and Cross, 1981) and has no physical basis but is a workaround to achieve the goal of setting the velocity to zero in a solid. As the results generated with the Darcy source term method have shown to be better than the other two methods, it is used in this thesis.

3

Solvers and benchmark cases

The effect of the inclination angle of the freeze-plug has been studied through CFD. To validate the working of the code, two benchmark cases have been created. These are explained in this chapter.

3.1. Solvers

Two separate solvers have been used in this thesis, both developed for *OpenFOAM 8*. *OpenFOAM* stands for Open Field Operation And Manipulation and is an open source software package for CFD using the FVM. The first code, *phaseChangeFoam*, is an inhouse code originally developed by Reus (2021) for *OpenFOAM 7* and adapted by Bouke Kaaks for *OpenFOAM 8*. To adapt it to the latest insights and to solve some inaccuracies, several changes have been implemented during this work as well. Examples of changes include the correct implementation of convection, and simplifying the case setup by reducing the amount of input files necessary to run the solver. *phaseChangeFoam* is based on the *pimpleFoam* solver and simulates phase change. The other code is *CHT_PhaseChangeFoam*, also developed by Reus (2021) and adapted for *OpenFOAM 8* and improved by Bouke Kaaks. The Author's contribution was testing the code and providing feedback. *CHT_PhaseChangeFoam* is based on the *chtMultiRegionFoam* solver, with the same way of implementing phase change as *phaseChangeFoam*; i.e. by tracking enthalpy and using the Linearised Enthalpy Method to calculate the temperature from said enthalpy. Besides phase change, it does account for conjugate heat transfer (CHT), which is heat transfer between different materials such as a salt and a metal or two different metals. This was achieved by splitting up the mesh into several different domains, with each domain corresponding with one material. On the interface, a coupling boundary condition ensured heat could flow between the materials. The domains were solved sequentially, with the liquid domains being solved before the solid ones.

Worth noting is the fact *CHT_PhaseChangeFoam*'s calculation of ℓ was slightly different from the one from Equation 2.48: ℓ was set to 0 for all $H \leq H_l$, so also if the PCM was undergoing phase change. This was done to better mimic physical behaviour, as ℓ can be only 0 or 1 in reality as well. This is especially important for non-eutectic materials, which use a mushy zone; this way of defining ℓ helps with possible future implementation of non-eutectic materials. This way of implementing ℓ would force the velocity to become zero during phase change. Considering the calculations in the LEM are based on H rather than ℓ, this method of calculating ℓ was not problematic.

Unless specified otherwise, the cases were all run with the same numerical schemes and tolerances. For temporal discretisation, the BDF2 scheme was used, while gradients were discretised with a linear scheme. For the convective terms, two different schemes were used: the divergence of velocity was calculated via a first order upwind scheme, the divergence of temperature via the limited linear upwind scheme with $\xi = 0.5$. This combination showed to produce the best results.

The equations of motion were solved using PIMPLE. In the PIMPLE loop, the temperature and velocity fields were solved with a multigrid solver using the symmetric Gauss-Seidel smoother. Pressure was solved via the Preconditioned Conjugate Gradient (PCG) solver with the Simplified Diagonal-based Incomplete Cholesky (DIC) preconditioner. Convergence was assumed for all fields when the tolerance dropped below 10^{-8}. The energy iteration of the phase change model was performed until the tolerance dropped below 10^{-6} or until 50 iterations, whichever occured first; in practice, the former criterion was satisfied first almost always.

3.2. Inclined gallium melting with *phaseChangeFoam*

3.2.1. Inclination

Method

The first benchmark case was simulating the melting of gallium in an inclination and was used to test the validity of the phase change implementation. Gallium is often used as a benchmark PCM because of its easy accessibility, safety, and melting point close to room temperature, allowing easy physical experiments. Because of this, many experimental results exist to assess the working of numerical methods.

In the benchmark, the experiment of Zennouhi et al. (2017) has been performed as it was a straightforward simulation of an inclined phase change material. Both Zennouhi et al.'s and the Author's simulations are 2D simulations of the melting of gallium in a rectangular cavity with length $l = 120$ mm and width $w = 50$ mm. In this work, the mesh was divided in 240x100 cells. The gallium originally was solid with a temperature of $T_c = 288.15$ K. The right wall was kept constant at $T_h = 308.15$ K, above gallium's melting point. The left wall, bottom and top were kept adiabatic. A visual representation of the case is given in Figure 3.1. As is the case with all other simulations, the Finite Volume Method was used in combination with the PIMPLE algorithm and the Linearised Enthalpy Method. The simulation was run for inclination angles $0°$, $30°$, $45°$, $60°$, and $90°$, where $0°$ was the case where gravity was aligned with the $-\hat{\boldsymbol{z}}$ direction. Positive inclination angles corresponded with clockwise rotation according to Figure 3.1, i.e. the case with $\theta = 90°$ had the hot wall below.

The solver used in this simulation was *phaseChangeFoam*.

OpenFOAM has a builtin time step control that lets the user specify the time step based on a maximum allowed Courant number. This allows for a faster simulation, as larger time steps can be made when little flow is present while maintaining physical results once more flow takes place. However, this time step control is not perfect as sometimes the maximum Courant number is higher than the maximum allowable one. Because of this, $Co_{max} = 0.8$ has been chosen via trial-and-error for time control. This ensured that the Courant number would never be higher than 1.

The thermophysical properties are given in Table 3.1. Simulation parameters and other data are given in Table 3.2.

Table 3.1: Thermophysical properties of gallium (Zennouhi et al., 2017).

	solid	liquid
T_m [K]	302.93	
L [J kg^{-1}]	$8.016 \cdot 10^4$	
ν [m^2 s^{-1}]	N/A	$2.9706 \cdot 10^{-7}$
β [K^{-1}]	N/A	$1.2 \cdot 10^{-4}$
λ [W m^{-1} K^{-1}]	32	32
c_p [J kg^{-1} K^{-1}]	381.5	381.5
ρ [kg m^3]	5907	6093
Pr [-]	N/A	0.0216

Results

At an inclination of $0°$, in the beginning, heat transfer takes place only via diffusion as too little gallium has molten for convection to take place. Around $t = 80$ s, vortices start to form and convection plays a role in the heat transfer. At the beginning, there are seven vortices, but they gradually merge until two are present. At the end of the simulation, the velocity behaviour is chaotic and the flow profile changes rapidly. With $Ra = 9.9 \cdot 10^6$, the flow is laminar. For higher inclinations, the same behaviour is visible, except Ra scales with $\cos\theta$ as explained in the theory.

The exception to this is the case of $90°$ inclination. This is because in this case, the case cannot be described as a vertical hot wall (see Section 2.4). For this case, the liquid fraction first lags behind the ones of the other angles, as vortices are not developed so heat transfer can take only place via diffusion. Around $t = 450$ s, vortices start to form and because of the geometry (heating from the bottom) the effect of convection is much stronger than for the other inclinations. The fact that at $90°$

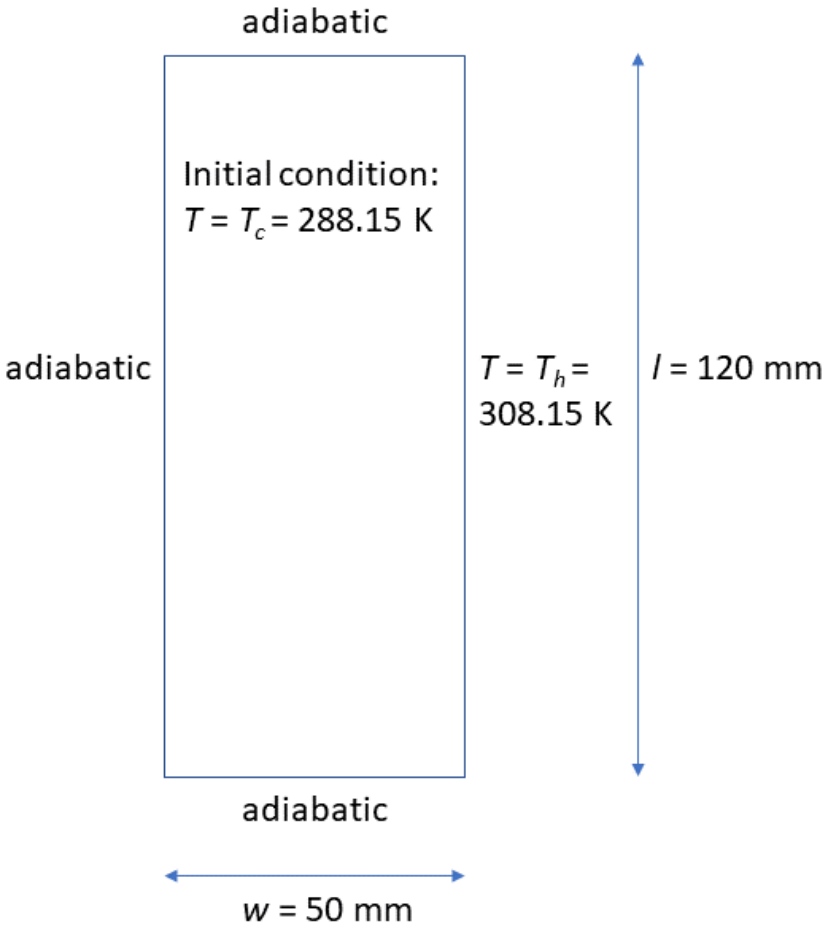
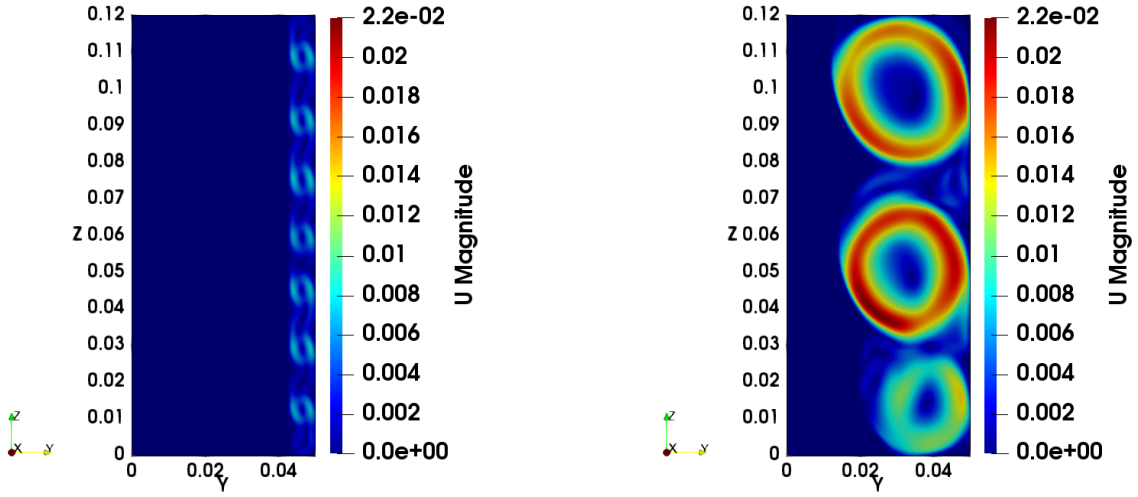


Figure 3.1: The setup of the inclined gallium melting case with *phaseChangeFoam*.

Table 3.2: Relevant parameters and simulation data for the inclined gallium melting simulations.

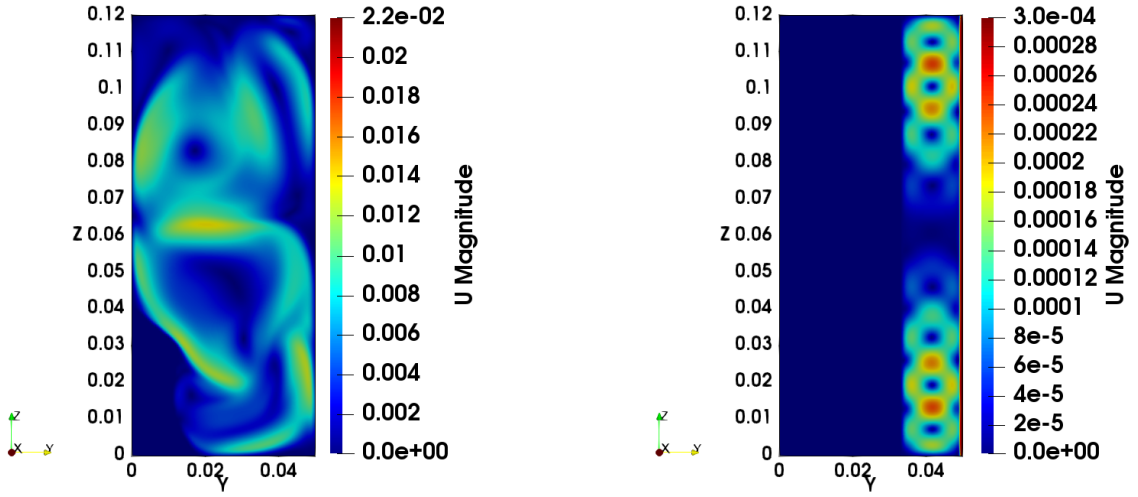
	This work	Zennouhi et al. (2017)
l [mm]	120	
w [mm]	50	
T_c [K]	288.15	
T_h [K]	308.15	
N_l [-]	240	unknown, but $N_l N_w = 92000$
N_w [-]	100	
Δt	such that $Co_{max} = 0.8$	10^{-4} s
numerical method	Finite Volume Method	
algorithm to solve equations of motion	PIMPLE	unknown
phase change model	Linearised Enthalpy Method	Source-Based Method
velocity switch-off technique in solid	Darcy source term	
D [kg m$^{-3}s^{-1}$]	10^{10}	10^{15}
ε	0.001	
θ	0°, 30°, 45°, 60°, 90°	0°, 30°, 60°, 90°

the gallium first melts slowly but then suddenly quickly is in qualitative accordance with Zennouhi et al. (2017), although the effect measured in this work is much bigger than in Zennouhi et al. (2017).



(a) $\theta = 0^\circ, t = 100$ s

(b) $\theta = 0^\circ, t = 850$ s



(c) $\theta = 0^\circ, t = 1400$ s

(d) $\theta = 90^\circ, t = 450$ s

Figure 3.2: The different types of fluid flow. At the beginning, seven small vortices started forming (a). These then gradually merged into three bigger ones (b). Even later, these would again merge into two vortices. At the end of the simulation, the behaviour was much more chaotic (c). At 90°, the behaviour was much different, as vortices started to form much later and in different ways than in the other cases (d); note the different scaling of the colour bar.

A plot of average liquid fraction as a function of time, for all angles, is given in Figure 3.3. Both results from this work and the one of Zennouhi et al. are present. Some snapshots of the flow field are present in Figure 3.2. It can be seen that Zennouhi et al. gives much faster melting than this work. In the simulation of Zennouhi et al. (2017), the melting rate increases with decreasing angle; in this work, this is not the case for $\theta = 90^\circ$, but it is true for the other four inclination angles. Since the case with $\theta = 90^\circ$ has significantly different melting behaviour, as described earlier, the different behaviour of the liquid fraction is not surprising.

The reason for the mismatch between the results of Zennouhi et al. (2017) and this work is unknown, but is perhaps due to the mesh being relatively coarse in this work's simulations. Another possibility is

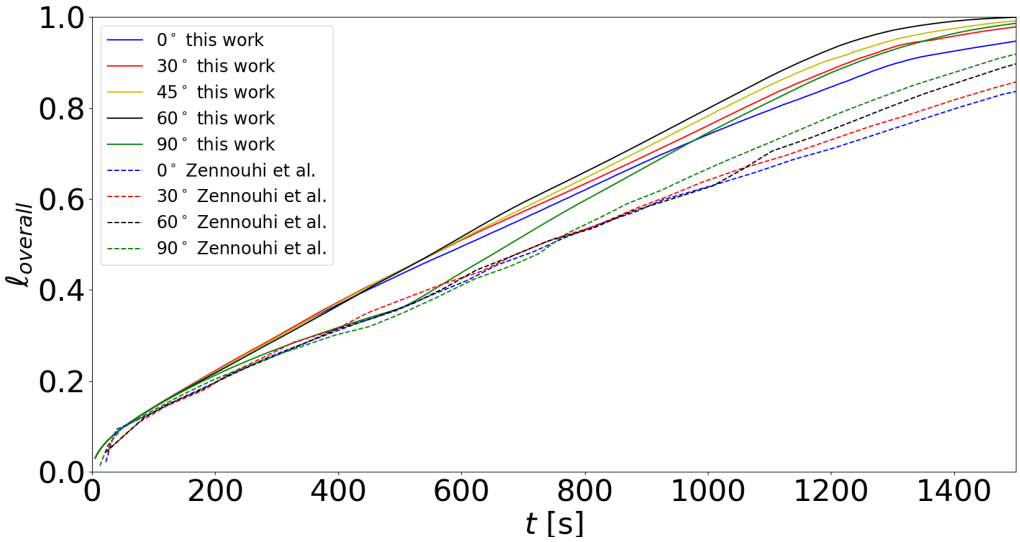


Figure 3.3: The average liquid fractions for different inclination angles from this work and Zennouhi et al. (2017).

the usage of the SBM as phase change model in the simulation of Zennouhi et al. (2017).

3.2.2. Multithreading

Method

OpenFOAM allows multithreading in simulations, which is running multiple tasks simultaneously. The method of multithreading - also called parallel processing - in *OpenFOAM* is domain decomposition. This means the geometry and associated fields are split up into several subregions, where every subregion is assigned to a separate processor. The case is then run in parallel on each processor, and in the end the mesh is recomposed and post-processed. As every processor has only a part of the domain to simulate, this significantly speeds up the simulation.

Reus (2021) encountered some problems regarding the parallelisation of the phase change simulations. Because of the massive computational benefit, resolving this issue was one of the goals of this thesis. One of the changes made to optimise the working of *phaseChangeFoam* and *CHT_PhaseChangeFoam* appeared to have ensured multithreading is always present, although it is not certain which change it was. Running a multithreaded case with *phaseChangeFoam* worked the same way as doing so with *pimpleFoam*, the solver *phaseChangeFoam* is based on. Likewise, running *CHT_PhaseChangeFoam* in multiple threads worked the same way as *chtMultiRegionFoam*.

After multithreading was incorporated in the solver, it was tested. Three separate simulations were run for the case with $\theta = 90°$: on one, four and six processors. In the case with four processors, the computational domain was decomposed in two parts in both length and width. In the case with six processors, the length was split in three parts and the width in two, bringing the length:width ratio of each partial domain closer to 1:1 than with any other configuration.

Worth noting is that the purpose of the different simulations was to test the validity of the parallel execution of the decomposed case. The difference in run time was of less importance, since parallel processing has a small or even negative effect for meshes with few cells as the one in question. This is because the overhead costs of communication between the different processors are higher than the gain of each processor having a smaller domain to compute. The reason multithreading was tested was the fact that the freeze plug described later in this chapter was a much bigger domain where multithreading was expected to have a major advantage. Multithreading was tested for this case instead of the freeze plug as this case was small enough such that run time on one, four and six cores were all reasonable.

Results

When multithreading, at the beginning the melting process happened the same way for one and for six threads. However, when vortices started forming, the simulation started changing. The reason for this is unknown, but it is possible the phenomenon occured due to numerical diffusion at a processor boundary when the flow field intersected with it, i.e. when particles crossed from one processor domain to another[1].

The difference between the average liquid fractions of the simulation on 1 core on the one hand and 4 or 6 cores on the other hand are found in Figure 3.4.

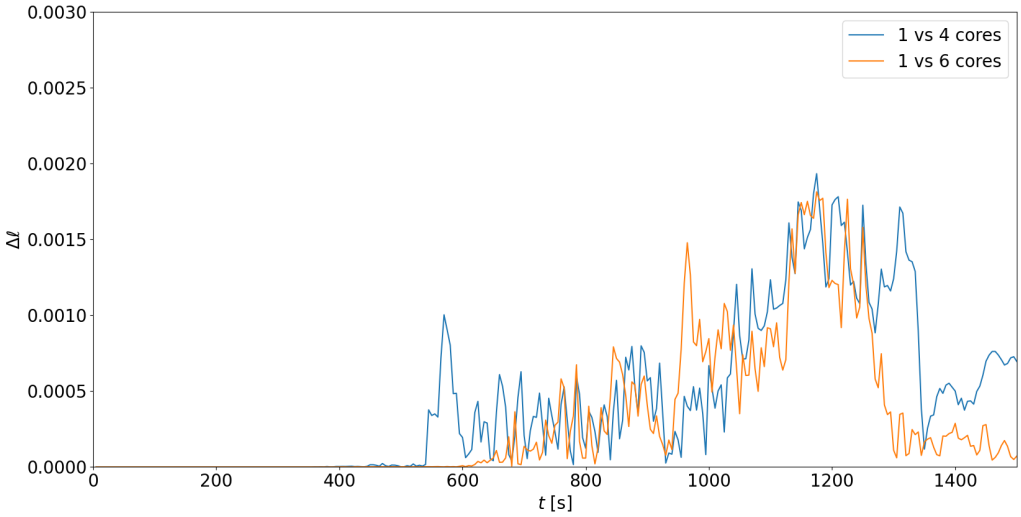


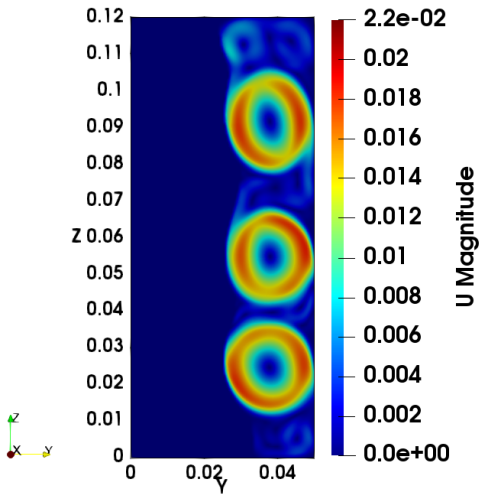
Figure 3.4: The absolute difference in average liquid fraction for the gallium melting with *phaseChangeFoam*.

The discrepancy between the simulations with different numbers of cores started mostly when vortices merged, as the behaviour was chaotic at that time and point. Because of this, the effect of a small initial variation quickly increased. This is shown in Figure 3.5.

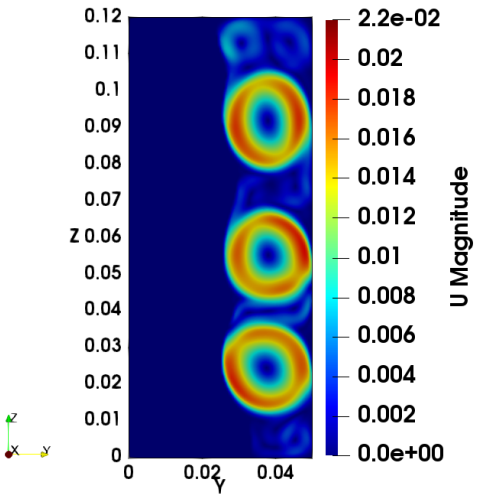
The relevance of the difference between the cases on one and multiple processors is limited. This is due to the fact that $\max(\Delta\ell) = 0.02$, and the time difference to overcome this is approximately 5 seconds, as can be seen in Figure 3.6. Also, qualitatively the same melting behaviour - vortex forming and merging - is visible for all numbers of cores.

Noteworthy is the fact that the error due to multithreading decreased when the matrix tolerances were set stricter. However, this increased the run time of the simulation, diminishing the effect of the multithreading, i.e. speeding up the simulation.

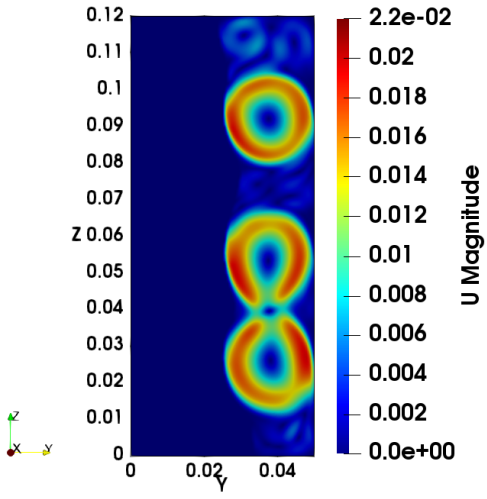
[1] As no Lagrangian particle tracking is performed during this thesis, there are no single particles that could potentially cross a processor boundary. Therefore, this formulation is only a conceptual one.



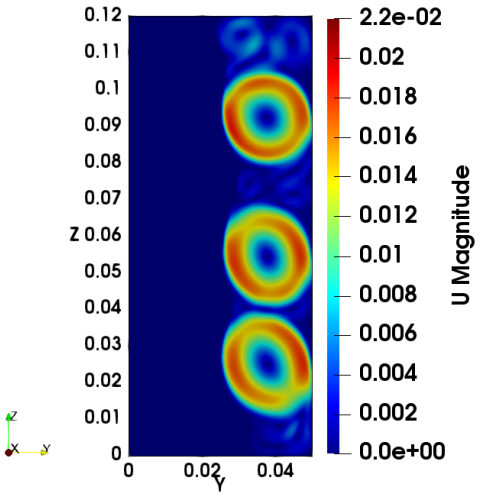
(a) $0°, t = 535$ s, single core



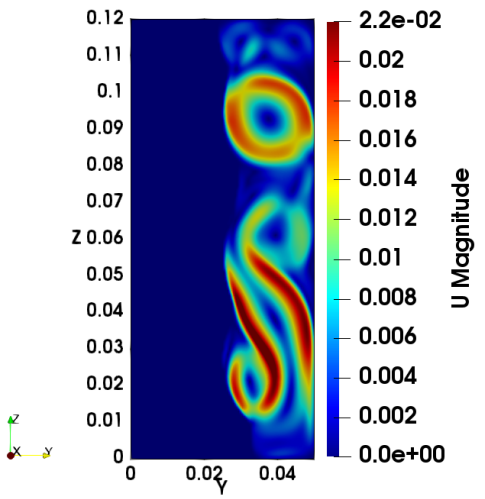
(b) $0°, t = 535$ s, four cores



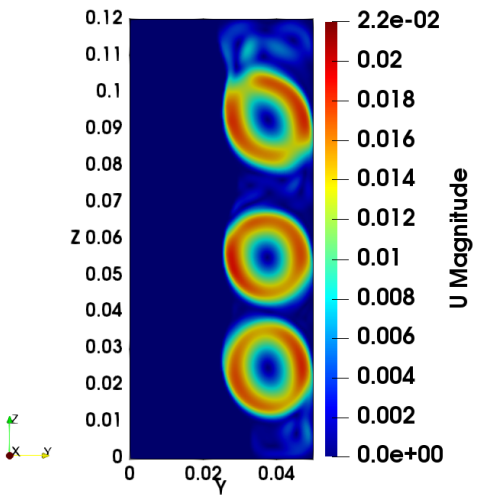
(c) $0°, t = 540$ s, single core



(d) $0°, t = 540$ s, four cores



(e) $0°, t = 545$ s, single core



(f) $0°, t = 545$ s, four cores

Figure 3.5: Several flow fields. At $t = 535$ s, the flow fields are almost equal for one and four cores (a and b). At $t = 540$ s, on one core, the start merging (c), while on four cores they remain (d). At $t = 545$ s, the flow fields are completely different (e and f). These times correspond with the first 4-core peak of Figure 3.4.

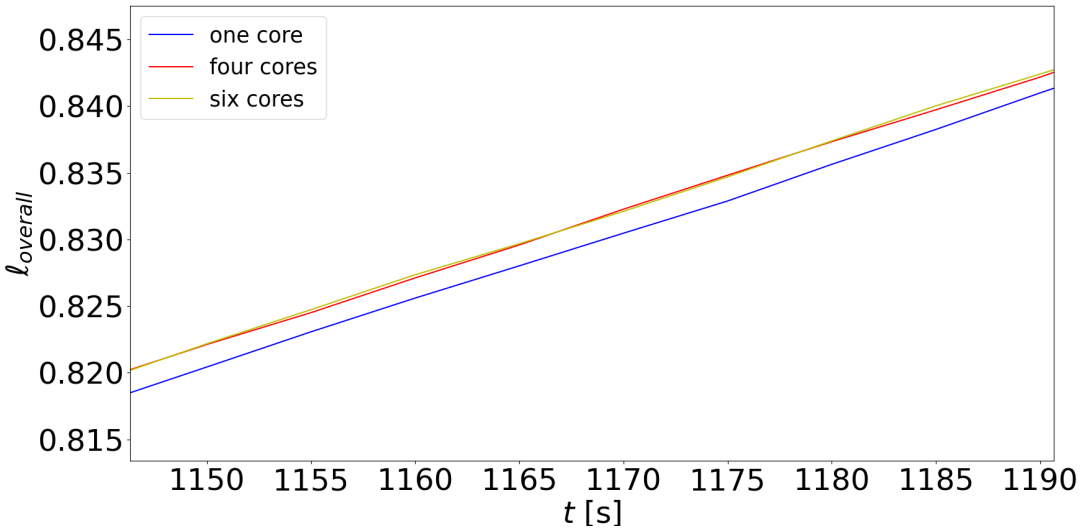


Figure 3.6: The average liquid fractions at 1, 4, and 6 cores, zoomed in around $t = 1170$ s.

3.3. Fin in air

Method

The other case used to validate the working of the code was a 2D simulation of a copper fin in air. This code was run with *CHT_PhaseChangeFoam*, but this time conjugate heat transfer was present, while phase change was not. The aim of this simulation was testing the implementation of the CHT.

The domain of the simulation was a 2D square cavity with $l = w = 100$ mm, divided into 100x100 cells. A copper fin was placed at the bottom of the cavity. The fin consisted of a rectangle between 20 mm $\leq y \leq 60$ mm, $z \leq 10$ mm, as well as a rectangle between 30 mm $\leq y \leq 40$ mm, 10 mm $< z < 50$ mm. The two copper rectangles thus formed an upside-down skewed T shape. The rest of the domain consisted of air. The gravity vector was pointing in the $-\hat{z}$ direction. The setup is visualised in Figure 3.7.

A case with finer domain was also run. It was set up the same way as the previously described case, except it consisted of 200x200 cells. This domain was run in parallel over sixteen processors, with both the y and the z direction being split up in four parts.

Initially, both the copper and the air had a temperature of $T_c = 293$ K. The bottom had a constant temperature of $T_h = 343$ K for the copper part of the boundary and was adiabatic for the air part. The top was also adiabatic, while the sides were kept at a constant $T_c = 293$ K. The thermophysical properties of copper and air are given in Table 3.3.

Table 3.3: The thermophysical properties of air and copper used in the benchmark case of the copper fin in air.

	air	copper
ν [m^2 s^{-1}]	$1.454 \cdot 10^{-3}$	N/A
β [K^{-1}]	$3.32 \cdot 10^{-3}$	
λ [W m^{-1} K^{-1}]	$2.60 \cdot 10^{-2}$	401
c_p [J kg^{-1} K^{-1}]	1003.62	377
ρ [kg m^{-3}]	1.276	8960

After the simulation was run, at $t = 2500$ s, the temperatures in the air have been sampled over the whole z length at $y = 15$ mm. They have been compared with the results of the same case run in *DGFlows*. This is a solver based on the discontinuous Galerkin method, an application of the Finite Element Method.

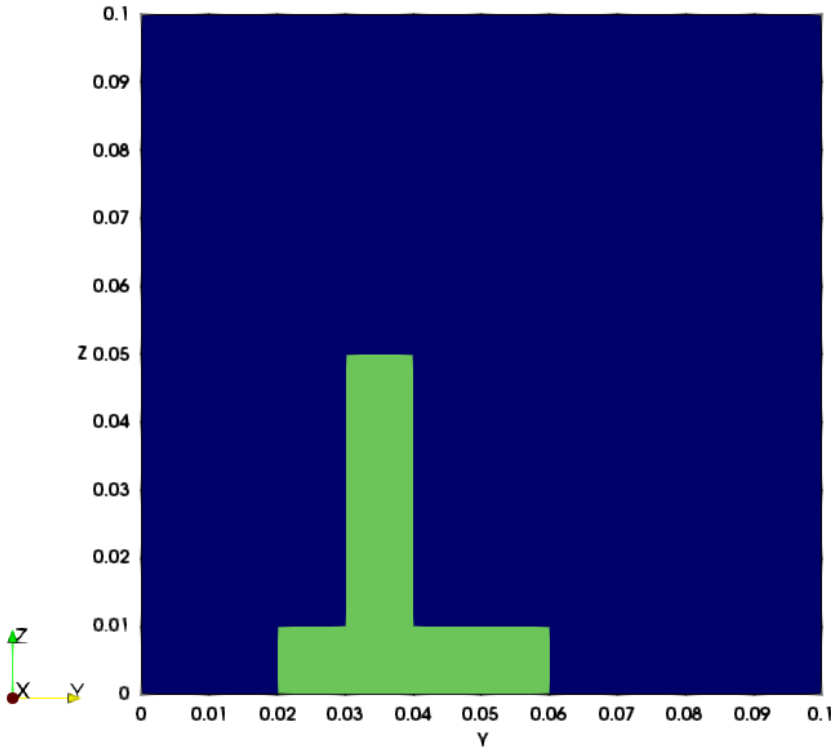


Figure 3.7: The geometry of the fin in air case. Air is shown as blue, copper as green. Coordinates are given in metres.

Results

A plot of the temperatures at $y = 15$ mm is found in Figure 3.8. These are compared with the results from *DGFlows*. Qualitatively, DGFlows and the *OpenFOAM* solver yield the same results, with the temperatures following roughly the same pattern using both solvers. However, the quantitative results are slightly different, with the *DGFlows* temperature profile being slightly broader, with higher maxima and lower minima than the 100x100 cell *OpenFOAM* case. The maximum difference is $\Delta T_{max} = 0.7$ K. The 200x200 cell mesh shows interesting behaviour: for $z < 10$ mm, the *OpenFOAM* case yields almost equal results to *DGFlows*. For $10\text{mm} < z < 45\text{mm}$ the results follow the result of the 100x100 cell *OpenFOAM* simulation. Then the results of the two *OpenFOAM* results start diverging until $z = 70$ mm, and the 200x200 cell mesh result comes closer to the *DGFlows* result. Then, for $z > 70$ mm, the results for the 200x200 *OpenFOAM* mesh are virtually equal to *DGFlows*. The maximum deviation between the *OpenFOAM* solution on 200x200 cells and the solution of the *DGFlows* simulation is $\Delta T_{max} = 0.2$ K, which is not much.

The temperature contour on the 100x100 cell mesh at $t = 2500$ s from the *OpenFOAM* solver is given in Figure 3.9.

For the 200x200 cell mesh, the results only slightly deviate from the *DGFlows* simulation. Possibly, the discrepancy would be even lower for even finer meshes, as there is no certainty the simulation on the 200x200 cell mesh was mesh convergent. Another possible explanation for the small difference is an inconsistency in the case setup. Therefore, it is plausible that the implementation of the conjugate heat transfer is correct in the *CHT_PhaseChangeFoam* code.

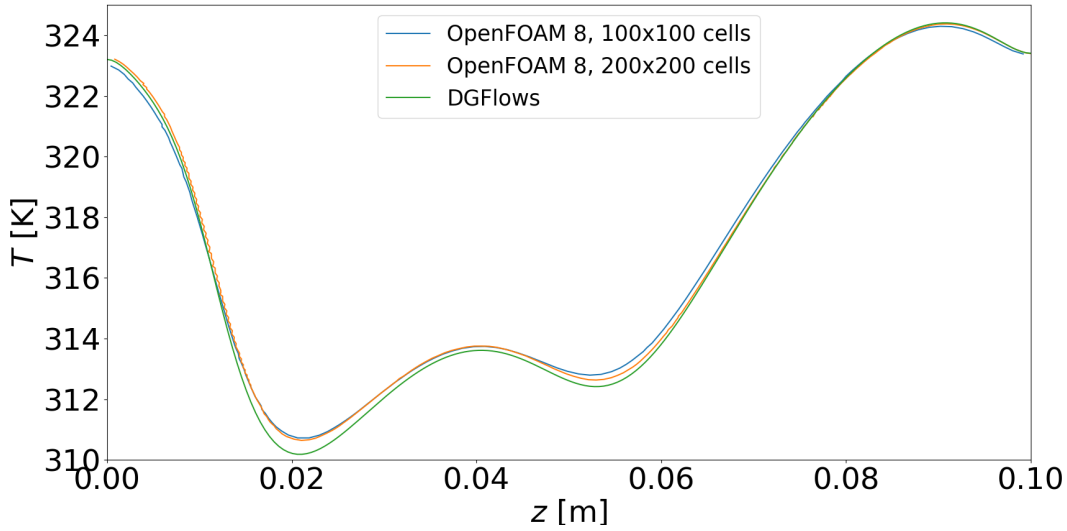


Figure 3.8: The temperatures at $y = 15$ mm of the fin in air case.

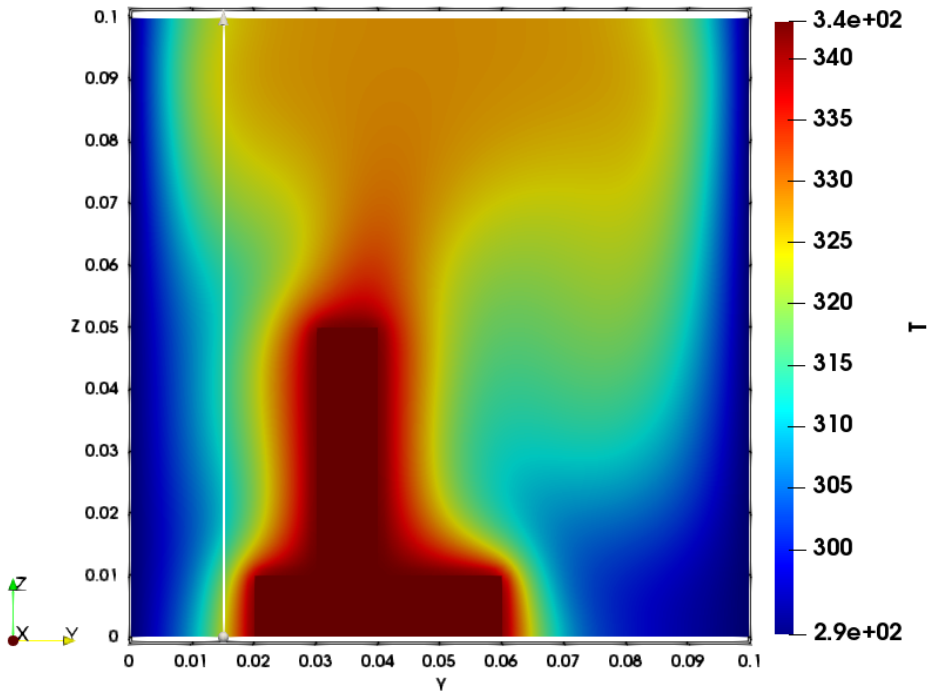


Figure 3.9: The temperature contour of the fin in air case at $t = 2500$ s on the 100x100 cell mesh. The line $y = 15$ mm, where the temperature is sampled, is indicated in white.

4

Freeze plug

After the benchmark cases, the final and main simulations of this thesis were performed. In this chapter, first the geometry and materials used for the freeze plug simulation are described, followed by the method and results.

4.1. Geometry and materials

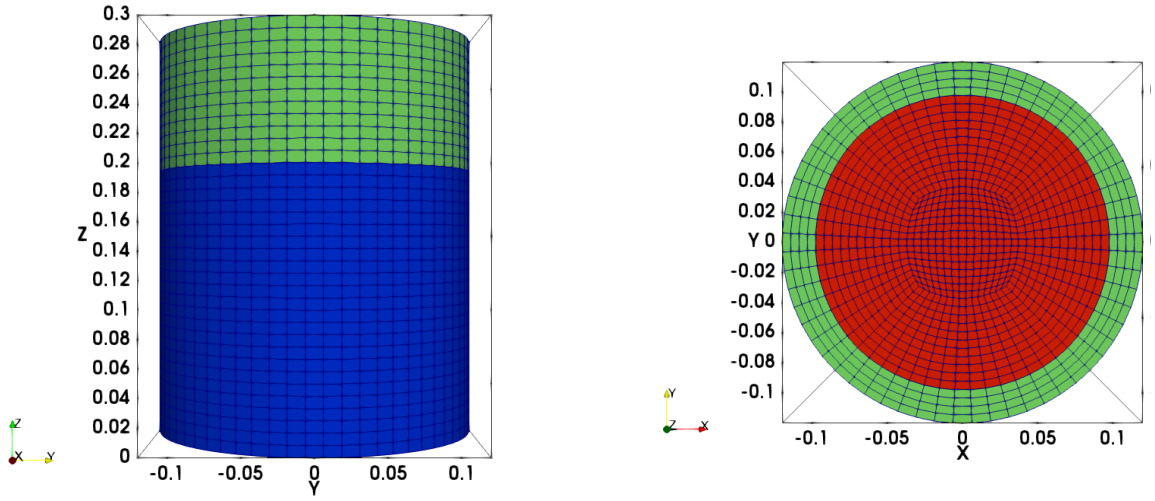
The final and main simulation in this thesis was the freeze plug. For the freeze plug, one of the designs from Tiberga et al. (2019) was chosen, which was also used in the thesis of Reus (2021). The freeze plug consisted of a salt cylinder with a radius of 100 mm and a height of 300 mm. The salt in question was LiF-ThF_4. This salt has similar properties to LiF-ThF_4-UF_4 but does not contain uranium, making it suitable in terms of safety, security, and licensing for experiments (Ghetta et al., 2017).

The cylinder was surrounded by two 20 mm thick annuli on top of each other. The lower 200 mm of the salt cylinder were surrounded by copper, which mimicked the cooling system of the freeze plug. The top 100 mm was surrounded by Hastelloy-N, simulating a pipe section. Hastelloy-N is an alloy consisting of 71 mass-% nickel, 16 mass-% molybdenum, 7 mass-% chromium, and 6 mass-% other elements. It has excellent resistance against oxidation, corrosion, and high temperature, making it suitable for applications with the high-corrosive and high-temperature LiF-ThF_4 (Hayes International, nd). The material properties of LiF-ThF_4, copper, and hastelloy can be found in Table 4.1. The described domain can be seen in Figure 4.1.

Table 4.1: The material properties of LiF-ThF_4, Hastelloy-N, and copper. β of LiF-ThF_4 has been extracted from Gheribi et al. (2014), all other data from Shafer (2018).

	LiF-ThF_4		Hastelloy-N	Copper
	solid	liquid	solid	solid
T_m [K]	841		N/A	
L [J kg^{-1}]	$1.59 \cdot 10^5$			
ν [m^2 s^{-1}]	N/A	$2.278 \cdot 10^{-6}$		
β [K^{-1}]		$2.55 \cdot 10^{-4}$		
λ [W m^{-1}]	1.5	1.5	23.6	401
c_p [J kg^{-1} K^{-1}]	815	1000	578	377
ρ [kg m^{-3}]	4390	4502	8860	8960

The simulation of the freeze plug consisted of multiple steps: firstly a steady-state run on a coarse mesh, a fine mesh and an even finer mesh to simulate the freeze plug's shape during normal MSFR operation. Every steady-state result - except the one for the finest mesh - would act as the initial condition for the mesh one step finer. Finally, a transient simulation was performed to simulate melting.



(a) (b)

Figure 4.1: Front view (a) and top view (b) of the physical domain on the 40.5K mesh, the coarsest mesh used in this thesis. The red region is the salt, green corresponds to Hastelloy-N and blue represents copper. Coordinates are given in metres.

4.2. Mesh

First, a cylindrical mesh consisting of 40,500 cells was made via the *blockMesh* facility in *OpenFOAM*. This mesh is referred to in the remainder of this thesis as the 40.5K mesh. The cylinder's axis was placed on the line segment between $(x, y, z) = (0, 0, 0)$ and $(0, 0, 300)$ mm, while its radius was 120 mm. The length (z direction) was divided in 36 cells. The cross-section was built up from five pieces, one of which was an 'inflated square', with vertices at $(x, y) = (\pm 29, \pm 29)$ mm, and whose sides were arcs passing trough the points $(x, y) = (\pm 37, 0)$ mm and $(0, \pm 37)$ mm. The rest of the domain was split up into the four other pieces, the boundaries between which were the line segments between $(x, y) = (\pm 29, \pm 29)$ mm and $\left(\pm\frac{120}{\sqrt{2}}, \pm\frac{120}{\sqrt{2}}\right)$ mm. Each of the five pieces consisted of 15x15 = 1125 cells.

The reason for splitting up the mesh in five regions was the prevention of singularities - cells with near-zero volume. The inflated square shape was chosen for the central section for efficiency reasons: if a regular square was chosen, the mesh inside the square would be very fine while the mesh outside would be coarse. This would be ineffective considering the main region of interest was the salt area just near the wall, since melting was expected to take place mostly there (Reus, 2021, Shafer, 2018, Tiberga et al., 2019). The other extreme for the central section would be a circle, but this would lead to singularities.

There were two main reasons why the numbers of cells were chosen as described. Firstly, the number of cells in the z direction was to be divisible by three, such that the upper third of the 'annulus cells' could be assigned to be Hastelloy-N, while the lower two thirds would be allocated to copper. Furthermore, the average cell aspect ratio should be as close to 1:1:1 as possible. Because of this, and to keep run times reasonable while also being able to provide a good starting steady-state solution for the finer mesh, the values described above were chosen.

After the mesh generation, every cell was assigned one of three regions: salt, copper or Hastelloy-N, depending on its position. This was done in *OpenFOAM* via the *topoSet* and *splitMeshRegions* facilities. A representation of the mesh can be found in Figure 4.1. Finally, the mesh was decomposed in two parts in the x and y directions (so in four parts in total).

Important to consider during mesh generation was the fact that the assignment of cells to one of the three material regions was imperfect; sometimes, a cell would be assigned to a wrong region. An example of such a mismatch - leading to an unphysical simulation - is shown in Figure 4.2. The values of 29 and 37 mm (as described earlier in this subsection) were chosen by trial-and-error to prevent any

such mismatch.

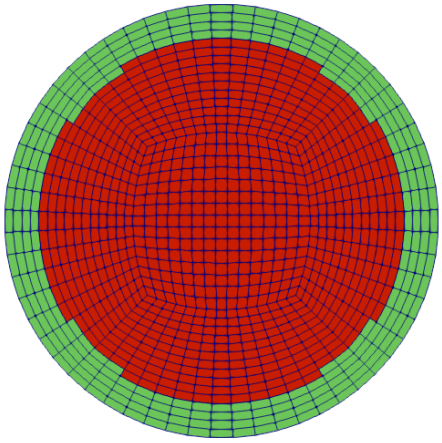


Figure 4.2: A mismatch between cells and regions, resulting in an incorrect geometry of the freeze plug.

4.3. Steady-state simulation on 40.5K mesh

Method

Once the mesh was generated, a steady-state simulation was started. Following Tiberga et al. (2019) and Reus (2021), initially, the freeze plug was set to have a temperature of $T(t = 0) = T_c = 821$ K. This is 20 K below the melting point of the salt of the freeze plug. As the simulation was ran until a steady-state situation was formed and only one steady-state solution was expected, the initial conditions were important only to ensure a quick run time, not for the steady-state solution. The top of the geometry, which corresponds to the bottom of the reactor during running, was kept at a constant $T = T_h = 923$ K. This is the average operating temperature of the MSFR according to Shafer (2018). The bottom as well as the side part of the Hastelloy-N were kept adiabatic, while the copper side was kept at a constant $T = T_c = 821$ K. This experiment was performed for each of the inclination angles $\theta = 0°, 15°, 30°, 45°, 60°, 90°$. The inclination angles were enforced by adjusting the gravity vector:

$$\boldsymbol{g} = -g \cos\theta \hat{\boldsymbol{z}} + g \sin\theta \hat{\boldsymbol{y}} \, . \tag{4.1}$$

The simulation ran for each inclination angle, with fields being saved every 5000 s. The time steps were chosen such that $Co_{max} = 0.8$ with an additional constraint that they could never exceed $\Delta t_{max} = 5$ s. This time step control would also be the case for the other simulations of the freeze plug described later in this section. The simulation lasted until a steady-state solution appeared to be present by visually inspecting the liquid fraction field mapped in *ParaView* for multiple different time steps. This is not a very accurate method, but that was not problematic considering this simulation was only performed to provide an initial condition for a finer mesh. The steady state for that mesh would also be analysed with other means as well, as explained later in this section.

Results

The freeze plug did not form at $\theta = 90°$, meaning this inclination angle is unsuitable for the MSFR, provided the given cooling and geometry. The case with $\theta = 90°$ is thus discarded in the remainder of the simulations. The liquid fraction fields for the steady-state results are given in Appendix C.

4.4. Steady-state simulation on 1.1M mesh

Method

After a steady-state solution was achieved on the 40.5K mesh, a new mesh was generated. This mesh was a 3x refinement of the 40.5K mesh in all three directions. As a result, the length was now divided

into 108 cells, while each of the five cross-section pieces now consisted of 45x45 = 2025 cells. The total mesh therefore consisted of $108 \cdot 5 \cdot 45 \cdot 45 = 1,093,500$ cells. This mesh is referred to in this thesis as the 1.1M mesh. The fields which resulted from the steady-state simulation on the 40.5K mesh were mapped on the 1.1M mesh via the *mapFields* utility in *OpenFOAM* for all inclination angles except $\theta = 90°$, as this yielded a non-forming freeze plug. The mesh was then decomposed to run on 48 processors; it was split in four parts in the x and y directions, and in three parts in the z direction.

After mapping and mesh-decomposing, the same steady-state simulation was ran as at the 40.5K mesh, except resulting fields were saved every 250 s. This was again stopped at steady-state. However, now a set of probes tracking temperature was defined on the locations as shown in Table 4.2 and Figure 4.3. Because of symmetry in the x direction, probes were only placed at places with nonnegative x coordinate. The temperatures were analysed and the simulation was stopped whenever the temperatures were seen to be constant over a longer range of time.

Table 4.2: Locations of the probes in the freeze plug.

Probe number	x [m]	y [m]	z [m]
1	0	0	0.0001
2	0	0.09	0.0001
3	0	-0.09	0.0001
4	0.09	0	0.0001
5	0	0	0.1
6	0	0.09	0.1
7	0	-0.09	0.1
8	0.09	0	0.1
9	0	0	0.2
10	0	0.09	0.2
11	0	-0.09	0.2
12	0.09	0	0.2
13	0	0	0.2999
14	0	0.09	0.2999
15	0	-0.09	0.2999
16	0.09	0	0.2999

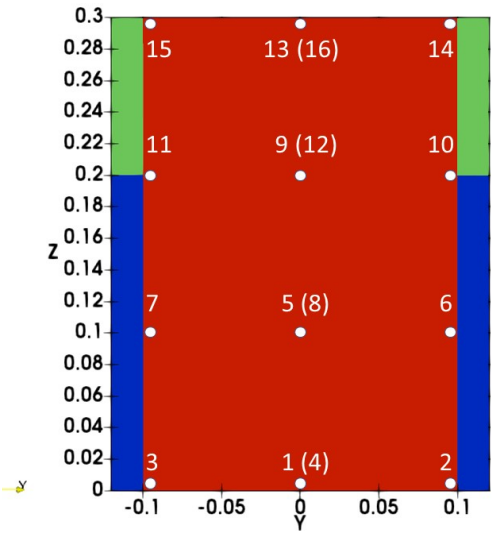


Figure 4.3: A cross-section of the freeze plug geometry. The red area indicates the salt, green is Hastelloy-N and blue corresponds to the copper part. The locations of the probes in the freeze plug simulation, as described in Table 4.2, are indicated. All probes are located at $x = 0$ m, except the ones noted in brackets which are located at $x = 0.009$ m.

Results

The steady-state freeze plugs on the 1.1M mesh are shown in Figure 4.4. As can be seen, there was a big discrepancy between the steady-state simulations for the freeze plug on the 40.5K and 1.1M meshes. The biggest difference was the fact that now the freeze plug at $\theta = 60°$ did not form, so this is not a valid inclination angle for the freeze plug. The case with $\theta = 60°$ is therefore also not taken into account in further simulations. Important to note is the fact this is valid only for the given parameters - with higher subcooling or for a longer freeze plug, formation would take place. This can be seen by the difference with the inclined TMSR freeze plug design, with different dimensions, subcooling and materials: Aji (2020) found that that freeze plug even forms up to an inclination angle of $\theta = 120°$, at which the 'top' of the freeze plug is underneath its 'bottom'.

Interesting to see is the fact that for all inclination angles, near the copper and Hastelloy-N wall, the solid-liquid interface forms close to $z = 0.2$ m, close to the boundary between the copper and Hastelloy-N. This is due to the fact that the copper is actively cooled below the salt's melting point, while the Hastelloy-N has a higher temperature during steady-state due to the heating from the top of the geometry.

Also notable is the fact that the solid-liquid interfaces are roughly horizontal[1]. This is mainly visible near the central axis of the freeze plug (the z axis in the model), as it is furthest away from the boundary between the copper and the Hastelloy-N, which has the same z coordinate in the model but due to the inclination is not horizontal for the inclined freeze plugs. The central axis therefore undergoes the least influence from the copper-Hastelloy-N boundary, allowing for a near-horizontal boundary.

The reason $x = 0$ is pictured is the fact it is representative for the freeze plug as a whole, so visual inspection is easy to do. When looking at the outside of the freeze plug, similar results are shown as with the cross section, but it has the disadvantage of only showing boundary cells and skewing data due to its curvature. This is shown - for a mesh of 3.6 million cells discussed later in this section - in Figure 4.5

Some temperature values at the probes for valid inclinations can be seen in Figure 4.6. For every inclination angle except $60°$ and $90°$, at which a freeze plug did not form, two figures have been made. One figure shows the temperature during the run at the bottom eight probes, the other one does the same for the upper eight probes. This division has been made to improve the clarity of the graphs. At the beginning of the run, the temperature values often drop significantly, but at longer times they stabilise. Note the scaling of the horizontal (temperature) axis differs significantly from plot to plot.

Interesting to see is the fact that some probes' temperatures do not become constant, but show periodic behaviour. This is mainly seen for probes 11 and 14 for $\theta = 15°$ and $30°$. At those points, the flow field is high, as can be seen in Figure 4.7. With an estimated $Ra = 2.2 \cdot 10^8$ in the freeze plug, the flow is turbulent. This means the 'steady-state' situation in fact is not a true steady-state. For simplicity and lack of better term, the - albeit sloppy - terminology 'steady-state' is used in the remainder of this thesis.

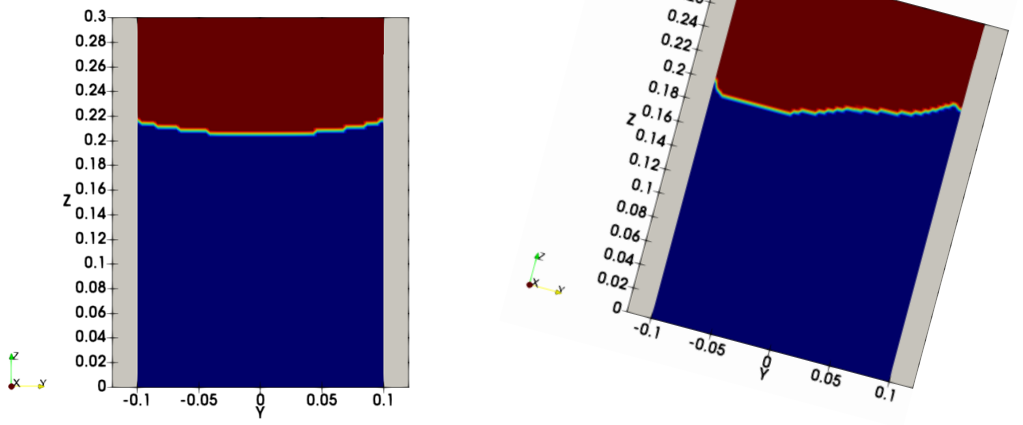
Zooming in on the temperatures at probes 11 and 14 at $\theta = 15°, 30°$ it is visible that the temperatures follow a periodic behaviour. This is shown in Figure 4.8. The top and bottom temperatures vary, but the period is constant; about 4.8 s for $\theta = 15°$ and 4.4 s for $\theta = 30°$. The graph's shape is roughly constant as well.

Another interesting graph is the one of probe 7 at $45°$. It does not form a single horizontal line but multiple smaller ones. This is due to the fact phase change is taking place at the probe; note the proximity of the graph to $T_m = 841$ K. Apparently there is no true steady-state for that probe, so once again there is a quasi-steady-state.

The average liquid fractions - both overall and near the hastelloy/copper wall - can be found in Table 4.3 and Figure 4.9. As can be seen, the liquid fraction decreases with θ, and this effect is stronger for the wall liquid fraction.

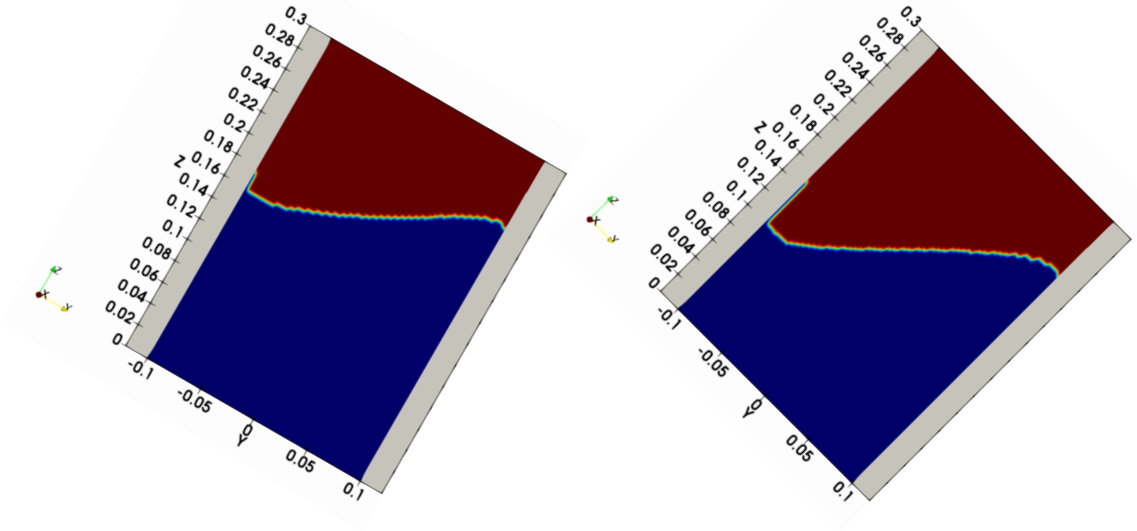
The solid-liquid boundary compared with the ones found by Reus (2021) and Shafer (2018) can be found in Figure 4.10. It can be seen that the shape is quite consistent with Reus (2021). The fact it is stepwise instead of smooth is due to the mesh quality. The location differs slightly, but that is possibly

[1] With 'horizontal', it is meant that the normal of the interface is (anti)parallel to the gravity vector $\boldsymbol{g}$, not the z axis.



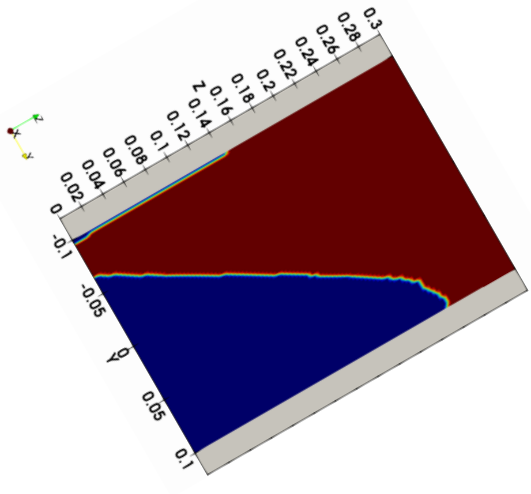
(a) 0°

(b) 15°



(c) 30°

(d) 45°



(e) 60°

Figure 4.4: The steady-state situations for all inclination angles on the 1.1M mesh. The liquid salt is indicated with red, the solid with blue. The freeze plug does not form at $\theta = 60°$, contrary to the 40.5K mesh.

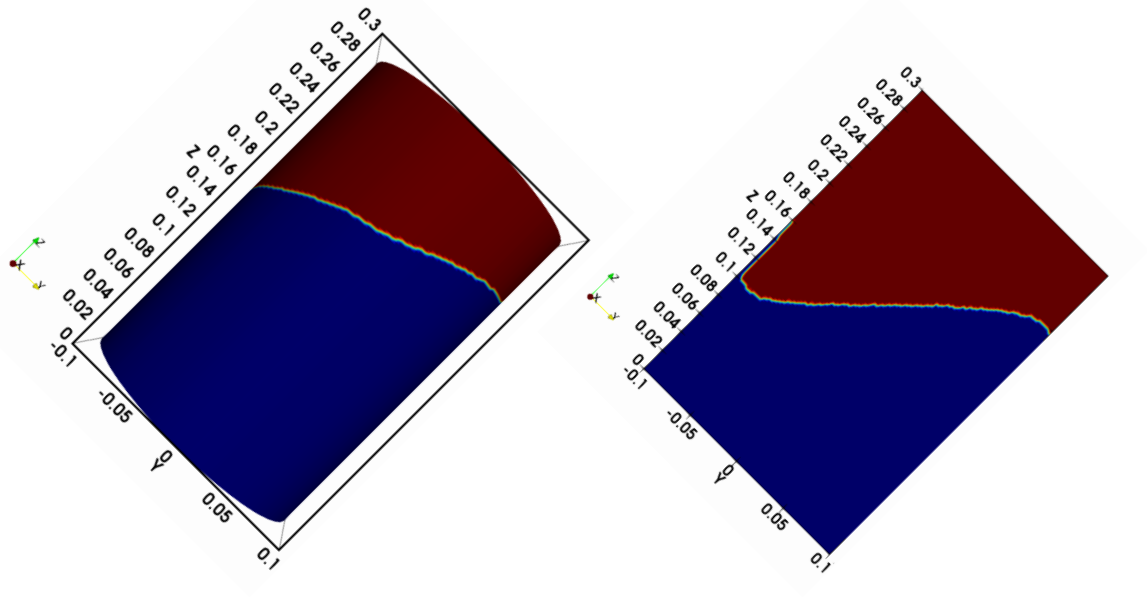
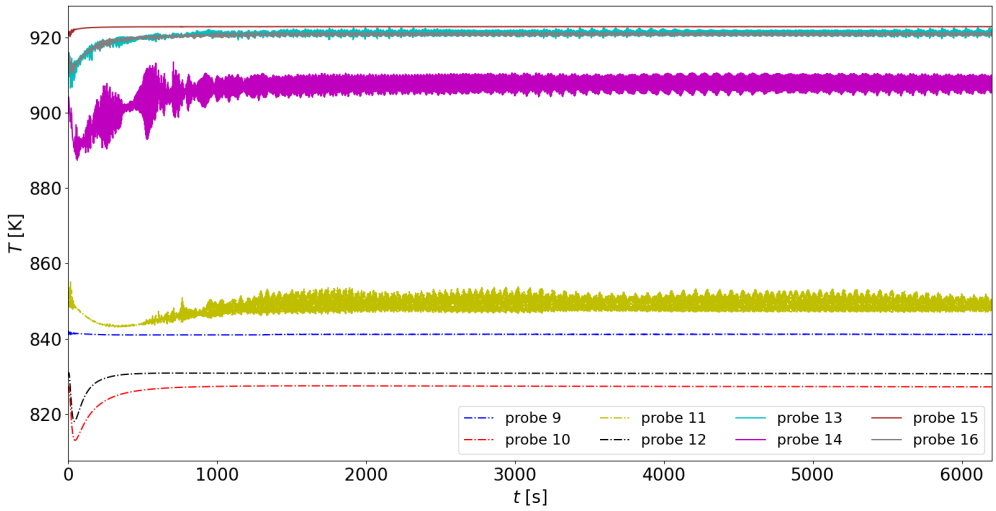


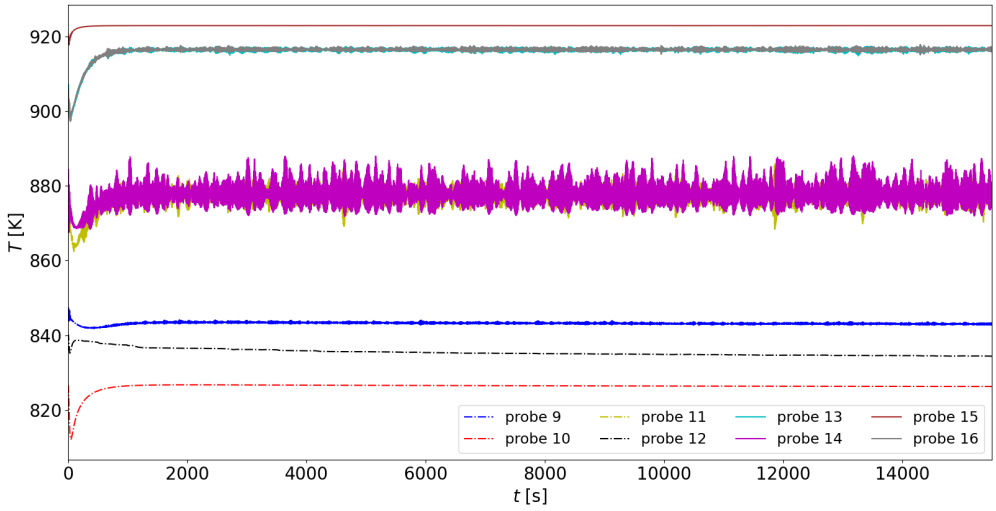
Figure 4.5: The liquid fractions of the steady-state freeze plug at $\theta = 45°$ shown at the outline (a) and slice at $x = 0$ (b). Only the salt part is visible as the copper and Hastelloy-N would otherwise obstruct the visibility of the outline results.

Table 4.3: The liquid fractions at steady-state for different inclination angles.

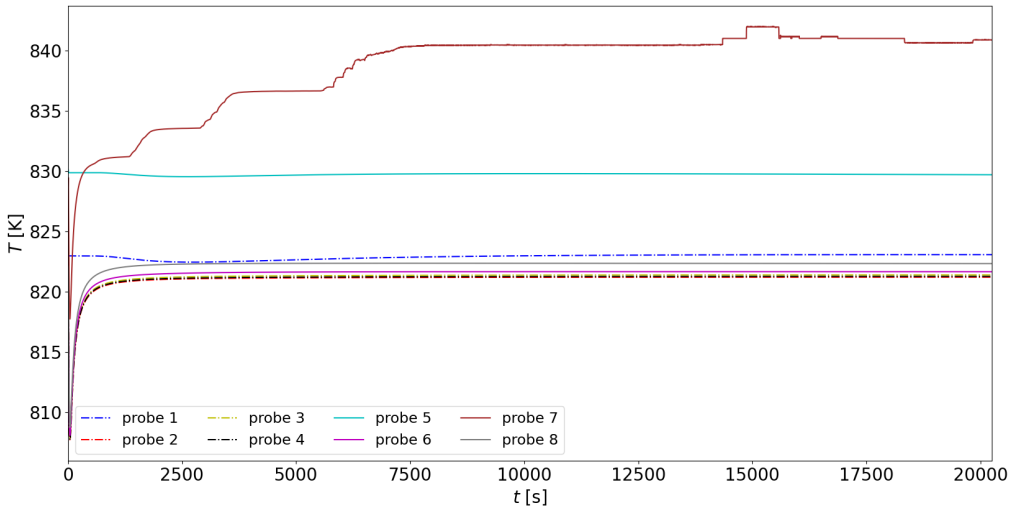
	$\bar{\ell}_{overall}$	$\bar{\ell}_{boundary}$
0°	0.2990	0.2868
15°	0.3162	0.3077
30°	0.3435	0.3416
45°	0.4250	0.429



(a) 15°, probes 9 to 16



(b) 30°, probes 9 to 16



(c) 45°, probes 1 to 8

Figure 4.6: Selected temperature values at the probes for the 1.1M steady-state simulation. The rest of the Figures can be found in Appendix D.

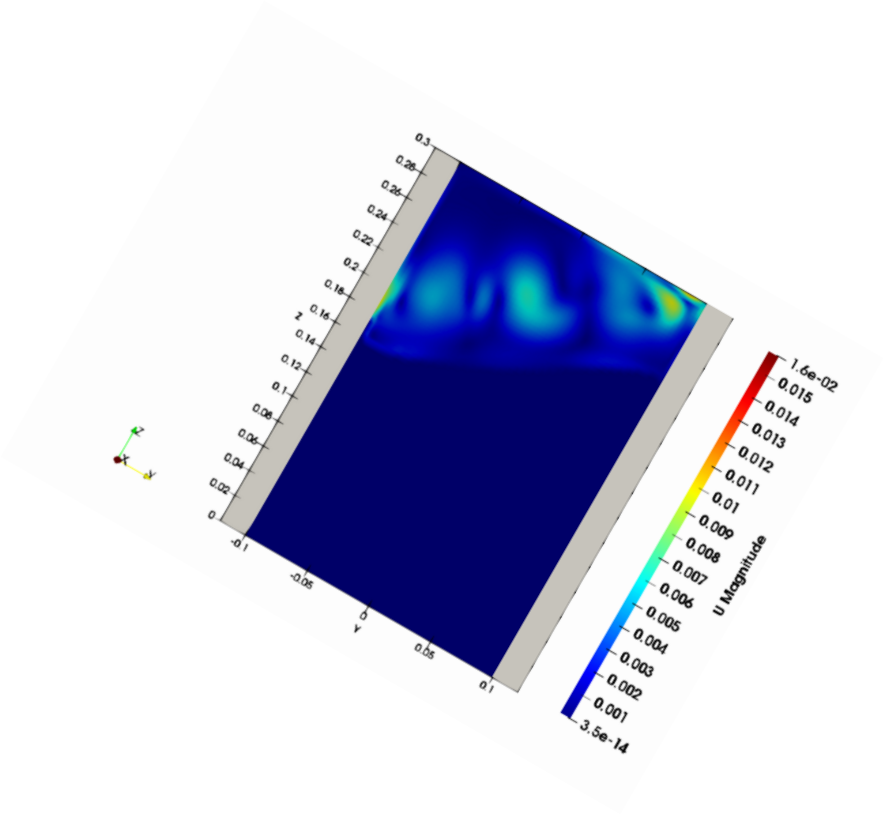
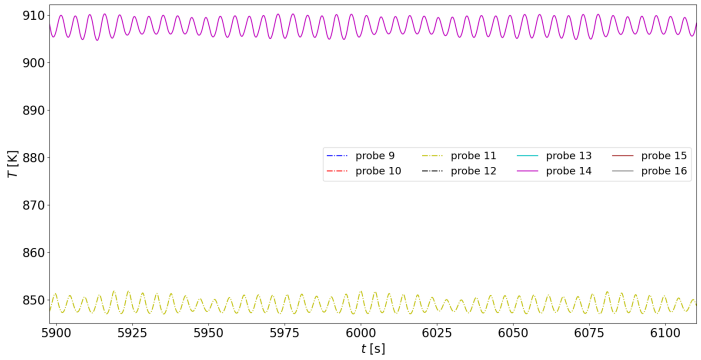
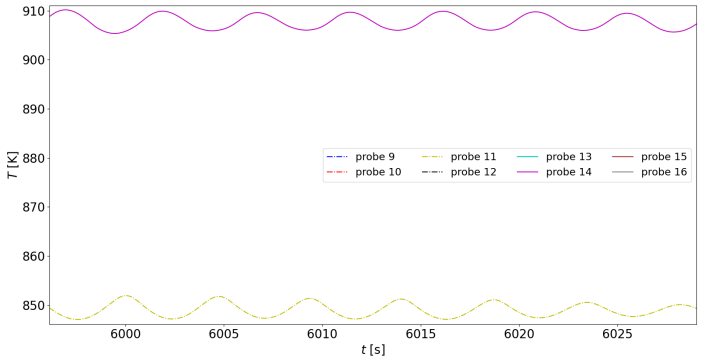


Figure 4.7: The magnitude of the velocity field at steady-state at $\theta = 30°$. The flow is especially high near probes 11 and 14 (see Figure 4.3), which cause the highest temporal oscillation.

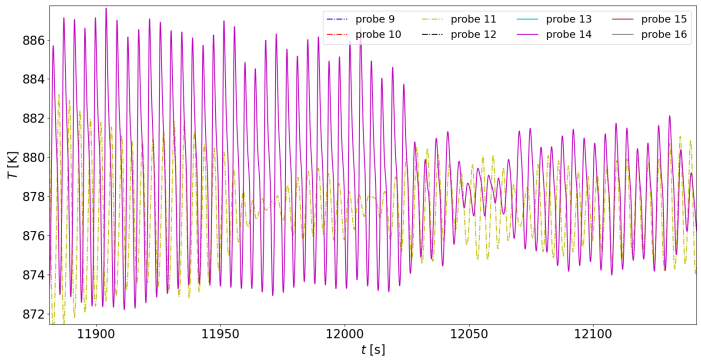
due to the imperfect extraction of the location. A difference in shape is visible for Shafer (2018); the shape of the solid-liquid interface in that work is much steeper than in this work and Reus's. This is possibly due to Shafer's usage of a FEM code with the AHCM as phase change model.



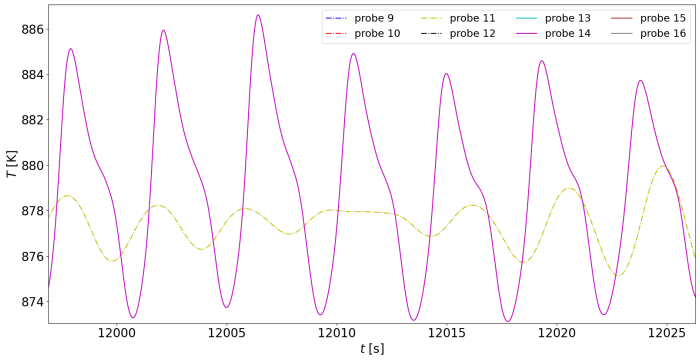
(a) 15°, zoom-in



(b) 15°, further zoom-in



(c) 30°, zoom-in



(d) 30°, further zoom-in

Figure 4.8: The zoomed-in version of the probes of the freeze plug.

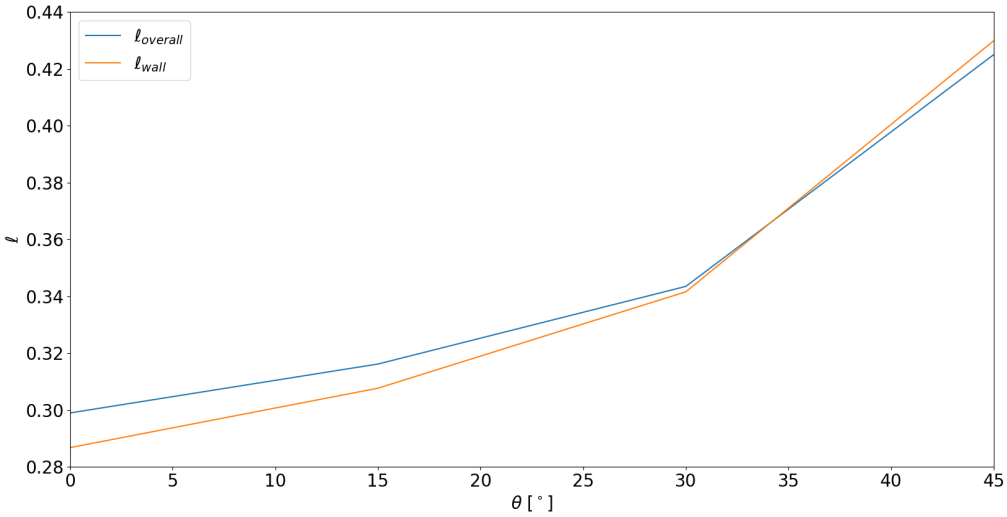


Figure 4.9: The liquid fractions at steady-state for different inclination angles.

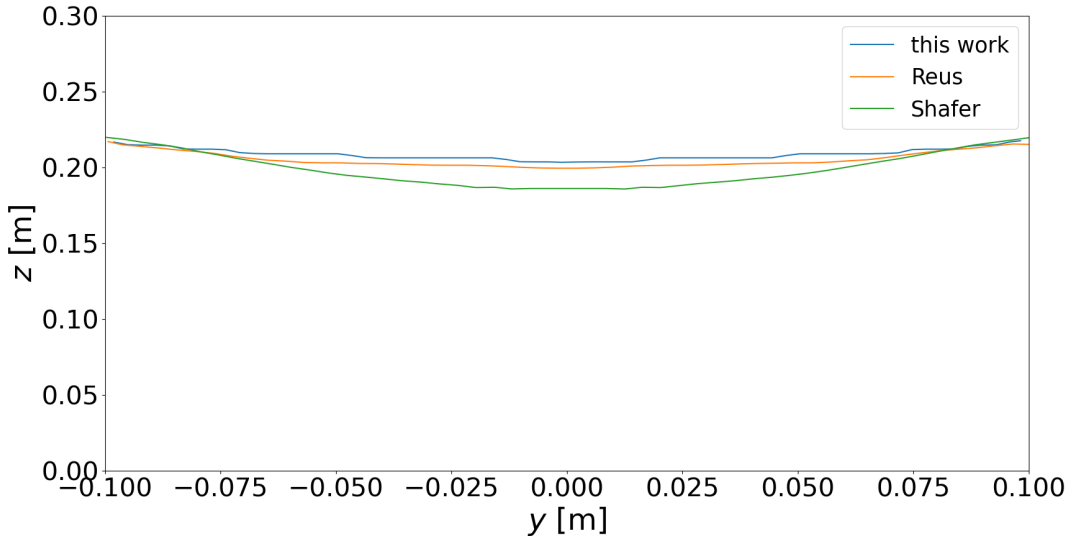


Figure 4.10: The location of the solid-liquid boundary from this work, Reus (2021), and Shafer (2018).

4.5. Steady-state simulation on 3.6M mesh

Method

After the steady-state situations was reached on the 1.1M mesh, yet another mesh was generated. The mesh was divided 162 cells in the length and every cross-section piece consisted of 67x67 = 4489 cells. The mesh therefore consisted of 3,636,090 cells and as a result this mesh is referred to as the 3.6M mesh. The geometry of the mesh was also changed a bit: the vertices of the inflated circle were not longer present at $(x, y) = (\pm 29, \pm 29)$ mm, but at $(\pm 27, \pm 27)$ mm. The vertices were still connected by arcs passing through $(x, y) = (\pm 37, 0)$ and $(0, \pm 37)$ mm. This was done as the original splitting of the mesh resulted in some cells being matched to the wrong region.

The steady-state result of the 1.1M mesh was again mapped on the 3.6M mesh via *mapFields*. The mesh was then split into 96 parts: four partitions in the x and y directions and six in the z direction. The steady-state simulation was ran on the 3.6M mesh until steady-state. The results were saved for every ten seconds of simulated time. Then, for every inclination angle the average overall liquid fractions in the salt as well as the average liquid fractions near the wall were calculated on the 3.6M mesh as well as the 1.1M mesh. They were subsequently compared to check if the 1.1M mesh was representative of reality. The method of calculating the average liquid fractions is explained in Appendix B.

From the steady-state runs on the 3.6M mesh, h_{min} were calculated. h_{min} is the smallest thickness of the solid part of the freeze plug in steady-state, as shown in Figure 4.11. Aji (2020) found that for the TMSR design of the freeze plug, a higher θ leads to a lower h_{min}, so it was assumed this qualitative relationship would also be the case for the MSFR freeze plug.

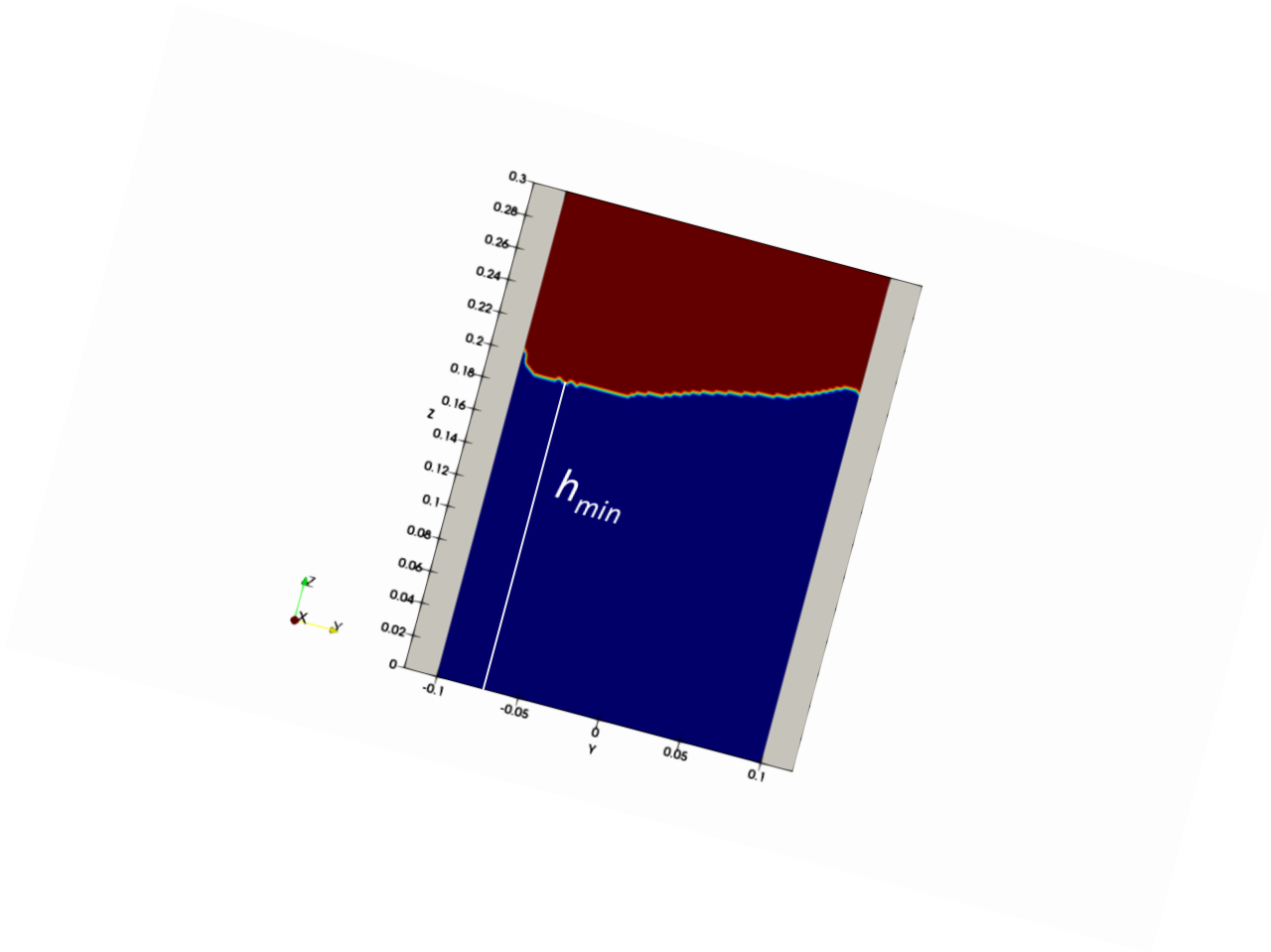


Figure 4.11: The steady-state case for $\theta = 15°$, with h_{min} indicated with the white line.

Results

The steady-states of the 1.1M and 3.6M meshes corresponded quite well with each other. Figure 4.12 shows the steady-states for the two meshes for $\theta = 45°$, for the other three θ's these are included in Appendix E.

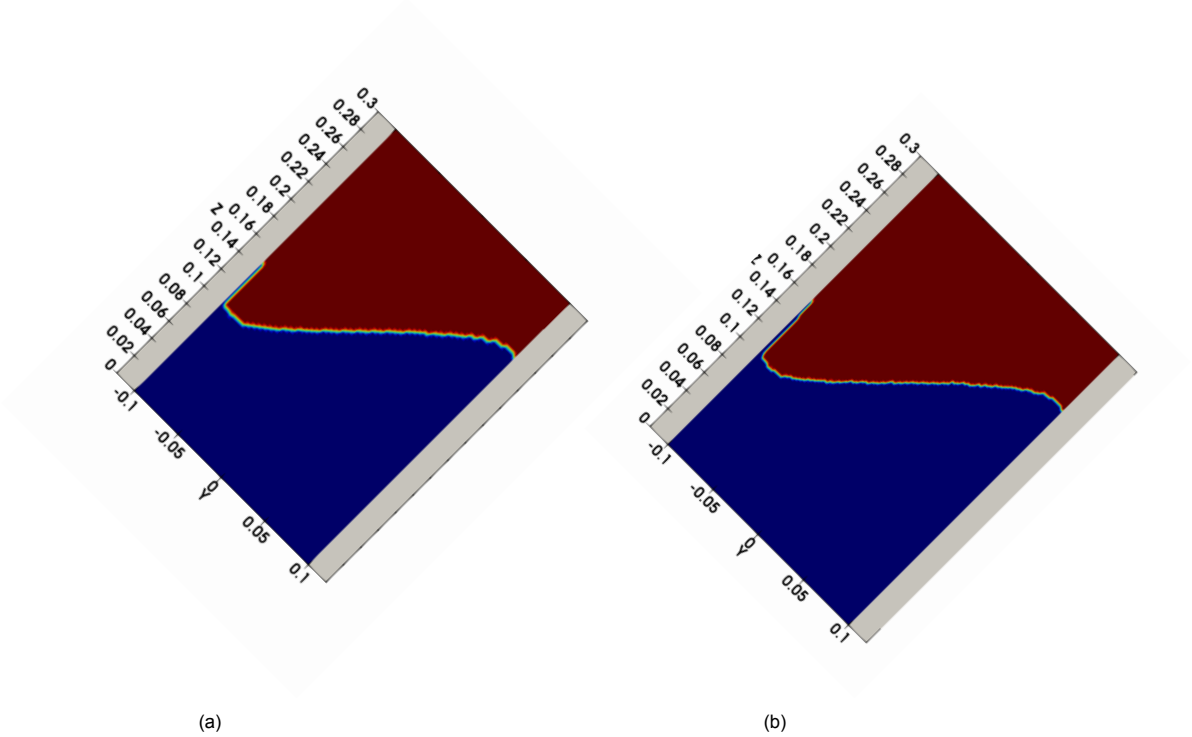


Figure 4.12: The steady-state situation at $45°$ for the 1.1M mesh (a) and the 3.6M mesh (b)

The difference between the steady-states has been quantified by calculating the overall liquid fraction for the two meshes for all θ's. The absolute difference in overall liquid fraction was about 0.4% for $\theta = 15°$ and less than 0.1% for the other inclination angles.

The average overall liquid fractions for the steady-states for all inclination angles is given in Table 4.4. Because of the small absolute difference and the visually found similarity between the sets of steady-states, the 1.1M cell mesh was decided to be mesh-convergent.

Table 4.4: The average overall liquid fractions for the steady-states on the 1.1M and 3.6M meshes.

	$\bar{\ell}_{overall}$ (1.1M mesh)	$\bar{\ell}_{overall}$ (3.6M mesh)	absolute difference
$0°$	0.2990	0.2981	0.0009
$15°$	0.3162	0.3120	0.0042
$30°$	0.3435	0.3432	0.0003
$45°$	0.4250	0.4238	0.0012

The values for h_{min} can be seen in Table 4.5. As the domain consisted of 300 mm divided into 162 cells, the uncertainty was $\frac{300\text{mm}}{162} = 2$ mm.

The values have been plotted in 4.13. In blue, the data points are plotted and interpolated linearly. In orange, a quadratic fit has been plotted through the data points. The quadratic fit assumes the relationship between h_{min} (given in mm) and θ (given in degrees) is $h_{min}(\theta) = a\theta^2 + c$, with $a = (-49 \pm 3) \cdot 10^{-3}$ and $c = 200 \pm 4$. Values behind the $\pm$ indicate the 95% confidence intervals.

The quadratic dependency between h_{min} and θ is a purely empirical one; as far as the Author knows, there is no theoretical basis for this relation. Important to note is the fact that from the given relationship it follows that $h_{min}(\theta = 60°) > 0$, while the non-formation of the freeze plug implies that in reality

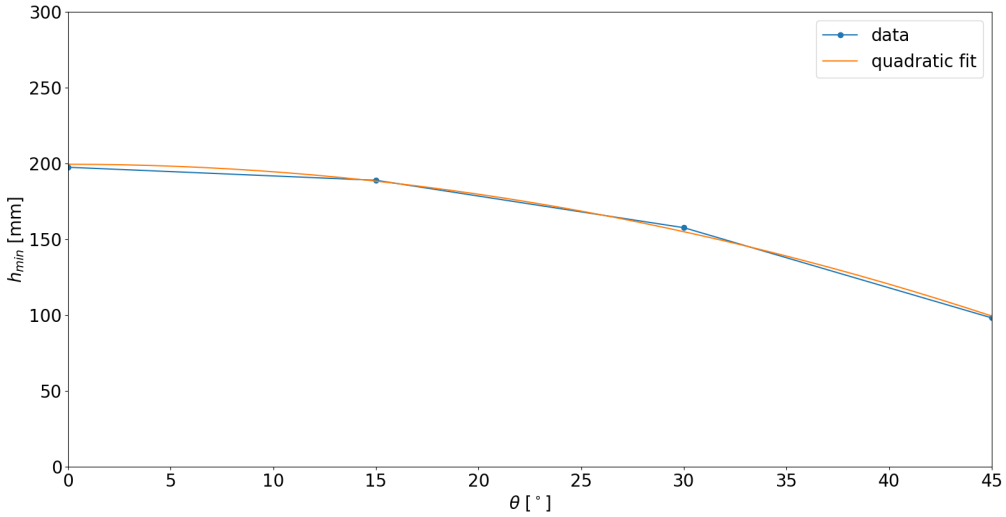


Figure 4.13: The values for h_{min} with and without fit

$h_{min}(\theta = 60°) < 0$. Therefore, more research regarding the question whether the quadratic relationship is indeed true, and if so for which values of θ, is necessary before any hard claims can be made. Should this relationship hold, it would provide interesting information for the design of the MSFR freeze plug as a criterion on h_{min} can then be easily imposed.

Table 4.5: The values for h_{min} with uncertainty of 2 mm.

	h_{min} [mm]
0°	198
15°	189
30°	158
45°	98

4.6. Transient simulation on 1.1M mesh

Method

After the steady-state simulations on the 1.1M mesh and the 3.6M mesh were found to yield reasonably equal results, a transient simulation was started on the 1.1M mesh. This transient simulated an incident in the reactor, resulting in a loss of cooling and additional heating from the top as the system is not cooled anymore while decay heat is added due to the ongoing nuclear reaction. The transient simulation could not be run on the 3.6M mesh, since this would take too long (about 30 days for one simulation, contrary to 1 to 2 days on the 1.1M mesh). The steady-state simulation of the 1.1M mesh was taken as the initial condition for the transient case. The boundary conditions were altered; the copper side was now made adiabatic, simulating the loss of cooling. The reactor heating was simulated with a polynomial (Shafer, 2018):

$$T_{top}(t) = -0.0001t^2 + 0.5244t + 923\ , \tag{4.2}$$

where T is given in K and t in s. The bottom and Hastelloy-N side were not changed, thus kept adiabatic. The different fields were saved every five seconds.

Different results were extracted from the transient runs. The main results were the opening times for the inclination angles. Two different opening times were defined: a start opening time $t_{open,start}$ and an end opening time $t_{open,end}$. At $t_{open,start}$, the melt front first reaches the bottom, while at $t_{open,end}$ all salt cells forming the boundary with the copper and Hastelloy-N were liquid.

The opening times represent the two most important events in the melting of the freeze plug: at the

start opening time, molten salt starts leaking, while at the end opening time the freeze plug falls down and the reactor core drains. Simulation data after the end opening time are thus inherently unphysical. Because of this, the opening times are more suitable as a quantitative measure of the melting behaviour than the total melt time of the whole freeze plug.

Important to note is the fact that generally in MSFR freeze plug literature, e.g. Reus (2021), Shafer (2018), Tiberga et al. (2019), the term 'melting time' is used for what is called opening time in this thesis (as the freeze plug was not inclined there, the start and end opening times were equal due to symmetry). However, one could erroneously think this term means the time needed for the whole freeze plug to melt. 'Opening time' was used by Aji (2020) to represent the start opening time for the TMSR freeze plug, while Aji did not have a specific term for the end opening time as it was not used for analysis in his work. The terms 'start opening time' and 'end opening time' have been coined by the Author.

The opening times were found by visual inspection of the liquid fraction. For the end opening time, this was aided by making a plot of the average liquid fraction in the salt cells forming the boundary with the copper or Hastelloy-N. This average boundary liquid fraction reached 1 exactly at the end opening time. These two methods of finding the end opening times turned out to correspond with each other every time.

Another result was the average salt liquid fraction in the whole domain during the run, instead of only for the cells at the boundary with the copper and Hastelloy-N.

Results

The opening times of the freeze plug have been found and can be found in Table 4.6. As the data were saved every 5 seconds, the opening times could only be found with 5 second accuracy. The opening times have all been rounded up, so e.g. $t_{open,start}$ for $45°$ has been found to lie between 395 and 400 s, so $t_{open,start}$ has been chosen to be 400 s rather than 395 s.

Table 4.6: The opening times for transient melting on the 1.1M mesh

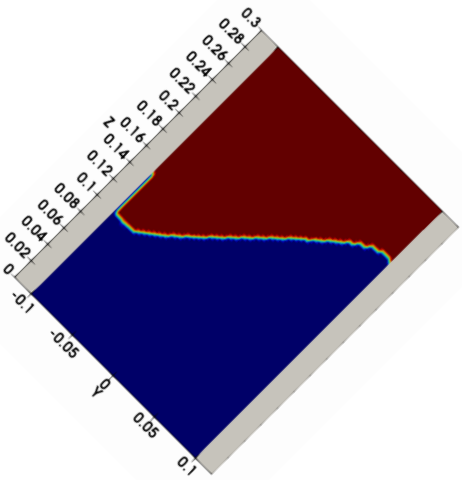
θ	$t_{open,start}$ [s]	$t_{open,end}$ [s]
$0°$	1520	1520
$15°$	1105	1390
$30°$	650	995
$45°$	400	730

From the results, it becomes clear that a stronger inclination results in smaller opening times. This is in accordance with Aji (2020) and is arguably the main result of this thesis. The start opening time for the freeze plug inclined by $\theta = 45°$ is reduced by 74%, while the end opening time is reduced by 52%. The freeze plugs inclined by $15°$ and $30°$ have smaller opening time reductions, but the opening time differences with respect to the $0°$ case are substantial.

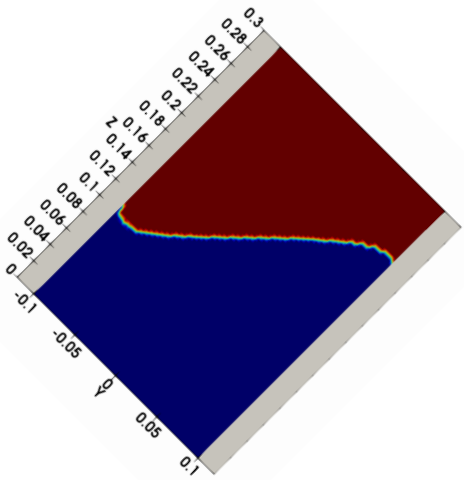
We also see the difference between the two opening times increases with increasing inclination angle. This is not surprising considering higher inclination leads to higher asymmetry in the system. The non-inclined freeze plug, with $\theta = 0°$, is fully symmetric and as a result the opening times are equal in that case.

Some figures captioning the melting behaviour of the $\theta = 45°$ freeze plug are given in Figure 4.14. It can be seen that melting takes place mostly near the boundary with the copper. The melt front first reaches the bottom at $t_{open,start}$ (as can be seen in Subfigures 4.14c and 4.14d, which are the freeze plugs at one time step before $t_{open,start}$ and $t_{open,start}$ itself). The freeze plug continues melting until $t_{open,end}$ is reached. Interesting to see is that the phase change interface is approximately horizontal, except for when the wall is close. This was already known for $\theta = 0°$ (Reus, 2021), but it is interesting that this holds for an inclined freeze plug as well. For the other θ's, the same behaviour takes place, except for the fact that the asymmetry between $t_{open,start}$ and $t_{open,end}$ decreases with decreasing θ.

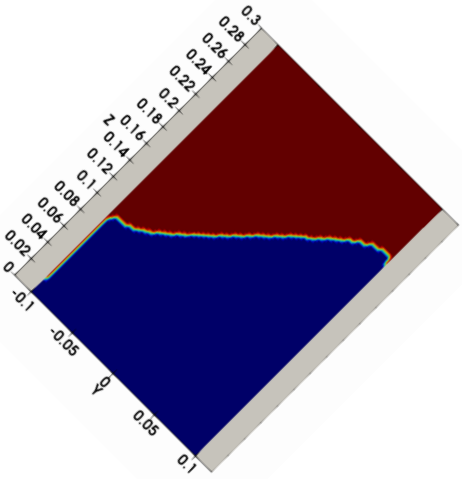
The opening times range from 400 to 1520 s. These are close to the required opening times, which according to literature range from 480 to 1600 s, as explained in section 1.4 and both more and less than the working point for this thesis, $t = 1000$ s.



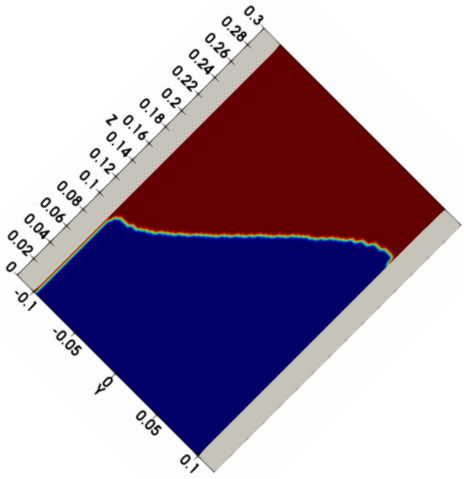
(a) $t = 5$ s



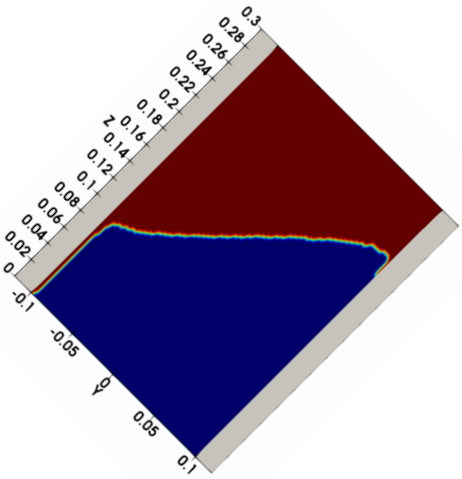
(b) $t = 150$ s



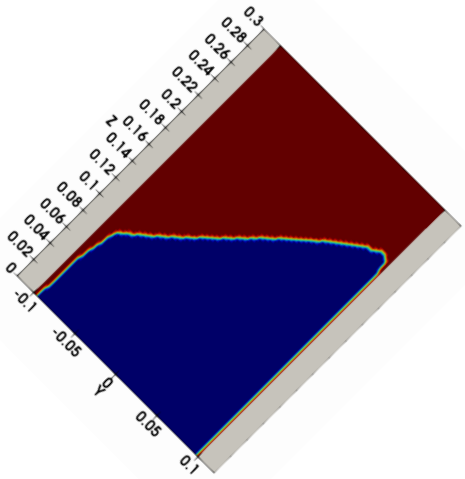
(c) $t = 395$ s



(d) $t = 400$ s



(e) $t = 550$ s



(f) $t = 730$ s

Figure 4.14: The freeze plug with $\theta = 45°$ at different times.

The fact that a high θ results in lower opening times is an important result. This either increases the safety directly, as an extra safety buffer is added to the design. This improves the freeze plug's adherence to the requirement that a freeze plug must melt quickly in case of a reactor malfunction. On the other hand, the freeze plug's design can be changed to decrease its proneness to thermal fluctuations, e.g. increasing the subcooling temperature or elongating the pipe between the reactor core and the freeze plug. This ensures the freeze plug will not melt during normal MSFR operation. The question which option is chosen lies outside the scope of this thesis, but can be investigated in further research.

Plots for liquid fraction have also been made. In Figure 4.15 the average overall liquid fraction and average liquid fraction near the wall are found. The plots start at $t = 0$ and stop at their $t_{open,end}$, since at that point in reality the freeze plug would fall down and the simulation - not allowing fluid from falling - became inherently unphysical. $t_{open,start}$ are indicated with vertical dotted lines. The higher θ, the higher the liquid fractions were at $t = 0$, but also the quicker melting became afterwards.

From the Figures - especially 4.15b, interesting information about the melting behaviour can be found. For a certain time, about 350 seconds for $\theta = 45°$ to 1000 seconds for $\theta = 0°$, melting goes slowly and the liquid fraction only barely increases. Then, melting suddenly accelerates and melting now takes place in near-constant rates until $t_{open,end}$. This is due to the fact that at that point part of the copper reaches a temperature higher than T_{melt} of the salt, enabling heating from the side instead of only from the top.

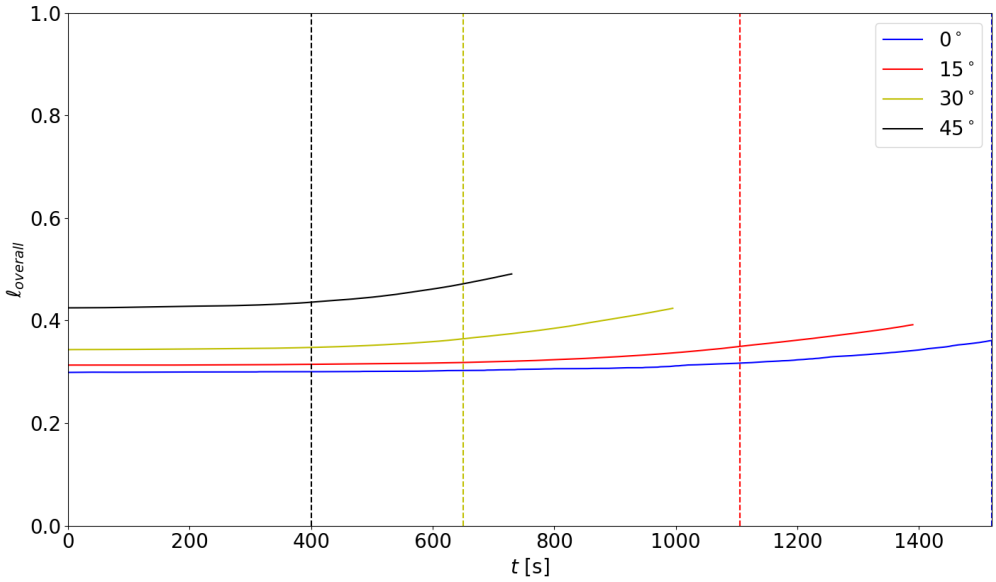
For $\theta = 0°$, the found opening times are $t_{open,start} = t_{open,end} = 1520$ s. This is unexpected, as it is much longer than the results found by Shafer (2018) and Reus (2021), which were 970 and 1140 seconds, respectively. The difference with Shafer's result may be due to the fact she used the Apparent Heat Capacity Method for his thesis, which is known to cause nonphysical behaviour. Also, his simulations were 2-dimensional instead of the 3-dimensional ones used in this work. The number of cells used in his simulations is unknown. The discrepancy with the results of Reus is more surprising, as he also used the Linearised Enthalpy Method and the code used in this work is based on his. The difference may also be caused by the fact his simulations were two-dimensional on an axisymmetric 400x200 cell mesh. This is finer than the 1.1M mesh used in this work, whose cross-section is not axisymmetric and has 108x135 cells. However, as mesh-convergence was found to be present when comparing the 1.1M mesh with the 3.6M mesh, the mesh fineness is not expected to be the problem. Another possiblity is the hypothesis that the difference is due to the fact the simulations from Reus (2021) (and Shafer (2018)) were 2-dimensional, contrary to the 3-dimensional ones in this work. 2-dimensional simulations are shown to have a higher melting rate than 3-dimensional simulations (Hu et al., 2021), plausibly explaining the faster melting in Reus's and Shafer's simulations.

Finally, an interesting phenomenon is the apparently strange behaviour of $\bar{\ell}_{boundary}$ for $\theta = 0°$: it increases stepwise, instead of smoothly as the other angles. The reason for that is the symmetry in the $0°$ case. Since the case is symmetric[2], at certain time points all cells with the same z coordinate melt - as explained in section 3.1, ℓ could be only 0 or 1, so cells underwent phase change instantaneously. The actual values of $\bar{\ell}_{boundary}$ for $\theta = 0°$ all correspond to $\frac{N}{108}$, where N is an integer. At every stepwise increment, N increases by one. As a reminder, the number of cells in the z direction is 108, verifying this explanation. For $\theta \neq 0°$ there is no symmetry so melting goes much more smoothly.

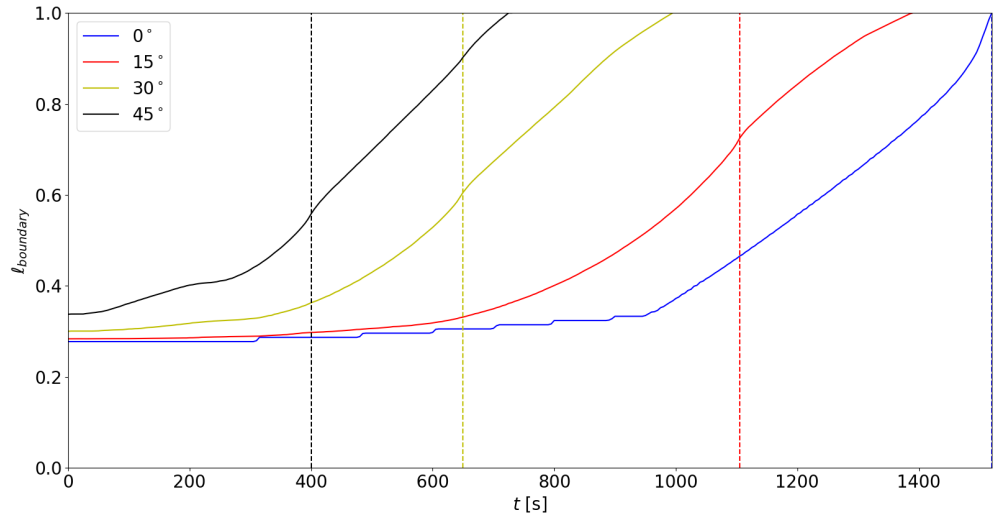
The Rayleigh number, calculated via Equation 2.14, was found to be equal to $Ra = 8.0 \cdot 10^9$ at the end of the simulation at $\theta_{freeze\ plug} = 45°$. Note that the θ in that equation is not equal to the θ of the freeze plug: at $\theta = 0$ in Equation 2.14, the hot wall stands upright. As the hot wall equivalent in the freeze plug is the top, this corresponds to a $\theta = 90°$ of the freeze plug. Therefore, $\theta_{Ra\ equation} = 90° - \theta_{freeze\ plug}$. At the end of the simulation with $\theta_{freeze\ plug} = 30°$, $\cos\theta_{Ra\ equation}$ decreases, while ΔT increases. There, $Ra = 6.9 \cdot 10^9$. However, when $\cos\theta_{freeze\ plug} = 0°$, the Rayleigh number must be calculated via Equation 2.11 with l_c. This leads to a much lower Rayleigh number of $Ra = 2.6 \cdot 10^8$.

According to Section 2.4, these Rayleigh numbers indicate that turbulence is present for $\theta_{freeze\ plug} = 30°$ and $45°$, while the flow for $\theta_{freeze\ plug} = 0°$ is laminar. For $\theta_{freeze\ plug} = 15°$, the theory is

[2] In reality the case is fully symmetric. In the simulation due to e.g. the formulation of the mesh this is not the case, but deviation from symmetry is only minimal.



(a)



(b)

Figure 4.15: The average overall liquid fraction (a) and the average liquid fraction at the boundary (b) during transient melting. $t_{open,start}$ is given by the dotted line, while the plots finish at $t_{open,end}$.

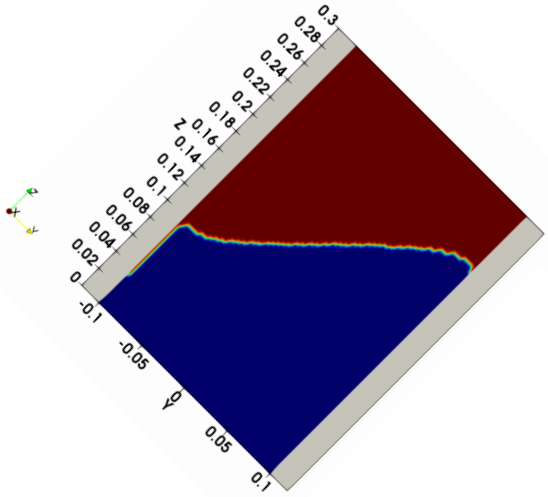
insufficient. A caveat is the fact that the theory assumes a vertical or inclined wall, i.e. a polyhedral cavity. On the other hand, the freeze plug is cylindrical with the liquid part having a complex shape. The validity of the results is therefore unknown, although they do give a good estimate on how to compare the results with one another; for instance the fact the calculated Rayleigh number increases by a factor 25 for the freeze plug inclined by $45°$ compared to the upright standing one does give good insight on the different thermodynamics of the system.

4.7. Error in simulation restarting

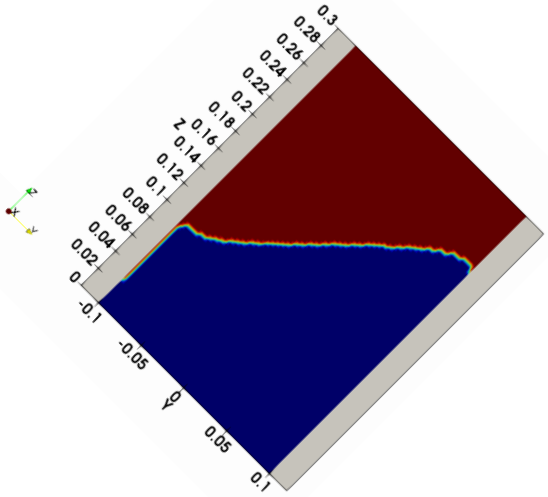
For the transient simulation of the freeze plug on the 1.1M mesh, stopping the simulation and rerunning it from its latest saved time step led to unphysical results, namely an instantaneous increase in melting. Usually restarting a simulation does not cause problems in *OpenFOAM*; neither did the restarting of the simulation in a steady-state simulation.

A wrong restarting instance is visible in Figure 4.16, where three instances of the transient 1.1M mesh simulation for $\theta = 45°$ are present. At $t = 380$ s, the phase change interface touches the copper around $(x, y, z) = (0, -0.1, 0.35)$ m, as is shown in Subfigure 4.16a. Without a restart, five seconds later the interface touches the wall at $(x, y, z) = (0, -0.1, 0.30)$ m, visible in Subfigure 4.16b. However, when stopping the simulation and restarting it from $t = 380$ s, the melting suddenly increases such that the interface touches the wall just above the bottom, which is at $(x, y, z) = (0, -0.1, 0)$ m (Subfigure 4.16c).

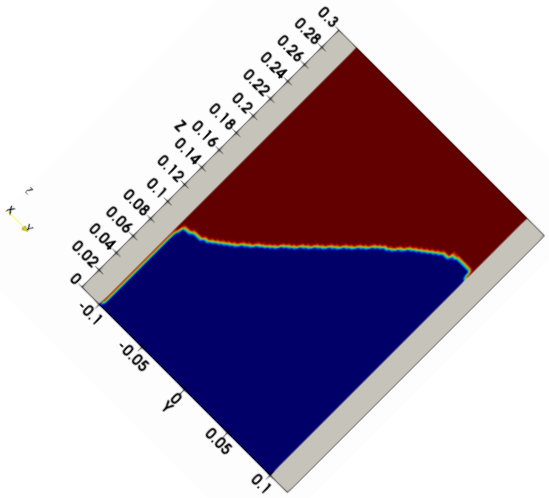
The reason for an error to happen is probably due to the initialisation of the enthalpy at the beginning of a simulation. The enthalpy is set to $H_{init} = \rho_s c_{p,s} T_{init}$ for $T_{init} \leq T_m$, which means that cells undergoing phase change have their liquid fraction set to $\ell = 0$. Checking if this is indeed the problem, and if so solving it, would be beneficial to the working of the code.



(a) Starting situation



(b) Without restart



(c) With restart

Figure 4.16: Three instances of the transient 1.1M mesh simulation for $\theta = 45°$.

5 Recommendations

The goal of this thesis was to investigate the effect of the inclination angle of a freeze plug on its melting behaviour. The results show opening times decrease strongly with increasing inclination angle. However, there are certain improvements in numerical modelling and design of the freeze plug that can be performed in future research.

The first improvement is the behaviour of the molten salt once the start opening time has been reached. The model assumes the molten salt to be contained within the physical domain, but in reality, the molten salt starts leaking out of the cylindrical physical domain once this happens. The severity of this effect is unknown: on the one hand, the opening area is small and the molten salt has a high viscosity. On the other hand, the pressure acting on the molten salt near the opening is high due to the molten salt above it. Moreover, leakage of the fluid increases the convective heat transfer, further enlarging the opening, leading to a positive feedback loop. The model's inability to take into account leakage therefore leads to an overestimation of $t_{open,end}$, especially for higher θ's as $t_{open,end} - t_{open,start}$ increases with θ. However, the size of this overestimation is unknown without further research.

The next possible model feature is the freeze plug's behaviour near its end melting time. The model assumes the freeze plug falls when it loses all contact with the wall. In reality, the freeze plug already falls earlier, because at some point the pressure of the molten salt acting on the freeze plug overpowers the adhesive forces between the freeze plug and the wall. The purpose of the freeze plug - draining the reactor in case of an incident - is then achieved before $t_{open,end}$. The severity of this inaccuracy of the model is conceivably low if there is no minimal time until draining in the freeze plug requirements. However, including adhesive forces by incorporating settling in the model would be an advantage as the model would be more realistic.

Another improvement is refining the mesh in the outer regions of the freeze plug. The current mesh was quite ineffective as overall it was finest near the cylindrical freeze plug's axis, while it was coarsest close to the boundary of the domain. However, this was the physically most interesting area, considering most melting took place there, and opening times were defined by the freeze plug's behaviour near the wall. Refining the mesh - at the beginning of the simulation, or adaptively - could be advantageous for the simulation. For adaptive mesh refinement, a possible refinement criterium is liquid fraction; Besseling (2022) showed are vorticity and shear rate are also viable. However, important to note is that his model used the Lattice Boltzmann method (based on the Finite Difference Method) and Source Based Method, so it is uncertain if these criteria would work for the simulation discussed in this thesis as well.

Another mesh-related improvement point is the execution of a mesh and time-step convergence test. In this thesis, a brief one was performed as described in Section 4.5. However, in order to better certify the results, the presence of mesh convergence should be validated.

The imperfect multithreading - as described in Subsection 3.2.2 should also be resolved for future simulations, as it would enhance their validity. Another inaccuracy is the fact that stopping the code and rerunning it from the last saved point results in a wrong continuation in the transient case. This problem should be solved before the code is further used.

The model could further be analysed by exploiting its symmetry. As the freeze plug is symmetric around $x = 0$, the freeze plug could be modelled only on the half-domain $x \geq 0$ (or $x \leq 0$). This would halve the amount of cells in the simulation. Exploiting this symmetry was not performed in this work, as it was not known *a priori* whether this symmetry would also be present in the (imperfect) model.

However, the simulation has shown that for big enough meshes this symmetry is indeed present, so when implemented it would increase simulation speed without negative effect on its accuracy.

In this thesis, the simulations used the laminar transport model in the *simulationType* file in *OpenFOAM*. It would be interesting to use other transport models there in future simulations; although qualitative results of this thesis are expected to be equal with another transport model, there could be a difference in quantitave results. Coupling the transport model with the phase change model could therefore lead to interesting insights.

Apart from improvements to the simulations, extra information could be extracted from the simulations as well. One interesting question is whether the relationship between θ and h_{min} is indeed a quadratic one, as explained in Subsection 4.5; and if so, for which θ range this would be valid. Even more attractive would be finding a theoretical basis for this relationship - if it is indeed true - as this would allow for generalisation for freeze plugs with other design parameters such as sub-cooling temperature or depth underneath the reactor core.

Also compelling is the proneness to thermal fluctuations and reactor pressure of inclined freeze plugs with respect to upright standing ones. As d_{min} decreases with θ, this increases the chance that the freeze plug melts without a reactor incident. The different convective behaviour of the molten salt above the freeze plug also may increase or decrease this chance. Regardless of the effect's size, it can probably be countered by changing design parameters such as sub-cooling temperature or distance between the reactor core and the freeze plug; this would partially or fully nullify the decrease in t_{open}. The desirability of increasing freeze plug stability at the expense of t_{open} also depends on the question how prone the inclined freeze plugs are to thermal fluctuations and pressure with respect to the upright freeze plug and each other.

6

Conclusion

In this thesis, the effect of inclination angle on the melting time of a freeze plug has been investigated. This was done via the Linearised Enthalpy Method in OpenFOAM 8, a software package using the Finite Volume Method. The phase change behaviour was implemented in an in-house code based on the *chtMultiRegionFoam* solver.

The research questions, as posed in Section 1.5, were as follows:

- How does the solver perform the benchmark cases with respect to previous results?
- What effect does the inclination angle have on the formation of a freeze plug for normal MSFR operation?
- What effect does the inclination angle have on the melting behaviour of the MSFR freeze plug?

To answer the first question, two benchmark cases have been set up and analysed: a gallium melting case showing the accelerated melting for inclined media, used to test the phase change implementation of the solver, as well as a case of a copper fin in air validating the solver's CHT implementation. The results of this work were qualitatively similar to the benchmark cases, although a quantitative difference was visible - mainly for the gallium simulation. This can be explained by the different implementation of phase change in the benchmark case, or the quality of the mesh.

For the second and third questions, cylindrical freeze plugs have been generated for inclination angles of $\theta = 0°, 15°, 30°, 45°, 60°$, and $90°$. From steady-state simulations on 3D meshes of 40,500 and 1.1 million cells the freeze plug did not form for $\theta = 60°$ and $90°$. For the other four inclination angles, an additional steady-state simulation on a 3.6 million cell grid showed mesh convergence. With a transient simulation on the mesh of 1.1 million cells, it was shown that a higher inclination angle resulted in a smaller minimal distance between liquid salt and freeze plug bottom, with an apparent quadratic relationship between this distance and inclination angle. Also, higher inclination angles resulted in a lower opening time; with respect to $\theta = 0°$, the start and end opening times for the freeze plug inclined by $45°$ were reduced by 74% and 52%, respectively. The opening times for $\theta = 15°$ and $30°$ also decreased, but by a smaller percentage. Inclining a freeze plug therefore is an efficient way to control opening times.

Improvements for the model are possible, such as including molten salt leakage after start opening time, and settling. Computational improvements include the refinement of the mesh near the salt boundary as well as the adaptive mesh refinement near the solid-liquid boundary. Also symmetry can be exploited to speed up results. Further improvements are the resolution of the discrepancies in results caused by the use of multithreading and reinitialisation of the enthalpy following a restart of the simulation. An interesting follow-up research question is the investigation of the proneness to thermal fluctuations of the different steady-state freeze plugs.

Bibliography

Aji, I. K. (2020). *Investigation of Basic Parameters in Developing High-Performance Freeze Valve for Molten Salt Reactor*. PhD thesis, The University of Electro-Communications.

Bejan, A. (2013). *Convection Heat Transfer*. Wiley, 4 edition.

Besseling, T. E. (2022). Investigating adaptive mesh refinement criteria for a double-distribution fmlb scheme in melting and solidification processes.

Bradbury, R. D. (1953). *Fahrenheit 451*. Ballantine Books.

Brent, A. D., Voller, V. R., and Reid, K. J. (1988). Enthalpy-porosity technique for modeling convection-diffusion phase change: application to the melting of a pure metal. *Numerical Heat Transfer*, 13(3):297–318.

Brovchenko, M., Heuer, D., Merle-Lucotte, E., Allibert, M., Ghetta, V., Laureau, A., and Rubiolo, P. (2013). Design-related studies for the preliminary safety assessment of the molten salt fast reactor. *Nuclear Science and Engineering*, 175(3):329 – 339. Cited by: 28.

Cadence PCB solutions (n.d.). Finite element method (fem) vs. finite volume method (fvm) in field solvers for electronics. https://resources.pcb.cadence.com/blog/2020-finite-element-method-fem-vs-finite-volume-method-fvm-in-field-solvers-for-electro (Accessed on 2023-01-17).

Chen, L. (2021). Finite volume methods. https://www.math.uci.edu/~chenlong/226/FVM.pdf. (Accessed on 2022-12-09).

Cheng, P., Tang, Z., Gao, Y., Liu, P., Liu, C., and Chen, X. (2022). Flexible engineering of advanced phase change materials. *iScience*, 25(5):104226.

Chilvers, J. (2014). The finite volume method: An introduction. https://tameaero.wordpress.com/2014/10/24/the-finite-volume-method-an-introduction/. (Accessed on 2022-12-09).

Chisholm, B. M., Krahn, S. L., and Sowder, A. G. (2020). A unique molten salt reactor feature – the freeze valve system: Design, operating experience, and reliability. *Nuclear Engineering and Design*.

Clay Mathematics Institute (2022). Navier-stokes equation. https://www.claymath.org/millennium-problems/navier-stokes-equation. (Accessed on 2022-11-28).

Dai, Z. (2017). 17 - thorium molten salt reactor nuclear energy system (tmsr). In Dolan, T. J., editor, *Molten Salt Reactors and Thorium Energy*, pages 531–540. Woodhead Publishing.

de la Cruz, L. M. and Monsivais, D. (2014). Parallel numerical simulation of two-phase flow model in porous media using distributed and shared memory architectures. *Geofísica Internacional*, 53(1):59–75.

Delis, A. and Mathioudakis, E. (2009). A finite volume method parallelization for the simulation of free surface shallow water flows. *Mathematics and Computers in Simulation*, 79:3339–3359.

Faden, M., Linhardt, C., Höhlein, S., König-Haagen, A., and Brüggemann, D. (2019). Velocity field and phase boundary measurements during melting of n-octadecane in a cubical test cell. *International Journal of Heat and Mass Transfer*, 135:104–114.

Fermigier, M. (2017). Transport equations: Mass and heat balances. https://blog.espci.fr/marcfermigier/files/2017/03/Heat_Mass_Balances.pdf. (Accessed on 2022-10-28).

Frey, P. (2017). The finite difference method. https://www.ljll.math.upmc.fr/frey/cours/UdC/ma691/ma691_ch6.pdf. (Accessed on 2022-12-09).

Gartling, D. K. (1978). Finite element analysis of convective heat transfer problems with change of phase.

GEN IV International Forum (2022). Generation iv goals. https://www.gen-4.org/gif/jcms/c_9502/generation-iv-goals. (Accessed on 2022-09-22).

Gheribi, A., Corradini, D., Dewan, L., Chartrand, P., Simon, C., Madden, P., and Salanne, M. (2014). Prediction of the thermophysical properties of molten salt fast reactor fuel from first-principles. *Molecular Physics*.

Ghetta, V., Giraud, J., Rubiolo, P., and Tano-Retamales, M. (2017). MSFR fuel salt conditions during typical draining transients.

Giraud, J., Ghetta, V., Rubiolo, P., and Tano-Retamales, M. (2019). Development of a cold plug valve with fluoride salt. *EPJ Nuclear Sciences & Technologies*, 5(9).

Greenshields, C. (2022). Notes on computational fluid dynamics: General principles; 5.21 the pimple algorithm. https://doc.cfd.direct/notes/cfd-general-principles/the-pimple-algorithm. (Accessed on 2022-11-18).

Guo, R., Shan, L., Wu, Y., Cai, Y., Huang, R., Ma, H., Tang, K., and Liu, K. (2022). Phase-change materials for intelligent temperature regulation. *Materials Today Energy*, 23:100888.

Hayes International (n.d.). Hastelloy® n alloy. https://haynesintl.com/docs/default-source/pdfs/new-alloy-brochures/corrosion-resistant-alloys/brochures/n-brochure.pdf?sfvrsn=18. (Accessed on 2022-12-12).

Hu, N., Fan, L.-W., and Zhu, Z.-Q. (2021). Can the numerical simulations of melting in a differentially-heated rectangular cavity be rationally reduced to 2d? a comparative study between 2d and 3d simulation results. *International Journal of Heat and Mass Transfer*, 166:120751.

Ignatiev, V. V., Feynberg, O. S., Zagnitko, A. V., Merzlyakov, A. V., Surenkov, A. I., Panov, A. V., Subbotin, V. G., Afonichkin, V. K., Khokhlov, V. A., and Kormilitsyn, M. V. (2012). Molten-salt reactors: New possibilities, problems and solutions. *Atomic Energy*, 112(3).

Intergovernmental Panel on Climate Change (2018). *Summary for Policymakers*, page 1–24. Cambridge University Press.

Intergovernmental Panel on Climate Change (n.d.). Global warming of 1.5 °c. https://www.ipcc.ch/sr15/. (Accessed on 2022-09-22).

International Atomic Energy Agency (n.d.). Molten salt reactors. https://www.iaea.org/topics/molten-salt-reactors. (Accessed on 2023-01-13).

Iserles, A. (1996). *A First Course in the Numerical Analysis*. Cambridge University Press, 1 edition.

Jasak, H. (1996). *Error Analysis and Estimation for the Finite Volume Method With Applications to Fluid Flows*. PhD thesis, Imperial College London.

Jiang, X., Lu, H.-J., Chen, Y.-S., Fu, Y., and Wang, N.-X. (2020). Numerical and experimental investigation of a new conceptual fluoride salt freeze valve for thorium-based molten salt reactor. *Nuclear Science and Techniques*, 31.

Kaaks, B., Introini, C., Tartaglia, D., Cammi, A., Lorenzi, S., and Lathouwers, D. (2022). Advanced phase change model representation and implementation.

Kaaks, B. J. and Lathouwers, D. (n.d.). Fixed-grid modelling of solid-liquid phase change with a symmetric interior penalty discontinuous galerkin method. Awaiting publication.

Kenjeres, S. (2021). Continuum physics lectures series, lecture 9: Special topics: Thermal convection. Lecture notes.

Korti, A. I. N. and Guellil, H. (2020). Experimental study of the effect of inclination angle on the paraffin melting process in a square cavity. *Journal of Energy Storage*.

Krishnan, G., Parhizi, M., and Jain, A. (2022). Eigenfunction-based solution for solid-liquid phase change heat transfer problems with time-dependent boundary conditions. *International Journal of Heat and Mass Transfer*, 189:122693.

Kundu, P. K., Cohen, I. M., and Dowling, D. R. (2016). Chapter 4 - conservation laws. In Kundu, P. K., Cohen, I. M., and Dowling, D. R., editors, *Fluid Mechanics (Sixth Edition)*, pages 109–193. Academic Press, Boston, sixth edition edition.

König-Haagen, A., Franquet, E., Faden, M., and Brüggemann, D. (2020). Influence of the convective energy formulation for melting problems with enthalpy methods. *International Journal of Thermal Sciences*, 158:106477.

König-Haagen, A., Franquet, E., Pernot, E., and Brüggemann, D. (2017). A comprehensive benchmark of fixed-grid methods for the modeling of melting. *International Journal of Thermal Sciences*, 118:69–103.

Lacroix, M. and Voller, V. (1990). Finite difference solutions of solidification phase change problems: Transformed versus fixed grids. *Journal of Numerical Heat Transfer*, 17:25–41.

Nature (2021). China prepares to test thorium-fuelled nuclear reactor. https://www.nature.com/articles/d41586-021-02459-w. (Accessed on 2022-05-15).

Nedjar, B. (2002). An enthalpy-based finite element method for nonlinear heat problems involving phase change. *Computers & Structures*, 80(1):9–21.

Oak Ridge National Laboratory (2016). Time warp: Molten salt reactor experiment—alvin weinberg's magnum opus. https://www.ornl.gov/blog/ornl-review/time-warp-molten-salt-reactor-experiment-alvin-weinberg-s-magnum-opus. (Accessed on 2023-01-13).

Organisation for Economic Co-operation and Development (2021). Climate change targets: The role of nuclear energy. https://www.oecd-nea.org/upload/docs/application/pdf/2021-10/nuclear_energy_and_climate_change_-_cop26_flyer.pdf.

Reus, J. W. A. (2021). A numerical investigation of freeze-valve melting based on a linearised enthalpy method.

SAMOSAFER (n.d.a). Concept. https://samosafer.eu/project/concept/. (Accessed on 2023-01-13).

SAMOSAFER (n.d.b). Samosafer. https://samosafer.eu/. (Accessed on 2022-10-07).

Shafer, T. D. (2018). Design and melting behavior of the msfr freeze plug.

Sjodin, B. (2016). What's the difference between fem, fdm, and fvm? https://www.machinedesign.com/3d-printing-cad/fea-and-simulation/article/21832072/whats-the-difference-between-fem-fdm-and-fvm. (Accessed on 2023-01-17).

Sonin, A. A. (2001). Equation of motion for viscous fluids. https://web.mit.edu/2.25/www/pdf/viscous_flow_eqn.pdf. (Accessed on 2023-01-18).

Squires, T. M. and Quake, S. R. (2005). Microfluidics: Fluid physics at the nanoliter scale. *Reviews of Modern Physics*, 77:977–1026.

Stefan, J. (1891). Ueber die theorie der eisbildung, insbesondere über die eisbildung im polarmeere. *Annalen der Physik*, 278(2):269–286.

Swaminathan, C. R. and Voller, V. R. (1993). On the enthalpy method. *International Journal of Numerical Methods for Heat & Fluid Flow*.

Tang, X., Bonner, III, R., Desai, T., and Fan, A. (2011). A 2-d numerical study of microscale phase change material thermal storage for gan transistor thermal management. pages 27 – 34.

Tiberga, M., Shafer, D., Lathouwers, D., Rohde, M., and Kloosterman, J. L. (2019). Preliminary investigation on the melting behavior of a freeze-valve for the molten salt fast reactor. *Annals of Nuclear Energy*.

van den Akker, H. and Mudde, R. (2014). *Fysische transportverschijnselen - denken in balansen*. Delft Academic Press, 4th edition.

van Kan, J., Segal, A., and Vermolen, F. (2014). *Numerical Methods in Scientific Computing*. VSSD.

Vliet, G. C. (1969). Natural Convection Local Heat Transfer on Constant-Heat-Flux Inclined Surfaces. *Journal of Heat Transfer*, 91(4):511–516.

Vliet, G. C. and Ross, D. C. (1975). Turbulent Natural Convection on Upward and Downward Facing Inclined Constant Heat Flux Surfaces. *Journal of Heat Transfer*, 97(4):549–554.

Voller, V. and Cross, M. (1981). Accurate solutions of moving boundary problems using the enthalpy method. *International Journal of Heat and Mass Transfer*, 24(3):545–556.

Voller, V. R., Cross, M., and Markatos, N. C. (1987). An enthalpy method for convection/diffusion phase change. *International Journal for Numerical Methods in Engineering*, 24(1):271–284.

Voller, V. R. and Swaminathan, C. R. (1991). General source-based method for solidification phase change. *Numerical Heat Transfer, Part B: Fundamentals*, 19(2):175–189.

Wimshurst, A. (2018a). [cfd] the simple algorithm (to solve incompressible navier-stokes) [video]. https://www.youtube.com/watch?v=OOILoJ1zuiw.

Wimshurst, A. (2018b). [cfd] what is the difference between upwind, linear upwind and central differencing? [video]. https://www.youtube.com/watch?v=JVE0fNkc540.

Wolf Dynamics (n.d.). Finite volume method: A crash introduction. http://www.wolfdynamics.com/wiki/fvm_crash_intro.pdf. (Accessed on 2022-12-09).

World Nuclear Association (2020). Outline history of nuclear energy. https://world-nuclear.org/information-library/current-and-future-generation/outline-history-of-nuclear-energy.aspx. (Accessed on 2022-09-22).

World Nuclear Association (2022a). Chernobyl accident 1986. https://world-nuclear.org/information-library/safety-and-security/safety-of-plants/chernobyl-accident.aspx. (Accessed on 2022-09-22).

World Nuclear Association (2022b). Fukushima daiichi accident. https://world-nuclear.org/information-library/safety-and-security/safety-of-plants/fukushima-daiichi-accident.aspx. (Accessed on 2022-09-22).

World Nuclear Association (2022c). Supply of uranium. https://world-nuclear.org/information-library/nuclear-fuel-cycle/uranium-resources/supply-of-uranium.aspx. (Accessed on 2022-09-22).

World Nuclear Association (2023). Molten salt reactors. https://world-nuclear.org/information-library/current-and-future-generation/molten-salt-reactors.aspx. (Accessed on 2023-01-17).

Yao, M. and Chait, A. (1993). An alternative formulation of the apparent heat capacity method for phase-change problems. *Numerical Heat Transfer, Part B: Fundamentals*, 24(3):279–300.

Zennouhi, H., Benomar, W., Kousksou, T., Ait Msaad, A., Allouhi, A., Mahdaoui, M., and El Rhafikia, T. (2017). Effect of inclination angle on the melting process of phase change material. *Case Studies in Thermal Engineering*.

Appendices

A. The PIMPLE algorithm

For the PIMPLE algorithm, the Navier-Stokes Equations, Equation 2.4, are expressed in matrix form:

$$\underline{\underline{M}}\boldsymbol{u} = -\nabla p \ . \tag{A.1}$$

The coefficients of $\underline{\underline{M}}$ are known as they follow from the NSE. Then, $\underline{\underline{M}}$ is split into diagonal and off-diagonal components:

$$\underline{\underline{A}}\boldsymbol{u} - \boldsymbol{H} = -\nabla p \ , \tag{A.2}$$

where $\boldsymbol{H} \equiv \underline{\underline{M}}\boldsymbol{u} - \underline{\underline{A}}\boldsymbol{u}$. Equation A.2 is rearranged to isolate $\boldsymbol{u}$:

$$\boldsymbol{u} = \underline{\underline{A}}^{-1}\underline{\underline{H}} - \underline{\underline{A}}^{-1}\nabla p \ . \tag{A.3}$$

Since $\underline{\underline{A}}$ is a diagonal matrix, inverting it is a straightforward operation. Finally, Equation A.3 is substituted into the mass continuity equation, Equation 2.8:

$$\nabla \cdot (\underline{\underline{A}}^{-1}\nabla p) = \nabla \cdot (\underline{\underline{A}}^{-1}\boldsymbol{H}) \ , \tag{A.4}$$

yielding a Poisson equation for pressure.

The steps of the PIMPLE algorithm itself are as follows:

1. Initialise the p and $\boldsymbol{u}$ fields; set them equal to the fields of the previous time step.
2. Solve Equation A.1 for the velocity field. This field does not satisfy the mass continuity equation but serves as a good first guess.
3. Solve Equation A.4 for the pressure field.
4. Correct the velocity field using the pressure field so it satisfies Equation A.3.

Steps 1, 2 and 3 are repeated until desired convergence. Important to note is that in step 1, the $\boldsymbol{u}$ and p fields are now not initialised from the previous time step, but the previous iteration.

B. Average liquid fraction calculation

The average liquid fraction cannot be evaluated directly in *OpenFOAM*. Therefore, a *Python* code has been written to calculate it.

For the overall average liquid fraction, first an array of the volumes of each cell inside the salt has been found $\boldsymbol{V_{cell}}$. This was done via the *writeCellVolumes* option in *OpenFOAM*. Then, in every simulation, for every time step an array $\boldsymbol{\ell_{cell}}$ was created via *Python*. The average overall liquid fraction was found via an inner product in *Python*:

$$\ell_{overall} = \frac{\boldsymbol{V_{cell}} \cdot \boldsymbol{\ell_{cell}}}{\sum_{\text{all cells}} V_{cell}} \tag{B.1}$$

For the liquid fraction for the boundary cells, the first step was finding the cells forming the boundary with the salt and hastelloy. These were extracted from the *polyMesh/boundary* file generated by *OpenFOAM* and their labels were then imported to *Python*. By mistake, the cell volumes were not taken into account in the calculation of the average boundary liquid fraction, so this was calculated via

$$\ell_{boundary} = \frac{\sum_{\text{boundary cells}} \boldsymbol{\ell_{cell}}}{N_{\text{boundary cells}}} \tag{B.2}$$

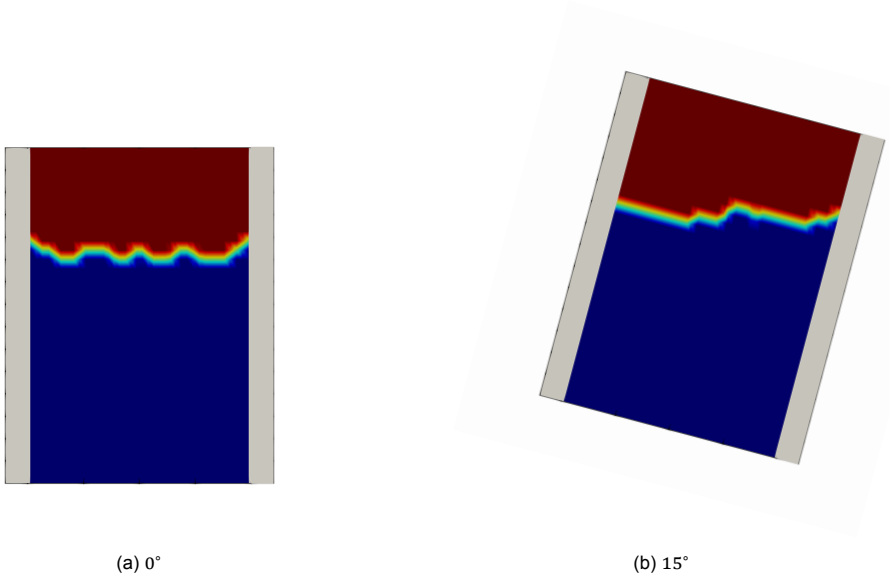
using the same array $\boldsymbol{\ell_{cell}}$ as from the calculation of $\ell_{overall}$. $\sum_{\text{boundary cells}} \boldsymbol{\ell_{cell}}$ indicates the summation over the boundary cells of the elements in $\boldsymbol{\ell_{cell}}$.

The failure to take the cell size of the boundary cells into account is of limited importance. Firstly, cell sizes are symmetrical over the cylinder, so small cells are close to big cells. Besides that, the cell size profile is consistent for every cell layer in the z direction, so e.g. at the top of the cylinder both the smallest and biggest cells are present. The effects of the small and big cells therefore cancel each other out.

Finally, the cell size discrepancy is relatively small. The mean cell size was $1.743 \cdot 10^{-8}$ m^3. The standard deviation was $2.7 \cdot 10^{-10}$ m^3, about 1.57% of the cell size. The minimum and maximum cell sizes were $1.686 \cdot 10^{-8}$ m^3 and $1.775 \cdot 10^{-8}$ m^3, a relative difference of 3.32% and 1.82% with respect to the mean, respectively.

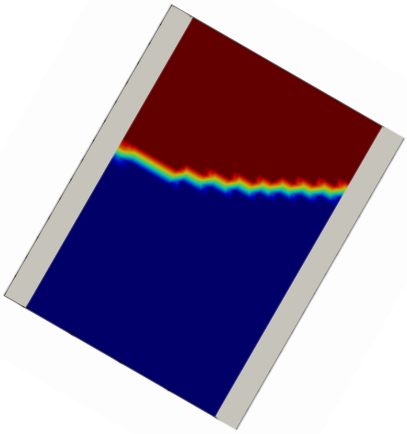
C. Steady-state results for steady-state 40.5K mesh

The liquid fractions at the cross section at $x = 0$ are shown in Figure C.1.

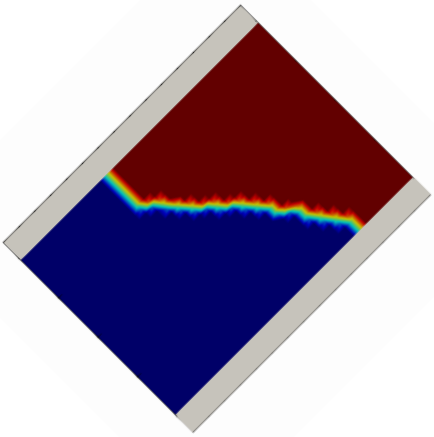


(a) 0°

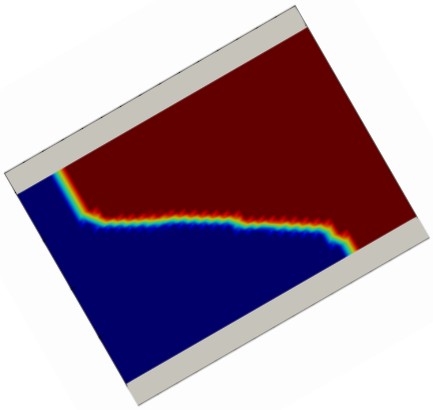
(b) 15°



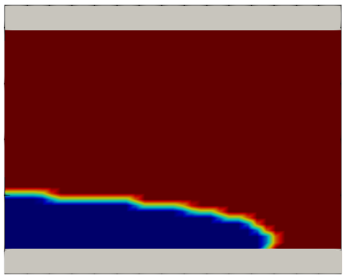
(c) 30°



(d) 45°



(e) 60°



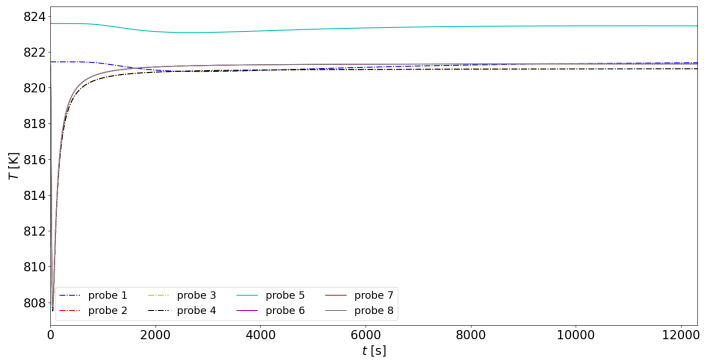
(f) 90°

Figure C.1: The steady-state situations for all inclination angles on the 40.5K mesh. The liquid salt is indicated with red, the solid with blue, while grey indicated the copper and Hastelloy-N for which no liquid fraction was defined. On this mesh, the freeze plug does not form at $\theta = 90°$, it does form at the other inclination angles.

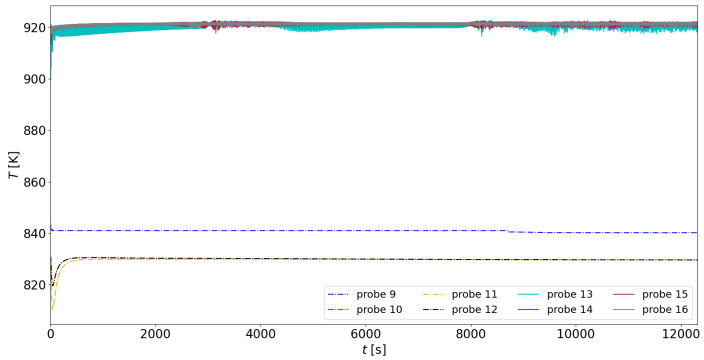
D. Probe results for steady-state 1.1M mesh

The temperatures resulting from the probes for the steady-state 1.1M mesh, except for the ones discussed in section 4.4, are given in Figure D.1.

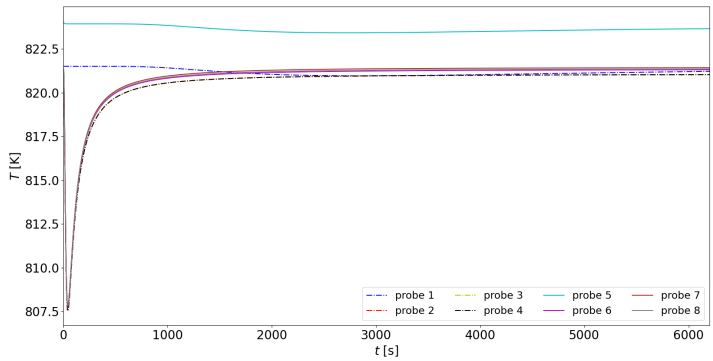
At $0°$, many graphs overlap. This is expected as the case is symmetric, so probes 2, 3, and 4 are geometrically equivalent, and so are 6, 7, and 8, et cetera.



(a) 0°, probes 1 to 8

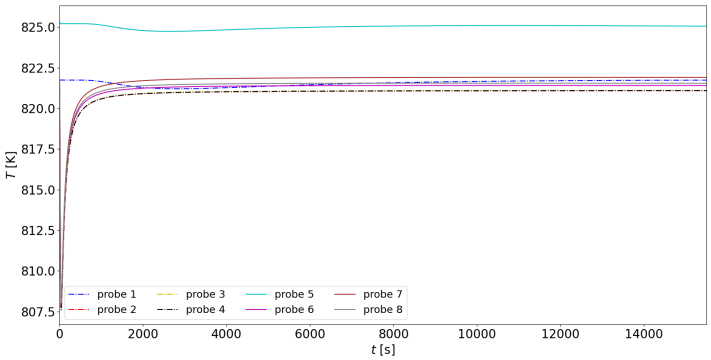


(b) 0°, probes 9 to 16

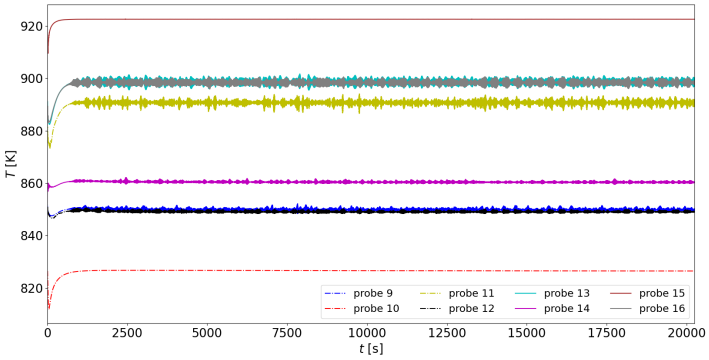


(c) 15°, probes 1 to 8

Figure D.1: The temperature values at the probes for the 1.1M steady-state simulation (to be continued).



(d) 30°, probes 1 to 8

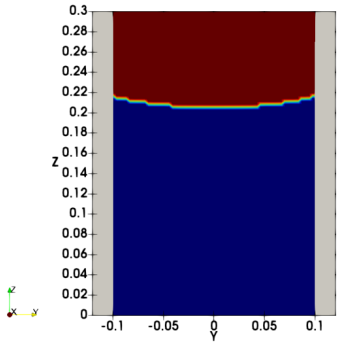


(e) 45°, probes 9 to 16

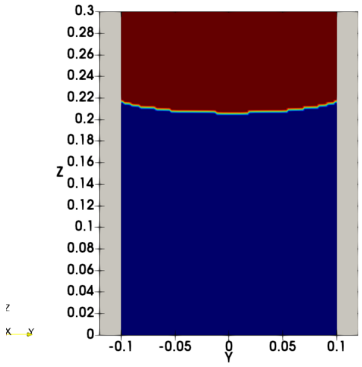
Figure D.1: The temperature values at the probes for the 1.1M steady-state simulation (continued).

E. Steady-state solutions for inclination angles of 0, 15, and 30 degrees.

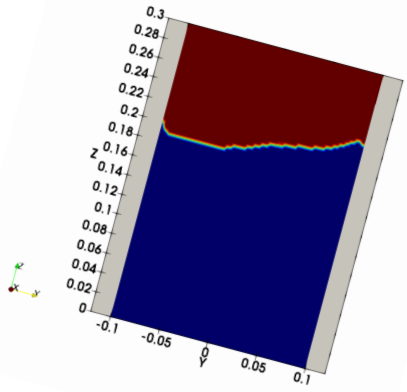
The steady-state solutions for $\theta = 0°, 15°$ and $30°$ are given in Figure E.1



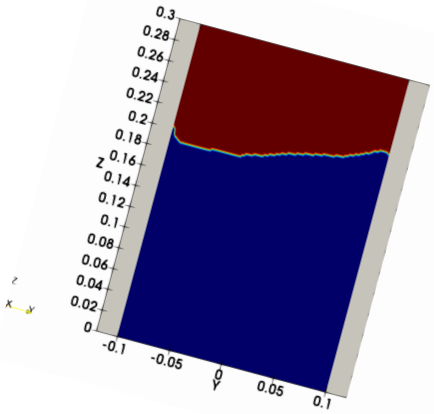
(a) $\theta = 0°$, 1.1M mesh



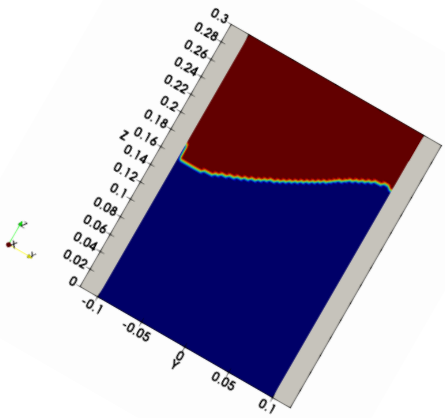
(b) $\theta = 0°$, 3.6M mesh



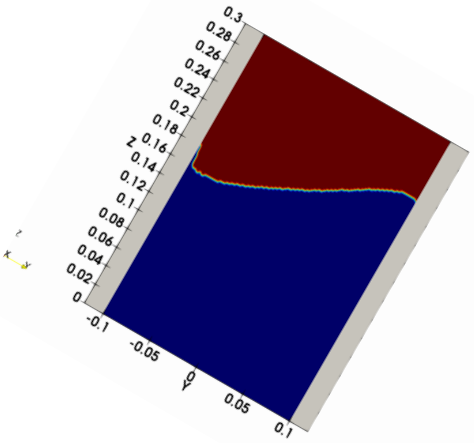
(c) $\theta = 15°$, 1.1M mesh



(d) $\theta = 15°$, 3.6M mesh



(e) $\theta = 30°$, 1.1M mesh



(f) $\theta = 30°$, 3.6M mesh

Figure E.1: The steady-state situation for $0°, 15°$, and $30°$ on the 1.1M and 3.6M meshes.