MODELLING AND INVESTIGATION OF MULTI-PHASE MIXING FOR VITRIFICATION APPLICATIONS USING SMOOTHED PARTICLE HYDRODYNAMICS (SPH)

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Science and Engineering

2022

Georgina R. Reece
School of Engineering

BLANK PAGE

Contents

Word count 42701

List of Published Works

Parts of this research project have been published in the following:

Peer-reviewed journals papers:

- Reece, G. R., Rogers, B. D., Lind, S. J. and Fourtakas G., New instability and mixing simulations using SPH and a novel mixing measure, Journal of Hydrodynamics 32 (2020) 684–698.

Conference papers:

- Reece, G. R., Rogers, B. D., Lind, S. J. and Fourtakas G., A Meshless Model of Two-Viscosity Mixing Using Smoothed Particle Hydrodynamics, in: Proceedings of the 2019 MACE PGR Conference.
- Reece, G. R., Rogers, B. D., Lind, S. J. and Fourtakas G., Appearance of interfacial instability using SPH and measures of mixing between phases, in: Proceedings of the 2020 SPHERIC Harbin International Workshop.

Conference posters and abstracts:

- Reece, G. R., Rogers, B. D., Lind, S. J. and Fourtakas G., Modelling Unstable Mixing of Two Viscous Fluids with Smoothed Particle Hydrodynamics, in: Proceedings of the SIAM UKIE National Student Chapter Conference 2019.
- Reece, G. R., Rogers, B. D., Lind, S. J. and Fourtakas G., Modelling Unstable Mixing of Two Viscous Fluids with Smoothed Particle Hydrodynamics, in: Proceedings of the 17th ERCOFTAC Osborne Reynolds Day 2019.
- Reece, G. R., Rogers, B. D., Lind, S. J. and Fourtakas G., Capturing Interfacial Instabilities with DualSPHysics, in: Proceedings of the 5th DualSPHysics Users Workshop 2021.

List of Tables

List of Figures

List of Acronyms

ALE Arbitrary-Lagrangian–Eulerian. 204

CFD Computational Fluid Dynamics. 32, 34, 36, 53–55, 59, 63, 72, 76, 198

CFL Courant-Friedrichs-Lewy. 9, 15, 46, 161, 167–169

CPU central processing unit. 9, 59, 73, 104, 170, 199

CUDA Compute Unified Device Architecture. 10, 72, 104, 169–171

DBC dynamic boundary condition. 9, 13, 89–91, 121, 122, 136–138, 144, 146, 163, 178, 184, 196

DEM Diffusive Element Method. 61

DNS Direct Numerical Simulation. 59

DPD Dissipative Particle Dynamics. 60, 62

EFG Element Free Galerkin. 61

FDM finite difference method. 54–57, 72

FE finite element. 54, 56, 57, 59

FTLE finite-time Lyapunov exponent. 14, 17, 66, 74, 75, 113, 114, 116, 117, 152, 153, 155, 180, 183, 189, 190, 196, 200–202, 205–207

FV finite volume. 32, 54, 56, 64, 69

FVPM Finite Volume Particle Method. 100

GPU graphics processing unit. 9, 10, 62, 65, 71–73, 104, 113, 169, 170, 199

HAL Highly Active Liquor. 37, 42, 76

HLW High Level Waste. 29–34, 36, 37, 39–41, 43–46, 48–52, 75–77, 80, 151, 173, 192, 194, 198

ISPH incompressible SPH. 63, 65, 100, 204

LBM Lattice Boltzmann Method. 61, 62

LES Large Eddy Simulation. 59

LSS Large Scale Simulation. 59

LUST Local Uniform Stencil. 71

MD Molecular Dynamics. 60, 62, 66

mDBC modified dynamic boundary condition. 13–15, 71, 89, 91, 92, 100, 105, 111, 121, 122, 142, 144–149, 157, 160, 162–166, 171, 172, 174, 175, 178, 184, 196, 199, 201, 203, 206

MLS Moving Least Squares. 61, 69

MPI message passing interface. 72

MPS Moving Particle Semi-implicit. 46, 60

MVA Melt Viscosity Alogirthm. 173, 174, 192, 194, 203

NNL National Nuclear Laboratory. 11, 30, 31, 76, 108

ODEs ordinary differential equations. 62

OpenMP Open Multi-Processing. 104

PDEs partial differential equations. 55

PIC Particle In Cell. 60

PPE pressure Poisson equation. 81

RANS Reynolds-Averaged Navier-Stokes. 58, 59

SPH Smoothed Particle Hydrodynamics. 13, 14, 33, 34, 36, 46, 54, 55, 60–72, 74–78, 82, 83, 87, 89–91, 93, 94, 101, 104–107, 111–115, 117, 118, 120, 122, 127, 129, 133, 134, 136, 138, 142–144, 147, 149–154, 156, 158, 159, 162, 163, 168, 171, 172, 174, 198, 199, 201–205, 207

SPHERIC the SPH rEsearch and engineeRing International Community. 89

VF volume fraction. 17, 65, 112, 116, 117, 151, 153, 180, 196, 200–202, 205, 206

VOF volume-of-fluid. 34, 57, 58, 60, 112, 115, 134, 135, 149, 150, 171, 201

VPC Vitrification Product Container. 39, 40, 206

VTR Vitrification Test Rig. 31, 42, 192, 194

WCSPH weakly compressible SPH. 63–66, 70, 77, 100, 104

WVP Waste Vitrification Plant. 37, 40

List of Symbols

$\langle \cdot \rangle$ SPH integral interpolant. 82

$\mathbf{a}$ Acceleration. 95

β Coefficient of thermal expansion. 47, 80, 160, 185

c Speed of sound. 47, 81, 82, 99, 157, 161

C_0 Courant number. 96

C_p Specific heat capacity at constant pressure. 81, 95, 157, 185

C_s Coefficient for artificial speed of sound. 82

D Shifting coefficient. 100

d Half channel width. 121, 124, 125, 128, 130, 136

$\frac{D}{Dt}$ Material derivative. 78

δ Density diffusion parameter. 99

dt Timestep size. 55, 90, 93, 95, 96, 161

dx Particle spacing. 9, 13–15, 55, 84, 89, 117, 124, 130, 132, 134, 137, 143, 145, 146, 154, 161–168, 170, 175

τ Stress tensor. 78, 97, 98

E_k Kinetic energy. 124, 144

ε Small constant threshold for steady state. 124, 127, 141, 142, 144

F Volume fraction. 112–114, 116, 151

$\mathbf{f}$ Body force $\mathbf{f} = (f_x, f_y, f_z)$. 54, 78, 123

g Gravitational acceleration. 47

$\mathbf{g}$ Gravitational force. 156

γ Polytropic index. 81, 82, 98

h Kernel smoothing length. 15, 82, 84, 87, 88, 99, 105, 117, 127, 161, 166–169, 185

i_g SPH ghost particle index. 93

K Bulk modulus. 81

k Thermal conductivity coefficient. 48, 79, 81, 92, 95, 157, 161, 162, 185

κ Thermal diffusivity. 47

L Characteristic length. 47, 48, 140, 142, 149, 156, 157, 160, 163

L^2 L^2 error norm. 9, 13, 14, 122, 127, 137, 139, 147, 148, 151, 158, 166

Λ Finite time Lyapunov exponent. 113, 116

λ Dilational viscosity. 98

M Robinson's local mixing measure. 114, 115

m Mass. 83, 85, 124

Ma Mach number. 47, 81

$\overline{M}$ New mixing measure. 116, 117

$\tilde{m}$ Viscosity ratio. 128, 130, 131

μ Dynamic viscosity. 44, 47, 79, 98, 108, 109, 121, 140

$\mathbf{n}$ Normal vector. 48, 92

n Number of particles. 114, 116

$\hat{\mathbf{n}}$ Unit normal. 146

$\tilde{n}$ Phase width ratio. 128, 130, 131

Nu Nusselt number. 48

ν Kinematic viscosity. 47, 79, 95–98, 110, 123, 128, 130, 136, 140, 149, 157, 176, 179, 185

P Pressure gradient. 136

p Pressure. 78, 81, 82, 94, 121, 123, 125

Pr Prandtl number. 47, 160, 188

$\mathbf{q}$ Heat transfer. 80, 92

$\dot{q}$ Rate of heat transfer. 79

q_θ Convective heat transfer coefficient.. 48

R Gas constant. 109

$\mathbf{r}_{ij}$ Difference in position between particles i and j: $\mathbf{r}_i - \mathbf{r}_j$. 83

$\mathbf{r}$ Position $\mathbf{r} = (r_x, r_y, r_z)$. 82, 90, 93, 97, 100, 114, 163

$\mathbf{v}_{ij}$ Difference in velocity between particles i and j: $\mathbf{v}_i - \mathbf{v}_j$. 84

r_{ij} Difference in position between particles i and j: $|\mathbf{r}_{ij}|$. 83

Ra Rayleigh number. 14, 47, 160, 162, 188

Re Reynolds number. 13, 47, 96, 136, 140, 142–149, 188

$\overline{\rho}$ Reference density. 81, 124, 157, 185

S Mixing entropy. 115, 116, 154

ρ Density. 47, 78, 83, 85, 91, 93, 94, 99, 110, 140

σ Surface tension. 48

t Time. 14, 15, 93, 94, 113, 116, 117, 153, 160, 162–165

Θ Thermal time constraint parameter. 95

θ Temperature. 14, 15, 47, 78, 80, 87, 89, 92, 93, 108–110, 135, 136, 157, 158, 160, 161, 163–169, 174

$\overline{\theta}$ Reference temperature. 54, 109, 160, 185

θ_g Glass transition temperature. 50, 109

θ_L Liquius temperature. 50

Ω Domain. 82

U Internal energy. 78, 81, 92

V Volume. 81, 83

$\mathbf{v}$ Velocity $\mathbf{v} = (v_x, v_y, v_z)$. 47, 63, 78, 85, 86, 89, 90, 93, 97, 98, 121, 147, 161, 163

$\overline{v}$ Characteric velocity magnitude. 47, 188

W Kernel. 82, 84, 87, 88

We Weber number. 47

Abstract

Georgina R. Reece
A thesis submitted for the degree of Doctor of Philosophy at the University of Manchester
April 2022

Modelling and Investigation of Multi-Phase Mixing for Vitrification Applications using Smoothed Particle Hydrodynamics (SPH)

A new smoothed particle hydrodynamics (SPH) scheme is presented to simulate multi-component fluid mixing with thermally-driven buoyancy and temperature-dependent viscosities. The scheme is developed for multi-phase flows in industrial tanks used in the vitrification process in the nuclear industry for waste immobilisation before long-term storage. These multi-component flows feature a mixing of high-temperature phases composed of molten glass and highly radioactive waste, with fluid properties changing with temperature. SPH is a meshless Lagrangian discretization scheme whose major advantage is the absence of a mesh, making the method ideal for highly non-linear flows with mixing of multiple components whose properties change with time. Emphasis has been given to modelling the thermally-induced buoyancy flows from the external heating and viscous flow instabilities, which are needed to predict accurately the mixing phenomena.

The numerical SPH scheme includes temperature-dependent viscosities, thermal conduction and convection using a Boussinesq approximation, and external heating. A new boundary treatment has been developed to implement adiabatic boundary conditions by extending the recently developed modified dynamic boundary condition (mDBC) to thermally-driven driven flows. The SPH scheme has been implemented to run on a graphics processing unit (GPU), showing speed-ups on the order of 50 compared to optimised code on a multi-core multi-thread central processing unit (CPU).

The code is extensively validated for a range of test cases including two-phase Poiseuille flow where each phase has a different viscosity, temperature dependent Poiseuille flow, two-phase lid-driven cavity flow, and a differentially-heated cavity in 2-D and 3-D. For the two-phase Poiseuille flow, an interfacial instability is observed for certain ranges of viscosity and channel-width ratios. Both stable and unstable flow regimes are correctly predicted. The instability growth rate predicted by the new SPH model agrees with linear analytical theory, which has not been seen before. The evolution of consequent non-linear flow is straightforwardly predicted, with the interface remaining clear.

A new mixing measure is proposed based on the combination of the finite time Lyapunov exponent (FTLE) and the volume fraction (VF) that enables both the degree of mixing at a point and how well-mixed the two phases are at that location to be quantified in a single value. The VF-FTLE mixing measure is applied to the two-phase lid-driven cavity flow, highlighting regions with high particle movement and where both phases are present. A new global mixing measure is established from this local measure, quantifying the change in overall mixing with time.

The new model is applied to a vitrification melter for single-phase and two-phase flows with temperature-dependent viscosities. The VF-FTLE measure is applied to interpret the mixing taking place due to buoyancy-driven circulation within the heated melter. A 3-D flow instability is observed, likely related to a thin thermal boundary layer existing at high Rayleigh and Prandtl numbers, suggesting distinct flow regimes in the melter which requires further investigation.

Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

Acknowledgements

Firstly, I would like to thank my supervisors, Prof. Benedict D. Rogers, Dr Georgios Fourtakas and Dr Steven Lind, for their advice and guidance throughout this PhD. Their knowledge and experience has been invaluable to my learning and progress in academic research. Additionally, my thanks go to the wider SPH group at the University of Manchester for discussions, advice and an encouraging environment to study in.

I would like to thank the Engineering and Physical Sciences Research Council (grant number 1961431) and the National Nuclear Laboratory (NNL) for their funding of this research project. Also, I thank NNL and my supervisors there, Dr Brendan Perry and Dr Mark Bankhead, for providing the motivation for this work and hosting me during my industrial placement.

I give my thanks to the support I have always received from my parents Alison and Simon and my sister Alexandra, and their continued confidence in me. I also appreciate my friends for their encouragement, and for providing a diversion from the desk at times. Finally, thank you to my wife Eleanor for her support and understanding, which allowed me to complete this PhD.

Chapter 1

Introduction

1.1 Background and Motivation

Nuclear energy will contribute to current and long-term energy requirements in the UK and across the world for the foreseeable future. Alongside renewable energy sources and carbon capture, the UK government is investing in nuclear energy to reduce carbon emissions and to meet Net Zero targets by 2050, and potentially to contribute to energy production into the more distant future [51]. Currently, 15 nuclear reactors generate approximately 20% of UK electricity [91]. With eight confirmed new sites in 2012 [52] as the third generation of nuclear power stations [51] (though only one has complete approval at present), there are 13 reactors currently in construction and planning [91].

The UK previously operated a closed fuel cycle, where spent nuclear fuel is reprocessed. Uranium and plutonium is removed to be reused and the remaining High Level Waste (HLW), containing highly active fission products, is immobilised by vitrification for long-term storage. In 2018 the THORP reprocessing facility ceased operation and spent fuel is now stored indefinitely without reprocessing [250], known as an open fuel cycle. With the increase in nuclear capacity, the economic case for restarting reprocessing is greater and so there continues to be research into this topic [3].

One of the priorities of the 2013 policy is for safe decommissioning and disposal of nuclear facilities at the end of their operative life, as well as planning for waste produced during operation of new and existing power stations [51]. The radioactive nature of this waste requires specialised treatment and, for higher level waste with long half lives, long-term storage. This is an ongoing and evolving challenge. Vitrification of waste from post-operational clean-out of nuclear facilities to dispose of residual radioactive components is an active area of research. The current policy for long-term storage is for deep geological disposal, but no site has yet been chosen [91]. Nuclear waste is therefore all currently stored at existing sites [91].

This project is motivated by improving simulation capability of the vitrification waste treatment process in collaboration with the National Nuclear Laboratory (NNL), who provide scientific and technical support for the UK nuclear industry. The main application is the vitrification process, where HLW is mixed with molten glass in a melter before being poured into stainless steel containers for long-term storage. In particular, this work is motivated by observations of cooled vitrified HLW. In some cases, the vitrified HLW had visibly separated into two different phases, as seen in Figure 1.1. Due to the highly radioactive nature of the HLW, it is not possible to observe the vitrification process or take measurements inside the tank. Therefore, alternative methods must be developed to investigate the mixing process and gain understanding of the conditions under which separated phases remain in the vitrified product.

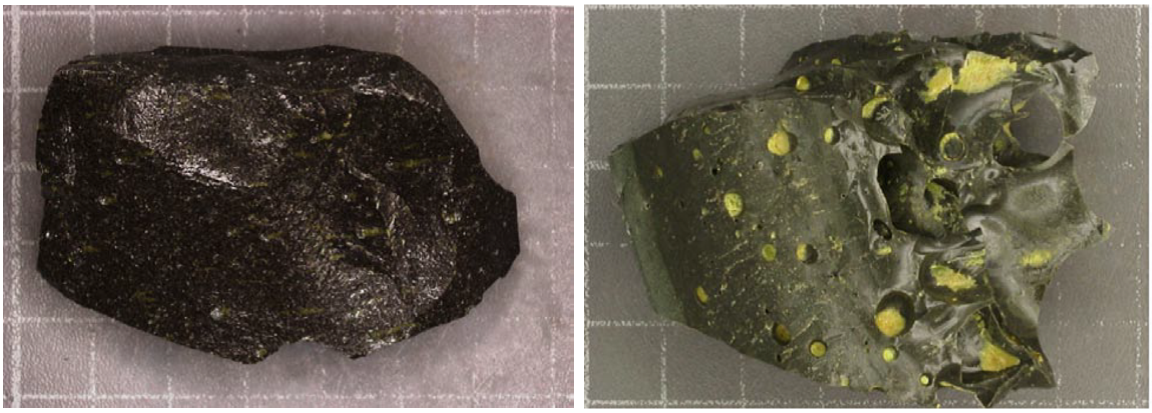


Figure 1.1: Photographs of non-radioactive vitrification experiments performed by NNL with homogeneous product (left) and separated product (right) [81].

Models of long-term waste durability, used to inform storage decisions, assume the vitrified HLW is homogeneous. Therefore, any separation of phases could impact the predictions made. Notwithstanding the large safety margins employed, it is important that durability models of nuclear waste are as accurate as possible to understand any potential weaknesses which may result in leaching of radioactive components. With government policy to maintain the current level of nuclear energy at around one fifth of the country's total after some reactors are retired in the next ten years and possibly increase the proportion to work towards renewable energy targets [249], it is important that this widely-used process for immobilising and storing HLW is well understood. Any adjustments to current predictive models, which may be required when vitrifying HLW from decommissioning, may also take into account these advances in understanding of vitrification.

1.2 Computational Modelling

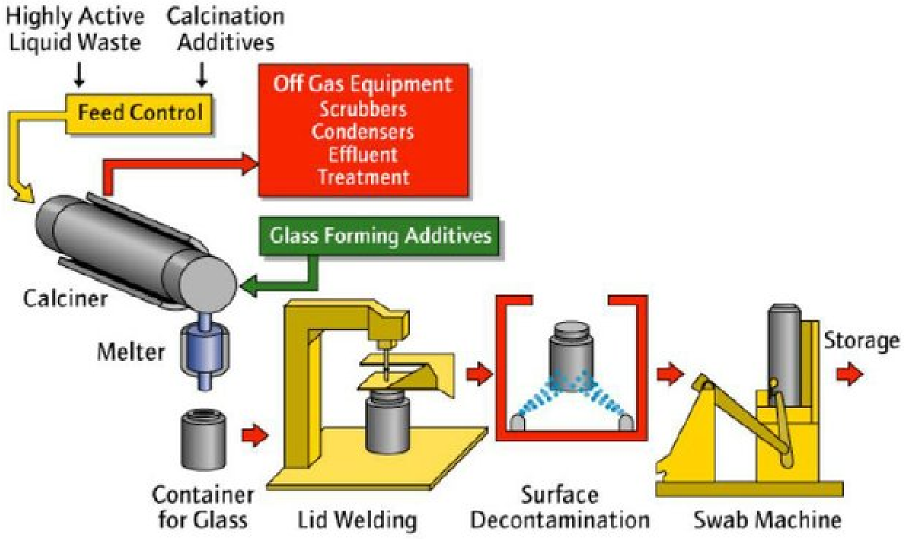


Figure 1.2: Diagram of the vitrification process used at Sellafield [90].

Due to the high temperatures involved in the vitrification process, as well as the radioactivity of the waste, it is not possible to take measurements during the process itself, or the vitrified product. A diagram of the complete vitrification process is shown in Figure 1.2. NNL undertake experiments with non-radioactive analogues at their Vitrification Test Rig (VTR), and are therefore able to take physical measurements of the product, as seen in Figure 1.1 where the appearance of phase separation is demonstrated. However, these experiments are costly and time-consuming to perform, and many difficulties of observing vitrification in process remain.

To gain greater insight into the flows and mixing taking place during vitrification, a computational approach is proposed. This has the advantages of being able to see the flow and mixing at any time and any point within the fluid and, once developed, can be run without any preparation in a short time. Therefore, this alternative approach is less expensive and more flexible. Any computational method needs to be able to simulate mixing by heating through buoyancy-driven flow, as well as deal with the large deformations involved. The inclusion of multiple phases adds extra considerations, including following a changing interface between the phases which may become disconnected or interconnected. Traditional Computational Fluid Dynamics (CFD) methods, such as finite volume (FV), struggle with unphysical diffusion at interfaces, carrying inaccuracies into industry estimations. The modelling method and assumptions must be thoroughly investigated and chosen to ensure the inclusion of all key physics, so that any results are trustworthy.

1.3 Measuring Mixing

During the vitrification process, the glass and HLW within are heated through the walls of the melter. This causes buoyancy-driven flow, which generates circulation and mixing. It has been observed that the product can sometimes separate into two phases, so in these cases it is important to understand the mixing process of any separated phases and how well-mixed they become.

Considering the potential for multiple phases within the flow, a key part of this work will focus on understanding the relative movement and mixing of phases. A number of methods are considered, giving information about the state of mixing at a given instant in time or throughout the simulation. Both local and global measures of mixing are investigated to determine a set of measures which can be used to fully understand the mixing of phases taking place.

1.4 Objectives of this Project

The aim of this thesis is to produce a model of multi-phase mixing, driven by natural convection, which can be applied to flow inside a melter during the vitrification of HLW. Using this model, investigation of conditions under which separated phases mix fully to produce a homogeneous mixture, or regions of separation remain, is undertaken. To realise this model, a number of objectives are set:

- Describe the physical process of vitrification and research previous computational models, motivating the requirement for further research in the modelling of mixing during HLW vitrification.

- Research previous models of multi-phase mixing of viscous fluids to determine the most appropriate numerical methodology and investigate relevant work.

- Describe the Smoothed Particle Hydrodynamics (SPH) method which will be used for the numerical model.

- Develop the model to include the appropriate physics for the application, validating all additions against analytical solutions or well-regarded experimental and numerical data in the literature.

- Develop the addition of new physics in the SPH model, specifically governing equations that account for thermally-driven buoyancy with multiple components where fundamental quantities such as viscosity are temperature-dependent with appropriate boundary conditions.

- Investigate appropriate measures of mixing and develop a measure which includes all key mixing information.

- Apply the model and mixing measures to a case simulating mixing during vitrification, to determine when the product will be well-mixed and when phase separation will occur.

This model can then be put into practice, where the components and mixing process can be adapted to achieve a greater degree of mixing.

1.5 Thesis Outline

In Chapter 2 of this thesis, a literature review is presented. Firstly, the vitrification process is described, including considerations for vitrified HLW, before reviewing current models relating to vitrification and the vitrified product. Next, significant dimensionless parameters for the key physical processes involved in vitrification are examined with the parameter ranges expected. More detail on the observed separation of vitrified HLW is given and existing empirical models are summarised. The current state of mixing models for two phases of similar density is assessed, including those with buoyant convection which drives the flow during vitrification. Current models of mixing are limited by their mesh-based formulation or development for high viscosity ratio or high density ratio applications.

A summary of common mesh-based CFD methods is presented, before investigating multi-phase models with these approaches and outlining meshless methods in general. Section 2.5 introduces the chosen method of SPH, justifying its suitability for modelling vitrification and reviewing current work concerning multi-phase flows and mixing. Next, topics of viscosity, heating and boundary conditions, which are pertinent for this work, are investigated, as well as the importance of hardware acceleration and available software options. Finally, measures of mixing used in related work are examined.

Chapter 3 then presents the equations governing the flow concerned, before describing the SPH discretisation scheme and associated methodology. In particular, the boundary conditions required for the cases in this work and the viscous operators used, as well as numerical stabilisation techniques implemented. Subsequently, code structure of the chosen open-source SPH DualSPHysics software is summarised, along with key modifications for viscosity, temperature, and multiple phases. Lastly, definitions are presented for mixing measures, including a new measure providing additional insight into these particular flows.

This model is validated in Chapter 4 against validation cases with analytical solutions or benchmark results in the literature. Firstly, Poiseuille flow is investigated with one and two phases of different viscosity, to validate multiple phases with different viscosities. Under certain conditions an instability forms at the interface between phases. Both onset and growth rate of the instability agree with results from perturbation analysis in the literature. The development of the interfacial instability can be observed in a level of detail not possible with volume-of-fluid (VOF). A temperature-dependent viscosity is introduced, along with a temperature gradient across the channel, and again compared with analytical solutions.

The next case of lid-driven cavity flow validates mechanically driven flow against widely used results. A two-phase case is introduced and simulated alongside STAR-CCM+ as a benchmark comparison. The existing and new mixing measures introduced in the methodology are assessed using this case and their various advantages and disadvantages examined. The final section of Chapter 4 validates heat conduction and buoyancy-driven flow by the introduction of the governing equation for temperature, using benchmark numerical data for the lid-driven cavity.

The validated model is then applied to the case of vitrification in Chapter 5. A simplified melter geometry is defined and interim cases are run with a single fluid phase and based on the differentially heated cavity to demonstrate the buoyancy-driven circulation. The fluid is then split into two phases of different constant density and the mixing measures applied to interpret the results, before the viscosity-temperature relationships are included.

Conclusions are then drawn in Chapter 6, summarising the work done and results found. Outcomes are compared with the objectives defined in Section 1.4 and the key findings highlighted, while points requiring further investigation are suggested.

Chapter 2

Literature Review

2.1 Introduction

In this chapter the vitrification of nuclear waste is described, with the physical processes summarised and the need for modelling highlighted. The equations governing the proposed model are then outlined, before critiquing current models of mixing for related cases. Next, Computational Fluid Dynamics (CFD) techniques are briefly reviewed, along with a more in-depth review of the chosen Smoothed Particle Hydrodynamics (SPH) method and work related to this application which uses SPH. Finally, measures of mixing are assessed for this work.

2.2 Vitrification of Nuclear Waste

Vitrification is a widely used and well-established method for immobilising and safely storing High Level Waste (HLW) (used for over 40 years in the UK [174]). Vitrification is relatively low cost compared to other storage methods, despite the initial high cost of equipment [174]. The process of vitrification bonds the atoms of nuclear waste into the glass matrix, making the majority of elements inert [24], so that the waste is incorporated both chemically and physically into the glass [174]. Current glasses are able to incorporate up to 40wt% waste oxides [90]. The resultant waste form is durable and stable and so can be stored safely [90], before being moved to geological disposal where the radioactivity decreases over thousands of years. Here vitrification prevents leakage of toxic species into the environment and shields against radioactivity emittance [102].

However, vitrified waste is spending longer in temporary storage than originally expected and designed for, as no geological disposal sites have yet been confirmed [102]. Thus, as more waste is produced, more is stored in temporary locations. In the USA, spent fuel is not reprocessed and most historical waste has not yet been vitrified [102]. Therefore, it remains in temporary storage in its original forms, creating some different issues in terms of temporary storage requirements from the UK.

The vitrification process itself, shown in Figure 1.2, begins with a concentrated liquid HLW slurry of spent nuclear fuel dissolved in nitric acid, along with fission products and residual actinides [90], which is stored in Highly Active Liquor (HAL) storage tanks. The slurry is produced by the reprocessing of spent nuclear fuel in order to remove any remaining plutonium and uranium for other uses. Its exact composition varies depending on the reactor type, use history and reprocessing [90]. In the UK, all civil nuclear fuel is reprocessed at the Sellafield Waste Vitrification Plant (WVP), where the first HLW was vitrified in 1990 [90], and involves a number of steps shown in Figure 2.1.

The slurry first goes through a calcination process, where the majority of chemical processes take place as the temperature is increased. The calciner is a long tube which is heated to 600-840°C electrically while it is rotated [90]. In the first two zones water and any nitric acid evaporate, while in the second two zones the remaining solid nitrates decompose as nitrous oxides and are released as gases [90, 194].

The remaining residue is made up of oxides and nitrate-free salt melts [194], making up a dry powder. It is combined with borosilicate glass frit (typical components given in Table 2.1) in a melter, such that the vitrified product contains 25wt% waste. The alkali borosilicate glasses used have a glass transition temperature of $\theta_g \approx 500°\text{C}$ [90]. The melter is constantly heated through its walls by induction to approximately 1050°C. At the WVP, an "elliptical Nicrofer melter made of 6025 HT alloy" is used, containing approximately 100 litres [90].

SiO_2	B_2O_3	Na_2O	Li_2O
63.42	22.50	11.35	2.74

Table 2.1: Percentage composition of Mixture Windscale - $\frac{1}{2}$Li base glass [90].

During the vitrification process, melting takes place in the cold cap – a floating

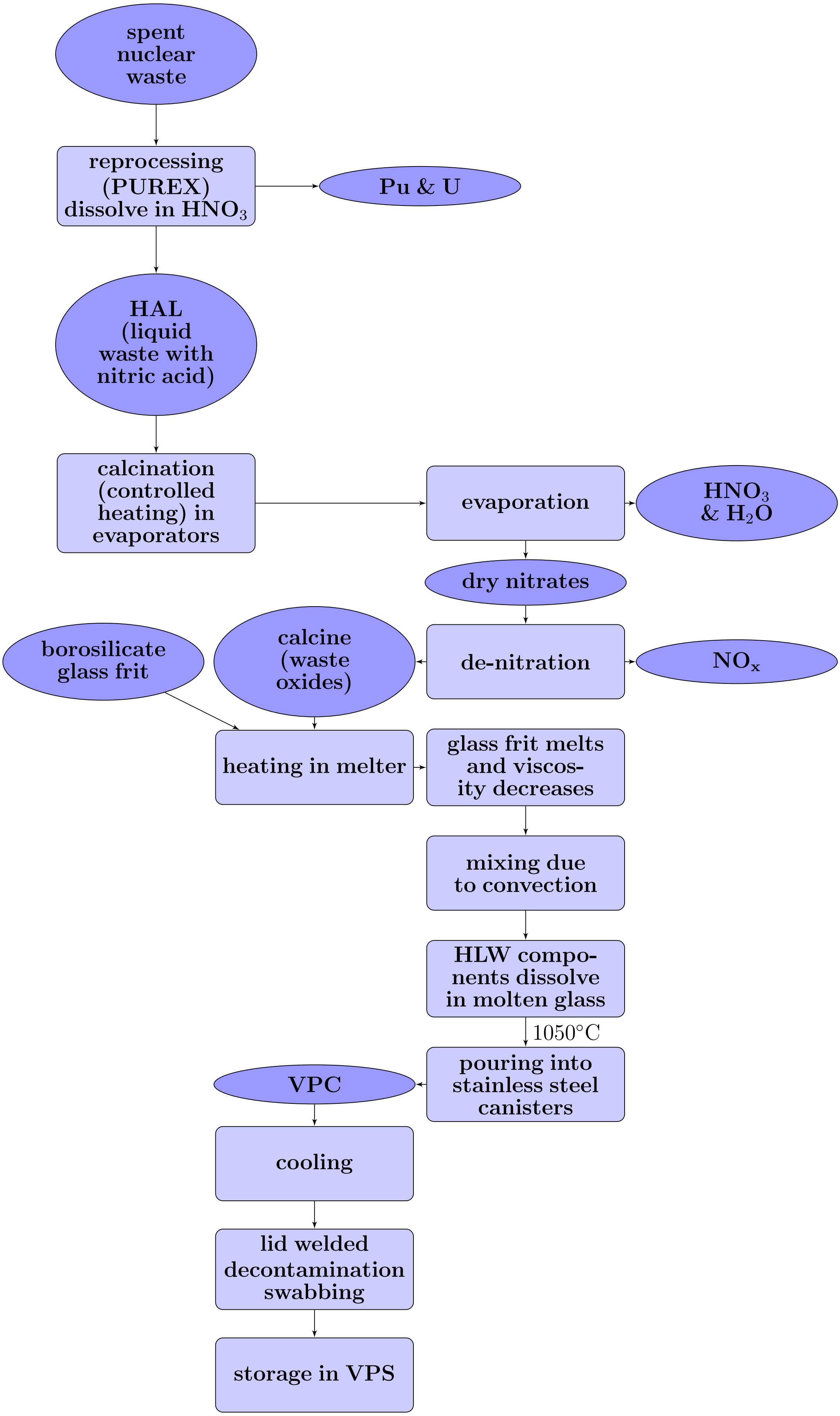

spent nuclear waste
reprocessing (PUREX) dissolve in HNO_3
Pu & U
HAL (liquid waste with nitric acid)
calcination (controlled heating) in evaporators
evaporation
HNO_3 & H_2O
dry nitrates
borosilicate glass frit
calcine (waste oxides)
de-nitration
NO_x
heating in melter
glass frit melts and viscosity decreases
mixing due to convection
HLW components dissolve in molten glass
1050°C
VPC
pouring into stainless steel canisters
cooling
lid welded decontamination swabbing
storage in VPS

Figure 2.1: Flow chart showing steps of vitrification process at Sellafield WVP.

layer of glass frit and powdered calcination product on top of the molten glass [97], as components are added over 8 hours [90]. A sketch of the cold cap is given in Figure 2.2. Hrma et al. [97] found that the dynamics and heat transfer through the foam layer below the cold cap are crucial to be able to predict the rate of melting. At around 600°C, the glass frit softens and begins to melt between 800 and 1200°C [194]. Physically, the molten glass acts as a solvent to dissolve the HLW components within its structure, thus immobilising them. As temperature increases through electrical heating the viscosity of the glass melt reduces, allowing convection flows to form which causes mixing within the melter. Mixing is amplified by 4 argon or air sparge tubes [90] (horizontal pipes with holes delivering spray). Chemical reactions also take place between the calcination product and glass, but it is not known to what extent this contributes to mixing.

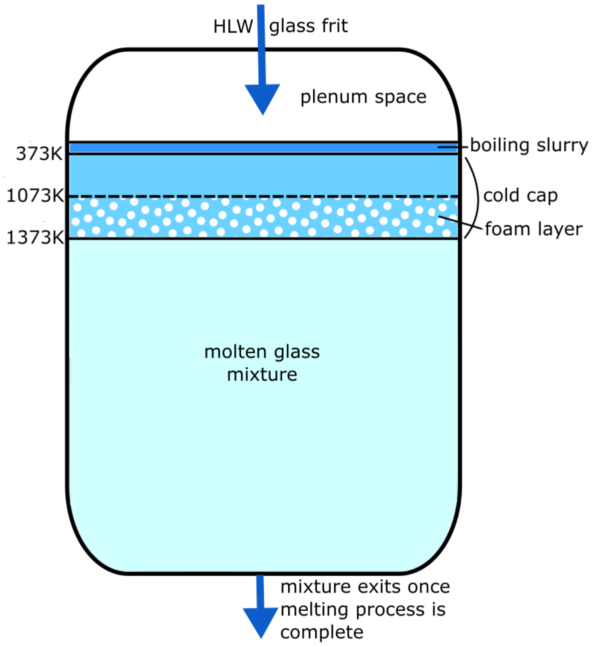


Figure 2.2: Representation of the cold cap in a melter (not to scale) with approximate temperatures labelled [97].

Mixing is deemed complete once the temperature inside the melter is above 1000°C, around which point a solid plug melts at the bottom of the melter and 200kg is poured into the Vitrification Product Container (VPC) [90]. Around 70kg remains in the melter to be included in the next cycle [90], whilst gas and aerosol waste leave separately [174]. Sometimes vitrified waste is also annealed by cooling slowly to avoid cracks forming, though long-term storage requirements are met either way [174]. Each

VPC contains 400 kg of vitrified product (2 pours), and is cooled for 24 hours before a lid is welded on [90] as shown in Figure 1.2. The container is then decontaminated and the outside swabbed to check for radioactivity before being moved to the vitrified product store [90] to await long-term storage in a geological disposal facility.

As well as producing a highly durable waste product, the vitrification process also reduces the volume of waste significantly [24]. Durability and the proportion of waste per volume of glass must be balanced when deciding on the additives to include in the melter [174], though waste loading can be increased by allowing some crystallisation [95]. Although 25wt% waste has historically been used, up to 32wt% can be produced at WVP [90]. Ojovan and Lee [174] found that toughened borosilicate glass takes 2.7×10^{11} years for permanent stress to decrease by 0.5%. Radiation from the contained waste can change the internal structure of the glass, but since it is originally disordered this has little effect [174]. Natural glasses have existed for approximately 300 million years with changes only of $\mathcal{O}(0.1\text{ mm})$ per million years [174].

The chemical durability of the product ensures it lasts for thousands of years, even in a corrosive environment [174] and undergoing high levels of self-radiation. Vitrified HLW is resistant to leaching when in contact with water (for example groundwater) [174], giving a further layer of security once the container has itself corroded. Stainless steel used for HLW containers itself takes thousands of years to breach, since high radiation and temperature reduces after a few hundred years [174]. However, recent research has found accelerated corrosion where water penetrates the very small space between the glass and steel canister after a breach [86]. The affected vitrified waste is only 10^{-6} m thick, but this corrosion mechanism is not typically taken into account in durability models [86]. The leaching rate can also be decreased by adding more aluminium oxide (Al_2O_3) to the mixture [174]. The normalised rate of leaching of radioactive nuclides ($\mathrm{g\,cm^{-2}}$) can be estimated as a function of radioactivity, specific activity, volume and surface area in contact with the water [174]. Therefore, for a reliable calculation regarding product durability it is important to have a good knowledge of the composition of the waste product. When there is inhomogeneity of the product, porosity within the separated phases can increase leaching. Therefore, distribution of the inhomogeneous phases is important to understand.

The effect of self-radiation is modelled experimentally on an inactive simulant glass using short lived emitters or external radiation, but both have drawbacks. Radiation damages the internal structure of glass, causing change in physical and chemical properties. There are three stages of glass alteration: initial fast alteration; transitions to residual rate; and potential alteration renewal [173]. The third stage may be initiated by iron silicates produced when the stainless steel container breaches. Composition also influences the rate of glass alteration [75]. It is important that the vitrified waste remains stable until all radiation has decayed, so potential inhomogeneity adds further uncertainty to the experimental limitations.

Borosilicate glasses are the most commonly used for HLW vitrification and those will be considered in this project. Phosphate glasses can be better at dissolving some waste products than borosilicate glasses but are not used so often [95]. The vitrification method must be 95% accurate without knowing the exact composition of the waste mixture, which contains about 75% of all elements and is highly variable [106]. HLW glasses contain 15-20 components, seven of which make up more than 95% of the chemical composition of the glass [105]. In addition, further components are included to aid the vitrification process. Silica (SiO_2) is added to the waste mixture to vitrify; a threshold amount of silica is required in the waste to avoid needing additives [95]. Boron oxide (B_2O_3) and alkali oxide additives are used to reduce the melting temperature, but increase the volume for storage [95].

Since the waste is highly radioactive, it is important that all equipment can be operated remotely. Therefore, simple and reliable methods and equipment are preferred to reduce the amount of maintenance required [174]. Many processes are involved in the melting, both physical and chemical [241], and happen concurrently therefore it is complex to model [97]. Melting is not the focus of the thesis and will not be considered further. This work will instead concentrate on flow within the melter once melting is complete.

Properties of the glass during vitrification are influenced by the form of the components inserted and by the type of melter [95]. For this project, we shall assume that the melter is externally heated with no internal moving components, similar to the diagram in Figure 2.3. Sun et al. [214] found that the kinematic viscosity of the outflow was dependent on the temperature, but all other physical properties were independent. The temperature is kept between 1150 and 1550 K, with a mixture viscosity

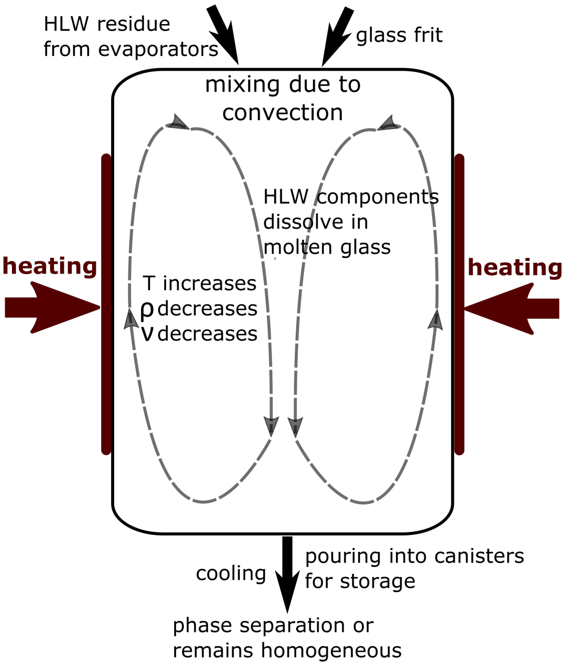


Figure 2.3: Simplified diagram of the melter and mixing within.

of approximately 10-60 Pa s [144]. It is important to maintain a temperature within this range as too high a temperature increases chemical reactions, causing refractory oxides to separate and species which should remain in the glass to boil off [143]. These reactions also weaken the melter, reducing its lifespan. Reducing viscosity by increasing temperature can also cause problems with blending the components [174]. Conversely, too low a temperature can also result in poor mixing if some components are not melted [144], in addition to inhibiting drainage into the storage container if the mixture is too viscous.

The vitrified product is not sampled, instead the process is controlled according to the 'HLWP Process Specification' which samples only the HAL [90]. Product composition and properties are controlled by carefully controlling the base glass and additives, as well as the calciner and melter conditions [90]. The specifications are determined by inspection of the vitrified product from non-radioactive experiments performed at the Vitrification Test Rig (VTR) [90]. The VTR is also used to optimise operational envelopes to increase waste loading whilst retaining an acceptable quality, as well as preparing for processing different waste such as produced by the post-operational clean-out of HAL storage tanks which contains more molybdenum [90].

This work will focus on modelling mixing within the melter. Since melting rates are in the order of hours [90], the borosilicate glass will be assumed to be melted to allow investigation of the flow at smaller timescales. The internal melter temperature will be maintained at the upper end of the borosilicate glass melting range of 1150-1550 K [144] to avoid any change of state. The glasses involved do not have a well-defined melting point, but instead increase in viscosity continuously as they cool [144]. Viscosity ranges of the molten mixture are given in the literature as 1-10Pa s [95, 174], 20-100Pa s [104] and 10-60 Pa s [144] with a density of 0.5-1.2×10^3kg m^{-3} [179]. Velocities within the convective flow are of the order of magnitude 10^{-3}m s^{-1} [215]. These will be discussed in greater detail in Chapter 5.

2.2.1 Current Models of Vitrification

There are numerous numerical models relating to the vitrification of HLW and of molten glass more generally. These include empirical models of homogeneity and computational models of the mixture itself, some of which are described in this section. However, the computational models usually focus on the melting in the cold cap or the stream of fluid leaving the melter, whereas it is the mixing and possible phase separation inside the melter which are key to this project. Homogeneity of the glass is affected by the melter at both macro and micro scales [241], so it will be important to consider both when selecting methods and mixing measures for this work.

Viskanta [241] split the models of commercial glass melters into three categories: empirical, physical/experimental, and computational based on physical and chemical processes. Models of different parts of the melter can be split into coupled submodels [241]. The focus of this project is the mixing of molten glass within the melter. Common assumptions taken to simplify models include: molten glass is incompressible, homogeneous and Newtonian; neglect of variations in composition; Boussinesq approximation for density variations; all other properties are (empirical) functions of temperature; heat transfer is diffusive; neglect of rheology and chemical reactions in the batch [241]. Some of these assumptions are not well-justified and are only taken to simplify the problem [241]. It needs to be assessed carefully which assumptions are appropriate for determining mixing within the melter, in order to determine mixing and separation of phases.

Since only temperature can be measured directly within the melter, known relationships are used by Jantzen [106] to model other properties and extrapolate them outside the range of composition. Jantzen [106] presented an overview of the models used to operate the Joule heated ceramic melter at the Savannah River Site Defense Waste Processing Facility to effectively immobilise HLW. The properties within the melter are therefore controlled by ensuring the composition of the glass stays within a prescribed range [106].

It is important to be able to model viscosity inside the melter, since it has a large effect on the rate of homogenisation [106]. Viscosity of a glass depends on its temperature [174] and also its composition. Hrma et al. [96] found a linear approximation for viscosity as a function of temperature and composition. For low viscosities ($\mu < 10^2$ Pa s) it is common to assume the dynamic viscosity (μ) of molten glass can be modelled by an Arrhenius function [96]:

$$\ln \mu = A + \frac{B}{\theta}, \tag{2.1}$$

where A is a constant, B is the activation energy and θ is temperature. B can be approximated as a sum for a linear mixture with x_i the mass fraction of component i, giving the viscosity as [96]:

$$\ln \mu = A + \frac{1}{\theta} \sum_{i=1}^{N} B_i x_i. \tag{2.2}$$

Therefore, only one variable independent of temperature is needed to describe the relationship between composition and viscosity – the activation energy [96]. Regression analysis was used to extrapolate data for the complete range of temperatures [96]. However, the Arrhenius model is too simplistic over the temperature ranges involved in vitrification [144]. This model does not take into account component interactions [96], which may also contribute to the viscosity. Jantzen [106] also found an empirical model for viscosity depending on temperature and composition. Jin et al. [108] found viscosity of the feed from an empirical model depending on concentration, in order to better understand movement and melting taking place in the cold cap. Differences in the contribution of undissolved gas and solid phases were found depending on the level of radioactivity of the waste [108].

Miller [144] developed a computational algorithm which computes parameters for viscosity-temperature relationships from experimental data. It is able to extrapolate parameters for individual components and extended temperature ranges. More details of the viscosity models used are given in Section 3.5.1, including an Arrhenius function and more complex extended empirical models. There are a number of viscosity models to choose from depending on conditions and composition, and therefore the relative significance of different physics [144].

Based on US government data from Hanford, Hrma [95] found a system of equations to represent the relationships between properties and composition. The composition itself can be split between waste and additive parts [95]. The aim therefore, is to find an additive composition which gives good properties whilst maximising the waste loading [95]. The relative product volume can also be calculated, and the specific volume of the waste glass can be estimated from the components [95]. These relationships can be used to determine the proportion of glass and additivies to be used in vitrification, in order to balance effectiveness and volume for storage. Generally, Hrma [95] suggests to aim for a product volume of approximately two times the calcined HLW volume, in order to minimise waste volume whilst remaining satisfactorily durable.

Jantzen [104] developed a systems model to optimise performance parameters against “processing considerations”, which has been used at the Savannah River Site in the USA. The internal parameters (viscosity, electrical resistivity, liquidus temperature) are difficult to measure during the vitrification process, therefore these are estimated from measurable parameters using empirical equations derived from linear regression of experimental data (given in Jantzen’s paper) [104]. The model itself is based on hydration free energy and gives a measure of glass quality from the composition of the input slurry that has been through an evaporator [104]. Jantzen [104] found that viscosity is the most important parameter for predicting quality of the vitrified HLW, therefore it is important to include a reliable model for viscosity in this work.

Chang et al. [34] developed a model to simulate multi-phase flow inside a glass melter with the aim of being able to predict the quality of the glass and the efficiency of the furnace. The model has three coupled Eulerian components, each with a conservation of mass equation, which are solved one by one for each time iteration [34]. The heating and melting component is particularly relevant to this project, although a Lagrangian scheme will be used in this work. More detail on the differences between Eulerian and Lagrangian methods, as well as the advantages and disadvantages of each in relation to this project, is given in Section 2.4. However, commercial glass melters are long and shallow, which is a very different shape to those used for vitrification of HLW so this model cannot be directly compared. Learnings can be taken from the assumptions which have been made of Newtonian laminar steady state flow and temperature remaining constant at the melting point once it is reached [34]. Here reaction and melting rates are empirical [34]. Again, viscosity was found to be highly dependent on temperature and higher temperature gradients increased recirculation (desirable for homogeneity) [34]. Models of reactions inside the cold cap, where the batch melts, have also been proposed by Hrma et al. [97], Jin et al. [108], Pokorny and Hrma [179] (with a view of adding it to a full melter model). The focus of this project, however, is mixing once all components are melted.

Sun et al. [214] modelled the molten mixture as it discharges from the bottom of a Joule-heated glass melter by introducing a new algorithm to an existing Lagrangian method, in order to understand observed “unpredictable abnormalities”. Since a free surface is involved in pouring, it is important that this model includes surface tension. However this work is concerned with the interface between two similar phases, rather than a free surface, and the significance of surface tension will be considered in Section 2.2.2. As explained later in Section 2.4, to treat the free surface, a mesh-free method is chosen, but the Moving Particle Semi-implicit (MPS) method is preferred to SPH since they claim that the compressible nature of SPH may reduce calculation efficiency [214]. Sun et al. [214] found that numerical stability is governed by the high viscosity, rather than the Courant-Friedrichs-Lewy (CFL) condition. Implicit techniques also remove the influence of the diffusion number on the time step constraint and were found to speed up their simulation by 10^4 times [214].

From the review of vitrification modelling, it is found that there is a gap in numerically modelling the flow inside a vitrification melter. Related models exist for commercial glass melters and for pouring vitrified HLW from a melter, but neither address the flow within the melter which contributes to mixing.

2.2.2 Key Dimensionless Parameters

Dimensionless numbers which characterise the multi-phase flows modelled in this work are considered here. These parameters which have been found to be important in the literature, as well as for the validation cases in Chapter 4. Typical values are given in Table 2.2.

The Mach number is a key variable for fluid mechanics, in particular all weakly compressible model formulations considered herein. The significance and how it is chosen is discussed further in Section 3.2.5. The Mach number is defined to be:

$$Ma = \frac{\overline{v}}{c}, \tag{2.3}$$

where $\overline{v} = |\overline{\mathbf{v}}|$ is the characteristic velocity magnitude of the object and c the speed of sound.

The Rayleigh number defines the onset of unforced convection through a ratio of buoyancy and viscous forces [248]. Below this value, heat transfer is dominated by conduction and above it by convection [259]. Clearly, this will be significant in determining when mixing due to convective flow is occurring. It is calculated as

$$Ra = \frac{g\beta}{\nu\kappa}\Delta\theta L^3, \tag{2.4}$$

where g is gravitational acceleration, β the coefficient of thermal expansion ν is the kinematic viscosity, κ is the thermal diffusivity, $\Delta\theta$ the temperature difference and L a characteristic length.

The Prandtl number is defined to be

$$Pr = \frac{\nu}{\kappa}, \tag{2.5}$$

and typically lies between 3 and 300 for liquids [82]. This ratio gives a measure of the relative significance of thermal and viscous dissipation in the flow. Therefore it is important for the case in Section 4.4, as well as the vitrification model.

The Reynolds number is commonly used to characterise viscous flow. It gives the ratio of inertial to viscous forces and so gives a measure of how laminar or turbulent the flow is [171]. Typically, turbulence is found for $Re > 5000$ but there is no theorem relating Reynolds number to turbulence [259]. It is calculated by [171]

$$Re = \frac{vL\rho}{\mu} = \frac{vL}{\nu}, \tag{2.6}$$

with density ρ and dynamic viscosity $\mu = \rho\nu$.

The relative importance of surface tension compared to the inertia of the fluid can be determined by the Weber number,

$$We = \frac{\rho\nu^2 L}{\sigma}, \tag{2.7}$$

where σ is surface tension. A large viscosity, as is the case for the application of this work, will result in a higher Weber number where inertial forces dominate surface tension. Therefore, surface tension will not be included in this model.

The Nusselt number gives the temperature gradient at a surface or boundary and is defined as

$$Nu = \frac{q_\theta L}{k}, \tag{2.8}$$

where q_θ is the convective heat transfer coefficient and k the thermal conductivity of the fluid. It can be approximated locally as [244]:

$$Nu_{\text{local}} = \left.\frac{\partial \theta}{\partial \mathbf{n}}\right|_{\text{wall}}, \tag{2.9}$$

where $\mathbf{n}$ is normal to the wall or surface. Therefore, it is the ratio of convective to conductive heat transfer [259] and represents non-dimensional heat transfer along a wall [244], since thermal conductivity of a wall can be measured by rate of heat transfer [178].

The high radioactivity of HLW generates significant heating [100], which requires cooling for approximately 50 years after vitrification before long-term storage [170]. HLW contains radionuclides with both long and short half lives [100]. The typical activity level of spent nuclear waste is 10^4-10^6 TBq m^{-3} [100], which reduces to around 10^4 TBq m^{-3} after cooling [14]. During vitrification, the activity level will be somewhere between these values, since the HLW is stored in tanks before being vitrified [100]. Considerable heating will still be taking place, which could influence flow during the vitrification process.

Source	Ra	Pr	Re	We	Nu
Sun et al. [214]					
higher θ	-	-	66.7	4.13×10^9	-
lower θ	-	-	6.67	4.13×10^{11}	-
Ojovan and Lee [174]					
high Na	-	-	1220	-	-
water-water energetic reactor	-	-	1430	-	-
glass composite material	-	-	1130	-	-
Kaushik et al. [113]	-	-	12.0	-	-
combined	1.60×10^9	1790	1200	1.00×10^6	1750

Table 2.2: Dimensionless values for molten borosilicate glass calculated from data from different sources (all to 3 significant figures).

As evidenced by the gaps in Table 2.2, it is difficult to find sufficient data to calculate the dimensionless numbers for the case of vitrifying HLW. An attempt has been made in the last row to consolidate data from different sources in order to obtain results, which will be used to form some general conclusions about the behaviour of the fluids. However, care must be taken in order to match parameters such as temperature so that the resulting values are reliable. This project would benefit greatly from a reliable set of parameter values which could be used to characterise the fluid dynamics involved.

From Table 2.2, it can be seen that Reynolds number is likely to be low and therefore viscous effects will be important to the fluid dynamics and must be included in the model. The Rayleigh number is high and therefore heat transfer is dominated by convection, which will also be key to the movement of the fluid. A Nusselt number well above one again demonstrates that convective heat transfer dominates conductive heat transfer. The high Prandtl number shows that the viscous diffusion rate is greater than the thermal diffusion rate, and therefore thermal boundary layers will be thinner than velocity boundary layers. As expected, the Weber number is very high and surface tension is insignificant in comparison to inertial forces for vitrification cases.

The results of this section determine the physics that must be prioritised in order to successfully simulate mixing during vitrification. These are: multiple phases, heating, and viscosity modelling. Approaches for modelling these components are reviewed in Sections 2.3 to 2.5.

2.2.3 Phase Separation of Vitrified Waste

Phase separation affects the physical and chemical properties of glasses, including viscosity, density, strength and chemical durability [103]. These are all important properties when considering storage of nuclear waste. The durability of vitrified waste is also affected by the chemistry of the glass and its thermal history [105]. As a glass is heated, its properties gradually change until the glass transition temperature, when they change more abruptly and become the properties of a liquid state [174]. There is no sharp transition in state like for a typical solid, as seen in the example phase diagram in Figure 2.4. The glass transition temperature depends on cooling and is approximated by Kauzmann's relation [174]:

$$\theta_g \approx \frac{2}{3}\theta_L, \tag{2.10}$$

where θ_L is the liquidus temperature (assuming no crystallisation). Above the liquidus temperature, the glass is a liquid completely. There are sharp phase boundaries representing the immiscibility boundaries in HLW glasses [105], although changes in state do not occur at a well defined point [144] due to the disordered internal structure.

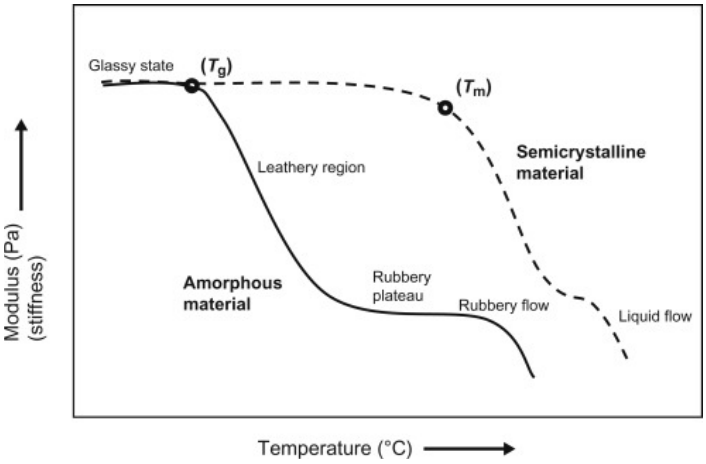


Figure 2.4: Typical phase diagram of semicrystalline and amorphous materials undergoing glass transition when heated [207].

Phase separation in glass occurs as it moves below the liquidus temperature and begins to solidify [67]. Understanding how well mixed the flow is before cooling then becomes essential and is addressed in the next Section 2.3. The separation is caused by ionic interaction which changes the thermodynamics of the system [229]. This critical temperature depends on the concentration of each component in the mixture and the thermal history [67]. Additives such as boron oxide are used to lower the liquidus temperature [95], therefore making phase separation less likely at the high temperatures inside the melter. Below the liquidus temperature, the glass gradually changes into a crystalline aggregate where the crystalline phase is thermodynamically stable [159] at the glass transition temperature (Equation (2.10)). Depending on the composition, crystallisation may only occur when the glass is rapidly cooled, or not at all [159]. Applications such as HLW vitrification require careful consideration of the conditions under which crystallisation occurs in order to avoid it due to the detrimental effects on durability, as outlined in Section 2.2.

Phase separation has been observed in HLW glasses but is not well understood [105]. Borosilicate glasses, like those used in nuclear waste vitrification, separate into phases of mainly borate and mainly silica [67]. Usually, the silica-rich phase is continuous and the borate-rich phase is only continuous if enough borate is in the mixture [67]. Otherwise, it consists of small droplets dispersed throughout the silica phase [67]. Many immiscibility boundaries have been found (including those of borosilicate glasses) [229]. Wakabayashi et al. [242] used conductivity to measure phase separation and found that increasing the addition of molybdenum oxide to nuclear waste glasses rose the immiscibility temperatures and increased the growth rate of phase separation. Stable immiscibility was found above the liquidus temperature, whereas metastable immiscibility was found below it [242].

Both separated phases in vitrified HLW are glassy (non-crystalline) and differ in composition, density and surface tension from the melt as well as generally being immiscible [105]. Two common types of phase separation are found under these circumstances: (i) spinodal decomposition has diffuse phase interfaces and is caused by small fluctuations in composition; (ii) homogeneous nucleation has sharp phase interfaces and requires a free energy barrier to be overcome for a nucleus of the critical size to form and begin the separation, it may lead to crystallisation [105]. Furthermore, phases may separate on a macro- or microscopic level, depending on factors including relative densities and viscosities [105]. Secondary phase separation of already separated phases during cooling can cause silica droplets within the borate phase (microstructures) [67]. Both scales of separation reduce the durability of the product [105].

Phase separation in HLW glasses can also occur when chemical reactions result in the formation of crystallite spinel phases [206]. These spinels are flexible crystallites which can incorporate different cations, including radionuclides, therefore can improve immobilisation. However, this type of phase separation can negatively impact the durability of the product. The use of glass-ceramic materials for HLW immobilisation, where spinel separation is promoted to increase the waste capacity is an active area of research [31, 194, 239]. This method is used in hot isostatic pressing [88], a derivative of vitrification that is being investigated within the nuclear industry.

Ohtsuki et al. [172] investigated the effect of the addition of phosphorus and aluminium oxides to borosilicate glass. They found that all samples exhibited phase separation through spinodal decomposition [172]. For a glass, a third component can increase phase separation if its immiscibility with one of the original components is greater than the combined immiscibility of the original two phases [116]. Extra components can also decrease phase separation [116]. Phase separation in vitrified nuclear waste is important to understand, especially when some parts of the waste are immiscible in glass [174]. In HLW glasses, separation can be prevented by including a minimum percentage weight of aluminium oxide in the composition (this was first suggested by Tovena et al. in 1993 [231]); this is also true of natural basalt [106].

A model was introduced by Jantzen [105] at the Defence Waste Processing Facility in the USA in 1994 to predict for which vitrification compositions (before entering the melter) phase separation would occur. The durability of the product with respect to water was then calculated, using a model based on first principles, in order to determine whether it met the benchmark for geological disposal [105]. This model assumed homogeneity because phase separation negatively affects durability in an unpredictable way, since the compositions of the separate phases are not well known [105].

Current methods of predicting separation of the vitrified product are empirical and depend only on the composition of components, not taking into account the mixing taking place during vitrification.

2.3 Mixing Models

Experiments in the literature which concern viscous mixing include those of Kusch and Ottino [118], who look at two mechanically driven mixing techniques. They found that time-periodic and spatially-periodic chaotic mixing showed fundamental differences in how the mixing takes place. Comparisons between experiments and numerical methods were undertaken by Unger et al. [233], regarding mixing of viscous fluids by an impinging jet contactor. The experiments allowed them to find steady state flow regimes in terms of Reynolds number.

Most relevant to the applications of this model, Swanson and Ottino [216] compared experimental and computational chaotic mixing of viscous fluids with two-dimensional, low Reynolds number flow with the aim of predicting the movement of passive tracers in chaotic experiments. They developed a new technique for tracking the boundaries of the tracer which held when it became more dilute due to large stretching. The movement of tracer was predicted using low-order unstable manifolds, combined with eigendirections of low-order hyperbolic periodic points. This is related to the finite time Lyapunov exponent used as a measure of mixing in Section 3.6.2, which can also identify stable and unstable manifolds in the flow.

Empirical models can give valuable insight to a mixing process but may not capture all the physics, particularly in cases where it is difficult to make internal observations for calibration. Additionally, extrapolating these to regions of untested composition risks missing key dependencies. Computational models can take a measurement at any time and location, without a physical probe disrupting or modifying the flow. Analytical models are the preferred option for simple flows since they can be quick and highly accurate. However, techniques such as linearisation, which make it possible to find a solution, are not applicable when the flow becomes unsteady or more complex, as in Section 4.2.2.2. These more complex non-linear flows can be approximated numerically, with many CFD models developed to solve such systems.

2.3.1 Mechanical Mixing

For flows with little or no density difference within or between phases, mechanical intervention can be used to aid mixing. Mechanically-driven flows are considered in Section 4.3, as well as using mixing measure to investigate the mixing achieved. However, the focus of this review is motivated by the main application of mixing during vitrification, and is therefore on non-mechanical mixing.

2.3.2 Buoyancy-Driven Mixing

Where there is a difference in density within the fluid or between phases, for example due to composition or temperature, flow can be induced by buoyant convection. The behaviour is characterised by the Rayleigh number in Equation (2.4). For natural convection, the body force acting on the fluid is due to gravity acceleration.

A common way of modelling buoyancy-driven flow is to use the Boussinesq approximation by neglecting changes in density, except in the body force term. It is also typically assumed that the fluid is incompressible (or weakly compressible), viscous dissipation is negligible and all other fluid properties are constant within the range of temperatures applied [82]. The approximation of body force is derived from a first order Taylor series of density as a function of temperature about a reference temperature $\overline{\theta}$ [82]. Then the body force acting on the fluid in the conservation of momentum equation (see Chapter 3) is

$$\mathbf{f} = -\mathbf{g}\beta(\theta - \overline{\theta}). \tag{2.11}$$

In this way, the momentum equation and energy equation are coupled. This form is used in many mixing models [46, 62, 167, 255], but some also include a background gravity [82, 121, 124]:

$$\mathbf{f} = -\mathbf{g}(1 - \beta(\theta - \overline{\theta})). \tag{2.12}$$

Some discussion on the Boussinesq approximation and its effectiveness when used with SPH models appears in Section 2.5.4.

2.4 Computational Fluid Dynamics Techniques

A wide range of Computational Fluid Dynamics (CFD) techniques exist, varying in both accuracy and computational efficiency. Different methods are more suited to different problems, depending on the geometry and the physics involved. Since the governing equations of fluid flow are non-linear, care must be taken in the discretisation order to obtain an accurate solution [9]. In all cases, discretisation of the governing equations gives a system of non-linear algebraic equations which are solved numerically [65].

The most common and mature of these discretisation methods are grid-based, for example the finite element (FE) method, finite difference method (FDM) and finite volume (FV) method. Of these methods, FE can be either Lagrangian or Eulerian and the others are typically Eulerian. A general mesh-based method is made up of multiple steps including [238]: defining a domain; dividing the domain into a grid of cells/volumes/elements; selecting the chemical and physical phenomena to model;

defining fluid properties; specifying boundary conditions; solving; and post-processing. However, all methods have limitations, even when combined [76].

Using a mesh makes it hard to keep the numerical compatibility condition equivalent to the physical compatibility condition when the problem is more complex [132], such as the large deformations which occur in the mixing of multiple components. It is also difficult and computationally expensive to produce a good mesh when the geometry of the problem is complex [132]. Versteeg and Malalasekera [238] note that in industry, defining and generating the grid can take more than half of the total time of a CFD project. All grids must balance accuracy and computational efficiency when choosing the coarseness [65], similarly to when choosing the particle spacing in meshless methods, such as Smoothed Particle Hydrodynamics (SPH), discussed later. An adaptive mesh, which is refined in areas where more detail is needed whilst still remaining coarser elsewhere to avoid unnecessary extra computation, can improve the accuracy of grid based methods [238].

All CFD methods can be split into two categories: explicit, where there is only one unknown in each equation (the field variable at the next timestep); and implicit, where multiple unknowns require simultaneous solutions of equations [8]. An explicit method is less complicated but, since the timestep size, dt, is dependent on particle spacing, dx, and a characteristic velocity such as the speed of sound, the timestep size must fall below a stability criterion for the simulation to remain stable [8]. This small dt increases runtime and may even be impracticable. Although parallel computing can reduce this impact, this constraint does not apply to an implicit formulation so a larger dt can be used whilst still maintaining stability of the solution.

2.4.1 Finite Differences

The FDM was the first method used for CFD (and for partial differential equations (PDEs) more generally [65]) and is typically used when the geometry is Cartesian (often this is not the case physically) [9], since a structured grid is always used. The method uses an “algebraic difference quotient” to approximate a partial derivative at grid points, as a function of nodal values of itself and neighbouring nodes [65], and therefore discretise the PDEs [8]. This discretisation is usually based on a Taylor series expansion, where the order of accuracy can be chosen [8]. Second-order accuracy is usually deemed sufficient [65]. These approximations create a system of difference equations from the governing equations [8].

The drawbacks to FDM are the special care needed to enforce conservation, as well as requiring a geometry which can be easily described by Cartesian coordinates [65]. Transformation to a complex grid can be more computationally expensive than the problem itself [132]. As with all methods using fixed grids, it is difficult to find the exact locations of physical phenomena such as moving interfaces [132].

2.4.2 Finite Elements

FE methods use a mesh but the elements of it can be almost any shape, allowing it to accommodate different geometries [9]. It is frequently used in engineering [53], especially for solid dynamics [9].

The unstructured grid means the FE method can deal with any geometry, but also that the matrices of system of equations are not well structured [65]. The generation of this mesh contributes a large part of the computational time [132]. Continuity across boundaries of the grid elements are guaranteed [65] but special techniques are needed for large deformations when the mesh becomes highly deformed [132].

2.4.3 Finite Volumes

FV methods are popular with engineers as the relationship between physical conservation principles and the numerical algorithm is clear to understand [238]. The numerical equations are generated by discretising the integral form of the governing conservation equations before being solved iteratively [8]. There are numerous discretisation schemes to choose from, depending on the application and order of accuracy required [9]. The solution domain is divided into cells, each of which is treated as a small volume element and the conservation equations are applied at each cell [9], with fluxes across cell interfaces ensuring conservation.

The primary benefit of this method is local conservation [9]. It is also suitable for complex geometries since any type of grid can be used [65]. Smaller cells can be used to increase accuracy, along with computation time, but adaptive meshing can increase the rate of convergence whilst minimising the numbers of cells added [9]. However, it is difficult to achieve higher than second order accuracy in three dimensions, due to the three stages of approximation during the process (interpolation, differentiation and integration) [65].

To track an interface, markers are used, but there can be difficulties when an interface breaks, interconnects or disappears [254]. Alternatively, for the volume-of-fluid (VOF) method, the interface may be captured by an algorithm which reconstructs the interface [254]. This can result in the loss of its exact position [224], particularly for a complex dynamic interface [254]. Since new volume fluxes for time integration depend on the inferred interface, this error grows with time [224]. The accuracy is related to grid cell size, so adaptive mesh refinement may be used to increase accuracy whilst minimising computational cost.

2.4.4 Mesh-Based Multi-Phase and Mixing Methods

Mesh-based methods have been used extensively for predicting mixing. Lagrangian mesh-based methods, such as FE, are able to maintain a sharp interface between phases if grid nodes are placed along it [130]. However, the accuracy of the solution is reduced when the mesh is subject to large deformations [130]. Increasing the resolution to mitigate against this can lead to an impractically small timestep [130]. A sliding mesh is sometimes used for stirring in the food industry, where parts of the mesh move relative to each other along an interface, such as the stirrer [169]. Re-meshing is another option to deal with large deformations but, since mesh generation is so computationally expensive, this can make the simulations take much longer to run. Additionally, each re-mesh introduces an error which contributes to diffusion of the material properties [130].

For these reasons, Eulerian mesh-based methods, such as FDM, are more popular in fluid dynamics [130]. These fixed grids do not suffer issues with deformation, but are difficult to apply to a complex geometry. Furthermore, a very high resolution is needed to resolve an interface with sufficient accuracy. For example, Tang and Wrobel [223] used ANSYS Fluent [11] to model the mixing of two immiscible fluid phases by using the VOF method with interface tracking. These interface tracking formulations were found to deform the shape of the interface, requiring computationally expensive reconstruction techniques while interfaces remain inexact [223]. Tartakovsky and

Panchenko [226] state that this can be avoided by a number of methods but then requires solving additional equations, adding extra computation. Pre-conditioning may also be required to reduce the time taken to solve [117]. Nevertheless, diffusion still occurs, as found in Section 4.2.2, and extra computation is required to track an interface fully.

Kunz et al. [117] developed a model of two- and three-dimensional incompressible two-phase flows by generating an implicit algorithm from the multi-phase Navier Stokes equations. They found that they needed to use pre-conditioning of the time derivatives to obtain eigenvalues that are well-conditioned and therefore take less time to solve [117]. The resulting model was found to be accurate for high density ratios [117]. Venkateswaran et al. [236] also found that their iterative method needed either preconditioning or artificial compressibility to be both accurate and efficient, though compressibility was preferred for their model.

In pharmaceutical mixing, which typically includes mechanical mixing such as stirring, the Reynolds-Averaged Navier-Stokes (RANS) equations are often used to find shear rates and determine mixing performance of turbulent flows [16]. Experiments with tracers are undertaken to determine time the taken to mix [16]. Multiple phases may be modelled by mesh-based techniques using VOF [16]. Montante et al. [158] modelled mixing in stirred tanks, including Newtonian fluids. A single phase was used with a tracer to determine mixing in experiments. Both sliding grid and multiple reference frame in ANSYS Fluent [11] were used with RANS equations, and found to accurately predict homogenisation curves and mixing times of experiments [158]. Moving frames of reference, rotating with the flow, are also used to reduce computation time [16]. RANS is commonly used for modelling two-phase flows inside nuclear reactors, due to its simplicity and computational efficiency [22]. However, the position of an interface must be computed as an average of other properties [22], losing information of a deforming interface. To determine the homogeneity of the fluid, concentration of the tracer was measured over time at probes taking measurements through the fluid at different heights [158].

Examples of mesh-based models for viscous mixing include Chang et al.'s [34] coupled Eulerian model of multi-phase flow inside a glass melter and Kunz et al.'s [117] implicit algorithm for incompressible two-phase flows. Bestion [22] considered four CFD methods to simulate thermohydraulic two-phase flows inside nuclear reactors, involving dispersed mixing between fluid and gas phases, and finding limitations to all of them. Large Scale Simulation (LSS) methods were found to work only for specific flows where phases are separate, but fell down when the interface is complex and RANS methods are also limited to simple flows [22]. Although a pseudo-Direct Numerical Simulation (DNS) approach can identify small-scale flow features, the additional central processing unit (CPU) cost due to interface tracking was so high that it made the pseudo-DNS unusable at an appropriate resolution, a hybrid Large Eddy Simulation (LES) method also required filtering of large interfaces [22]. For the low Reynolds numbers expected in the case of vitrification from Section 2.2, it is not necessary to consider a RANS or LES approach for this work. DNS has also been applied to interfacial flows, in particular those involving air and water [197]. However, the computational cost of DNS is high, due to the high resolution required to fully resolve any turbulent fluid structures.

Hybrid methods, involving mapping between an Eulerian and a Lagrangian grid, aim to combine the benefits of both approaches. However, the additional layer of computation can introduce instability and reduce accuracy [129]. Smolianski [211] combined FE simulation of the continuum with a level set representation of the interface to model interfacial flows of two fluid phases, including a Rayleigh-Taylor instability.

Mesh-based methods are limited by the extra computational cost to track the interface, as well as the extra layer of computation introducing additional error. Even then, interface tracking may not be accurate enough and further computation needed for reconstruction of the interface. Without these techniques, flows with complex interfaces cannot be simulated successfully by mesh-based methods. Meshless methods, discussed next, do not have these issues, their particle-based formulations naturally following an interface. Complex domains can also be modelled without any difficulties or additional time required for meshing.

2.4.5 Meshless Methods

Meshless methods are able to deal with some of the limitations of mesh-based methods described in Section 2.4.4. In particular for this work, large deformations do not require expensive re-meshing, and moving interfaces can be followed without additional computation or reconstruction, since particles belong to a particular phase. Most meshfree methods are Lagrangian, with computation points moving with the flow, but there are also Eulerian approaches with fixed particles [130]. Hybrid methods, such as Particle In Cell (PIC) [89], also exist, where a background mesh is used along with particles.

Dissipative Particle Dynamics (DPD) uses a stochastic approach to simulate fluid flows. Both SPH and DPD are well-suited to modelling fluids with complex flow structures [256], and DPD has similar accuracy to SPH when used to model unsteady flows [66]. However, SPH is more suitable for macroscale problems, whereas DPD is better suited to mesoscale problems. Since DPD takes into account thermal fluctuations, which SPH typically does not, it requires a smaller timestep size and significantly greater computation time [66].

Molecular Dynamics (MD) [4] is a particle-based method on the molecular scale, where individual molecules and their interactions are directly modelled. Whilst this is advantageous for capturing physical phenomena on a small scale, simulations with size on the order of metres or larger are computationally prohibitively expensive. Moreover, the small-scale effects may have negligible impact on the overall flow.

The Moving Particle Semi-implicit (MPS) method is an incompressible particle method, based on a Taylor series expansion, where movement depends on neighbouring particles [115]. Koshizuka and Oka [115] found good agreement with experimental data for free surface flows. Similar to SPH, pressure can be calculated by solving the pressure Poisson equation, or through an equation of state [125]. It has been applied to a number of thermal hydraulic cases in nuclear engineering, including melt behaviour inside a reactor [125]. However, it is limited in its ability to model complex multiphase flows and in its original form is only zero-order accurate [125]. No additional computation is required to track an interface, but special treatments are needed to deal with discontinuities in density and viscosity across an interface between phases [125]. Density smoothing results in the same problem of a decrease in interface sharpness as for VOF, while separate computation of phases is unable to deal with complex instabilities [125]. A number of hybrid methods have been introduced to overcome these disadvantages, including MPS-VOF [60], grid-particle [141] and MPS-level set [166].

Nayroles et al. [163] developed the Diffusive Element Method (DEM) by applying Moving Least Squares (MLS) approximations with a Galerkin method. Many variations have since been introduced, including the Element Free Galerkin (EFG) method [21], which is widely used for modelling solids [130].

The Lattice Boltzmann Method (LBM) is capable of modelling multi-phase flows [85] and has been used for the mechanically-driven mixing of air and liquid phases within bioreactors [247]. Its Euler-Lagrange formulation gives it more flexibility to apply to complex geometries and flows with large deformations, whilst it can also be highly parallelised, leading to increases in computational efficiency [16]. LBM is also capable of modelling interfaces between phases [37]. The method simplifies kinetic models on a particle scale in order to reproduce the correct physics at a larger scale [37], and has also been extended to model phases with different viscosities and densities [83]. A forcing term is used to maintain immiscibility of phases [37]. However, this is computationally expensive and can lead to unphysical surface tension [37]. Alternatively, the surface tension can be modified to automatically separate phases [204], but momentum is no longer conserved [37]. To include thermodynamic properties, a free-energy approach was introduced by Swift et al. [217]. Spinodal decomposition, one of the phase separation mechanisms highlighted in Section 2.2, has been modelled using LBM for two fluid phases [5], demonstrating the capability of LBM to model multi-phase flows with complex changing interfaces.

As one of the oldest meshless methods, having been first developed in 1977 [74, 134], SPH is a particularly good choice of method for multi-phase and non-linear flows [129]. The advantages and disadvantage of SPH, as well as a summary of related previous work, are now discussed in the following Section 2.5.

2.5 Smoothed Particle Hydrodynamics

Smoothed Particle Hydrodynamics (SPH) is a Lagrangian discretisation scheme, originally developed in 1977 by Gingold and Monaghan [74] as well as Lucy [134] for astrophysical modelling, but now used in a number of areas including fluid dynamics [76]. Since it is a particle method, no grid is needed [148].

SPH is based on the convolution integral of an interpolation function, enabling expressions for derivatives to be derived from the analytical differentiation of interpolation formulae [76]. The expressions for derivatives enable a system of ordinary differential equations (ODEs) from the conservation equation which govern the model [148] to be represented as particles. The fluid is discretised as a set of points, each representing a point of the fluid with fixed mass, which are free to move and carry material properties. These ODEs can be interpreted mechanically and thermodynamically, using physical intuition, particularly when a Gaussian kernel is used [154].

A small timestep size and a large number of particles are often needed to achieve high accuracy [76], therefore increasing computation time. Increases in computation speed have been made by using parallel and graphics processing unit (GPU) computing [76], discussed in Section 2.5.6.

2.5.1 Suitability for this Thesis

The particle-based formulation of SPH makes it particularly well-suited for modelling fluids with a constantly changing interface between phases undergoing large deformations. It is also well-suited to modelling any breaks in continuity of phases [76], or an interconnected interface [128]. It is able to predict motions even when high levels of strain are involved [240].

SPH is a well established meshless method with a large quantity of literature rigorously investigating and improving its capabilities. The Lagrangian nature of the methods also means that mass is conserved, giving a fair comparison between initial and final states of mixing. In fact, the method is fully conservative [240], provided some care is taken with the formulation of the governing equations.

SPH can more easily be extended to a three-dimensional complex physical model than finite difference methods [148], since there are no extra considerations for generating a mesh. The particle nature of SPH also allows more straight forward coupling with other mesh-free methods, such as MD [132], than for mesh-based methods.

Any hybrid meshless-mesh based methods still suffer from the disadvantages of generating and updating a mesh. SPH has an advantage of lower computational expense over both DPD and MD, as both methods focus on smaller scale phenomena.The LBM is capable of modelling interfaces between multiple phases but needs special treatments to retain immiscibility, which are not required in SPH when the density ratio between phases is close to one. SPH has also previously been used to model viscous flows, including some examples in the next section where more complex viscosity models are used.

The first formulations of SPH for engineering applications involving liquids were weakly compressible (WCSPH), where the pressure is computed from an equation of state rather than a Poisson equation, and it is still commonly used [151]. A newer development is incompressible SPH (ISPH) where the incompressibility constraint $\nabla \cdot \mathbf{v} = 0$ is enforced to calculate pressure [133]. This produces pressure fields closer to well-established CFD methods, whereas weakly compressible SPH (WCSPH) can generate unphysical pressure variations [122], although significant improvements have been made over the past 10 years [12, 64, 147]. ISPH is also more computationally efficient in serial since the timestep size restriction from the speed of sound does not apply, although it does use more memory and computational cores so this may not be the case for large numbers of particles computed in parallel [219]. WCSPH, however, is more simple to code and gives results close to ISPH when smaller particle spacing is used for higher Reynolds number flows, and surface profiles have been found to be similar for both methods [122].

Szewc et al. [219] found in 2012 that the state-of-the-art ISPH produced errors resulting from density accumulation, though new techniques such as particle shifting [127] have been introduced to mitigate this. At this time, WCSPH produced a better model for multi-phase flows, since it is straightforward to vary the equation of state between phases [219].

2.5.2 Multi-Phase and Mixing in SPH

The SPH technique was first applied to two-phase flow by Monaghan in 1997 [152]. Although the interface between phases is not tracked explicitly, it is natural to trace (even for complex movement) as it moves with the particles [132]. Sharp interfaces are maintained between fluid phases, even without surface tension, although methods have been proposed to include a surface tension force [1, 160, 225]. Phases influence each other simply through interactions between particles of different phases. Colagrossi and Landrini [40] developed the method for two-dimensional flows with sharp interfaces separating immiscible fluids with low density ratios, but use artificial viscosity which is empirical. They also used variationally-consistent operators to treat instabilities at the interface caused by the sharp density gradient, as well as an additional term in the equation of state to maintain a sharp interface.

Hu and Adams [98] used a WCSPH method to develop a model for at least three immiscible phases. The discontinuity in viscosity at the interface is treated by assuming a layer in the region of the interface with a different viscous term. However, mass is not conserved because of the methods used for discontinuous density. Their model is capable of modelling multiple surface tensions [98]. A high density ratio SPH model with an incompressible phase was produced by Lind et al. [128], with the two phases coupled through surface pressure and surface velocity. In this thesis the phases will have comparable densities and compressibilities, therefore the model of Lind et al. [128] will not be directly applicable.

A two-dimensional SPH solver was developed by Jimeno et al. [107] to simulate mixing in oscillatory baffled reactors, where a second phase can develop due to crystallisation. Validation performed with FV found that the SPH method was able to reproduce the flow characteristics for a single phase case [107]. When density smoothing was employed to mitigate against the density fluctuations that are present for WCSPH, the method was able to obtain a more consistent quasi-steady-state and greater cycle-repeatability than FV [107]. The nature of SPH means no additional models are required to quantify the mixing which is taking place, since individual SPH particles can be tracked over time [107]. By contrast, FV requires Eulerian-Lagrangian coupling to gain this information, which is more computationally expensive [107].

A water-soil model, including suspension, using SPH was presented by Ulrich et al. [232] in 2013. SPH has been adapted for two-phase mixing between a Newtonian water phase and non-Newtonian sediment phase by Fourtakas and Rogers [68]. Zubeldia et al. [260] developed this work further by including a second yield criterion for erosion. A non-Newtonian two-phase SPH model was applied to debris flow by Han et al. [87]. Work by Ghaïtanellis et al. [72] used SPH to model granular flows with a solid phase and a liquid phase representing the yielded soil. Previously, Manenti et al. [137] used weakly compressible SPH to model both liquid and granular phases as viscous fluids, as did Kwon and Monaghan [119, 120] who had two sets of SPH particles representing both liquid and dust phases as fluids. A non-Newtonian landslide interacting with water is considered by Xenakis et al. [252] by using incompressible multi-phase SPH and smoothing density and viscosity over the interface. The flushing of sediment in a reservoir, with both liquid and granular phases modelled as SPH fluids, has been studied by Manenti et al. [138], while Nguyen et al. [168] applied SPH to sediment scour and transport under a submerged bridge. ISPH was also used by Pu et al. [182] to model sedimentation of a fine dust phase, along with a unified pressure equation for interaction between phases. Bed load transport is modelled using SPH by Amicarelli et al. [7] with a liquid phase and a solid granular phase mixing. The model successfully simulated both the free surface and bed-load transport [7].

Mokos et al. [146] also considered two or more phases when modelling a free surface and mixing effects. Here, WCSPH is used, along with a modified Fickian shifting to reduce the formation of unphysical voids [146]. The model is extended to three dimension with the aid of a GPU to speed up computation [146]. A high density gradient at the interface has been found to lead to inaccuracies when compared to well-documented results [220], or a discontinuous velocity which is not physical [226]. Tartakovsky and Panchenko [226] solve this by using either a continuum boundary force or a pairwise forces at the interface, in addition to prescribing a surface tension. Grenier et al. [80] treat discontinuities at an interface by determining particle volumes through the continuity equation. However, the very small timestep required makes it prohibitively computationally expensive to run [220]. For this work, there will be discontinuities at the interface, although material ratios will not be so high, since phases will represent regions of the same material with differing compositions.

A model which focusses on mixing of viscous phases is Monaghan and Kocharyan's [156] two-dimensional model, which allows interpenetration of phases by means of a volume fraction (VF). Fragmentation of the interface (particularly for negligible surface tension) was mitigated by Szewc et al. [221] by either introducing repulsive forces close to the interface or by making density and viscosity smooth across the interface, in order that phases remain immiscible. Robinson [188] modelled viscous mixing using SPH by including a viscous term with the viscosities of both interacting particles, and produced results comparable to published experiments for 2-dimensional mixing at low Reynolds numbers. In order for phase separation to occur, Tartakovsky et al. [227] replace the equation of state with a modified van der Waals equation similar to a MD approach. This generates attractive and repulsive forces for each fluid component, keeping different phases separated.

SPH has been applied to a number of mixing processes in industry, for example production of foods, pharmaceuticals or polymers. These processes tend to be mechanically-driven, since SPH is better able to manage a moving boundary than a meshed method, and target homogeneity of the end-product. Robinson and Cleary [189, 190] modelled a two-dimensional twin cam mixer using single-phase SPH. Comparisons with experiments were made by computing a finite-time Lyapunov exponent (FTLE) field to determine whether the location of manifolds in the flow agreed with the distribution of tracer dye [190]. FTLEs have also been employed in three-dimensions in order to locate Lagrangian coherent structures in flows [196]. In Section 3.6.2, FTLEs are defined and later applied to validation cases in Chapter 4 in order to observe and quantify the mixing flows. The same twin cam mixer was simulated by Eitzlmayr et al. [57] with a modified boundary method for curved walls.

Shamsoddini et al. [203] investigated three particle methods for modelling micro-mixing in a cylindrical paddle mixer. A modified WCSPH formulation was able to successfully dampen unphysical instabilities [203]. Homogenisation under high pressure to produce an emulsion from a continuous phase and a droplet phase was simulated with SPH by Wieth et al. [246]. The droplet deformation and fluid velocity displayed the same behaviour as experimental results, but the resolution was too low for quantitative comparisons [246].

The modelling approaches described in this section each model elements of the mixing occurring during the vitrification process. Ideas from these will be combined, focussing on the physical processes found most significant in Section 2.2.2, to produce the method in Chapter 3 with the aim of gaining insight into mixing during the vitrification process.

2.5.3 Modelling Viscosity with SPH

To include a viscous term in the momentum equation, it is necessary to calculate a second order derivative. However, the second order kernel derivative is very sensitive to particle disorder [135]. Since this is also computationally expensive, a numerical approximation is often used and based on a Taylor series expansion.

An artificial approximation of the Newtonian viscous term was introduced by Monaghan and Gingold [155] to help maintain stability in simulations. However, this term does not have a physical derivation and is therefore suitable when only diffusive properties are important, rather than accurately modelling viscosity. Although an approximate viscosity can be obtained from the empirical parameter, the resulting Reynolds number is only as high as the physical Reynolds number for very high resolution [27].

Morris et al. [161] developed a viscous term which approximates the second derivative. This form does not exactly conserve angular momentum and does not lead to total dissipation [27]. A viscous component due to compressibility may be neglected for small Mach number [27], as with the assumption of weak compressibility, so no bulk viscosity is included. Fatehi and Manzari [61] developed a second-order accurate formulation for a second-order derivative. This can be applied to the Laplacian viscosity or temperature terms.

Stokes' hypothesis states that in a Newtonian fluid the bulk viscosity is zero. However, when Colagrossi et al. [42] followed an SPH consistent formulation they found it contradicts Stokes' hypothesis. The value of bulk viscosity obtained was close to that for monatomic gases and common liquids such as water, but for many liquids should be higher [42]. Therefore, it is important to take into account the fluid being modelled, as well as compressibility, when considering whether to include bulk viscosity or follow Stokes' hypothesis. The weakly compressible SPH formulation will be used in this work, so any compressibility will be small. Additionally, the Boussinesq approximation employed for buoyancy-driven flow assumes that density only varies in the body force term. Therefore, bulk viscosity is not expected to have a significant effect.

A viscous term of the same form as introduced by Monaghan and Gingold [155] can be derived physically without any empirical parameters [27]. This term conserves angular as well as linear momentum, but includes an extra viscosity term and does lead to total dissipation [27]. When bulk viscosity assumptions are made, the extra term is found to represent a bulk viscosity which is a constant multiple of the dynamic viscosity [27]. The extra term is connected with compressibility dissipation [27], and is therefore important to include when modelling processes such as explosions where compressibility is physically significant. Bonet Avalos et al. [27] developed a new viscous term which, when included with the Monaghan and Gingold term, adds a bulk viscosity which is independent of the shear viscosity. In this way they are able to model phenomena such as sound waves whilst conserving both linear and angular momentum [27].

For this project, viscous terms derived from approximations of the second derivative will be used. The significance of including the bulk viscosity term when modelling variable viscosity is discussed in Section 3.3.6.2. Throughout the validation cases in Chapter 4 the Morris [161], and Monaghan and Gingold [155] operators shall be considered in terms of their ability to simulate flows with two viscous phases.

2.5.4 SPH with Heating

To include heating in the SPH model, a governing equation for temperature or energy must be included. Additionally, heat transfer is included by prescribing a temperature or temperature gradient at the boundary. A governing equation for temperature was first introduced by Monaghan [149], before Cleary [38] developed a more sophisticated form, which is commonly used today.

Cleary and Monaghan [39] developed one of the first models for thermal conduction using SPH, improving on a previous method for thermal conduction in stars [28]. The improvements allowed for variation in the conductivity, whilst maintaining continuity of the heat flux [39]. The method was demonstrated to perform well for thermal conduction in solid domains [39]. Later, Schwaiger [200] developed a new approximation for the Laplacian operator, present in the governing equation for temperature, in order to better simulate thermal diffusion with SPH.

Buoyancy-driven flow is typically modelled in SPH through the Boussinesq approximation [38, 218, 255], coupled with the temperature governing equation, including in incompressible SPH [124] since the approximation states density only varies in the Boussinesq term. Leroy et al. [124] found good agreement with mesh-based methods for Poiseuille flow, but struggled to reproduce the local Nusselt number in regions with high temperature gradients. This was attributed to the error in SPH gradient approximations [124].

Ng et al. [167] used the Adami et al. [2] ghost boundary method with a Dirichlet condition for temperature to model heat transfer through a wall. This was preferred to a MLS method with mirroring, which does not retain its higher order of accuracy once particles are disordered and is more computationally expensive [167]. Sikarudi and Nikseresht [209] also found MLS to be more sensitive to disordered particles than SPH when modelling heat conduction. They introduce a new boundary method to implement an adiabatic boundary condition, which ensures no heat flux through a boundary [209]. The intuitive method of excluding boundary particles from interpolated values is unable to completely stop heat transfer due to the incomplete kernel [209].

SPH has been used to model heat transfer over longer periods of time when modelling the growth of semiconductor crystals [192]. In order to increase the time step size, and therefore reduce computation time, an explicit time integration scheme was employed [192]. It was shown that SPH was able to compete with FV, both in terms of results and simulation time [192].

2.5.5 SPH Boundary Conditions

Since the vitrification process takes place within a bounded domain, accurate boundary conditions will be key to capturing the details of the flow, and therefore the mixing occurring. Although one of the key benefits of SPH is its versatility with complex geometries, the simplest boundary conditions are of a low order of accuracy and are known to produce poor velocity gradients near the boundaries. Therefore, care will need to be taken when choosing the boundary conditions for this work in order to capture the complex mixing taking place inside a vitrification melter.

Boundaries are simple to implement, but complicated to perfect in SPH. In fact, boundary conditions were named as one of the current grand challenges for SPH in the recent paper by Vacondio et al. [234]. The choice of boundary method influences convergence of first and second order differential operators in the limit as the kernel smoothing length tends to zero [135]. This is important for viscous and dissipative or diffusive terms, which will influence the flows modelled in this work, independent of the chosen viscosity model [135].

Solid boundaries were first modelled simply by including an extra layer of particles which do not move, known as dummy particles, at the edge of the fluid domain [114]. However, this results in an incomplete kernel support for any fluid particle near the boundary. This can allow penetration of boundaries by fluid particles, as well as inaccurate interpolation of fluid properties near the boundary. Early solutions employed a potential-based boundary force along the inward normal to prevent penetration [151]. The truncated kernel can be completed by using layers of boundary particles [184], or numerically using a boundary integral method [136, 222]. With fixed dummy particles, viscous forces diverge near the boundary as the smoothing length tends to zero, and a slip velocity error is present at steady state [135]. Schwaiger [200] corrected the SPH discretisation of an incomplete kernel with a Taylor series. The method was found to work well for gradients of functions, but needed further refinement for second order derivatives [200].

Adami et al. [2] introduced a boundary condition using dummy particles whose quantities do not evolve through the timestepping method, but depend on both surrounding fluid particles and a boundary condition. The technique is designed to prevent any fluid particles from penetrating the wall by increasing density as fluid particles near the wall. A no-slip boundary condition is implemented by extrapolating the fluid velocity field onto the dummy particles in the boundary. Although this method is only first order accurate, it retains this level of accuracy when particles are disordered [167], as in common in WCSPH. However, Szewc et al. [220] found that dummy boundary particles were not sufficient for multi-phase flows and instead employed ghost particles. Dummy boundary particles are initially used in this work, due to their simple implementation, but are found to be inadequate for modelling flow within a cavity, an integral part of the end application of this project, in Section 4.3.

Basa et al. [20] evolved density and pressure of boundary particles with the governing equations, but calculated boundary particle velocity by reflecting fluid particles into the boundary to avoid penetration. Boundary particles are given opposite normal velocity to their origin fluid particle, while tangential velocity is extrapolated along a normal to the wall. The computation of boundary normals is more challenging and expensive to implement with complex geometries.

Ghost particles were introduced by Colagrossi and Landrini [40] for modelling interfacial flows. An antisymmetric mirroring of velocity into the boundary is used. Although viscous forces are independent of smoothing length at the boundary, they are “incorrectly calculated” [135]. This boundary method also introduces a local error in velocity [135]. Symmetric mirroring of velocity can be used to enforce a free-slip boundary condition. However, they are not appropriate for modelling a no-slip boundary condition as a slip velocity error causes velocity to diverge near the boundary [135].

The more recent Local Uniform Stencil (LUST) method of Fourtakas et al. [69] uses a stencil of particles around each fluid particle. These are included in SPH interpolations when they lie inside the boundary, so that fluid particles always have a full kernel support [69]. This approach needs no modification to extend to complex geometries and can be easily parallelised to run on a GPU [69]. However, this method is complex to implement initially, and can otherwise be computationally expensive.

The recently introduced modified dynamic boundary condition (mDBC) method [59] extrapolates density onto boundary particles using the corrected kernel approximation of Liu and Liu [131]. No-slip boundary conditions are enforced by mirroring the negative of a Shepard filter sum of the fluid velocity [55]. This method can be extended for a free-slip boundary condition [59]. The mDBC method is chosen to use in this project for its simple and efficient implementation. Additionally, it is straightforward to apply to other variables. In Section 3.3.4.4, this method is extended to model an adiabatic boundary condition for temperature.

For thermal boundary conditions, Dirichlet or Neumann boundary conditions may be used, as summarised in Section 3.3.4. For a Dirichlet condition, any boundary particles can simply be given the prescribed boundary value. A straightforward way of enforcing a homogeneous Neumann condition (adiabatic for temperature) is to exclude boundary particles from the SPH sum [39]. However, this truncates the kernel near the boundary and can lead to problems similar to those outlined above for dummy particles. Alternatively, the values in the fluid may be reflected across the boundary [200]. On the other hand, using the negative of the fluid value in the boundary gives the homogeneous Dirichlet condition [200], where the value is zero at the boundary (for example, no-slip velocity). This method can increase the accuracy of Laplacian term approximations near the boundary [45].

For a heterogeneous Neumann boundary condition the flux across the boundary must be prescribed, which is more complicated still to implement in SPH. For example, the unified semi-analytical wall boundary approach [63, 142] uses an extrapolation onto boundary particles, so that the segment between these points has mean value of the flux, to produce a first-order Neumann boundary condition. Recently, Joubert et al. [109] applied a gradient correction at the boundary surface in order to enforce Neumann boundary conditions for pressure and velocity. Alternatively, Bai et al. [17] developed a new SPH-FDM method which converts a Neumann condition to a Dirichlet condition, in order for it to be applied to the SPH part of the method.

2.5.6 Acceleration and Software Options

For a high-resolution SPH simulation in three-dimensions, a large number of particles are required. This takes up a large amount of memory, and the resulting interactions require considerable computation time to complete. In particular, when there are high viscosities and thermal conductivity coefficients involved, as is the case for this project application, the timestep size (defined in Section 3.3.5.1) can become very small. Hardware acceleration, in terms of parallelisation, as well as efficiently written code, can reduce the runtime of simulations. There are several options to accelerate CFD codes and SPH: (i) multi-core shared memory (using openMP), (ii) multi-node distributed memory (using the message passing interface (MPI)), (iii) accelerator devices such as graphics processing units (GPUs), or (iv) a combination of these. Thanks to the compact support of SPH with limited numbers of neighbours around each particle, the SPH method is well-suited to acceleration via parallelisation using one of these aforementioned approaches. In particular, a graphics processing unit (GPU) can be exploited to run many small calculations concurrently, such as the SPH summations.

For astrophysical applications, the SPH options include GADGET [212], SWIFT [199], and PHANTOM [181]. Additionally, there are a number of commercial software with SPH capabilities, including: IMPETUS Afea solver [99], LS-DYNA [10], Altair nanoFluidX [6], Neutrino [32], SimPARTIX [71], and Nextflow [165].

The open-source SPH software options which make use of GPU acceleration are AQUAgpusph [33], GPUSPH [23] and DualSPHysics [55]. AQUAgpusph uses OpenCL parallelisation, suited to AMD GPUs, whereas GPUSPH and DualSPHysics are written with Compute Unified Device Architecture (CUDA) for use with Nvidia GPUs, which are available for this project. Of the three solvers, DualSPHysics was found to run the fastest [33]. All options feature a number of choices of boundary method, while GPUSPH and DualSPHysics include options for density diffusion and shifting, as well as different viscosity models. The multi-phase formulation in GPUSPH is tailored to simulate granular flows found in lava flows. In DualSPHysics [55], however, there are multi-phase options for liquid-gas formulations, as well as Newtonian and non-Newtonian fluid phases.

Currently, none of the release versions of these open-source codes run on multiple GPUs, but it is an ongoing area of research in order to further increase computational capacity.

The open-source SPH solver DualSPHysics [44] is chosen to be adapted for this model for its current capabilities, the capacity for modification, as well as the knowledge and expertise present within the research group. The DualSPHysics code is rigorously validated and widely used in both academia and industry, providing a highly optimised code that runs on different hardware including CPUs or GPUs [44]. An overview of the structure of the DualSPHysics code is given in Section 3.4. Currently the solver does not include a governing equation for temperature or any heating effects. A multi-phase option with non-Newtonian phases has recently been released [55]. Details of the modifications and additions made are given in the later sections of Chapter 3.

2.6 Measures of Mixing

Mixing measures are essential to assess the degree of mixing achieved by applications such as vitrification. Experimentally, tracers are typically used to demonstrate mixing. Examples include dyes, radioactive isotopes, and small beads. In practical situations, chemical engineers often use residence time or standard deviation [176].

The Shannon entropy is a measure of disorder, dependent on the probability of finding a specified phase in a certain area. Camesasca et al. [29] applied it to the mixing of two fluid phases in order to quantify the mixing of polymers over time. In atmospheric science, Riemer and West [187] developed a mixing state index for distribution of a chemical species in an aerosol. The measure is defined to be the ratio of the average diversity of species and the bulk diversity [187], so gives a global value of mixing for the whole fluid domain.

The variance of a concentration, which quantifies how close the mixture is to homogeneous, is regularly used as a measure of mixing in pharmaceutical and food processing industries, as well as civil engineering [228]. For a concentration field $\Phi(\mathbf{r}, t)$, the variance is calculate as [228]

$$\mathrm{Var}\Phi = ||\Phi||^2 - \langle \Phi \rangle^2 , \tag{2.13}$$

where $|| \cdot ||$ denotes a norm and $\langle \cdot \rangle$ denotes the mean. An L^2 norm is often used, but other norms, such as the mix-norm [140] and negative Sobolev norms, may be applied [228]. The decay of the variance, as the norm of the concentration becomes closer to the mean, can be investigated through statistical analysis of FTLE, as well as analysing the advection-diffusion equation describing the flow. Again, no information is given about where separate phases remain or how mixing is taking place.

Mixing measures typically used in oscillatory baffled reactor research were adapted by Jimeno et al. [107] to apply to SPH simulations. Firstly, a velocity ratio is calculated as the sum of axial velocities divided by the sum of radial velocities of all SPH particles [107]. Additionally, a stretch rate is obtained by assigning each SPH particle an infinitesimal line and corresponding initial orientation [107]. A stretch rate can then be computed for each particle, depending on orientation angle and velocity, and a global stretch rate found by taking a volume average and plotting against time [107]. However, neither of these measures account for the presence of multiple phases.

A new measure is introduced by Jimeno et al. [107] which quantifies the phase of neighbouring particles for two immiscible fluids simulated using SPH. For a particle i, the neighbouring mixing index is

$$NM_i(t) = \sum_{j=1}^{N} 2\frac{m_j}{\rho_j} \left| J_i - J_j \right| \tilde{W}_{ij}, \tag{2.14}$$

where J_i is the phase number (0 or 1) of particle i and the kernel has a Shepard filter applied to correct for particles near boundaries [107]. A global value (NM) is calculated by averaging over all fluid particles and can be plotted with time [107]. The domain is fully mixed when $NM = 1$ [107]. This measure is limited to use with two phases and particle-based methods, therefore it cannot be used to directly compare SPH with mesh-based methods.

A plot of volume fraction (VF) gives the locations of phases, and can be used to see where regions of single phases remain. Without any knowledge of the flow though, it may be unclear why these separated regions remain and if they will persist no matter how long the fluid is mixed.

Lyapunov exponents can be used to locate manifolds in the flow [19], which show regions where mixing is promoted or inhibited (manifolds act as separatrices of the flow). From these, the quality of mixing can be inferred. The largest Lyapunov exponent is also used to characterise chaotic flow characteristics and mixing performance [84]. It is formally defined in the limit as time tends to infinity, which must be approximated when applied to numerical data.

Robinson [188] used finite-time Lyapunov exponent (FTLE) to identify manifolds in SPH mixing simulations, as well as developing a measure to quantify local mixing. Recently, FTLE were used by Dauch et al. [47] to find Lagrangian coherent structures in an SPH model of a fuel spray nozzle. By using the backward-in-time FTLE to find unstable manifolds, they were able to identify flow structures. However, FTLE does not show the locations of different phases.

A number of mixing measures, both local and global, which will be used to compare cases both numerically and visually are presented in Section 3.6. Additionally, a new mixing measure, which gives further insight into the mixing of two phases by combining phase location and manifolds of the flow, is introduced. These measures are demonstrated and assessed for the validation cases in Chapter 4, before being applied to the final application in Chapter 5.

2.7 Concluding Remarks

Vitrification is a complex procedure, especially of High Level Waste (HLW) which contains a great number of components and involves both chemical and physical reactions. Care will need to be taken when deciding which processes are important and which we can assume negligible for the particular application we are modelling. Accurately determining the values of the relevant dimensionless numbers in Section 2.2.2 will be key to these assumptions. From Section 2.2 it is clear that the mixing process is driven by convection due to heating. Other processes, such as chemical reactions, will be neglected for this work. Multiple phases will be characterised by their viscosity, so the methods in Section 2.5.3 will be key to capturing their interactions. Heating, and the subsequent effect of temperature on viscosity and density, is also fundamental to the model. It is anticipated from Section 2.2.2 that surface tension will not be need to be included in the model to capture the physics, though this will be reviewed throughout the validation cases in Chapter 4. By its nature, HLW has high levels of radioactivity. As discussed in Section 2.2.2, the main impact on the waste is heating. Therefore, this may need to be taken into account when designing the application case in Chapter 5.

CFD models described in Section 2.2.1 cover a number of aspects of vitrification of HLW but do not focus on the influence of mixing on the product homogeneity. Although flow within a commercial glass melter has been modelled using CFD a number of times, the different shape of the melter and composition of the glass mean that these techniques cannot be directly applied.

SPH is an appropriate choice for modelling the mixing of multiple phases, since it is ideal for multi-component flows that change properties and flow behaviour with time. The particle-based nature of this method also allows for natural tracking of an interface which is subject to large deformations. Of the mesh-free methods, SPH is well-established and has been demonstrated to be accurate in capturing fluid dynamics. Currently, a number of multi-phase models exist for a variety of applications, with relevant ones highlighted in Section 2.5.2. The basic SPH model, as well as the additional features included for this work, is described in more detail in Chapter 3.

The advantages of SPH outlined in Section 2.5.1 highlight the reasons why National Nuclear Laboratory (NNL) is interested in using the method to model mixing during vitrification of Highly Active Liquor (HAL). It is imperative that the chosen method is computational rather than experimental since, due to the highly toxic and radioactive nature of the waste, no measurements can be taken of the inside of the tank during the process. However, a model is needed to help to understand what is causing the vitrified waste to be visibly separated. This needs to be prevented since current predictive models for storage assume the waste is homogeneous and it is impossible to predict how the inhomogeneity affects the waste properties without knowing the compositions of the separated phases.

The open-source solver DualSPHysics will provide a basis for the code, ensuring time is not wasted reproducing work that has already been done but spent adding new multi-phase models. The current DualSPHysics code uses WCSPH, which is appropriate for modelling buoyancy-driven flow. Although it has been found to produce less accurate pressure fields than ISPH, it still creates a good representation of the global behaviour of an interface or free surface, which is key to this project.

A different multi-phase formulation than the one currently available in DualSPHysics is required to simulate this application. It must be applicable to phases with similar densities, and include boundary phases in order to specify different boundary conditions. Motion is driven by convective forces, through variations in viscosity and density due to changes in temperature. Current SPH formulations in DualSPHysics do not include thermodynamics in their standard governing equations, so this needs to be added in order to capture the relationship between temperature, viscosity and fluid dynamics.

The next chapter presents the SPH methodology which will be used to model mixing during the vitrification of HLW, as well as the post-processing measures to determine the level of mixing achieved within the melter.

Chapter 3

Numerical Methodology

3.1 Introduction

This chapter presents the Smoothed Particle Hydrodynamics (SPH) model used throughout the simulations presented in later chapters of this thesis. The SPH formulation is derived, and additional equations and methods are described. Furthermore, the method for modelling multiple phases is outlined, as well as the relationships for variable viscosity. Finally, measures of mixing are defined and advantages of each discussed.

3.2 Governing Equations

Viscous fluid flow is governed by the Navier-Stokes equations, derived from three fundamental principles of conservation: continuity equation from conservation of mass; conservation of momentum equation from Newton's second law; conservation of energy equation from the first law of thermodynamics. They can be written in Lagrangian form as [8]:

Continuity equation: $$\frac{D\rho}{Dt} + \rho\nabla\cdot\mathbf{v} = 0; \quad (3.1)$$

Momentum equation: $$\rho\frac{D\mathbf{v}}{Dt} = -\nabla p + \nabla\cdot\tau + \rho\mathbf{f}; \quad (3.2)$$

Energy equation: $$\rho\frac{DU}{Dt} = \rho\dot{q} + \nabla\cdot(k\nabla\theta) - \nabla\cdot(\mathbf{v}p) + \nabla\cdot(\mathbf{v}\tau) + \rho\mathbf{f}\cdot\mathbf{v}. \quad (3.3)$$

where $\frac{D}{Dt}$ is the material derivative, ρ density, $\mathbf{v}$ velocity, p pressure, τ is the shear stress tensor, $\mathbf{f}$ the body force acting on the fluid, U the internal energy, θ the temperature,

k the thermal conductivity and $\dot{q}$ the rate of heat transfer.

3.2.1 Conservation of Mass

The fluid is assumed to be weakly compressible (where compressibility variations are limited to within 1%), and so an equation of state is used to calculate pressure from density and complete the set of governing equations. This is described in Section 3.2.5. Since there is no assumption of incompressibility, the continuity equation (Equation (3.1)) cannot be simplified further.

$$\frac{D\rho}{Dt} = -\rho\nabla \cdot \mathbf{v}. \tag{3.4}$$

3.2.2 Conservation of Momentum

For a Newtonian fluid, the shear stress is assumed to depend linearly on the velocity gradients. Since the fluid is being modelled as weakly compressible, it is further assumed that any compression is insignificant and Stokes hypothesis that there is no viscous dissipation due to compression is adopted (discussion of when this may not be valid is in Section 3.3.6). Shear stress is related to viscosity via

$$\tau = \mu(\nabla\mathbf{v} + \nabla\mathbf{v}^T), \tag{3.5}$$

where μ is dynamic viscosity. Since kinematic viscosity $\nu = \frac{\mu}{\rho}$, the momentum equation is used in the form

$$\frac{D\mathbf{v}}{Dt} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{v} + \mathbf{f}. \tag{3.6}$$

For cases where there are no viscous or body forces acting on the fluid

$$\frac{D\mathbf{v}}{Dt} = -\frac{1}{\rho}\nabla p. \tag{3.7}$$

In a number of the cases considered in this work, the only body force acting on the fluid will be acceleration due to gravity.

3.2.3 Boussinesq Approximation

For any fluid, a change in temperature leads to a change in density. Therefore, when modelling a non-isothermal case, such as in Section 4.4 or the application in Chapter 5, it is important to take this into consideration.

The Boussinesq approximation states that, assuming variations in other properties with temperature are small, changes in fluid density may be neglected in the governing equations, except in the body force term of Equation (3.35). The density difference then drives buoyant convection, whilst avoiding extra computation which has negligible effect on the result when these assumptions hold. For natural convection, as in this work, the body force is due to gravity only.

The Boussinesq approximation has been determined to be accurate for limited temperature ranges [79]. The low thermal expansion coefficient of borosilicate glass means that, although large temperature differences are involved in vitrification, the resulting density differences within the melt remain small. Therefore the effect of heating on density outside the Boussinesq term will remain small. Although viscosity is a function of temperature, viscous contributions to heating are assumed negligible in this model. Gray and Giorgini [79] also determined a characteristic length of less than $\mathcal{O}(10^3)$ m (for water) is required for viscous dissipation to be insignificant compared to thermal diffusion. The length scales of High Level Waste (HLW) vitrification fall well below these limits.

In the body force term, density varies according to

$$\mathbf{f} = \rho(-\beta(\theta - \overline{\theta}))\mathbf{g}. \tag{3.8}$$

where β is the thermal expansion coefficient and θ is temperature. Therefore, the momentum equation when using the Boussinesq approximation is

$$\frac{D\mathbf{v}}{Dt} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{v} - \rho\beta(\theta - \overline{\theta})\mathbf{g}, \tag{3.9}$$

with the other governing equations unchanged.

3.2.4 Conservation of Energy

Beginning with Equation (3.3), the assumption of weak compressibility means that the pressure term and viscous heating terms are very small relative to heat flux and can be neglected. Therefore,

$$\rho\frac{DU}{Dt} = -\nabla \cdot \mathbf{q}. \tag{3.10}$$

Since heat flux is related to temperature θ through

$$\mathbf{q} = -k\nabla\theta, \tag{3.11}$$

with k the coefficient of thermal conductivity, Equation (3.10) can be rewritten as a heat diffusion equation:

$$\frac{DU}{Dt} = \frac{-1}{\rho}\nabla \cdot (-k\nabla\theta) = \frac{k}{\rho}\nabla^2\theta. \tag{3.12}$$

Temperature is directly proportional to internal energy according to $U = C_p\theta$, for specific heat C_p at constant pressure. Therefore, the energy equation used for this model is

$$\frac{D\theta}{Dt} = \frac{k}{C_p\rho}\nabla^2\theta. \tag{3.13}$$

3.2.5 Equation of State

As mentioned in the literature review, assuming that an incompressible fluid can be approximated by a weakly compressible fluid avoids solving a pressure Poisson equation (PPE), which is computationally expensive. With weakly compressible SPH, the system of governing equations is closed by an equation of state for pressure. Compressibility, which is on the order of the Mach number squared, is kept below 1% by constructing an artificial equation of state, such that the Mach number is approximately 0.1 [148]:

$$Ma = \frac{|\mathbf{v}|}{c} \approx 0.1, \tag{3.14}$$

for speed of sound c.

For this model, the equation of state commonly known as Tait's equation is used [76]:

$$p = B\left[\left(\frac{\rho}{\overline{\rho}}\right)^{\gamma} - 1\right]. \tag{3.15}$$

Here B is a constant, $\gamma \in [1, 7]$ (chosen to be 7 unless otherwise specified) is a constant known as the polytropic index of the fluid and $\overline{\rho}$ reference density (chosen to be initial density) [150].

Tait's equation is a rearrangement of the equation of state derived by Murnaghan [162] for solids at high pressure:

$$\frac{\overline{V} - V}{V} = 1 - (1 + \overline{K}p)^{-\frac{1}{C\overline{K}}}, \tag{3.16}$$

where $\overline{V}$ is the reference volume, $\overline{K}$ the corresponding bulk modulus and C a constant.

The overbar $\bar{\cdot}$ denotes a reference value. Rearranging to find pressure,

$$p = \frac{1}{\overline{K}}\left(\left(\frac{V}{\overline{V}}\right)^{C\overline{K}} - 1\right). \tag{3.17}$$

Since mass is constant by definition for weakly compressible SPH and so $\frac{V}{\overline{V}} = \frac{\rho}{\overline{\rho}}$, this is in the same form as Equation (3.15) when $B = \frac{1}{\overline{K}}, \gamma = C\overline{K}$.

The timestep required for SPH calculations is impractically small if a realistic speed of sound is used. To meet the weakly compressible condition Equation (3.14), an artificial speed of sound is set to be

$$c = C_s \max\{|\mathbf{v}|\} \tag{3.18}$$

for a constant coefficient C_s, usually chosen to be around 10. The speed of sound can be calculated from

$$c^2(\rho) = \frac{\partial p}{\partial \rho} = \frac{B\gamma}{\overline{\rho}}\left(\frac{\rho}{\overline{\rho}}\right)^{\gamma-1}, \tag{3.19}$$

and therefore, B is found to be

$$B = \frac{\overline{c}^2\overline{\rho}}{\gamma}, \tag{3.20}$$

where $\overline{c}$ is the speed of sound for $\overline{\rho}$ [157].

3.3 SPH Formulation

In the SPH method, particles (which may be irregularly spaced) are used to represent a medium. Each particle, which is a computational node, has physical properties and interacts with nearby particles, thus changing its properties [76]. To facilitate this, a particle has a domain of influence with radius proportional to the smoothing length of the kernel used.

3.3.1 Derivation

Following Monaghan, the integral interpolant is denoted by $\langle\cdot\rangle$ and for a general function $\Phi(\mathbf{r})$ at position $\mathbf{r}$ is [148]:

$$\langle\Phi(\mathbf{r}_i)\rangle = \int_\Omega \Phi(\mathbf{r})W(\mathbf{r}-\mathbf{r}', h)d\mathbf{r}' \tag{3.21}$$

over the whole space Ω. W is an interpolating kernel with radius αh, where α is a coefficient and h is a characteristic length known as the smoothing length, which satisfies five conditions [76]:

1. $\int\limits_{\Omega} W(\mathbf{r}-\mathbf{r}', h)d\mathbf{r}' = 1$ – partition of unity;

2. $\lim\limits_{h\to 0} W(\mathbf{r}-\mathbf{r}', h) = \delta(\mathbf{r}-\mathbf{r}')$ – the kernel is equal to the delta function in the infinitesimal limit;

3. $W(\mathbf{r}-\mathbf{r}', h) \in C^k$ – the kernel is k times differentiable with a continuous derivative;

4. $W(\mathbf{r}-\mathbf{r}', h) > 0$ over the domain Ω and $W(\mathbf{r}-\mathbf{r}', h) = 0$ outside Ω – the kernel has a compact support and is positive within it;

5. $W(\frac{|\mathbf{r}-\mathbf{r}'|}{h}, h)$ decreases monotonically with distance $\frac{|\mathbf{r}-\mathbf{r}'|}{h}$.

Lind and Stansby [126] have shown that by relaxing conditions 4 and 5 higher-order convergence can be obtained, but this is not considered in this thesis. SPH uses a numerical approximation of the integral interpolant [148]. For an arbitrary particle i, sum over all particles j in the domain,

$$\langle \Phi(\mathbf{r}_i) \rangle = \sum_j \frac{m_j}{\rho_j} \Phi_j W(\mathbf{r}_i - \mathbf{r}_j, h). \tag{3.22}$$

where subscripts i and j denote SPH particle numbers, m_j is the mass of particle j and ρ_j its density. The mass of a particle is fixed but its density can vary, therefore the volume also changes through the relationship $V_j = m_j/\rho_j$ [76]. The kernel ensures that particles j have more influence on particle i the closer they are to it. For a scalar function Φ, this has derivative [148]

$$\langle \nabla\Phi(\mathbf{r}_i) \rangle = -\sum_j \frac{m_j}{\rho_j} \Phi_j \nabla_j W(\mathbf{r}_i - \mathbf{r}_j, h), \tag{3.23}$$

where ∇_j represents differentiation with respect to particle j, and for a vector function $\mathbf{\Phi}$,

$$\langle \nabla\cdot\mathbf{\Phi}(\mathbf{r}_i) \rangle = -\sum_j \frac{m_j}{\rho_j} \mathbf{\Phi}_j \cdot \nabla_j W(\mathbf{r}_i - \mathbf{r}_j, h). \tag{3.24}$$

For irregularly spaced particles, Equations (3.23) and (3.24) do not guarantee zeroth-order consistency, hence, when constructing differentials of interpolants, higher accuracy is achieved by using $\nabla(\rho\Phi) - \Phi\nabla\rho$ in place of $\rho\nabla\Phi$ [148], similarly $\rho\nabla\cdot\mathbf{\Phi} = \nabla\cdot(\rho\mathbf{\Phi} - \mathbf{\Phi}\cdot\nabla\rho)$.

The condensed notation $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ is introduced and the distance $r_{ij} = |\mathbf{r}_{ij}|$ defined. Additionally, the notation is simplified in a standard way such that $W_{ij} =$

$W(\mathbf{r}_i - \mathbf{r}_j, h)$ and $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$. Since W_{ij} is a symmetric function, the partial derivatives of the kernel $\frac{\partial W_{ij}}{\partial \mathbf{r}_j} = -\frac{\partial W_{ij}}{\partial \mathbf{r}_i}$. In more compact notation,

$$\nabla_j W_{ij} = -\nabla_i W_{ij}. \tag{3.25}$$

This substitution is commonly used when deriving SPH equations.

For a general function Φ, the following expressions with the gradient operator are also useful:

$$\langle \nabla \Phi \rangle_i = \sum_j (\Phi_j - \Phi_i) \nabla_i W_{ij} \frac{m_j}{\rho_j}; \tag{3.26}$$

$$\rho_i \langle \nabla \cdot \mathbf{\Phi} \rangle_i = \sum_j m_j (\mathbf{\Phi}_j - \mathbf{\Phi}_i) \cdot \nabla_i W_{ij}. \tag{3.27}$$

An error of $\mathcal{O}(h^2)$ or smaller is introduced by truncating the Taylor series (smoothing error) [148], as well as a discretisation error of $\mathcal{O}\left(\frac{dx}{h}\right)$ [183], where dx is the distance between particles at $t = 0$, from numerical approximation of the integral as a sum. In terms of SPH accuracy, Quinlan et al. [183] discovered that the truncation error of the SPH method is composed of two parts, each of which dominates under different circumstances. The smoothing error, introduced by the kernel chosen for the integral interpolation, is the limiting error as $\frac{dx}{h} \to 0$. Whereas when $h \to 0$, the discretisation error caused by approximating the integral as a sum over neighbouring particles dominates. These errors lead to numerical instabilities in the process. Thus, the size of both h and $\frac{dx}{h}$ must be considered carefully when investigating instabilities using SPH. Particle disorder also contributes to error and techniques such as shifting (Section 3.3.5) are used to restore a more uniform distribution of particles [128]. Particle disorder can also lead to poor accuracy of the second derivative, especially when different material parameters are involved [39].

3.3.2 SPH Governing Equations

The governing equations in Section 3.2 are rewritten in SPH form using the derivations in Section 3.3.1. Henceforth the angular brackets will be dropped for ease of reading.

3.3.2.1 Conservation of Mass

Beginning with the definition of conservation of mass for a particle i

$$\frac{Dm_i}{Dt} = 0, \tag{3.28}$$

which means the mass of each particle is constant. Using Equation (3.22) with $\Phi = \rho$ produces the simple SPH estimate for density

$$\rho_i = \sum_j \frac{m_j}{\rho_j} \rho_j W_{ij} = \sum_j m_j W_{ij}. \tag{3.29}$$

However, this equation is not used for simulations with an interface, such as a free surface, since it can cause drops in density at the edge of a fluid due to the kernel support not being full, thus altering the pressure and resulting in oscillations of the edge [148].

An improvement is to use the rate of change of density, as in Equation (3.4),

$$\frac{D\rho}{Dt} = -\rho \nabla \cdot \mathbf{v} \tag{3.30}$$

along with Equations (3.23) and (3.25) to work out

$$\begin{aligned}\frac{D\rho_i}{Dt} &= -\rho_i \nabla_i \cdot \mathbf{v}_i \\ &= -\sum_j \frac{m_j}{\rho_j} \rho_j \mathbf{v}_{ij} \cdot \nabla_j W_{ij} \\ &= \sum_j m_j \mathbf{v}_{ij} \cdot \nabla_i W_{ij}.\end{aligned}$$

Therefore, the conservation of mass can also be written in SPH form as

$$\frac{D\rho_i}{Dt} = \rho_i \sum_j \frac{m_j}{\rho_j} \mathbf{v}_{ij} \cdot \nabla_i W_{ij}. \tag{3.31}$$

This equation avoids the drops in density at boundaries and also can be calculated in the same loop as other parameters, making it more computationally efficient [148]. This form is also Galilean invariant [38]. However, it does require an initial density to be set [148].

3.3.2.2 Conservation of Momentum

Beginning from Equation (3.7) (viscous forces will be considered in Section 3.3.6 and body forces at the end of this section), and applying Equation (3.26) gives

$$\frac{D\mathbf{v}_i}{Dt} = -\sum_j \frac{m_j}{\rho_j} (p_j - p_i) \nabla_i W_{ij}. \tag{3.32}$$

This form avoids tensile instabilities [26] which lead to particle clumping, but is not conservative. Therefore, a symmetric formulation is preferred by using Equation (3.23) to find

$$\frac{D\mathbf{v}_i}{Dt} = \frac{1}{\rho_i}\sum_j m_j \frac{p_j}{\rho_j}\nabla_i W_{ij}. \tag{3.33}$$

The identity [130]

$$\sum_j m_j \frac{p_i}{\rho_i \rho_j}\nabla_i W_{ij} = \frac{p_i}{\rho_i}\left(\sum_j \frac{m_j}{\rho_j}\nabla_i W_i j\right) = 0 \tag{3.34}$$

follows from the kernel definition and is added to give the momentum equation used:

$$\frac{D\mathbf{v}_i}{Dt} = -\sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2}\right)\nabla_i W_{ij}. \tag{3.35}$$

However, this form of the pressure gradient has been found to be unstable, particularly for multiphase flows with small density ratios $\left(\frac{\rho_1}{\rho_2} \leq 0.1\right)$ [40]. A variationally consistent pressure gradient formulation can help to retain stability for interfacial flows with small density ratios. That is, the pressure gradient is derived from variational principles and leads to conservation of linear and angular momentum. Bonet and Lok [26] determined the conservation of momentum equation

$$\frac{D\mathbf{v}_i}{Dt} = -\sum_j m_j \left(\frac{p_j + p_i}{\rho_j \rho_i}\right)\nabla_i W_{ij} \tag{3.36}$$

to be variationally consistent with Equation (3.31), whereas Equation (3.35) is variationally consistent with Equation (3.29).

When a body force is present, for example gravity, then the momentum equation has an additional term:

$$\frac{D\mathbf{v}_i}{Dt} = -\sum_j m_j \frac{p_j + p_i}{\rho_j \rho_i}\nabla_i W_{ij} + \mathbf{f}_i. \tag{3.37}$$

With the Boussinesq approximation of Section 3.2.3, the body force term is included, driven by the temperature gradient

$$\frac{D\mathbf{v}_i}{Dt} = -\sum_j m_j \frac{p_j + p_i}{\rho_j \rho_i}\nabla_i W_{ij} - \rho\beta(\theta - \overline{\theta})\mathbf{g}. \tag{3.38}$$

According to the approximation, the other governing equations remain unchanged.

3.3.2.3 Conservation of Energy

Starting from the temperature diffusion Equation (3.13), Monaghan [149] determined that this can be written in SPH form for a particle i with neighbouring particles j as

$$C_{p,i}\frac{D\theta_i}{Dt} = \sum_j \frac{m_j}{\rho_i\rho_j}\{(k_i + k_j)(\theta_i - \theta_j)\}\nabla_i W_{ij}, \tag{3.39}$$

where the form of the term inside curly brackets is chosen to approximate the Laplacian.

To ensure that heat flux remains continuous for discontinuous k, the sum of thermal conductivity coefficients is replaced by twice the harmonic mean. Then the temperature evolution equation in SPH form, as implemented in this model, is:

$$\frac{D\theta_i}{Dt} = \frac{1}{C_{p,i}}\sum_j \frac{m_j}{\rho_i\rho_j}\frac{4k_ik_j}{k_i + k_j}(\theta_i - \theta_j)\frac{\nabla_i W_{ij}}{r_{ij}}. \tag{3.40}$$

3.3.3 Kernels

As discussed in Section 3.3.1, the kernel is chosen so that in the limit as the smoothing length h tends to zero it approximates the Dirac delta function. For larger h, the delta function is smoothed over the surrounding area, such that closer neighbouring particles have greater influence. The kernel function must also be differentiable so that the interpolants of gradients of functions can be computed [148].

An obvious candidate is the Gaussian function, shown here for two dimensions:

$$W(\mathbf{r}, h) = \frac{1}{\pi h^2}e^{-\left(\frac{r_{ij}}{h}\right)^2}. \tag{3.41}$$

However, since the Gaussian has infinite support, it is computationally expensive and not often employed in practice. Instead, commonly used kernels approximate the Gaussian but with compact support, as shown in Figure 3.1. Examples include the quadratic, cubic spline and Wendland kernels.

Cutting off the kernel at a finite radius excludes contributions from particles further away, introducing a truncation error. Since the kernel value reduces with distance, the error can be reduced by increasing smoothing length h. However, this also increases computational cost. Accuracy increases with the order of the kernel and, in general, computation time also increases [148]. Although the quadratic kernel is lower order than the others, the first derivative of the quadratic kernel has no point of inflexion, therefore the force between particles increases as the distance between particles decreases and it does not exhibit tensile instability [76]. A gradient kernel correction is sometimes employed to ensure correct evaluation of the gradient of the velocity field [26].

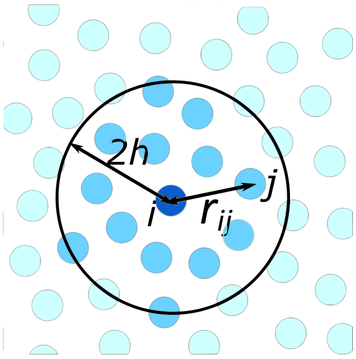


Figure 3.1: Kernel around particle i of radius $2h$, with interacting particles within the compact support coloured a darker shade.

The Wendland kernel [245] is chosen for this work due to its high accuracy [50]. It is a fifth-order polynomial function with compact support. In two-dimensions, the kernel with radius $2h$ is

$$W(\mathbf{r}_{ij},h)=\begin{cases}\frac{7}{4\pi h^2}\left(1-\frac{r_{ij}}{2h}\right)^4\left(\frac{2r_{ij}}{h}+1\right) & r_{ij}\in[0,2h],\\ 0 & \text{otherwise.}\end{cases} \tag{3.42}$$

with derivative

$$\frac{\partial W(\mathbf{r}_{ij},h)}{\partial r_{ij}}=\begin{cases}-\frac{35}{4\pi h^2}q\frac{r_{ij}}{2h}\left(1-\frac{r_{ij}}{2h}\right)^3 & \text{if } r_{ij}\in[0,2h],\\ 0 & \text{otherwise.}\end{cases} \tag{3.43}$$

This kernel is shown for smoothing length $h=1$ in Figure 3.2.

In three dimensions, the Wendland kernel is given by

$$W(\mathbf{r}_{ij},h)=\begin{cases}\frac{21}{16\pi h^3}\left(1-\frac{r_{ij}}{2h}\right)^4\left(\frac{2r_{ij}}{h}+1\right) & r_{ij}\in[0,2h],\\ 0 & \text{otherwise.}\end{cases} \tag{3.44}$$

3.3.4 Boundary Conditions

The derivation of the SPH equations of Section 3.3.2 does not naturally include boundary conditions. If a fluid particle is located near a boundary, its kernel support can

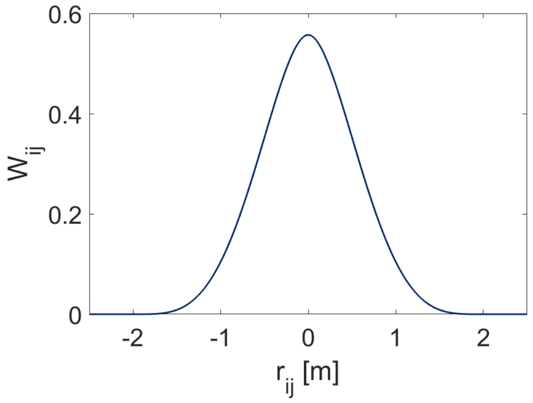


Figure 3.2: Wendland kernel with smoothing length $h = 1$, showing the weighting on particle contributions at different distances from particle i at $\mathbf{r} = 0$.

extend beyond the fluid domain into the boundary where there are no particles and hence the kernel support is incomplete. There are two commonly used types of boundary conditions with partial differential equations, such as the governing equations here. Dirichlet boundary conditions specify a value for a variable at the boundary, for example a stationary boundary with no slip has $\mathbf{v} = 0$. For a Neumann boundary condition, the normal derivative of a variable at the boundary is given. An example is an adiabatic boundary for temperature, where there is no heat flux through the boundary, which is equivalent to $\frac{\partial \theta}{\partial \mathbf{n}} = 0$ at the boundary.

If a single layer of particles are placed at the boundary then fluid particles near the boundary will still have an incomplete kernel support, leading to a truncation error in any SPH summation at these particles. As discussed in Section 2.5.5, there are a number of ways of resolving this problem, including numerical corrections. Here, in order to maintain a full kernel support for all fluid particles, multiple layers of boundary particles are placed (using the same intial particle spacing dx as the fluid), with the number depending on the size of the kernel radius. There are a number of approaches to prescribe variable values to the boundary particles, also discussed in Section 2.5.5. One of the current Grand Challenges for SPH, chosen by the SPH rEsearch and engineeRing International Community (SPHERIC), is the development of accurate and robust boundary conditions which maintain the advantages of SPH [234]. The options in DualSPHysics are the dynamic boundary condition (DBC) and the recently introduced modified dynamic boundary condition (mDBC), which are described below.

3.3.4.1 Dynamic Boundary Condition (DBC)

Dummy particles have position and velocity prescribed and fluid particles are allowed to interact with them, as with other fluid particles [43]. For a stationary boundary, as time is increased by dt:

$$\mathbf{r}(t+dt) = \mathbf{r}(t); \qquad \mathbf{v}(t+dt) = \mathbf{v}(t) = 0. \tag{3.45}$$

All other variables are calculated by solving the governing equations, as for the fluid particles. This is not entirely physically accurate and is associated with issues such as particles penetrating through and sticking to the wall, if the density of the boundary particles becomes lower than that of the fluid. This can be overcome by resetting boundary particle density at each timestep.

The DBC is quick to implement and computationally efficient to run. For many validation cases in Chapter 4 it gives sufficiently accurate results. Where this is not the case, it is specified that a more sophisticated boundary condition is required.

3.3.4.2 Modified Dynamic Boundary Condition (mDBC)

Recently, English at al. [58] developed a modification to the DualSPHysics DBC based on the work of Marrone et al. [139], which reduces penetration of boundaries as well as allowing for partial slip conditions. It is found to reduce fluctuations on boundary density, as well as giving more physical results. It can also reduce the gap that often develops between fluids and a boundary when using DBC.

Layers of boundary particles are used, as with DBC, but the physical boundary is midway between fluid and boundary particles. Boundary particles are mirrored into the fluid as ghost particles along the boundary normal into the fluid. Density and its gradient are found at a boundary particle i for its corresponding ghost particle i_g using SPH interpolation over surrounding fluid particles, along with the first order consistent correction of Liu and Liu [131]. When there are too few particles within the kernel of the ghost particle, the correction is not possible and a Shepard filter, as in Equation (3.75), is used instead. Density at a boundary particle i is then calculated through

$$\rho_{i_g} = \rho_i + (\mathbf{r}_i - \mathbf{r}_{i_g}) \cdot \begin{bmatrix} \partial_x \rho_i \\ \partial_y \rho_i \\ \partial_z \rho_i \end{bmatrix} . \tag{3.46}$$

Boundary particles are given zero velocity, unless a moving boundary is specified, as for the DBC method.

When using mDBC, the origin of the Cartesian grid which is used to place particles at initiation must be adjusted so that particles do not fall on the line of the boundary. The mDBC boundary interface should be mid-way between boundary and fluid particles. Since mDBC relies on boundary normals, it takes extra consideration to apply to more complex geometries. The most relevant for this work are corners, where boundary particles are reflected onto ghost particle along a line through the vertex.

For a number of validation cases in Chapter 4, mDBC are tested and, where significant improvements are found, are used.

3.3.4.3 Viscosity Boundary Conditions

For the SPH formulation used in this work, each particle must have its own viscosity, rather than there being a global viscosity for the whole domain. Therefore, boundary particles must also have a viscosity for when fluid particles near the boundaries interact with them through the viscous operator Equation (3.70).

With the newly introduced viscosity phases, boundaries assigned as a phase can have any of the same viscosity options as a fluid phase. Namely, they can be set at a constant viscosity, or can follow a chosen viscosity model as a function of temperature with specified parameters. In this way, a Dirichlet boundary condition for viscosity can be set, or the fluid viscosity can be extrapolated into the boundary. Otherwise, if no phase viscosity options are chosen, the boundary will have the constant default viscosity for the case.

There is potential to extend the mDBC method to mirror viscosity into the boundary, as temperature is in the following Section 3.3.4.4.

3.3.4.4 Temperature Boundary Conditions

Boundary phases can follow the temperature equation 3.40, as with fluid phases. Alternatively, a simple Dirichlet condition may be set by choosing

$$\theta = \overline{\theta} \tag{3.47}$$

constant (as default).

It is more complex to prescribe a Neumann boundary condition (see Section 2.5.5), i.e. the normal temperature gradient, in particular when there is heating present. An adiabatic boundary (no heat transfer) can be approximated by setting the thermal conductivity to

$$k = 0 \tag{3.48}$$

for the boundary phase, so that particles in this phase do not contribute to the temperature evolution equation (eq. (3.40)). This is equivalent to the naturally adiabatic condition of no boundary particles for solids described by Cleary [38], giving a zeroth order approximation, so is expected to have similar limitations. However, the existing boundary particles are able to contribute to SPH sums for other variables, so no additional truncation error is introduced here.

The capability of an isothermal condition (internal energy U constant) can be checked by computing the heat flux over boundary particles along the inward normal $\mathbf{n}$ [38]:

$$\mathbf{q}_i = \frac{1}{\rho_i}\sum_j 2m_j k_j \theta_{ij} \mathbf{r}_{ij} \cdot \mathbf{n}_i \frac{\mathbf{r}_{ij} \cdot \nabla_i W_{ij}}{\mathbf{r}_{ij}^2 + \eta^2}. \tag{3.49}$$

Clearly, this is affected by the same errors as other SPH interpolations, but will be equal to zero for the adiabatic condition $k = 0$. Cleary [38] states that heat flux is always conserved with the SPH temperature equation.

For a homogeneous Neumann condition (such as an adiabatic boundary for temperature) the symmetric boundary condition can be used. A more accurate approach than Equation (3.48) can be derived by extending the first-order consistent mDBC method, as described in Section 3.3.4.2 for density. The gradient term is not included, as no extrapolation is required here. Following the mDBC method for computing normals and ghost particles, the temperature at a boundary particle i is the temperature interpolated at its corresponding ghost particle i_g. As with density, the correction of

Liu and Liu [131] is applied to θ_{i_g}, then

$$\theta_i = \theta_{i_g}. \tag{3.50}$$

This is the same as Equation (3.46) without the gradient term. In this way, temperature is reflected across the line of the adiabatic boundary.

3.3.5 Timestepping Methods

Values at time $t + dt$ are calculated from those at time t using the governing equations in Section 3.3.2. Since the calculations depend on particle interactions, the values at time t must not be discarded until all particles have been progressed by dt. The conservation equations are autonomous, but the variables such as $\mathbf{v}$ and ρ depend on time. A variety of timestepping schemes have been used for SPH. Here we shall consider the two different schemes included in DualSPHysics – the Verlet method and the symplectic predictor-corrector method.

Verlet [237] introduced the following second order time integration scheme for molecular dynamics:

$$\mathbf{v}_i(t + dt) = \mathbf{v}(t - dt) + 2dt\frac{d\mathbf{v}_i(t)}{dt}; \tag{3.51}$$

$$\mathbf{r}_i(t + dt) = \mathbf{r}_i(t) + dt\mathbf{v}_i(t) + \frac{dt^2}{2}\frac{d\mathbf{v}_i(t)}{dt}; \tag{3.52}$$

$$\rho_i(t + dt) = \rho_i(t - dt) + 2dt\frac{\rho_i(t)}{dt}. \tag{3.53}$$

Although there is only one step per iteration, an additional step must be performed approximately every 40 iterations to reduce divergence due to decoupling of velocity and density.

$$\mathbf{v}_i(t + dt) = \mathbf{v}(t) + dt\frac{d\mathbf{v}_i(t)}{dt}; \tag{3.54}$$

$$\mathbf{r}_i(t + dt) = \mathbf{r}_i(t) + dt\mathbf{v}_i(t) + \frac{dt^2}{2}\frac{d\mathbf{v}_i(t)}{dt}; \tag{3.55}$$

$$\rho_i(t + dt) = \rho_i(t) + dt\frac{\rho_i(t)}{dt}. \tag{3.56}$$

This additional step is first order only in velocity and density, and the number of iterations at which it is required can vary between cases. Thus, an empirical parameter which needs tuning is introduced.

The symplectic predictor-corrector method for position, defined generally by Leimkuhler and Matthews [123] and applied to density by Parshikov et al. [175], is also of second order. Following the formulation in the paper of Gomez-Gesteira et al. [78] on the SPHysics code, the governing equations in Section 3.3.2 can all be written in the form:

$$\frac{D\mathbf{y}}{Dt} = \mathbf{x}(t). \tag{3.57}$$

The first step is to predict the values at half the timestep length, then this is corrected using the newly predicted values. Finally, the corrected result is used along with the value at t to find the new value at time $t + dt$ [78]:

$$\text{Predict:} \qquad \mathbf{y}\left(t + \frac{dt}{2}\right) = \mathbf{y}(t) + \frac{dt}{2}\mathbf{x}(t); \tag{3.58}$$

$$\text{Correct:} \qquad \mathbf{y}\left(t + \frac{dt}{2}\right) = \mathbf{y}(t) + \frac{dt}{2}\mathbf{x}\left(t + \frac{dt}{2}\right); \tag{3.59}$$

$$\text{Recalculate:} \qquad \mathbf{y}(t + dt) = 2\mathbf{y}\left(t + \frac{dt}{2}\right) - \mathbf{y}(t). \tag{3.60}$$

This scheme involves a half step but does not require any additional adjustment steps.

It is important to use at least a second order timestepping method to retain the second order accuracy of the SPH formulation. Higher order time integration methods, such as the fourth order Runge-Kutta scheme, are significantly more computationally expensive. Therefore, any potential increase in timestep size is negated by the longer time taken to complete each step. The half-step included in the predictor-corrector method, however, results in the scheme being stable for larger timesteps [237], with a smaller computational cost overall.

After each step for either scheme, pressure p must be calculated from ρ using the equation of state (Equation (3.15)) so that it is at the same stage as the other variables. Viscosity is updated similarly, since it is a function of only temperature (Section 3.5.1), if not constant.

3.3.5.1 Timestep Size

A sufficiently small timestep is required for an accurate solution, but also increases computation time. A variable timestep is employed which reduces runtime to 40% and helps to keep the simulation stable [78]. Based on a Courant-Friedrichs-Lewy (CFL) condition, it depends on the physical properties of the simulation at that instant in time, multiplied by a chosen constant Courant number C_0 (usually $\mathcal{O}(10^{-1})$).

DualSPHysics already includes two constraints on the timestep size. The first depends on the acceleration or body force:

$$dt_1 = \sqrt{\frac{h}{|\mathbf{a}|_{\max}}}, \tag{3.61}$$

where $\mathbf{a}_{\max}$ is the maximum acceleration of all particles. Maintaining weak compressibility, as in Equation (3.14), gives a second timestep constraint from the speed of sound:

$$dt_2 = \frac{h}{\max\{c, 10|\mathbf{v}|_{\max}\} + h\overline{\nu}}, \tag{3.62}$$

where $\overline{\nu}$ is the default viscosity for the case.

The third timestep constraint was introduced into DualSPHysics by Fourtakas and Rogers [68] to account for viscous dissipation in SPH:

$$dt_3 = \frac{h^2}{\nu_{\max}\lambda}, \tag{3.63}$$

where λ is a relaxation parameter set for each case. For this work, with each SPH particle having its own viscosity, $\nu_{\max}$ is taken to be the maximum viscosity of all particles as interaction forces are computed in each timestep.

Finally, a fourth timestep constraint must account for thermal dissipation when temperature evolution (Equation (3.40)) is included in the model. Cleary [38] introduced the Courant condition due to heat transfer for SPH:

$$dt_\theta = \frac{0.1\rho C_V h^2}{k}, \tag{3.64}$$

where the coefficient 0.1 corresponds to a Courant-type number. This work uses specific heat at constant pressure, rather than constant volume. Although $C_p > C_V$, the difference is small relative to their magnitudes so will not have a significant effect on the timestep size. A thermal time constraint parameter

$$\Theta_i = \frac{k_i}{\rho_i (C_p)_i} \tag{3.65}$$

is defined. Since k, C_p can vary between phases and density varies with time and between particles, $\Theta_{\max} = \max_i \Theta_i$ is computed at each timestep as with $\nu_{\max}$ and the final timestep constraint is

$$dt_4 = 0.1\frac{h^2}{\Theta_{\max}}. \tag{3.66}$$

The resultant timestep for each time integration iteration is

$$dt = C_0 \min\{dt_1, dt_2, dt_3, dt_4\}, \tag{3.67}$$

where C_0 is the chosen Courant number. A lower limit to dt is typically set as

$$dt_{\min} = 0.05\frac{h}{c}, \tag{3.68}$$

or may be chosen explicitly, in order to avoid the timestep becoming prohibitively small.

3.3.6 Viscosity

The momentum conservation equation derived in Section 3.3.2.2 neglects the viscous terms. This may be justified for certain flows with high Reynolds number but, since molten glass is a viscous fluid, it is important that viscosity is included in the SPH formulation for this model. For low Reynolds number, small variations in pressure (relative to the hydrostatic pressure gradient) cause movement within the fluid and viscous forces dominate the flow [161]. A commonly used method for astrophysics and free-surface flows is an artificial viscosity [154], where its diffusive properties aid stability. Although an equivalent viscosity can be approximated, it has no physical derivation. Another popular approach is the Morris viscosity operator which approximates the Laplacian term in the Navier Stokes momentum equation. For cases where viscosity is not constant in time, it may be necessary to consider other approaches.

To account for variations in viscosity between phases, and within phases with varying temperature, each SPH particle i has its own viscosity ν_i. This is achieved by introducing a new array (`Visc`) into DualSPHysics, rather than assigning a single viscosity to the case. Viscosities are allocated by phase when setting up the case and can remain constant or follow an equation dependent on another variable such as position or temperature, as in Section 3.5.1.

3.3.6.1 Morris Viscosity Operator

The Morris viscosity operator was introduced in 1997 by Morris et al. [161] for low *Re* flow, in particular non-Newtonian mixing. It approximates the Laplacian viscosity term in the Navier Stokes conservation of momentum Equation (3.7), with the assumption that viscosity is constant as in Figure 3.3. The term is used in the form introduced by Lo and Shao [133]:

$$\nu \nabla^2 \mathbf{v}_i = \sum_j m_j \frac{4\overline{\nu}}{\rho_i + \rho_j} \left(\frac{\mathbf{r}_{ij} \cdot \nabla_i W_{ij}}{\mathbf{r}_{ij}^2 + \eta^2} \right) \mathbf{v}_{ij}, \tag{3.69}$$

where η is a small constant included to avoid singularities as $\mathbf{r}_{ij} \to 0$. This form of the viscous term conserves linear but not angular momentum [101].

To include phases of different viscosities in the model, the viscous formulation is changed by replacing $\overline{\nu}$ with the arithmetic mean of both viscosities involved in the interaction, such that

$$\nu \nabla^2 \mathbf{v}_i = \sum_j m_j \frac{2(\nu_i + \nu_j)}{\rho_i + \rho_j} \left(\frac{\mathbf{r}_{ij} \cdot \nabla_i W_{ij}}{\mathbf{r}_{ij}^2 + \eta^2} \right) \mathbf{v}_{ij}. \tag{3.70}$$

Other means could be chosen, but this simple option is found to perform well in the two-phase Poiseuille flow case in Section 4.2.2.2, where viscosity difference at the interface is key to initiation of the instability. Equation (3.70) can deal with particles of different viscosities interacting, as in Figure 3.4, and reduces to the original Equation (3.69) when interacting particles have the same viscosity.

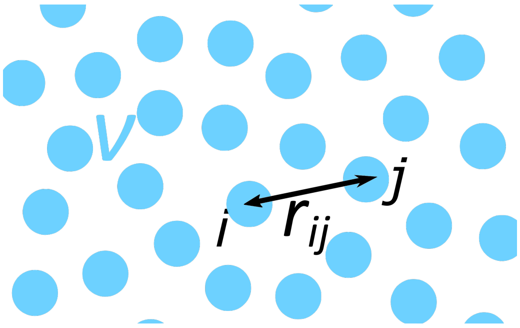


Figure 3.3: Interaction between particles of constant viscosity.

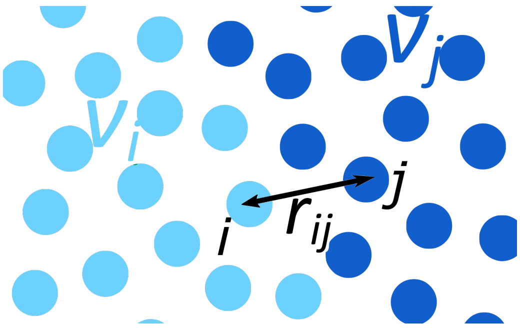


Figure 3.4: Interaction between particles of different viscosities.

3.3.6.2 Alternative Operators for Viscosity

The viscous term in Equation (3.7) is equivalent to the stress tensor term in Equation (3.2) for compressible Newtonian fluids, assuming that viscosity is constant and independent of other variables. However, for this work viscosity will not always be constant for the whole domain and may also change with temperature. If the Stokes hypothesis is dropped and the bulk viscosity term reinstated then

$$\nabla \cdot \tau = \nu \nabla^2 \mathbf{v} + \left(\nu + \frac{\lambda}{\rho} \right) \nabla (\nabla \cdot \mathbf{v}). \tag{3.71}$$

It should be determined whether the second term is significant in this case, or if Equation (3.70) will suffice.

A viscous term including bulk viscosity ($\frac{2}{3}\mu + \lambda$) was introduced by Avalos et al. [15] for compressible flows (when the Mach number is not small, as in Equation (3.14)). Bulk viscosity is also important to include when there are phenomena dependent on compressibility, even if the fluid is approximately incompressible [193]. The new term, including a bulk viscosity is [15]

$$\nabla \cdot \tau = \sum_j \frac{m_j}{\rho_i \rho_j} \left[4\nu \left(\frac{\mathbf{r}_{ij} \cdot \nabla_i W_{ij}}{\mathbf{r}_{ij}^2 + \eta^2} \right) \mathbf{v}_{ij} - \lambda^{\text{SPH}} \left(\frac{\dot{\rho}_i}{\rho_i} + \frac{\dot{\rho}_j}{\rho_j} \right) \frac{\nabla_i W_{ij}}{r_{ij}} \right] \tag{3.72}$$

where, assuming a Newtonian fluid, $\lambda^{\text{SPH}} = \lambda - \mu$.

As discussed in Section 2.5.3 of the literature review, Bonet Avalos et al. [27] determined that the viscous term of Monaghan and Gingold [155] can be derived from the assumptions of bulk viscosity. The Monaghan and Gingold operator is [155]

$$\nu \nabla^2 \mathbf{v}_i = \sum_j m_j \frac{4\overline{\nu}}{\rho_i + \rho_j} \left(\frac{\mathbf{r}_{ij} \cdot \mathbf{v}_{ij}}{\mathbf{r}_{ij}^2 + \eta^2} \right) \nabla_i W_{ij}, \tag{3.73}$$

where $\lambda = \mu$ is fixed. Further, Bonet Avalos et al. [27] developed a viscous term with λ independent of μ, which will not be used in this work.

For this work, the constant viscosity is replaced with the average viscosity of the two interacting particles, as in Equation (3.70), so that

$$\nu \nabla^2 \mathbf{v}_i = \sum_j m_j \frac{2(\nu_i + \nu_j)}{\rho_i + \rho_j} \left(\frac{\mathbf{r}_{ij} \cdot \mathbf{v}_{ij}}{\mathbf{r}_{ij}^2 + \eta^2} \right) \nabla_i W_{ij}. \tag{3.74}$$

The advantages of using the Monaghan and Gingold operator are considered for the differentially heated cavity case in Section 4.4.2.

3.3.7 Density Treatment

The Lagrangian nature of SPH means that particles move and, with the weakly compressible formulation, densities may vary. Through the equation of state (Equation (3.15)), pressure changes with density. Choosing $\gamma = 7$ means that pressure is sensitive to particle distribution and significant fluctuations can occur, even for simple flows such as still water. A number of techniques have been proposed to mitigate this problem through filtering or introducing density diffusion.

3.3.7.1 Shepard Filter

The Shepard filter is a zeroth order filter over density. Every 20 to 50 timesteps the density is adjusted using [76]

$$\rho_i^{\text{new}} = \frac{\sum_j \rho_j W_{ij} V_j}{\sum_j W_{ij} V_j}. \tag{3.75}$$

3.3.7.2 Density Diffusion

Molteni and Colagrossi [147] introduced a diffusive term to the density evolution Equation (3.31) which smooths the density, and therefore pressure, fields. The additional term is

$$\delta h c \sum_j 2 \frac{m_j}{\rho_j} (\rho_j - \rho_i) \frac{\mathbf{r}_{ij}}{r_{ij}^2} \cdot \nabla_i W_{ij}, \tag{3.76}$$

where the constant parameter is typically chosen to be $\delta = 0.1$.

Antuono et al. [12] also introduced a correction to Equation (3.76) to account for truncated kernels near boundaries. However, it requires a large additional computational cost and is designed for free-surface flows, which are not considered in this work. A density diffusion term was also used by Ferrari et al. [64] instead of adding an artificial viscosity term to the momentum equation.

Fourtakas et al. [70] improved this term for pressure near wall boundaries by introducing the hydrostatic density difference

$$\rho_{ij}^H = \rho_0 \left(\left(\frac{\gamma(\rho_0 g (r_z)_{ij} + 1)}{c^2 \rho_0} \right)^{\frac{1}{\gamma}} - 1 \right). \tag{3.77}$$

The adjusted density diffusion term is

$$\delta h c \sum_j 2 \frac{m_j}{\rho_j} (\rho_{ij} - \rho_{ij}^H) \frac{\mathbf{r}_{ij}}{r_{ij}^2} \cdot \nabla_i W_{ij}. \tag{3.78}$$

The density diffusion term of Molteni and Colagrossi [147] is used for the lid-driven cavity case in Section 4.3, as well as single and two-phase Poiseuille flow in Section 4.2, which were simulated in DualSPHysics v4.4. The temperature-dependent Poiseuille flow in Section 4.2.3 and differentially heated cavity case in Section 4.4, as well as the vitrification cases in Chapter 5, are run with the Fourtakas et al. [70] density diffusion term included in DualSPHysics v5.0. Values of δ are specified for each case, or where changed for a simulation.

3.3.8 Particle Shifting

As mentioned in Section 3.3.7, the weakly compressible nature of this method can lead to irregular particle distributions. Particle clumping can also occur, due to the kernels used (Section 3.3.3) having small gradient as particles approach each other closely, which introduces error to the SPH interpolations and can lead to numerical instability. Nestor et al. [164] introduced particle shifting in the Finite Volume Particle Method (FVPM) to maintain a more uniform particle distribution throughout the simulation. This was closely followed by Xu et al. [253] for incompressible SPH (ISPH), and then Shadloo et al. [202] for weakly compressible SPH (WCSPH). Lind et al. [127] then extended this idea to free surfaces, introducing the Fickian-based approach.

The method used by Lind et al. [127] is based on Fick's law of diffusion. Particles are moved a small distance, depending on the particle concentration gradient and shifting coefficient D,

$$\delta \mathbf{r}_i = -Ddt \sum_j \frac{m_j}{\rho_j} \nabla_i W_{ij}. \tag{3.79}$$

Skillen et al. [210] modified the shifting term to account for variations in particle velocity,

$$\delta \mathbf{r}_i = -Dh|\mathbf{v}_i|dt \sum_j \frac{m_j}{\rho_j} \nabla_i W_{ij}. \tag{3.80}$$

This form of shifting is used for this model, with D chosen for each case.

The approach of Skillen et al. [210] in Equation (3.80) is used for shifting in the majority of this work to maintain a more regular particle distribution and mitigate against related numerical errors. The method of Lind et al. [127] was employed for the lid-driven cavity case in Section 4.3.2.1, where particles were found to penetrate the boundary. Removing the velocity factor from the shifting term reduced the number of particles entering the boundary where velocity magnitudes are typically smaller. Particle loss is not a problem with the released version of mDBC and Equation (3.79) is not used for any other cases.

For multi-phase flows, Xenakis et al. [251] used a combination of Equation (3.79) for low D and Equation (3.80) for higher D and found both methods to be stable. However, care should be taken to ensure shifting is not causing unphysical mixing between phases, in particular when there is a large difference in density. For example, in the violent free-surface flows investigated by Mokos et al. [146].

For two-phase cases of the lid-driven cavity and Poiseuille flow, Equation (3.80) does not introduce additional mixing at the interface. Hence, the modifications of Mokos et al. [146] for air and water phases are not required for this work, where phase densities are identical or very close.

3.3.9 Neighbouring Particle Searches

Neighbour lists, where records are kept of neighbouring particles, improve efficiency of the SPH method. SPH summations need only consider particles within the kernel support of the particle concerned, and can neglect those who do not contribute due to their kernel being equal to zero. A neighbour list keeps a record of particles which are close to each other in space, enabling the method to disregard others. It also reorders all variables relating to the particles so that those close to each other (and therefore interacting with each other) are also close in the memory, which improves access to the CPU memory [77]. The link list method described in this section was found to speed up the lid driven cavity case in a previous self-written code by approximately 10 times. Therefore, this is highly effective tool for improving the efficiency of the SPH method. First, the domain must be split into cells. These are chosen to be squares of side length $2h$, the radius of the kernel support. The particles are then assigned the number of the cell they lie in and a gridding algorithm is used to sort them so that particles in nearby cells are stored next to each other. In DualSPHysics, the linked list method of Domínguez et al. [54] is used, where pointers are used to link from the cell to the first particle in it and then between all particles in the cell in a chain. This is then used to reorder all particle variables so they are sorted by the cell number the particle lies in.

3.3.9.1 Calculate number of cells

Firstly, all particles are looped over to find the maximum and minimum position in each dimension, and therefore the extents of the domain. A margin of $2h$ is added around the outside of the geometry, unless otherwise specified in the case file. Any particles which exceed these limits are excluded. The number of cells in each dimension is found by dividing by cell width and rounding up. The total number of cells is then the product of these two numbers and memory can be allocated to store cell information.

3.3.9.2 Split domain into cells

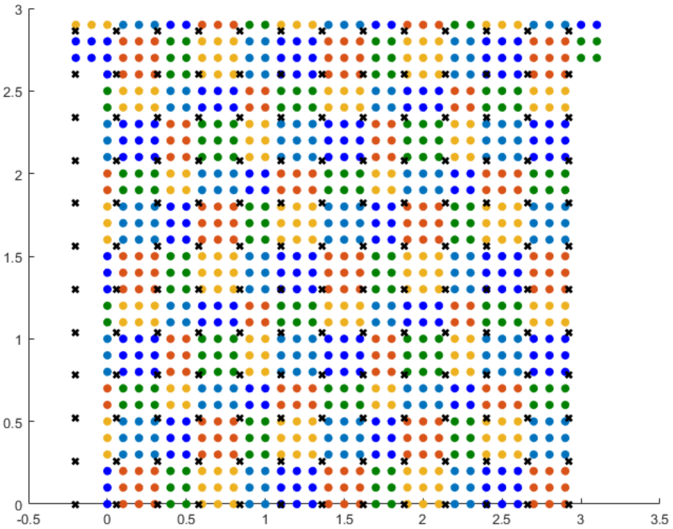
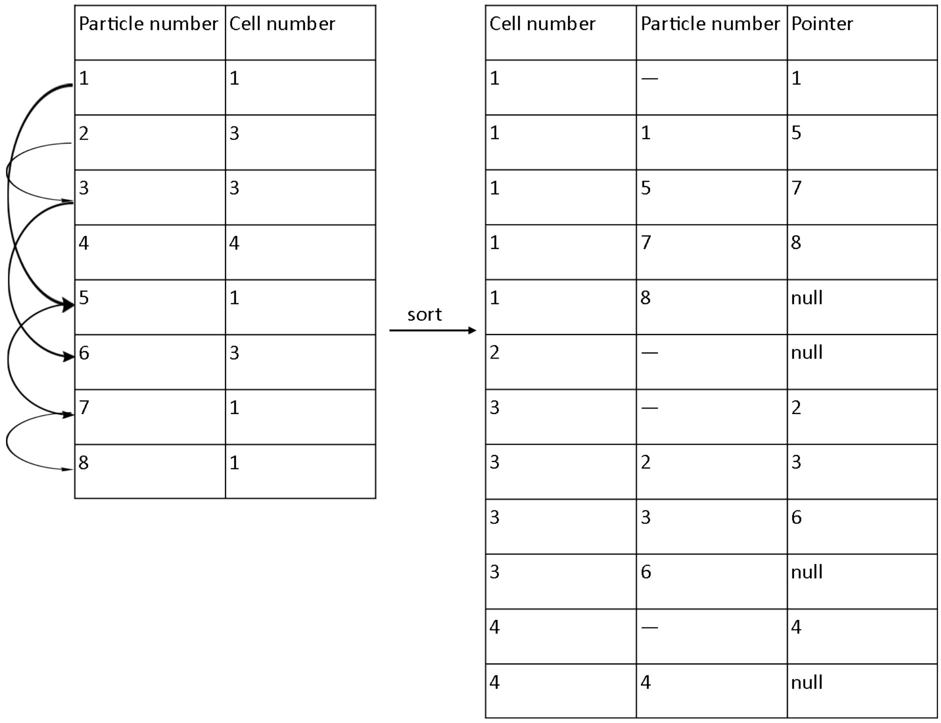


Figure 3.5: Example lid driven cavity domain (as in Section 4.3) split into cells for neighbour list method, with the bottom left corner of each cell marked by a black cross.

The domain is next split into the calculated number of cells, which are number according to a sorting method. For example, SortX sorts first in the x direction then the y direction and finally the z direction, so the cells are ordered along rows moving upward. An example of a domain split into cells of side-length $2h$ is shown in Figure 3.5, where each cell is defined by the location of the black cross in its bottom left corner and the particles in adjacent cells are coloured differently. Calculating the number of cells and splitting the domain need only be performed only once at the start of the simulation.

3.3.9.3 Place particles in cells and set up pointers

Looping over all particles, the cell each particle lies within is determined by its position and the corresponding cell number stored. Excluded particles lying outside the domain are given a value not corresponding to any cell numbers in use. Each cell and particle has a pointer which must be reset at the start of each timestep to avoid errors. To keep track of the linked list of pointers, the index of the current last particle in the chain for each cell is temporarily stored.



Particle number	Cell number
1	1
2	3
3	3
4	4
5	1
6	3
7	1
8	1

Cell number	Particle number	Pointer
1	—	1
1	1	5
1	5	7
1	7	8
1	8	null
2	—	null
3	—	2
3	2	3
3	3	6
3	6	null
4	—	4
4	4	null

Figure 3.6: Figure to show set up of pointers and sorting of vectors during link list process.

In the same loop as the cell number is found for a particle, the last particle index is checked against for this cell. If it is not equal to a particle number then this is the first particle in the cell and the cell's pointer is set to point to this particle. Otherwise, the last particle in the cell is j and the pointer of particle j is set to point to the current particle. In this way, a chain of 0 or more pointers is set up to link the particles in each cell. The chain of pointers for a small example set of particles can be seen on the left hand side of Figure 3.6. The final particle in a cell retains its initial null value, as does an empty cell.

3.3.9.4 Reorder particles

The linked list of pointers for each cell can now be used to reorder the particles so that data for nearby particles are also stored close in the memory. One cell at a time, in order of cell number, all variables for particles (including particle id) are put into temporary vectors by following the chain of pointers. When a null pointer is reached, the next cell is considered. Once complete, the particle variables are updated. The resulting sorted vector for the example particles can be seen in the middle column of the right-hand table in Figure 3.6.

The functions to give particles cell numbers and then sort them must be performed every timestep, as particles move throughout the domain and their neighbours change.

3.3.9.5 Flag particles in neighbouring cells

Before all functions involving multiplication by the kernel, which therefore only have contributions from particles within the kernel support, particles in neighbouring cells are flagged. The numbers of neighbouring cells can be determined, depending on the sorting algorithm used. Particles in these cells are flagged to be considered in kernel sums, whilst all other particles can be skipped as they will not contribute.

3.4 DualSPHysics Code Structure and Modifications

DualSPHysics [55] is an open-source SPH code, which provides a robust and well-validated solver for research and industrial applications. Originally developed to simulate free-surface flows for coastal engineering applications, it has since been extended to model a number of other physical processes. These include fluid-structure interactions and multi-phase flows, all of which can be run in two or three dimensions. The code undergoes continual development and has regular releases with new features and updates.

DualSPHysics is written to run on either a multi-core central processing unit (CPU) or graphics processing unit (GPU), and is optimised for both [55]. The initialisation and output are written only in C++, whereas the SPH formulation within the loop is implemented in both C++ and Compute Unified Device Architecture (CUDA) to run on either CPU or GPU respectively, as chosen. The CPU components of the code use Open Multi-Processing (OpenMP) to reduce computation time by parallelising on multi-core and multiple CPUs. Since the C++ code is more accessible, it allows quicker testing of additions and modifications to the code, before moving to CUDA to take advantage of the reduction in runtime provided by the GPU.

A WCSPH formulation, as presented in this chapter, is employed by DualSPHysics. Mass is conserved using Equation (3.31), and momentum via Equation (3.36).

Viscous term options include an artificial viscosity or the Shao and Lo [205] operator in Equation (3.69), but in both cases a constant global viscosity is used. Currently, DualSPHysics is isothermal with no governing equation for temperature. SPH particles positions are updated depending on their velocity and the system of equations is closed with the equation of state in Equation (3.15) for pressure. The Verlet and predictor-corrector timestepping methods presented in Section 3.3.5 are available, with the predictor-corrector method used for this work, and the Wendland kernel in Equation (3.42) is chosen from the kernel options.

Recent developments include the density diffusion term in Equation (3.78) [70] and the mDBC [59] described in Section 3.3.4.2, both of which will be used in this work. Two multi-phase options are included, namely an air-water model [145] and a Newtonian-non Newtonian model [68]. The code structure of the latter is modified for the types of phases concerned in this work.

DualSPHysics uses object-oriented programming, where data is stored as a structure of arrays (SoA), to structure the code and ensure efficient performance. An array is defined for each variable (`Pos` – three-dimensional position, `Velrhop` – velocity and density), with an entry for every particle. Pressure is not stored, but computed at each timestep from density through the equation of state, Equation (3.15). Additionally, the array `idp` stores particle numbers, which is key to the neighbour list procedure described later in this section. Additionally, some flags exist to determine whether or not certain functions are run. For example, when using mDBC `UseNormals` is set to `true` so that normals are computed for all boundary particles, to be used when extrapolating variables into the boundary.

The configuration is read in by GenCase, which produces output files that drive DualSPHysics, from an xml file where parameters, geometry and case options are specified. The geometry of fluids and boundaries is created by drawing `mk` blocks of different shapes. Each `mk` block is numbered to give it, and any particles that lie within it, an identifier. All is read into the main code, before particles are placed within the geometry and given the corresponding initial values.

The neighbour list is formed using a linked list method, which Dominguez et al. [54] found to be most efficient for SPH. The domain is divided into cells of size $2h$, whose number and location are stored. Before each timestep, a loop over all particles is performed to determine in which cell each particle is located. Then pointers are updated, such that each cell points to the first particle inside that cell and each particle points to the next particle in the cell, until the final particle in the cell is left with a null pointer. In this way, a chain of pointers is set up for each cell. When reaching the end of the chain, the algorithm can move onto the next cell. Then particles and all corresponding variables are sorted according to the linked list, so that data for neighbouring particles is stored close in the memory. The neighbour list procedure does take time to set up, but this is significantly outweighed by the resulting savings in computation time for interactions [55].

Particle interactions are then computed though solving the governing equations in Section 3.3.2. With the neighbour list previously set up, only contributions from particles in neighbouring cells are computed. This saves computation time and makes no difference to the results, providing the chosen kernel is compact with radius $2h$. Next, the timestep size is computed with Equation (3.67) and particle values are updated at the next timestep, using the timestepping procedure in Section 3.3.5 and the results of the interactions. The code then loops back to update the neighbour list.

Data is output to vtk files at specified time intervals until the maximum time is reached. Post-processing is performed using MeasureTool. The locations and variables required are specified, with values calculated through a SPH interpolation of the output data.

3.4.1 Addition of Viscosity and Temperature Variables

Currently, DualSPHysics admits a single viscosity for a simulation. Arrays `Visc` for viscosity and `Temp` for temperature are added so that each SPH particle may have its own viscosity and temperature. Options for computing these values are described in Sections 3.5.1 and 3.5.2 respectively. Although viscosity is not computed through a governing equation, both are included as global arrays, so they can be accessed and updated at any point during the simulation, as well as output to files.

3.4.2 Multiple Phases

A `ViscPhases` option flag has been introduced to set multiple phases with different viscosity and temperature properties, based on the code changes made by Fourtakas and Rogers [68] for the non-Newtonian multi-phase branch. If the option is not chosen, then the case will run with constant viscosity and temperature, as in the isothermal case with viscosity operator Equation (3.69).

When setting up a case, phase parameters and options are assigned within the special section of the xml case file, examples of which are in Appendix A. Any `mk` block created when setting up the geometry may be designated as a phase, and each phase is given a `phaseid` number to uniquely identify it. The parameters in Table 3.1 can be set, or left to the default value, and are stored in an array ordered by `phaseid`. For viscosity models and corresponding parameters see Section 3.5.1, and options for temperature are outlined in Section 3.5.2.

To allow boundary conditions to be set, any boundary `mk` block can have any of the same properties allocated. A count of fluid phases is kept so that boundary phases can be given a unique `phaseid` without any repetition. Initial viscosity and temperature are given to each SPH particle according to which phase they lie in during the initialisation of the case, as outlined in Section 3.4. For the lid-driven cavity case in Section 4.3, the moving lid boundary can be modelled either with moving particles and a periodic boundary condition, or by keeping boundary particles in the lid stationary whilst giving them a velocity.

If the `ViscPhases` option is selected then the code forks to use particle interaction functions including the modified viscous term in Equation (3.70) and temperature evolution Equation (3.40). For each SPH particle, the phase number is found through its `mk` and corresponding phase properties are looked up from the phase arrays and used in the interaction equations.

3.5 Closure Submodels

A number of closure submodels are introduced to compute values for the newly introduced viscosity and temperature variables. To include variable viscosity in the

parameter	definition	units	default
`phaseid`	phase number	-	-
`mk`	mkfluid (if fluid)	-	-
	mkbound + no. fluid phases (if bound)	-	-
`rho0`	initial density	kg m^{-3}	ρ_0
`visc0`	initial kinematic viscosity	m^2 s^{-1}	ν_0
`viscop1`	choice of viscosity model	-	constant
`viscop2`	choice of viscosity model	-	constant
`cst1`	viscosity model parameter	-	0
`cst2`	viscosity model parameter	-	0
`cst3`	viscosity model parameter	-	0
`cst4`	viscosity model parameter	-	0
`temp0`	initial temperature θ_0	K	0
`Tref`	reference temperature	K	θ_0
`tempop`	choice of temperature model	-	constant
`cp`	specific heat capacity C_p	J K^{-1}	0
`k`	thermal conductivity k	W m^{-1} K^{-1}	0
`beta`	thermal expansion coefficient β	K^{-1}	0
`Tu`	upper wall temperature	K	0
`Tl`	lower wall temperature	K	0

Table 3.1: Phase parameter options with `ViscPhases` option.

model, there are a number of equations to choose from. Viscosity depends only on temperature, so is computed after the timestepping process is applied to the governing equations. Alternatively, a constant viscosity can be specified for each phase. Temperature can be determined by the newly added Equation (3.40), or through an empirical equation.

3.5.1 Viscosity Models

Previous work for National Nuclear Laboratory (NNL) by Miller [144] developed an algorithm for the melt viscosities of glasses involved in vitrification of nuclear waste. Experimental input data is passed through a Levenberg-Marquardt curve-fitting algorithm to compute parameters for the selected viscosity equation. There are 10 equations for viscosity as a function of time to choose from (with capital letters the parameters to be computed):

- Cornelissen

$$\ln \mu = A + \frac{B}{\theta^x}; \tag{3.81}$$

- Arrhenius
$$\ln \mu = A + \frac{B}{R\theta}, \tag{3.82}$$
with gas constant $R := 8.31446 \mathrm{JK}^{-1}\mathrm{mol}^{-1}$;

- Jenckel3
$$\ln \mu = A + \frac{B}{\theta} + \frac{C}{\theta} e^{\frac{B}{\theta}}; \tag{3.83}$$

- Jenckel4
$$\ln \mu = A + \frac{B}{\theta} + \frac{C}{\theta} e^{\frac{D}{\theta}}; \tag{3.84}$$

- OjovanLee4
$$\ln \mu = A \ln(\theta) + \frac{d_{\mathrm{Hm}}}{R\theta} + \ln\left(1 + e^{-\left(\frac{d_{\mathrm{Sb}}}{R}\right)} e^{\left(\frac{d_{\mathrm{Hh}}}{R\theta}\right)}\right), \tag{3.85}$$
with d_x verified from experiments;

- Vogel Fulcher Tamman
$$\ln \mu = \mu_0 + \frac{B\overline{\theta}}{\theta - \overline{\theta}}; \tag{3.86}$$

- Avramov Milchev
$$\ln \mu = A + \ln(\mu_{\theta_g} - A)\left(\frac{\theta_g}{\theta}\right)^C, \tag{3.87}$$
with θ_g glass transition temperature, $\mu_{\theta_g} = 10^{12}$Pa s;

- Myega2
$$\ln \mu = u + \frac{K}{\theta} + e^{\frac{C}{\theta}} \tag{3.88}$$
(u independent of composition);

- Myega3
$$\ln \mu = \mu_\infty + \frac{K}{\theta} + e^{\frac{C}{\theta}}; \tag{3.89}$$

- Smedskjaer
$$\ln \mu = u + \frac{B}{\theta}(\ln(\mu_{\theta_g}) - u) e^{\frac{B}{\theta}\left(\frac{C}{\ln(\mu_{\theta_g}) - u} - 1\right)}. \tag{3.90}$$

According to Miller [144], the preferred models are Jenckel4, Avramov Milchev and Myega3. These models have a maximum of four constant parameters which may be specified for each phase in the xml case file, along with the choice of model as seen in Table 3.2, all of which have been added to the DualSPHysics code. Note that dynamic viscosity in these equations must be converted to kinematic viscosity to be used in the governing equations in Section 3.3.2.

Viscosity is computed from temperature during the particle update, after all variables have been progressed by the timestepping iteration, as with pressure through the equation of state (Equation (3.15)). Simple viscosity models for the cases in Section 4.2.3 are also included, in the forms used by Wall and Wilson [243]:

- exponential
$$\nu = \frac{1}{\rho} e^{-K_1 \theta}; \tag{3.91}$$
- linear
$$\nu = \frac{1}{\rho}(1 - K_2 \theta). \tag{3.92}$$

viscop1	viscop2	model	cst1	cst2	cst3	cst4
0	-	constant [default]	-	-	-	-
-	0	constant [default]	-	-	-	-
1	1	exponential	K_1	-	-	-
1	2	linear	K_2	-	-	-
2	1	Cornelisson	A	B	x	-
2	2	Arrhenius	A	B	-	-
2	3	Jenckel3	A	B	C	-
2	4	Jenckel4	A	B	C	D
2	5	OjovanLee4	A	d_{Hm}	d_{Sb}	d_{Hh}
2	6	VogelFulcherTamman	η_0	B	$\overline{\theta}$	-
2	7	AvramovMilchev	A	μ_{θ_g}	θ_g	C
2	8	Myega2	u	K	C	-
2	9	Myega3	μ_∞	K	C	-
2	10	Smedskjaer	u	B	η_{θ_g}	C

Table 3.2: Viscosity model options and parameters available in phase setup.

Miller's [144] algorithm is also capable of extrapolating the results into a greater range of temperatures than the experimental data, although this is usually unnecessary for vitrification test rig data. Additionally, parameters are fit for chemical composition and parameters for glasses can be decomposed into into sums of component amount and activity coefficient A_i according to Schott's activity principle, so that for a parameter A
$$A = \sum_{i=1}^{n} A_i y_i, \tag{3.93}$$
where y_i is the weight or mole fraction of component i.

The density of each phase also depends on composition. This can be calculated if the fraction of all components is known as a sum weighted by component fraction. Densities will not be so different as to cause the issues seen in air-water models (Section 2.5.2).

3.5.2 Temperature Options

Options for temperature can be assigned to any phase and are: constant, temperature evolution through Equation (3.40), linear temperature field, or adiabatic. If no option is specified, the phase is given constant temperature θ_0. When using the temperature governing equation, particle interactions are computed at the same time as the other governing equations, and particle temperatures updated similarly.

The linear field is used along with `Tu` and `Tl` from Table 3.1 (temperatures at upper and lower boundaries respectively) in Section 4.2.3 to form a linear temperature gradient across the channel. If a linear field is chosen, temperature is computed during the particle update section from the updated particle positions.

The adiabatic option is included to apply to boundary phases, in order to satisfy an adiabatic boundary condition. If any of the phases is chosen to be adiabatic then the flag `UseNormals` is set to `true`, so that boundary normals are computed and stored as for mDBC. Since normal calculation takes place only once at the beginning of the simulation, additional computation time is negligible. However, further modifications could make it possible to calculate normals only for the adiabatic boundary. Boundary particle temperatures are then calculated as described in Section 3.3.4.4.

3.6 Mixing Measures

In order to compare the degree of mixing between cases and methods both qualitatively and quantitatively, a number of mixing measures are introduced. They can give a numerical assessment of mixing in the whole domain or a value at a particular location, which can be used to give an overall value for the degree of mixing. Each of the measures has different characteristics, as discussed below, which may be advantageous for different cases.

Post-processing code is written in C++ to apply to the SPH output data from

DualSPHysics. This includes the kernel functions in Section 3.3.3, which are used in the calculation of volume fraction (VF). A neighbour list, using the link-list approach [54], is also employed to increase computational efficiency, as well as parallelisation using OpenMP. In a previous self-written SPH code this neighbour list was found to reduce computation time by around 10 times. Results are outputted for each local mixing measure below, as well as the global mixing measures at each input time, to be plotted to show the degree of mixing.

3.6.1 Volume Fraction

For comparison with volume-of-fluid (VOF) methods, it is natural to use a volume fraction (VF). This can be taken directly from the STAR-CCM+ simulation, but must be computed for the SPH model from output data. In the two-phase cases considered here, the fraction of one fluid gives the full picture since the fraction of the other fluid in any volume is one minus this.

An SPH interpolation (Equation (3.22)) is used to calculate the VF for SPH simulation data in post-processing. The volume fraction of phase A at point i is defined to be:

$$F_i^A = \frac{\sum\limits_{j \in \widetilde{J}} V_j^2 W_{ij}}{\sum\limits_{j \in J} V_j^2 W_{ij}}, \tag{3.94}$$

where the superscript A denotes phase A, J is the set of all particles j in a neighbourhood (kernel radius) of point i and $\widetilde{J} \subset J$ is the subset of these particles which are also of phase A. For comparison with STAR-CCM+ data, the points i are chosen to be at the centre of each mesh cell, where there is not necessarily an SPH particle at any given time.

A global comparison to quantify the difference in VF of the two modelling methods is given by the L^2 norm of volume fraction, calculated over all cells in the STAR-CCM+ mesh by

$$L^2(F^A) = \sqrt{\frac{\sum_i (\overline{F}_i^A - F_i^A)^2}{\sum_i (\overline{F}_i^A)^2}}, \tag{3.95}$$

where $\overline{F}_i^A$ is the volume fraction of phase A in STAR-CCM+ and F_i^A the volume fraction of phase A in DualSPHysics, calculated with Equation (3.94).

An understanding of global mixing cannot be gained from averaging volume fractions throughout the domain. Instead, the difference in local volume fraction of a phase A (F_i^A) to the global volume fraction of phase A (F^A) is calculated using

$$F_{\text{global}}^A = \frac{1}{N} \sum_{i=1}^{N} |F^A - F_i^A|, \tag{3.96}$$

summed over N measurement points. This number gives a value of how close, on average, the local concentration of a phase is to its global concentration.

3.6.2 Finite Time Lyapuov Exponent

For numerical simulations, it is impractical to use the Lyapunov exponent described in Section 2.6 in which time tends to infinity. The finite-time Lyapunov exponent (FTLE) at a time T is calculated using eigenvalues of the right Cauchy-Green deformation tensor at that instant. However, for a meshless Lagrangian method (such as SPH), this can be approximated as a ratio of distances between particle pairs over time [30]. Grid-based methods must use tracer particles to calculate FTLE in this way. Dauch et al. [48] developed a more computationally efficient method of calculating FTLEs on a GPU, consequently determining the FTLE field of three-dimensional SPH data without a prohibitively high computational cost.

This is implemented in the post-processing of SPH data. The maximum distance at the initial location of particle i is defined to be the maximum ratio of distances between particle i and particles j which begin within its neighbourhood at $t = 0$. Following the approach of Tóth and Szabó [230],

$$d_{\max}(\mathbf{r}_i(t=0), T) = \max_j \left\{ \frac{r_{ij}(t=T)}{r_{ij}(t=0)} \right\}. \tag{3.97}$$

Thus, the computationally expensive construction of the deformation tensor is avoided. The FTLE Λ at time T is then calculated by

$$\Lambda_i(T) = \frac{\ln(d_{\max}(\mathbf{r}_i(t=0), T))}{T}. \tag{3.98}$$

Each FTLE value is a measure of the exponential divergence of two closely spaced points. The ridges of maxima can be used to locate stable manifolds, in the sense of dynamical systems, across which pairs of particles move apart more rapidly than elsewhere. Similarly, unstable manifolds can be found by computing the FTLE for negative time. In this way, the magnitude of FTLE gives a rate of separation of trajectories [213], and so can show the flow structure at a point in time [201]. Where FTLE are constant with time, such as a periodic flow, these manifolds separate regions between which little mixing occurs. Particularly useful for this work, high relative movement between SPH particles identifies where mixing is promoted.

3.6.3 Robinson's Local Mixing Measure

While the FTLE field gives a measure of the divergence of particles at a point, Robinson [188] defines a local measure to quantify the mixing between different phases at position $\mathbf{r}_i$:

$$M_i = \frac{\mathbf{s}_i^l \cdot \mathbf{s}^g - M_{\min}}{1 - M_{\min}}. \tag{3.99}$$

N is the number of phases, n_i^A is the number of particles of phase A locally and n^A globally, with

$$\mathbf{s}_i^l = \frac{1}{\sqrt{\sum\limits_{j=1}^{N}\left(\frac{n_i^j}{n^j}\right)^2}}\left(\frac{n_i^A}{n^A}, \frac{n_i^B}{n^B}, ..., \frac{n_i^N}{n^N}\right), \tag{3.100}$$

$$\mathbf{s}^g = \frac{1}{\sqrt{N}}(1, 1, ..., 1), \tag{3.101}$$

$$M_{\min} = \frac{1}{\sqrt{N}}. \tag{3.102}$$

Robinson [188] designed this measure to compare the local ratio of phases to that of the whole domain, and it is able to admit any countable number of phases.

For the purposes of this work, the number of particles in a neighbourhood n_i^A is replaced by local volume fraction F_i^A (n^A for F^A respectively), so that the measure at a point i is

$$M_i = \frac{\mathbf{s}_i^l \cdot \mathbf{s}^g - M_{\min}}{1 - M_{\min}} \tag{3.103}$$

$$= \frac{\sqrt{N}+1}{N-1}\left(\frac{1}{\sqrt{\sum\limits_{j=1}^{N}\left(\frac{F_i^j}{F^j}\right)^2}}\sum_{k=1}^{N}\frac{F_i^k}{F^k} - 1\right), \tag{3.104}$$

where F_i^A is the local volume fraction of phase A at point $\mathbf{r}_i$ and F^A is the global volume fraction of phase A.

For cases with two phases, assume $N = 2$. Then Equations (3.100) to (3.102) can be rewritten as

$$\mathbf{s}_i^l = \frac{1}{\sqrt{\left(\frac{F_i^A}{F^A}\right)^2 + \left(\frac{F_i^B}{F^B}\right)^2}} \left(\frac{F_i^A}{F^A}, \frac{F_i^B}{F^B}\right), \tag{3.105}$$

$$\mathbf{s}_g = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right), \tag{3.106}$$

$$M_{\min} = \frac{1}{\sqrt{2}}. \tag{3.107}$$

Then the Robinson measure in this case is

$$\begin{aligned} M_i &= \frac{\frac{1}{\sqrt{2}}\left(\frac{\frac{F_i^A}{F^A}+\frac{F_i^B}{F^B}}{\sqrt{\left(\frac{F_i^A}{F^A}\right)^2+\left(\frac{F_i^B}{F^B}\right)^2}} - 1\right)}{1 - \frac{1}{\sqrt{2}}} \\ &= (\sqrt{2}+1)\left(\frac{\frac{F_i^A}{F^A} + \frac{F_i^B}{F^B}}{\sqrt{\left(\frac{F_i^A}{F^A}\right)^2 + \left(\frac{F_i^B}{F^B}\right)^2}} - 1\right). \end{aligned} \tag{3.108}$$

The further assumption of $F_i^A + F_i^B = F^A + F^B = 1$ (but in general $F^A \neq F^B$) allows Equation (3.108) to be rewritten in terms of only phase A, such that

$$M_i = (\sqrt{2}+1)\left(\frac{\frac{F_i^A}{F^A} + \frac{1-F_i^A}{1-F^A}}{\sqrt{\left(\frac{F_i^A}{F^A}\right)^2 + \left(\frac{1-F_i^A}{1-F^A}\right)^2}} - 1\right). \tag{3.109}$$

This approach is able to quantify mixing at a local level for any number of phases. Through the modification above to use volume fraction, it can be computed for both SPH and VOF methods for comparisons of results. It clearly shows regions where phases are mixed, and those where separated phases remain. However, no information about the mixing flow or particle movement can be inferred from the value.

3.6.4 Mixing entropy

Uniformity of phase distribution over the whole system can also be quantified by the global entropy measure of Kang et al. [110]. This sums information entropy over all cells, such that

$$S = -\sum_{i=1}^{N} n_i \log(n_i), \tag{3.110}$$

where n_i is particle number density. For comparisons between methods in this work, n_i is substituted for the volume fraction F_i of the chosen phase at the cell centre points defined in Section 3.6.1. The entropy sum is then normalised to be

$$S^* := \frac{S - \overline{S}}{S_{\max} - \overline{S}}, \tag{3.111}$$

where $S_{\max} = \log(N)$ is the maximum possible entropy and $\overline{S}$ is the entropy at the inlet (entropy at $t = 0$ here). This definition can be used to find a global mixing entropy value, S^*, for each phase.

The mixing entropy is calculated for the lid-driven cavity and vitrification cases. It shows whether the rate of mixing is increasing or where it tends to constant. However, it is computed for each phase separately, so does not give a complete picture of multiphase mixing.

3.6.5 New Measure of the Degree of Mixing Between Phases

A new measure is introduced as a combination of VF and FTLE with the aim of producing a measure which demonstrates both the degree of mixing at a point and how well-mixed the two phases are at that location. The measure was recently included in a publication [186], where it is demonstrated for two-phase flow in lid-driven cavity and Poiseuille flow cases. It is defined in Equation (3.114) for two phases, as used in this work, and will need modification for measuring a greater number of phases.

Firstly, both components are normalised at each point i, such that the value is 1 if most mixed and 0 if least. This is done according to

$$\overline{\Lambda}_i = \frac{\Lambda_i}{\max\limits_j \Lambda_j}, \tag{3.112}$$

$$\overline{F}_i^A = \begin{cases} \frac{F_i^A}{F^A} & \text{if } F_i^A \leq F^A \\ \frac{1-F_i^A}{1-F^A} & \text{if } F_i^A > F^A \end{cases}, \tag{3.113}$$

where the maximum FTLE is taken over all points in the domain at the time T. Then the new mixing measure is defined to be:

$$\overline{M} = \overline{\Lambda} \cdot \overline{F} \in [0, 1]. \tag{3.114}$$

$\overline{M} = 1$ if the local volume fraction is equal to the global volume fraction and the finite-time Lyapunov exponent is the maximum for T. The mixing measure $\overline{M} = 0$ if there is only one phase locally or if the distance to neighbouring particles is unchanged. $\overline{M}$ is calculated at each point where an SPH particle is located at $t = 0$, so that the FTLE makes sense.

By combining the VF and FTLE, this new measure is able to give information about the presence of local phases as well as relative movement of fluid volumes, thus giving a more informative picture of the mixing process. A global value of mixing is obtained by taking an average of $\overline{M}$ over all N fluid particles in the domain, for example

$$\overline{M}^G = \frac{1}{N}\sum_{j=1}^{N}\overline{M}_j \tag{3.115}$$

which also ranges from 0 to 1.

3.6.6 Mixing measures for periodic flows

For cases with periodic boundaries, such as in Section 4.2.2, computing mixing measures is more complex. For example, to the calculate volume fraction at a point near a periodic boundary SPH particles near the corresponding boundary must be included in the Equation (3.94) sum since they may also be neighbours. In addition, for the FTLE, the distance calculation in Equation (3.97) must take into account how many times both SPH particles in question have passed through the periodic boundary.

3.6.7 Mixing measures in three-dimensions

For the three-dimensional cases in Chapter 5, the mixing measures are computed on a $y = 0$ plane through the domain. The Freedraw option of the DualSPHysics pre-processing tool GenCase used means that particles do not necessarily lie on a Cartesian grid at initialisation, therefore particles initially lying within half a particle spacing of the plane ($r_y \in (-0.5dx, 0.5dx)$) are selected. This ensures the plane is fully populated whilst remaining only one particle thick.

Since the FTLE are defined at the point a particles begins its trajectory, these particle locations at $t = 0$ are chosen to be the locations for all local mixing measures. For VF, only particles within a kernel radius ($2h$) of the $y = 0$ plane are retained, as no others can contribute to the SPH summation. Since this corresponds to the size of the neighbour list cells, a two-dimensional link-list is still employed. For computing FTLE at any time, the particles chosen at $t = 0$ must be kept to compare their updated distances. These are sorted into the same order as the original neighbour list before calculations are performed.

3.7 Concluding Remarks

The governing equations for this model have been presented at the start of this chapter, with assumptions of the physics to be included and neglected. Next, the SPH numerical methodology is introduced, before being applied to generate the governing equations in SPH form. Other key aspects of the SPH formulation are described, including the interpolation kernel and timestepping methods used. Different boundary conditions are reviewed in terms of each of their advantages to this work. The dynamic boundary condition (DBC) is quicker and less computationally expensive, but the modified dynamic boundary condition (mDBC) produces more accurate densities in the boundary. Boundary conditions for viscosity and temperature are also considered, and a new method for enforcing an adiabatic boundary defined. Viscosity operators are discussed, taking into account particles with different viscosities and viscosities which may vary. Methods used to reduce numerical error through density treatments and particle shifting were presented, and the approaches used in this work identified.

Next, the computational method was presented, first by giving an overview of the current functionalities of DualSPHysics, before introducing the changes for this model. Variables for temperature and viscosity of each particle are added, along with the governing equation for temperature and the modified viscous operator previously discussed. The method for including multiple phases was explained, plus the addition of boundary phases to allow boundary conditions to also be prescribed for temperature and viscosity. The options for computing temperature in a phase have been outlined, and the viscosity models introduced. These correspond to the models used in the algorithm produced for vitrification glasses [144], so that realistic parameters can be computed using the Melt Viscosity Alogirthm (MVA) and used in this model.

Finally, a number of mixing measures have been introduced and discussed in the context of the model and application. A novel local measure has been proposed, which combines the volume fraction (VF) and finite-time Lyapunov exponent (FTLE), giving further insight into the mixing taking place. This new measure is extended to a global measure, which may be plotted against time to find the rate of change of mixing. It can also be used to determine whether a steady state has been reached, and therefore whether the mixture will ever become homogeneous.

The method introduced in this chapter is validated with a number of cases to test each of the different features in Chapter 4, before being applied to the motivational model of vitrification in Chapter 5.

Chapter 4

Validation and Results

4.1 Introduction

This chapter presents different cases which are chosen to validate each aspect of the model. In each section the case is described, before results are compared with data from literature and key findings are described. The aim of each test case is to ensure that each element of the Smoothed Particle Hydrodynamics (SPH) model can reproduce the expected behaviour of the application presented in Chapter 5. The first test cases present the validation of two phases of constant viscosity and temperature-dependent viscosity. This is followed by single and two-phase flow within a cavity, and validation of the mixing measures introduced in Section 3.6. Finally, the chapter presents the validation of the temperature evolution equation, buoyancy-driven flow through the Boussinesq approximation, and three-dimensional simulations.

4.2 Poiseuille Flow in Two Dimensions

The purpose of this test is to investigate and validate the convergence of the SPH simulation compared with an analytical solution for flow. This is shown to be the case for both one and two phases of constant viscosity, along with viscosity dependent on a linear temperature gradient across the channel. Furthermore, SPH is able to accurately model the onset of interfacial instabilities, as well as provide insight into the growth rate and development of these instabilities where other methods cannot.

The numerical domain is described as follows. Fluid fills the space between two fixed walls. Each end is open and subject to periodic boundary conditions. Flow is driven by a constant body force representing a pressure gradient acting on the fluid.

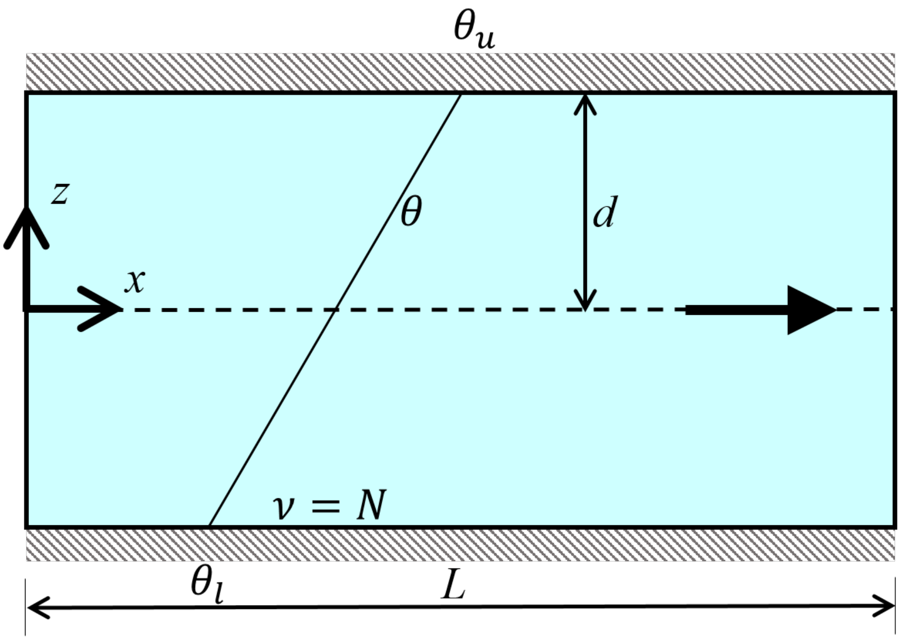


Figure 4.1: Diagram of Poiseuille flow with arrow showing direction of motion.

All cases in this section are run with dynamic boundary condition (DBC) since no significant difference was found with modified dynamic boundary condition (mDBC), and the key physics in Section 4.2.2 take place away from the boundaries in the channel.

4.2.1 Single-Phase Poiseuille Flow

When there is a single fluid phase in the channel, at steady state the governing equation of motion is

$$\frac{\partial^2 \mathbf{v}}{\partial r_z^2} = \frac{1}{\mu}\left(\frac{dp}{dr_x} - \rho f_x\right), \tag{4.1}$$

where r_x is position along the channel and r_z position across it. Let d be the half-channel width and note that $\mu = \rho\nu$, then integrate to find velocity along the channel, v_x. When flow is driven by a body force, f_x, along the channel

$$v_x = \frac{-f_x}{2\nu}(r_z^2 - d^2), \tag{4.2}$$

and when driven by pressure gradient

$$v_x = \frac{1}{2\rho\nu}\frac{dp}{dr_x}(r_z^2 - d^2). \tag{4.3}$$

parameter	value
ρ	$1000\ \mathrm{kg\,m^{-3}}$
ν	$1 \times 10^{-6}\ \mathrm{m^2\,s^{-1}}$
f_x	$0.0001\ \mathrm{kg\,m\,s^{-2}}$
d	$0.0005\ \mathrm{m}$

Table 4.1: Parameters for single phase Poiseuille flow.

For comparison, the parameters in Table 4.1 are chosen to match DualSPHysics example case 15 [56]. This results in a Reynolds number of approximately 30. SPH results overpredict channel velocity slightly when compared to the analytical solution. However, the modifications described in Section 3.4.2 and addition of timestep constraint for viscosity in Equation (3.63) reduce the error, defined for a general variable ϕ by

$$L^2(\phi) = \sqrt{\frac{\sum(\phi_{\text{analytical}} - \phi_{\text{SPH}})^2}{\sum \phi^2_{\text{analytical}}}}. \tag{4.4}$$

The modifications described in Chapter 3 have been made, except those relating to temperature. As can be seen in Table 4.2, the error quantified by the L^2 norm is comparable for the modified code developed for this work to the rigorously validated release version of DualSPHysics. In fact, it is slightly closer to the analytical solution, and the error is further reduced when using mDBC.

configuration	$L^2(v_x)$
DualSPHysics v5.0 release with DBC	0.0250
modified DualSPHysics v5.0 with DBC	0.0200
modified DualSPHysics v5.0 with mDBC	0.0185

Table 4.2: L^2 error norm of velocity along channel, compared with analytical solution for different code configurations.

4.2.2 Two-Phase Poiseuille Flow with Constant Viscosity

The classic Poiseuille flow case of flow along a channel with fixed walls is altered so that the fluid is composed of two phases. This case validates this SPH model for simulating flows of two phases with different viscosities but similar densities. Later, in Section 4.2.2.2, it is shown that the model is also able to capture a complex and changing interface.

The channel is initially split into two phases of widths d_1 and d_2, as in Figure 4.2.

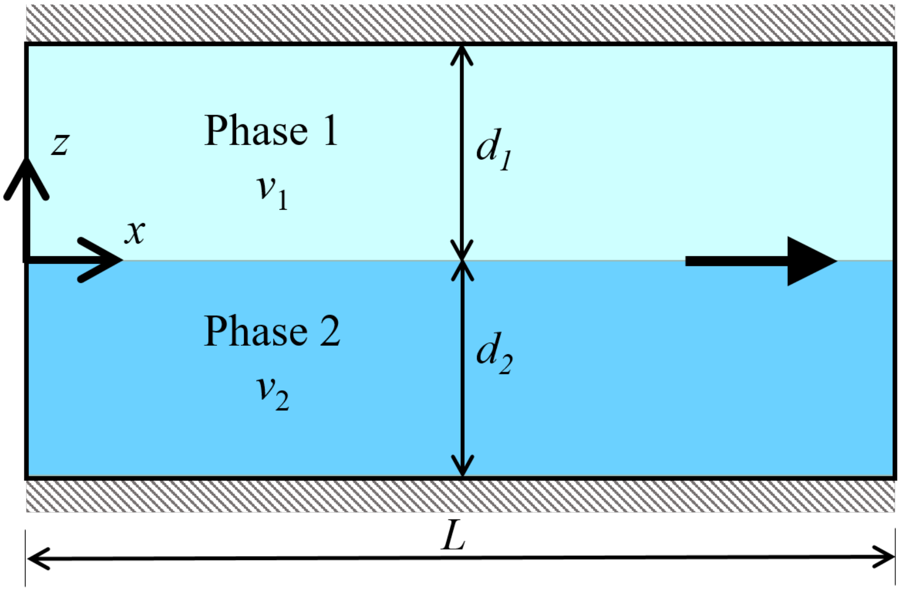


Figure 4.2: Two-phase Poiseuille flow set-up, showing kinematic viscosities ν_1 and ν_2 and width of each fluid phase. Motion is from left to right.

The fluid phases each have constant viscosities ν_1 and ν_2 respectively, and equal constant density ρ. Motion is driven by a constant body force or equivalent pressure gradient along the channel,

$$\mathbf{f} = (f_x, f_z) = (0.8, 0)\mathrm{m\,s}^{-2} = \frac{p_0 - p_L}{L}, \tag{4.5}$$

where $p_0 = p(x = 0), p_L = p(x = L)$ and L is the length of the channel.

For the two-phase cases, an approximate Reynolds number is calculated as in Equation (2.6) with the characteristic length the channel width, characteristic velocity the maximum velocity along the channel and characteristic viscosity the mean viscosity of the two phases. This allows comparisons of flow behaviour and time-scales to be made between single and multi-phase cases.

When $\nu_1 = \nu_2$, the cross-sectional velocity profile is symmetric, as expected for the single phase case, as in Figure 4.3. However, if $\nu_1 \neq \nu_2$ then the velocity profile can look like Figure 4.4 where symmetry no longer holds and the maximum velocity does not lie at the interface. These figures are produced with $d_1 = d_2 = 1$ m, $\nu_1 = 0.1$ $\mathrm{m}^2\,\mathrm{s}^{-1}$, $\nu_2 = 0.025$ $\mathrm{m}^2\,\mathrm{s}^{-1}$ and $\rho = 1000$ $\mathrm{kg\,m}^3$. Therefore, the Reynolds number is 64.

An analytical solution exists for the velocity along the channel in each fluid phase

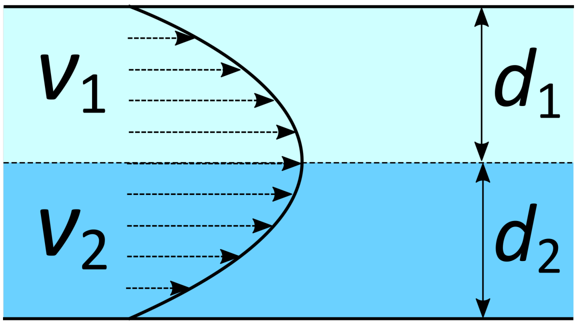


Figure 4.3: Sketch of velocity profile along channel when viscosities of each phase are equal.

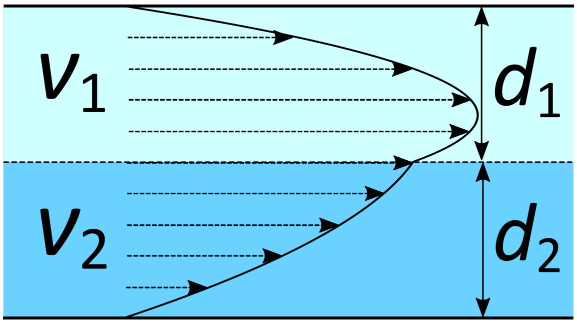


Figure 4.4: Sketch of velocity profile along channel when phases have different viscosities.

when $d_1 = d_2 = d$ and is given by Bird et al. [25] for a channel of length L:

$$\begin{aligned}(v_x)_1 =& \frac{f_x d^2}{2\rho\nu_1}\left[\left(\frac{2\nu_1}{\nu_1+\nu_2}\right) + \left(\frac{\nu_2-\nu_1}{\nu_1+\nu_2}\right)\left(\frac{r_z}{d}\right) - \left(\frac{r_z}{d}\right)^2\right];\\ (v_x)_2 =& \frac{f_x d^2}{2\rho\nu_2}\left[\left(\frac{2\nu_2}{\nu_1+\nu_2}\right) + \left(\frac{\nu_2-\nu_1}{\nu_1+\nu_2}\right)\left(\frac{r_z}{d}\right) - \left(\frac{r_z}{d}\right)^2\right].\end{aligned} \tag{4.6}$$

To determine when steady state has been reached, a measure of total kinetic energy is calculated over the whole fluid domain as the sum of kinetic energy of each particle. Since SPH particle mass remains constant, each particle in the two-dimensional simulation has the same mass $m = \overline{\rho}(dx)^2$, where $\overline{\rho}$ is the initial density and dx is the particle spacing. Therefore, total kinetic energy is computed from particle velocity along the channel by

$$E_k = \sum_j \frac{m_j}{2} v_{x,j}^2 = \frac{\overline{\rho}(dx)^2}{2} \sum_j v_{x,j}^2. \tag{4.7}$$

The change in kinetic energy is tracked, with steady state reached once

$$\left(\frac{dE_k}{dt} \max\{E_k\}\right) < \varepsilon. \tag{4.8}$$

where ε is a small constant threshold.

The velocity profile produced by the SPH model, once it has reached a steady state, after approximately 20 s, for the initial case described above is shown in Figure 4.5. This simulation gives close results to the analytical solution, capturing the qualitative behaviour of the velocity well – the shapes of the velocity profile in each phase and the discontinuity at the interface. The difference between the analytical value and maximum velocity or velocity at the interface are both less than 3%. The differences are explained by the discrete approximation of the numerical scheme and error from the interpolation kernel, as well as the assumption by Bird et al. [25] of incompressibility.

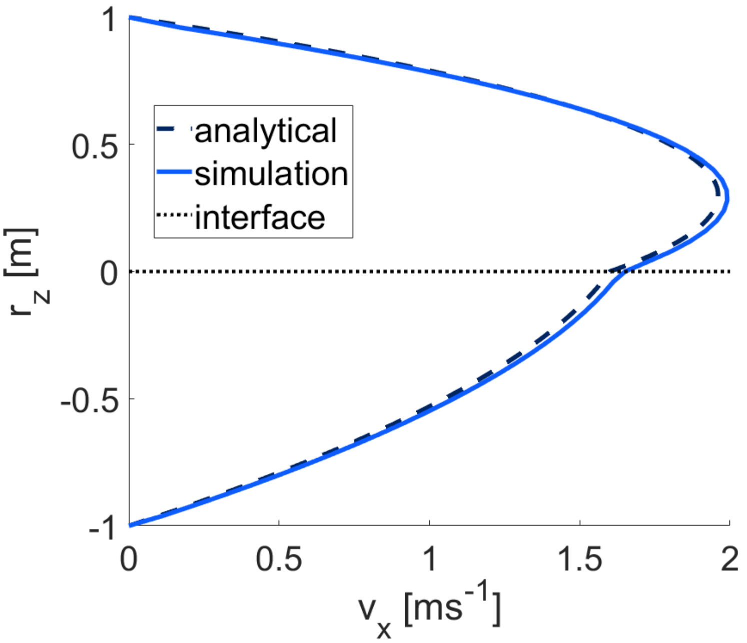


Figure 4.5: Velocity along the channel for a cross section of the flow at steady state with $\nu_1 = 0.1\text{m}^2\text{s}^{-1}$, $\nu_2 = 0.025\text{m}^2\text{s}^{-1}$, $d_1 = d_2 = 1\text{m}$ and interface located at position $z = 0\text{m}$.

For the following cases, $d_1 \neq d_2$ and, therefore, Equation (4.6) no longer holds. An analytical solution for three-phase steady Poiseuille flow with differently size phase widths has been derived by Rowlatt and Lind [195], also for incompressible flow. Velocity is assumed to be along the channel only ($v_z = 0$) and a constant pressure gradient along the channel $\frac{\partial p}{\partial x} = P = f_x$. Converting their solution to two phases, the analytical solution for velocity in each phase is:

$$\begin{aligned}(\tilde{v}_x(r_z))_1 &= \frac{P}{2\rho\nu_1} r_z^2 + \frac{C_1}{\rho\nu_1} r_z + C_2^1;\\ (\tilde{v}_x(r_z))_2 &= \frac{P}{2\rho\nu_2} r_z^2 + \frac{C_1}{\rho\nu_2} r_z + C_2^2.\end{aligned} \tag{4.9}$$

The constants are calculated as

$$\begin{aligned}C_1 &= -\frac{P\left(\frac{d_2^2}{\nu_2} + \frac{d_1^2 + 2d_1 d_2}{\nu_1}\right)}{2\left(\frac{d_1}{\nu_1} + \frac{d_2}{\nu_2}\right)};\\ C_2^1 &= -\frac{d_1 + d_2}{\rho\nu_1}\left(\frac{P(d_1 + d_2)}{2} + C_1\right);\\ C_2^2 &= 0.\end{aligned}$$

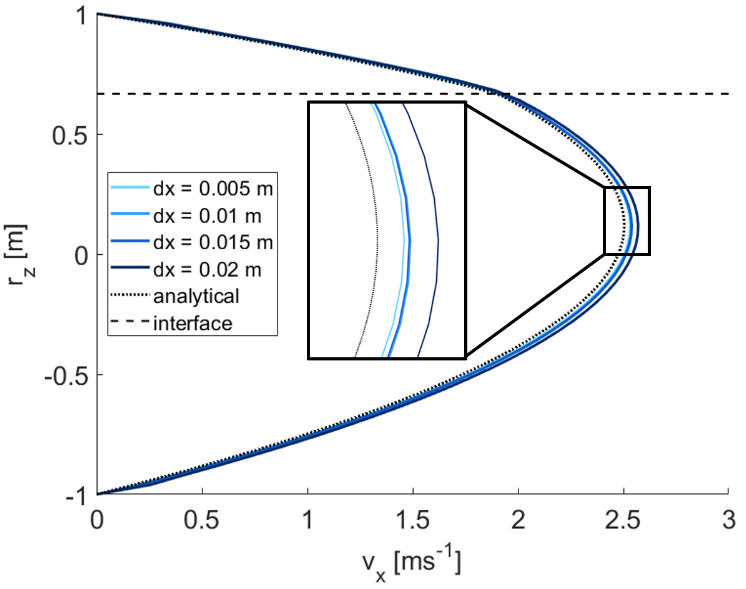


Figure 4.6: Comparisons of velocity profiles with varying particle spacing for $\nu_1 = 0.10, \nu_2 = 0.200, d_1 = 0.3333, d_2 = 1.6667$. Box shows zoom of peak velocity.

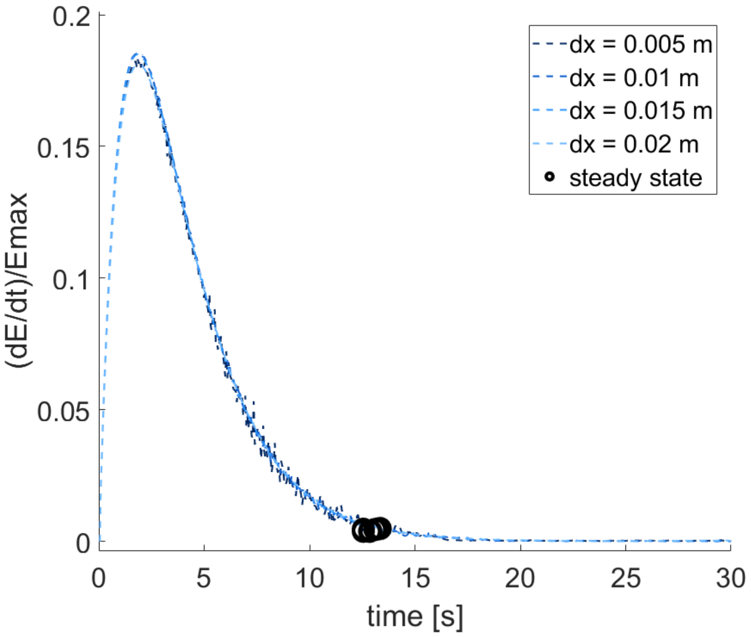


Figure 4.7: Steady state measures with varying particle spacing for $\nu_1 = 0.10, \nu_2 = 0.200, d_1 = 0.3333, d_2 = 1.6667$, with time of steady state marked.

4.2.2.1 Convergence of Velocity Profile

Multiple resolutions are run for an example case in Figure 4.6, with the maximum particle spacing the minimum required for a physically correct flow. This case has a Reynolds number of 33, close to that of the single phase case in Section 4.2.1. The

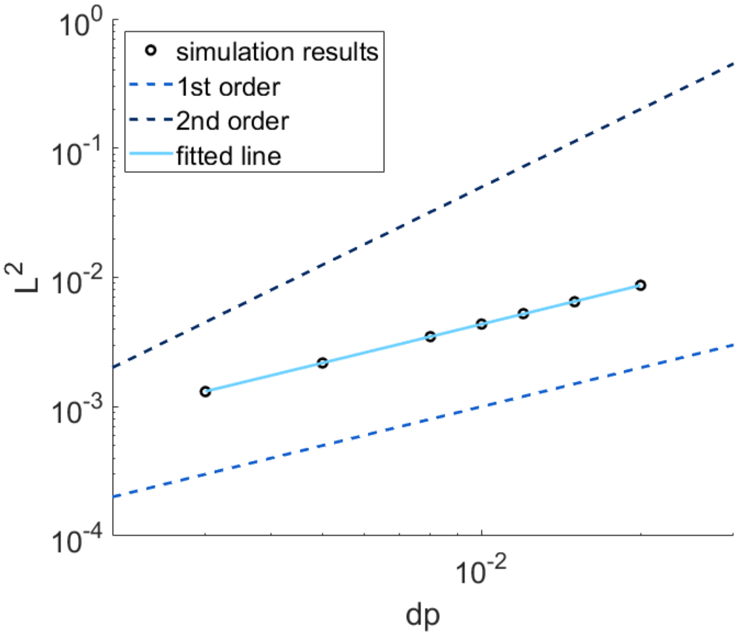


Figure 4.8: L^2-norm of velocity for with varying particle spacing for $\nu_1 = 0.10, \nu_2 = 0.200, d_1 = 0.3333, d_2 = 1.6667$.

position of the interface is marked and a change in velocity profile can be seen at this point. A steady state threshold of $\varepsilon = 0.1$ is chosen with Equation (4.8) and these points are marked in Figure 4.7. Convergence is quantified using the analytical solution in Equation (4.9) by an L^2 norm over all N SPH particles in the fluid, using Equation (4.4) divided by N with v_x in the place of ϕ. The resulting norms are plotted against particle spacing on logarithmic axes in Figure 4.8. A line through these points gives the rate of convergence to be of first order, $\mathcal{O}(h)$. This is the order of convergence expected to be achieved when using DBC. This figure also confirms the results from Figure 4.6 that the simulations are converging.

4.2.2.2 Interfacial Instability

After a small time, the case in Figure 4.5 becomes unstable at the interface between phases, causing the velocity profile to develop fluctuations. Numerous papers in the literature consider the interface instability which appears in two-phase Poiseuille flow [13, 35, 92, 93, 177, 235, 257, 258]. Here, the SPH model is used to provide insight into the formation of the instability in the particular case of two phases of different viscosity but constant density.

The first person to characterise this instability was Yih in 1967 [258], who derived an Orr-Sommerfeld equation which governs stability. This leads to an eigenvalue problem with respect to the wavespeed c, such that

$$c \begin{cases} > 0 & \text{unstable;} \\ = 0 & \text{neutrally stable;} \\ < 0 & \text{stable.} \end{cases}$$

Assuming constant density and equal widths of phases, Yih [258] derived an equation in terms of viscosity ratio ($\tilde{m} = \frac{\nu_2}{\nu_1}$) which governs when the flow becomes unstable.

Instability occurs when the velocity gradient at the interface is discontinuous and can be caused by changing only the viscosity in each phase [258]. This is an example of a Rayleigh instability, caused by an inflection point in the velocity profile [185]. The pressure gradient across the channel remains linear, but is non-zero at the interface. If the velocity profile across the whole channel is stable, then the flow remains stable [36]. Most of the analytical investigations (including Pinarbasi and Liakopoulos [177], Valluri et al. [235], Hooper [92, 93], Anturkar et al. [13], Charru and Fabre [36]) use perturbation theory to find a relationship between wavenumber and formation of an interfacial instability.

Experiments have also been performed to investigate the instability. Charles and Lilleleht [35] used oil and water in a rectangular channel to identify that interfacial waves occurred at the turbulent transition in the less viscous (water) phase. These waves increased when the flow rate of water and ratio of water to oil were both higher. Kao and Park [112] used a similar set up to find a critical Reynolds number, below which instability growth at the interface is fully damped.

A relationship between viscosity ratio and phase width ratio for instability development from long-wavelength disturbances was found by Yiantsios and Higgins [257] using perturbation analysis. Gravity is included but is insignificant for the case of constant density. Conditions were found for interfacial wave growth in terms of the relationship between phase width ratio $\tilde{n} = \frac{d_2}{d_1}$ and viscosity ratio $\tilde{m} = \frac{\nu_2}{\nu_1}$. The corresponding neutral stability diagram with curves of $\tilde{m} = 1, \tilde{m} = \sqrt{\tilde{n}}$ can be see in Figure 4.9. The case in Figure 4.5 ($\tilde{n} = 1, \tilde{m} = 0.25$), which became unstable, lies clearly in an unstable region, therefore the observed instability agrees with the theoretical results.

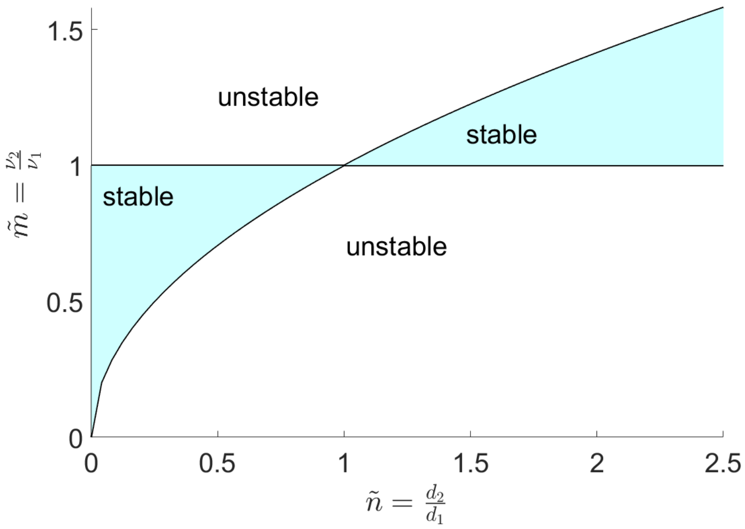


Figure 4.9: Neutral stability curves for two-phase Poiseuille flow, with stable regions shaded.

To investigate further the ability of the SPH model to capture stable and unstable behaviour in each of the four regions in Figure 4.9, four extra test points are chosen. The parameters used are given in Table 4.3 and the points are marked on the stability diagram in Figure 4.10, along with point 0 from Figure 4.5.

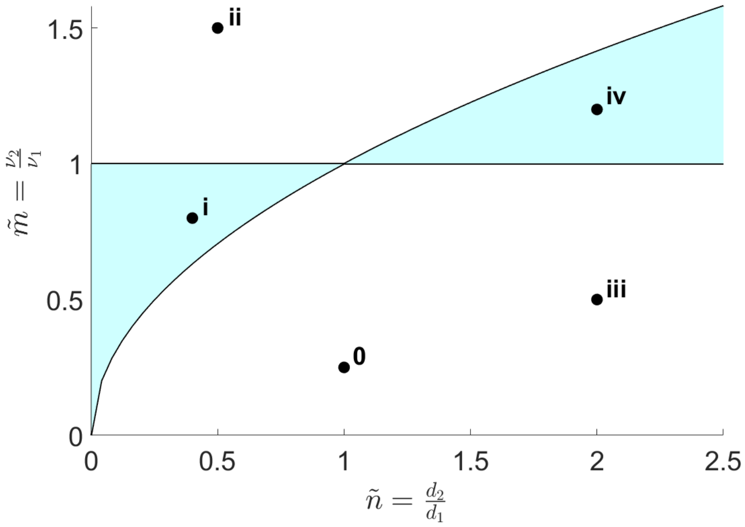


Figure 4.10: Test points for instability formation.

The resulting development of instabilities (or not) can be seen in Figure 4.11. These agree with the perturbation analysis in the literature. Points *i* and *iv* remain stable for the full 60 s run. For point *ii*, the instability is observed as a deformation of the interface beginning to grow. The instability of point *iii* is much clearer and the deforming shape of the interface can be clearly seen as the instability continues to

test point	$\tilde{n}$	$\tilde{m}$	ν_1	ν_2	d_1	d_2	Re
i	0.4	0.8	0.1	0.08	1.2	0.8	100
ii	0.5	1.5	0.1	0.15	4/3	2/3	56
iii	2.0	0.5	0.1	0.05	2/3	4/3	160
iv	2.0	1.2	0.05	0.06	2/3	4/3	255

Table 4.3: Table of parameters for instability test points.

grow. An investigation revealed that the use of the particle shifting technique does not prevent the onset of instability and only influences the shape of the growing instability at later time, when perturbation theory is no longer valid.

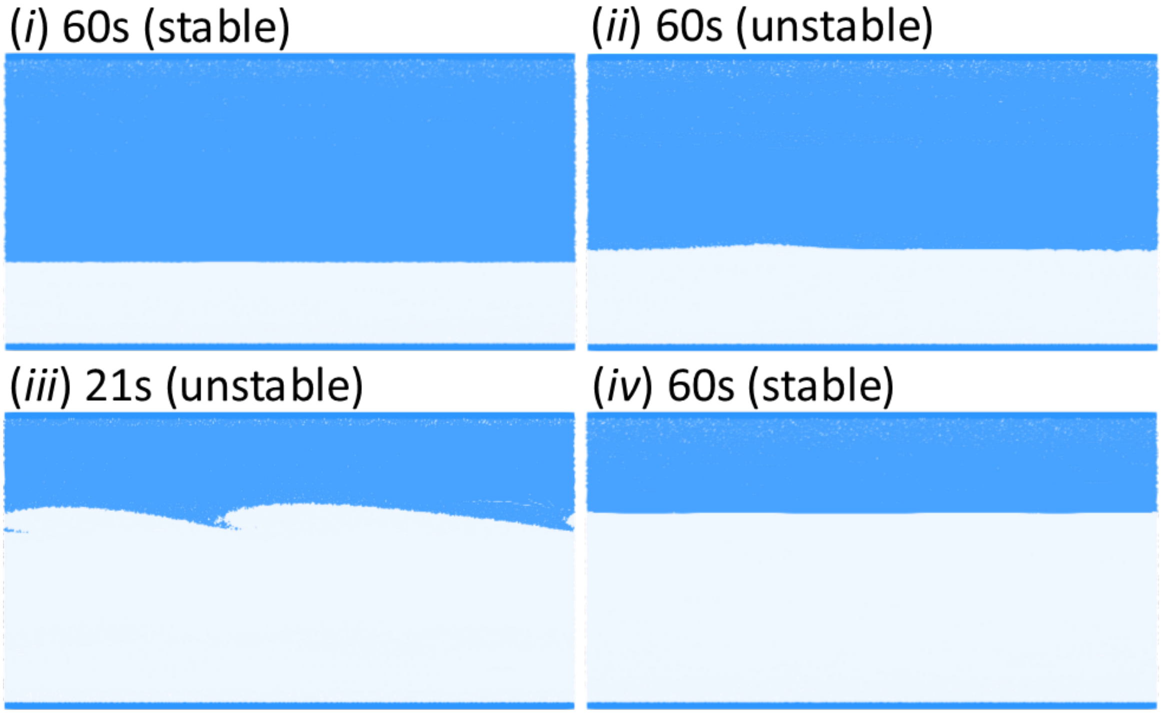


Figure 4.11: Two-phase Poiseuille flow for instability test points *i*-*iv*, particles coloured by fluid phase.

Points very close to the stability boundaries in Figure 4.9 appear to remain stable and may require a very high resolution for instability to appear. The onset of interfacial instabilities is clearly sensitive to particle resolution. For case *iii* in Table 4.3 and Figure 4.11, the interface does not deform at all for $\overline{dx} = \frac{dx}{d_1+d_2} > 0.01$ and, although some deformation is seen for $\overline{dx} > 0.005$, there is no growth of instability. Therefore, the results suggest that a threshold $\overline{dx}$ of 0.005 is required to capture the instability. As $\overline{dx}$ is decreased, the interface becomes sharper, as expected from the convergence study in Section 4.2.2.1.

Using Equation (4.9), the velocity profiles of the four test points chosen in Table 4.3 are plotted, along with the corresponding simulation results, in Figure 4.12. All four points show close agreement with the analytical velocity, with points *iii* and *iv* being slightly higher and points *i* and *ii* slightly lower.

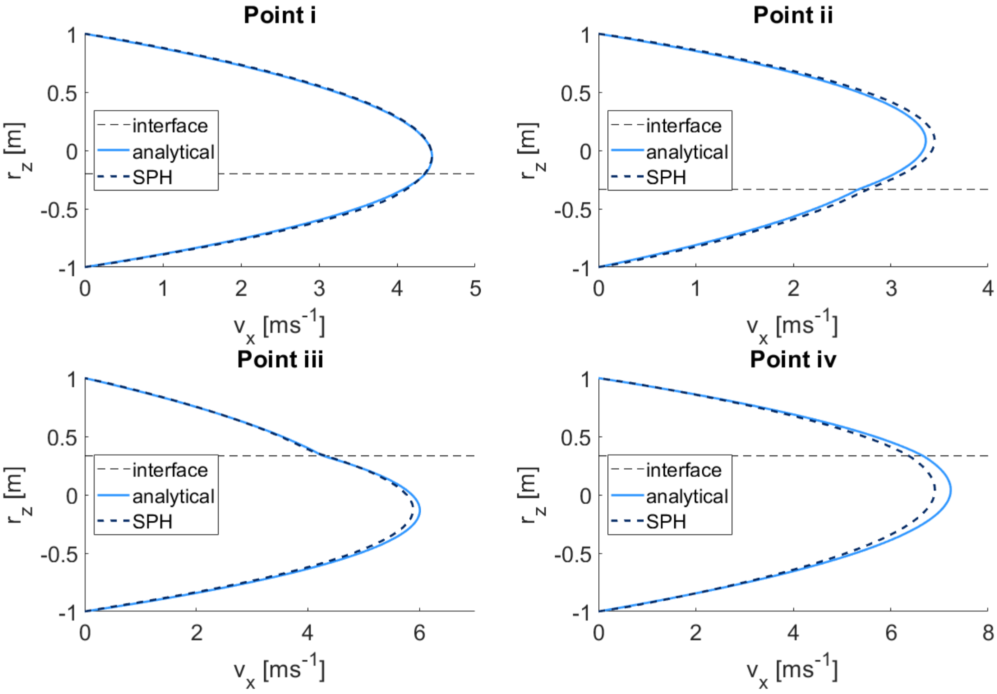


Figure 4.12: Comparisons of velocity profiles for the 4 test points with analytical solution.

4.2.2.3 Instability Growth Rate

There are no results in the literature which show the shape of the interface as the instability grows, since most studies focus on the initiation and initial growth of the instability using analytical techniques. However, comparisons can be made of the growth rate of the instability for small times after instability onset when the linear theory is valid.

The growth rate of the instability is estimated by following the position across the channel of particles which are initially located at the interface. A case investigated by Valluri et al. [235] with $\tilde{m} = 30, \tilde{n} = \frac{3}{7}$ and $Re = 6.45$, which has theoretical growth rate 0.34 is reproduced using the SPH model described in Chapter 3 and is shown in Figure 4.13. The average fluid velocity of 5 $\mathrm{m\,s^{-1}}$ is obtained with a constant body force of $f_x = 16.07\ \mathrm{m\,s^{-2}}$. A maximum vertical position r_z is taken over M points,

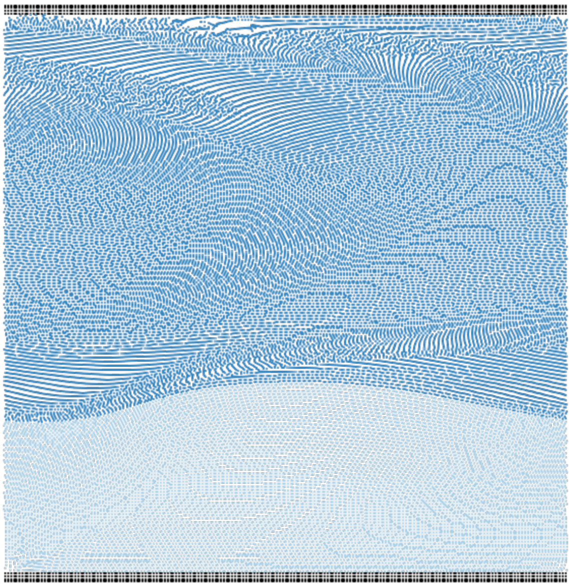


Figure 4.13: Deforming interface for growth rate case at 0.5 s with particles coloured by phase.

equally spaced along the line of the initial interface, at all output times and plotted in Figure 4.14. The onset of instability is defined to be at the first time the maximum vertical position of the chosen particles max r_z exceeds dx above the interface. To focus on linear growth only, a period of 0.2 s after instability onset is plotted in Figure 4.15 and an exponential line fitted.

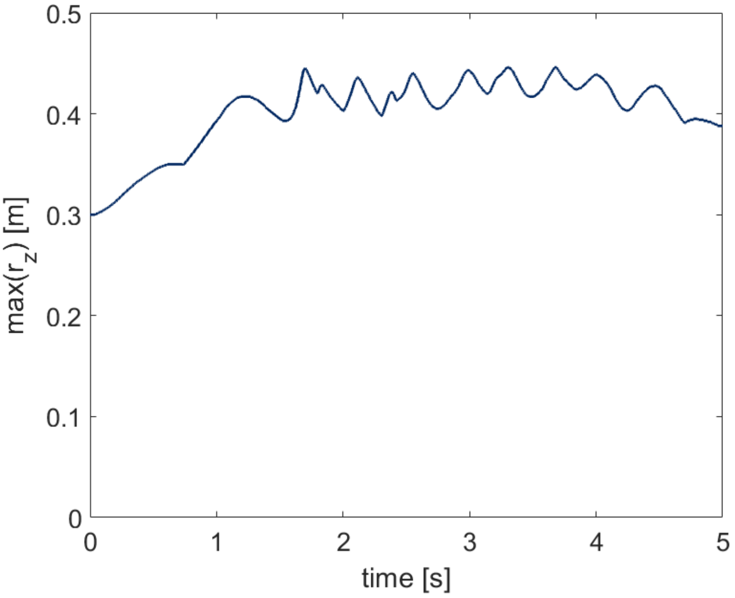


Figure 4.14: Maximum vertical displacement of interface points with time to measure growth rate of instability.

The instability growth rate results for different numbers of M points are given in Table 4.4. The points are regularly spaced throughout the domain in order to capture all amplitudes. It is found that 2 points are required per wavelength (≈ 1 m here) in order to accurately capture the growth rate. This gives a growth rate of 0.35 for 2 or 4 particles, demonstrating the ability of this SPH method to model the early stages of instability growth. The growth rate for $M = 8$ is still within 10% of the theoretical value. To the author's knowledge, this is the first time these results have been achieved with SPH. Unlike perturbation theory, the shape of the deforming interface can also be seen. Both are clear advantages over the STAR-CCM+ simulations presented next for comparison.

Number of points along initial interface M	growth rate
1	0.2834
2	0.3512
4	0.3514
8	0.3715
Theoretical [235]	0.3400

Table 4.4: Growth rate of interfacial instability for different numbers of sample points.

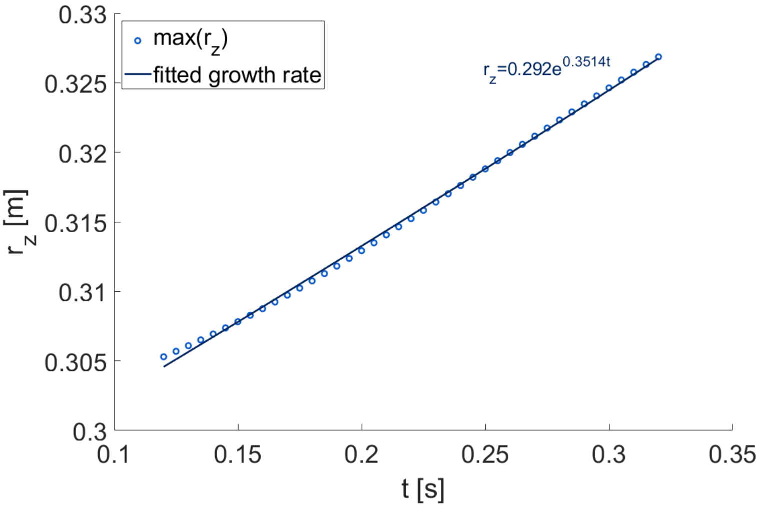


Figure 4.15: Growth rate of Poiseuille flow instability.

4.2.2.4 Comparison with Mesh-Based Method

Without a reference solution for the growth and development of the instability, the commercial code STAR-CCM+ [208] is used to compare the flow behaviour. Results

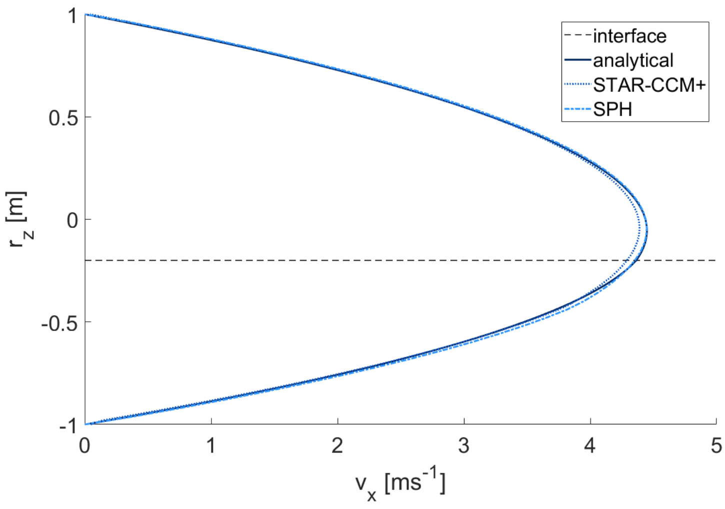


Figure 4.16: Velocity profiles for different methods, point *i*.

are in agreement for velocity along the channel, as shown in Figure 4.16. For comparison, the grid size is set to be equal to particle spacing, dx. However, it is unable to produce the deforming shape of the interface in Figure 4.17 as SPH does for the same case in Figure 4.11 case *iii*, and therefore no comparisons of the interface shape can be made. For any instability that does develop at the interface using STAR-CCM+, growth is in the form of mixing of the two phases and as the instability grows it becomes more difficult to capture the position of the interface. This is due to the diffusive nature of the volume-of-fluid (VOF) method used for multi-phase models. Hence, it is also not possible to capture the growth rate and evolution of the interfacial instability accurately.

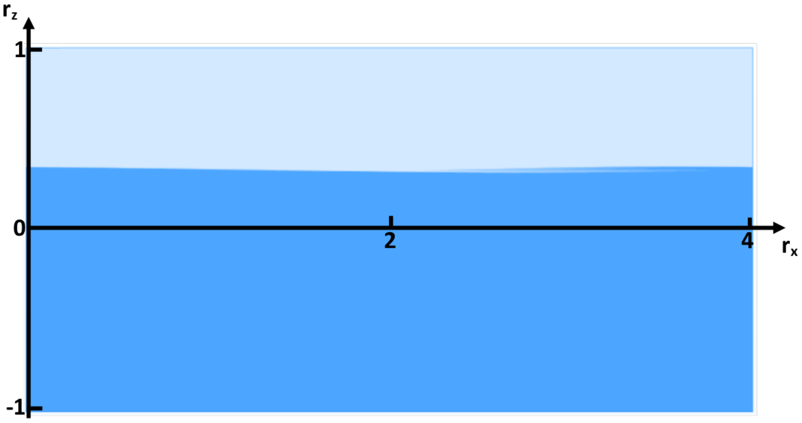


Figure 4.17: Deforming interface for point *iii* with STAR-CCM+.

Since VOF is a mesh-based method, the exact position and shape of the interface must be approximated using reconstruction techniques which can be computationally expensive [223] and the accuracy depends on the resolution of the mesh. This is particularly difficult for a complex dynamic interface [254] and may require the VOF method to be coupled with another method [224]. Since exact information for the interface is lost and the new volume fluxes for time integration depend on the inferred interface, this error grows with time [224].

4.2.3 Temperature-Dependent Poiseuille Flow

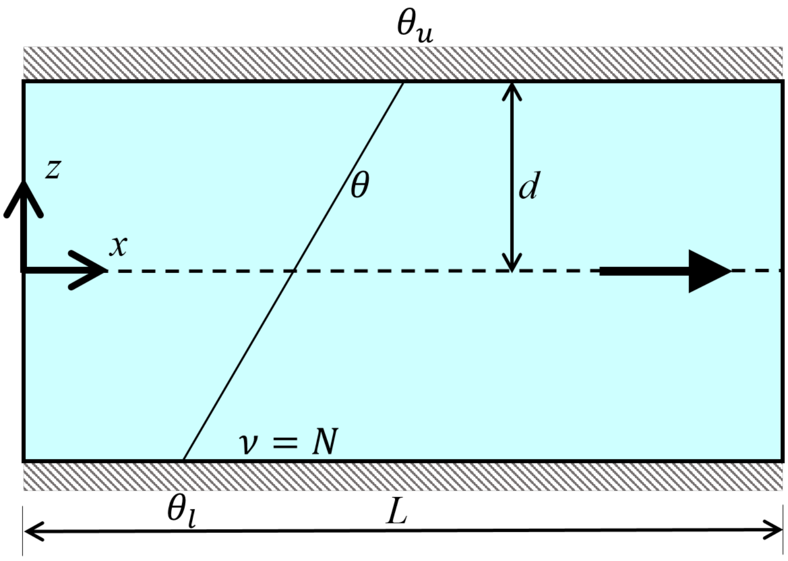


Figure 4.18: Temperature-dependent Poiseuille flow, showing temperature gradient across channel and direction of flow.

A number of studies have considered the stability of Poiseuille flow with temperature-dependent viscosity and heat transfer [180, 198]. Further cases are run for a single phase Poiseuille flow case with phase viscosity dependent on temperature to attempt to replicate the analytical velocity profile. This case validates the addition of temperature as a variable, as well the inclusion of a variable viscosity which depends on temperature. The temperature difference between upper and lower boundaries influences the stability of the flow [180], and flow is stabilised when viscosity is increased non-uniformly [243]. All cases considered here are stable.

The channel is discretised into 100 fluid particles across with periodic boundaries at each end. The bottom boundary in Figure 4.18 has temperature θ_l, viscosity

$\nu = N$ and the upper boundary has temperature θ_u. The boundaries have fixed temperatures with no-slip velocity boundary condition. Following Wall and Wilson [243], temperature is non-dimensionalised by

$$\theta^* = 2\left(\frac{\theta - \theta_l}{\theta_u - \theta_l}\right) \tag{4.10}$$

and is linearly dependent on the position across channel through

$$\theta^*(r_z) = \left(1 + \frac{r_z}{d}\right), \tag{4.11}$$

where d is half the channel width. This is implemented by setting the temperature option of the fluid phase to be a linear temperature gradient, as described in Section 3.5.2. A characteristic average velocity for the flow is given by [243]

$$\overline{v} = \frac{PL^2}{2N} \tag{4.12}$$

with $-P(< 0)$ a constant pressure gradient along the channel. The flow can be characterised by a Reynolds number through [243]

$$Re = \frac{\overline{v}L}{N} = \frac{PL^3}{2N^2}. \tag{4.13}$$

Firstly, an exponential relationship for viscosity is used where [243]

$$\nu(\theta^*) = Ne^{-K_1\theta^*} \tag{4.14}$$

for some constant K_1. Since viscosity is dependent on the non-dimensionalised temperature Equation (4.10), the temperature difference between channel boundaries has no effect on the viscosity across the channel. According to Wall and Wilson [243], this has an analytical solution for velocity along the channel of

$$\tilde{v}_x(r_z) = -\frac{2}{K_1}\overline{v}\left[1 + \coth K_1 + (r_zd - \coth K_1)e^{K_1(1+r_zd)}\right]. \tag{4.15}$$

To avoid turbulent effects, $Re < 10$ is maintained by choosing $P = 0.1, d = 0.5, N = 0.1$. This results in $Re = 0.625, \overline{v} = 0.12$. The effect on the error of increasing the Reynolds number above 10 can be seen in Table 4.5.

SPH slightly over-predicts velocity in Figure 4.19, but the maximum is at the same position across the channel as the analytical velocity. Boundaries are modelled using DBC, which gave good results for the two-phase case in Section 4.2.2. Temperature is extended linearly into boundary particles through the given temperature

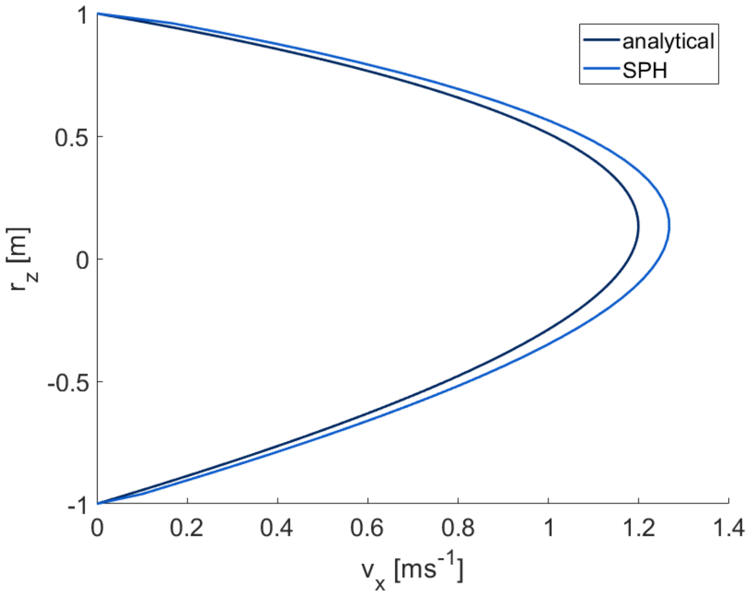


Figure 4.19: Velocity along channel of Poiseuille flow with temperature-dependent viscosity according to Equation (4.14) and $K_1 = 0.4$.

Viscosity equation	constant	N	Re	$L^2(v_x)$
Exponential	$K_1 = 0.2$	0.1	6.25	0.0279
Exponential	$K_1 = 0.4$	0.1	6.25	0.0279
Exponential	$K_1 = 1.0$	0.1	6.25	0.0266
Exponential	$K_1 = -0.4$	0.1	6.25	0.0277
Exponential	$K_1 = 0.4$	0.01	62.5	0.2562
Exponential	$K_1 = 0.4$	1.0	0.625	0.0268
Linear	$K_2 = 0.4$	0.1	6.25	0.0285
DualSPHysics example (isothermal)	-	0.000001	0.05	0.0250

Table 4.5: L^2 norm of error in velocity along channel compared to analytical solution. All results shown with DBC, $dx = 0.02$ m.

boundary conditions at the line of the boundary by assigning the boundary phases the same temperature option as the fluid phase, and viscosity computed according to Equation (4.14). This ensures that boundary particles have viscosity close to that of neighbouring fluid particles. The choice of boundary viscosity treatment influences the velocity profile. For example, when all boundary particles are given the maximum viscosity for the whole domain, the maximum of the velocity profile is shifted across the channel.

Computing the error using an L^2 norm over all fluid particles, as in Equation (4.4), gives a result comparable to those for an isothermal single phase case. For example, the L^2 norm for the DualSPHysics example Poiseuille flow case is listed in Table 4.5

along with those for the temperature-dependent cases defined in this section. The effect of varying constant K_1 on the error is negligible. The error of approximately 3% is within the expected range for Lagrangian SPH with DBC, and is approximately consistent across the channel.

dp	particles	$L^2(v_x)$
0.005	200	0.0234
0.01	100	0.0255
0.02	50	0.0291
0.04	25	0.0366
0.1	10	0.0582

Table 4.6: L^2 norm of error in velocity along channel compared to analytical solution for varying resolution with $K_1 = 0.4, P = 0.8, d = 0.5m, N = 0.1$.

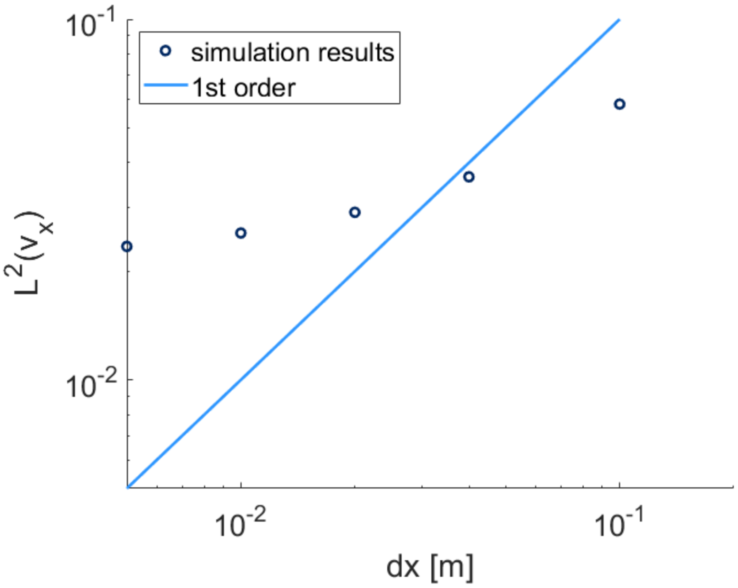


Figure 4.20: Convergence of velocity along channel of Poiseuille flow with increasing resolution for temperature-dependent viscosity according to Equation (4.14) and $K_1 = 0.4$, with first order line plotted.

A convergence study is undertaken with $K_1 = 0.4, P = 0.8, d = 0.5m, N = 0.1$. The results are shown in Table 4.6 and plotted in Figure 4.20, where the convergence rate is less than first order. There is clear convergence of just under first order for coarser resolution, which is tending towards a limiting error. The velocity profiles are plotted in Figure 4.21, where a limit can clearly be seen as the velocity approaches the analytical solution with increasing resolution. In Figure 4.22, the error norm is plotted across the channel for the same resolutions. For all particle spacings the error

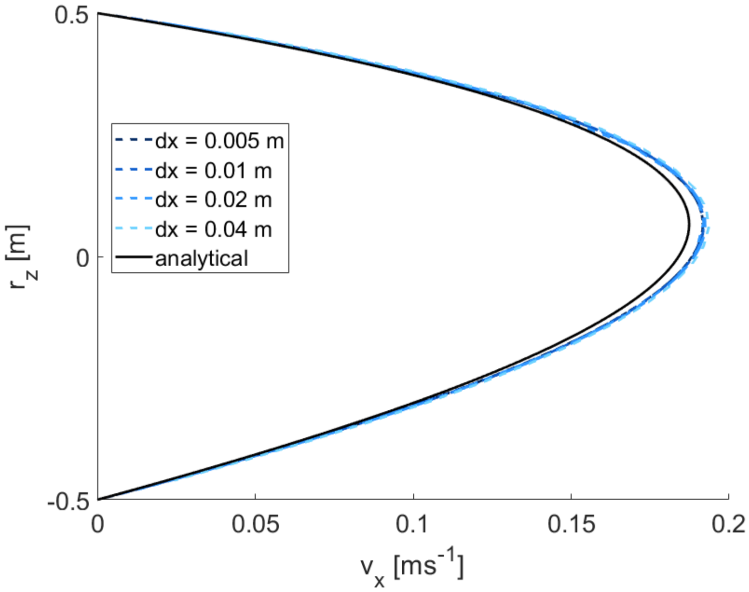


Figure 4.21: Velocity along channel of Poiseuille flow with increasing resolution for temperature-dependent viscosity according to Equation (4.14) and $K_1 = 0.4$.

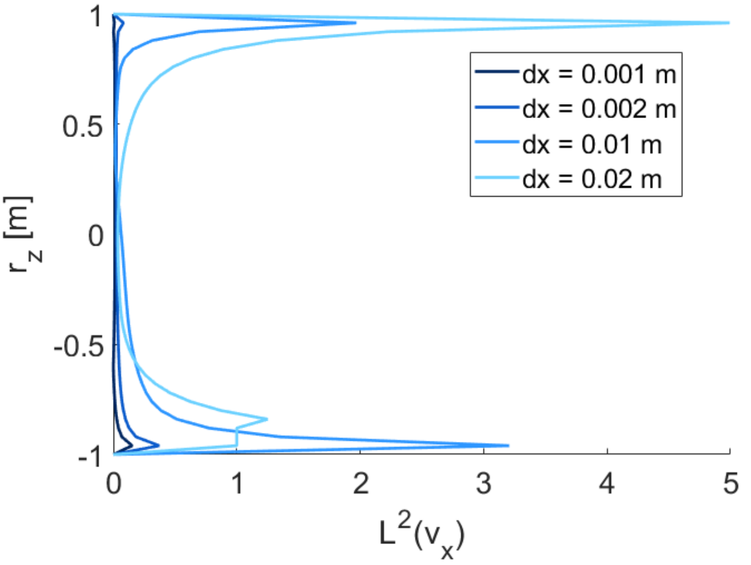


Figure 4.22: $L^2(v_x)$ error across channel for different resolutions.

is greatest near the boundary, but these errors clearly reduce as the particle spacing is decreased.

A viscosity linearly dependent on temperature is investigated using

$$\nu(\theta^*) = N(1 - K_2\theta^*), \tag{4.16}$$

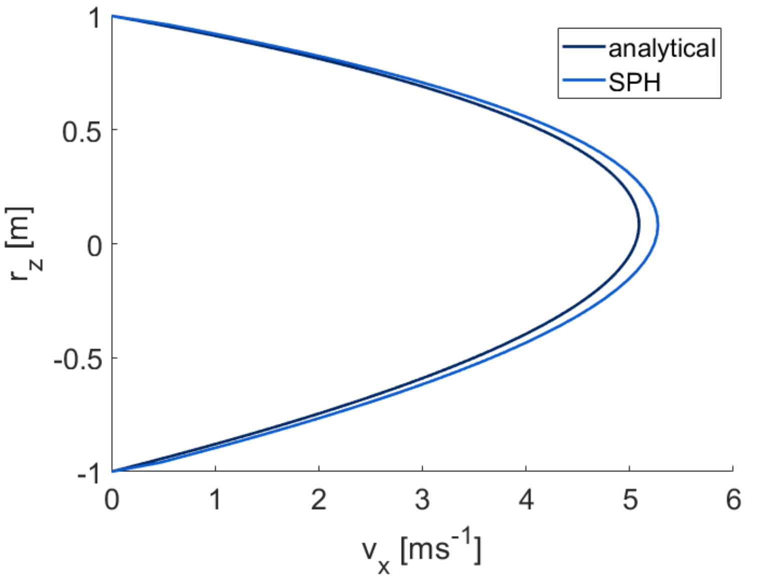


Figure 4.23: Velocity along channel of Poiseuille flow with temperature-dependent viscosity according to Equation (4.16) and $K_2 = 0.2$.

for a chosen constant $K_2 < \frac{1}{2}$, with analytical velocity [243]

$$\tilde{v}_x(r_z) = -\frac{2}{K_2}\overline{v}\left[\frac{-2\log\left[\frac{1-2K_2}{1-K_2(1+r_zd)}\right]}{\log\left(1-2K_2\right)} + 1 - r_zd\right]. \tag{4.17}$$

In Figure 4.23, the velocity is again close to the analytical solution for the full width of the channel, only marginally under-predicting the maximum velocity. The result in Table 4.5 shows this linear viscosity has a similar error to the exponential viscosity.

4.3 Lid-Driven Cavity

This case is chosen to identify and understand the different effects of the viscosity operators and density filters, as well as assess the capabilities of boundary methods. A square cavity of width $L = 1$ m, surrounded by boundary walls, is filled with a fluid which is driven by moving the top wall at velocity $v_x = v_{\text{lid}} = 1\ \text{m}\,\text{s}^{-1}$. It is assumed that there is no gravity acting on the model [240]. A diagram of the case can be seen in Figure 4.24. The flow behaviour is characterised by the Reynolds number

$$Re = \frac{v_{\text{lid}}L}{\nu} = \frac{v_{\text{lid}}L\rho}{\mu}, \tag{4.18}$$

where ρ is density, ν is kinematic viscosity and μ dynamic viscosity. Therefore, changes in viscosity influence the flow within the cavity. The chosen viscosity operator is key to

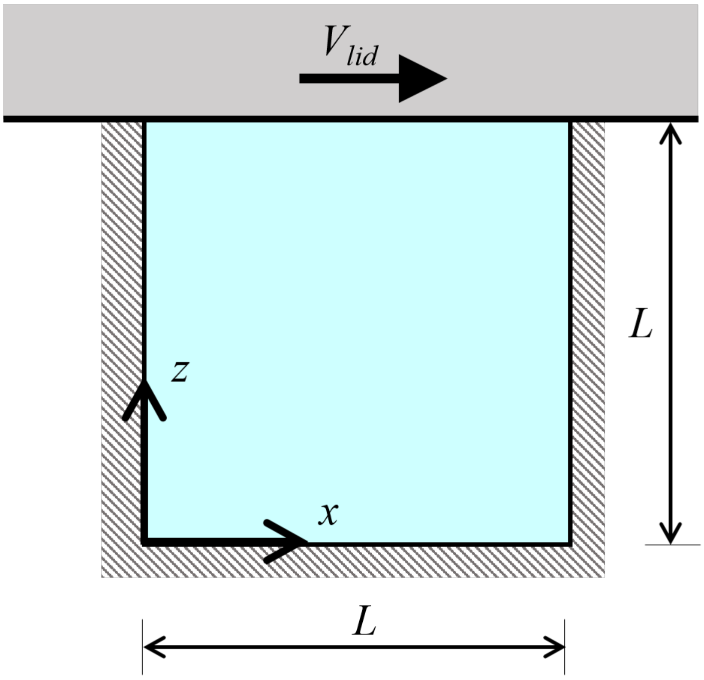


Figure 4.24: Diagram of lid-driven cavity case (adapted from [191]).

determining the flow behaviour since, through the conservation of momentum equation (3.35), the diffusion of momentum depends on the modelling of the viscous term [240].

4.3.1 Steady State

For comparison and validation of flow behaviour, it is important to know when the system has reached a steady state. The change in velocity components between timesteps can be used – this should tend to zero as a steady state is reached. However, this approach is sensitive to small fluctuations and could be improved by averaging over a period of time. If data exists in the literature for comparison then the change in error, for example computed by Equation (4.32), can be used similarly.

Energy measures are also a popular way to determine a steady state. Colagrossi et al. [41] use an energy-decomposition analysis in their work on viscous free-surface flows. Total energy is split into a mechanical force component and an internal viscous component, and a steady state is defined to be reached when the total change in these quantities is zero.

For this case, the total energy of the system shall be monitored and a steady state defined to be reached when the change of energy in time drops below some small constant threshold ε. Since there is no gravity or other external forces acting on the fluid, the total energy in the fluid can be calculated using Equation (4.7). Then the flow is determined to be at steady state by approximating the rate of change of kinetic energy, such that

$$\frac{\dot{E}_k}{\hat{E}_k(t=0)} \approx \frac{E_k^{n+1} - E_k^n}{\hat{E}_k(t^{n+1} - t^n)} < \varepsilon, \tag{4.19}$$

where $\hat{E}_k$ is the total energy in the fluid when the lid first reaches maximum velocity v_{lid}. $\hat{E}_k$ is used for comparison between different simulations in this case since it varies less than maximum particle E_k once the flow is approaching steady state. The threshold ε must be chosen carefully for each case so that it is small enough to identify a reasonably steady state but not too small to be impaired by fluctuations present due to the discrete nature of the method.

4.3.2 Single Phase

The case when the fluid within the cavity is of a single phase has been comprehensively studied in the literature using various numerical techniques and used as a test case. Notably, Ghia et al. [73] provide a large set of data in their paper, which is recommended by SPHERIC [191] as a benchmark to assess code performance. Both Ghia and Violeau [73, 240] take v_{lid} to be constant, but in this work the lid is accelerated from stationary to v_{lid} in the first second of simulation before remaining constant to avoid initially applying instantaneous high velocity to the flow. Comparisons with this data for $Re = 100$ with $L = 1$ m, once the flow has reached a steady state, can be seen in Figure 4.25. Boundary walls are thick enough so that the SPH kernel of any fluid particle is full, including at moving boundary corners.

The SPH model follows the general shape of the velocity profile, but is inaccurate at the lower boundary. Increasing the resolution is seen in Figure 4.26 to produce converging results, and make the curve smoother, but the incorrect behaviour near the bottom boundary remains. Therefore, this underestimation of velocity near the boundary is likely as a result of the simple dynamic boundary conditions used in the model. In Section 4.3.2.1, the effect of using the more sophisticated mDBC method is investigated. STAR-CCM+, however, is able to replicate the velocity in Figure 4.25 near the bottom boundary much more closely.

Streamlines of the same $Re = 100$ case are plotted in Figure 4.27, by interpolating

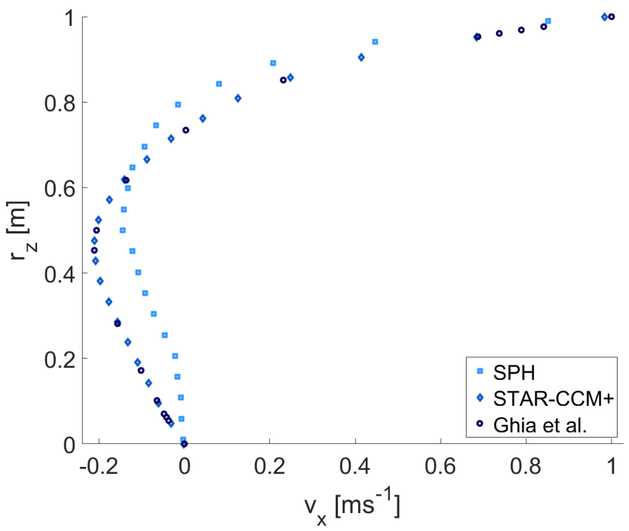


Figure 4.25: Horizontal component of velocity along a vertical line through the centre of the cavity. $Re = 100, dx = 0.02$ m.

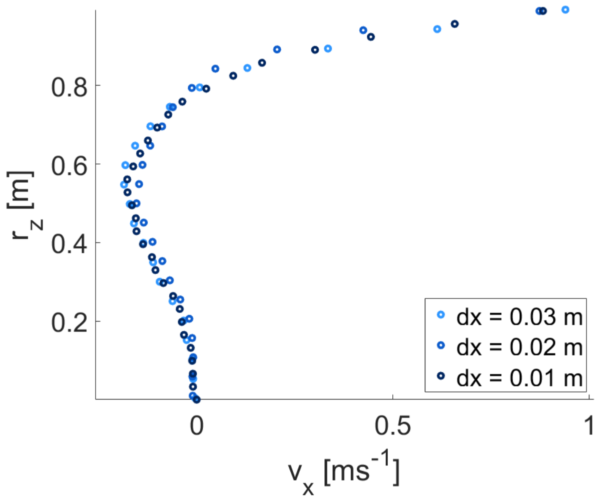


Figure 4.26: Centre line velocity at steady state ($t = 10$s) for various particle spacing dx.

the SPH output data onto a grid and using the MATLAB streamlines function. Despite the difference in velocity near the boundary in Figure 4.25, the main features of this flow are captured by the SPH model. The main large clockwise vortex has formed at the top of the cavity, slightly to the right. The formation of the smaller anticlockwise vortices is captured at the bottom corners. Therefore, the model is capable of capturing the general flow behaviour near the boundaries. An increase in particle number, as well as the advanced boundary condition, could also improve the resolution of these smaller vortices.

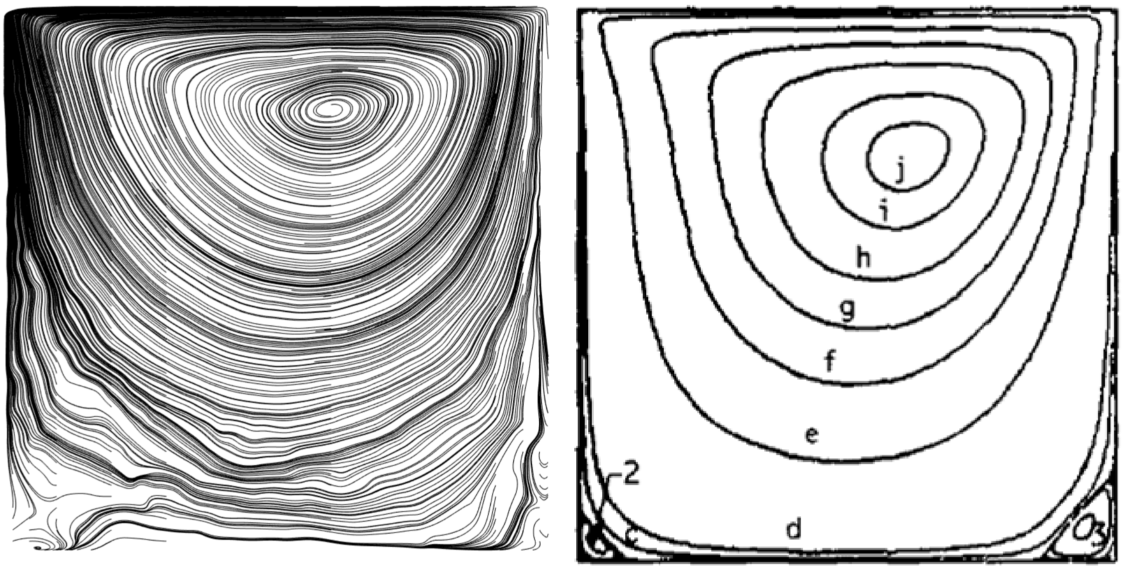


Figure 4.27: Streamlines of lid-driven cavity flow at steady state for $Re = 100$. SPH model on left, Ghia et al. [73] on right.

The case is determined to have reached a steady state according to Equation (4.19), which has been added to the main DualSPHysics code with the option to be output at each timestep. Figure 4.28 shows how both E_k and $\dot{E}_k$ change with time for a given case, as well as highlighting the point of steady state. An approximately steady state (determined by setting $\varepsilon = 0.1$) is reached for this case at 9.66 s, before decaying gradually over time. This decay is found to decrease as particle spacing is reduced. When shifting is included the change in energy is smoother, but with no significant quantitative differences.

When run with lower Reynolds number ($Re \leq 10$), particles are excluded due to particle clumping causing regions of high density or lost through the boundary at the top right-hand corner. These problems are mitigated when the lid is accelerated, rather than starting instantaneously at maximum velocity, v_{lid}, but not eliminated. In the following subsection, the capability of the more sophisticated mDBC method to resolve this is explored.

4.3.2.1 Improved Boundary Conditions

Results in the previous section show that simulations of flow in a lid-driven cavity with DBC produce incorrect velocity profiles. Therefore, to identify whether this is solely due to the default boundary condition in DualSPHysics, the cases using mDBC

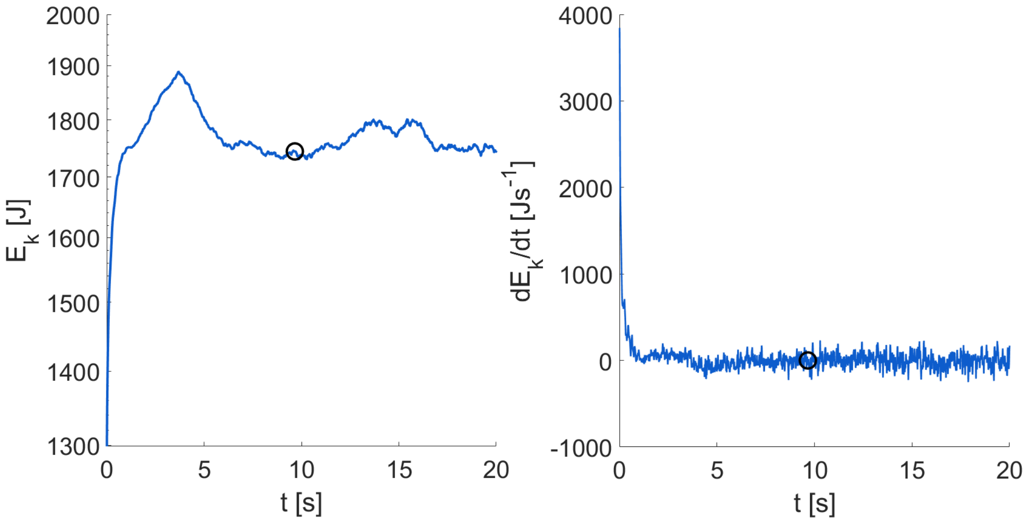


Figure 4.28: Energy (left) and change in energy (right) over time to determine steady state of lid-driven cavity case. The point at which steady state is reached is marked with a black circle. $Re = 100, dx = 0.02$.

with moving and fixed lid are also compared with the results of Ghia et al. [73] at $Re = 100$. For the fixed lid case, boundary particles in the lid are given a velocity, but their position is not updated. This allows the normals required for mDBC calculations to be pointing through the vertex of each top corner. With a moving lid, all normals must point straight down since they are not updated throughout the simulation.

A clear improvement is seen in Figure 4.29, with the velocity being closer to the results of Ghia et al. [73]. In particular, the velocity gradient toward the bottom boundary is now approximately correct. The approach with fixed lid particles was found to have greater difference from the benchmark solution at the bottom of the cavity, so a moving lid is used for this case going forward. The corresponding energy plots to find steady state can be seen in Figure 4.30. The streamlines in Figure 4.31 demonstrate improved agreement with the literature in Figure 4.27.

When working with an in-development form of mDBC which extrapolated both density and velocity into the boundary, particles were lost through the upper right corner of the cavity when full shifting (taking into account boundary particles) was applied. Therefore, the issue is related to shifting within a kernel radius of the boundary and occurs when particles are forced more closely together here than is typical. A tensile correction, as introduced by Monaghan [153] and used for particle shifting by Lind et al. [127] was found to decrease the number of particles lost but not prevent

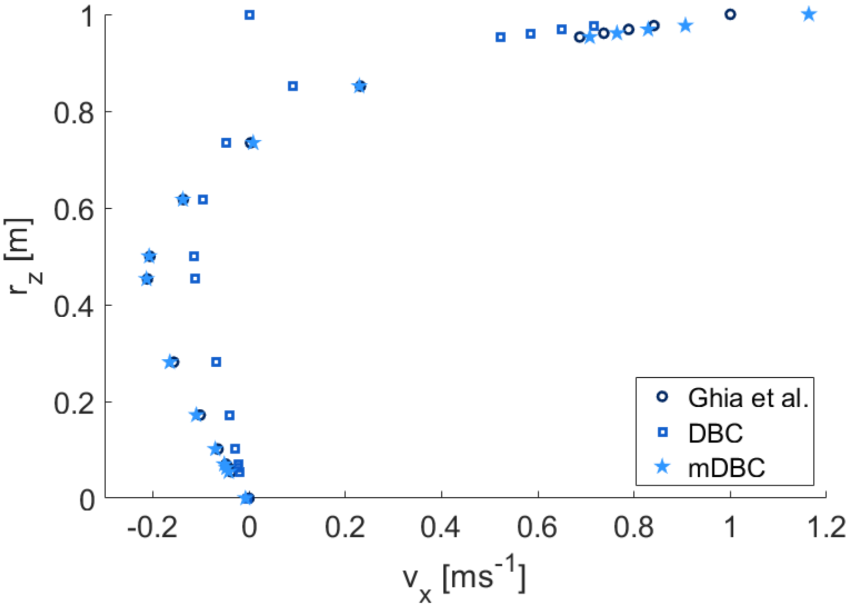


Figure 4.29: Horizontal velocity component along line through centre of cavity, comparing DBC and mDBC approaches, with resolution $dx = 0.01$ m, with benchmark solution of Ghia et al. [73].

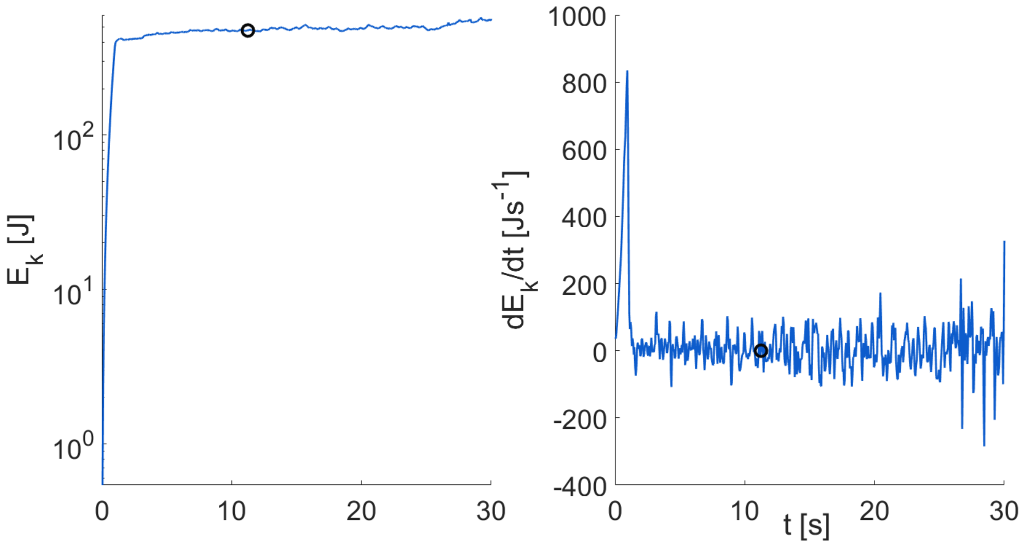


Figure 4.30: Energy (left) and change in energy (right) over time to determine steady state of lid-driven cavity case with mDBC. The point at which steady state is reached is marked with a black circle. $Re = 100, dx = 0.02$.

it completely. A fix was performed by reversing the component of velocity normal to the boundary of any fluid particles found beyond the line of the boundary. The unit normal for a particle i is computed as

$$\hat{\mathbf{n}}_i = \frac{\mathbf{r}_i - \mathbf{r}_C}{||\mathbf{r}_i - \mathbf{r}_C||}, \tag{4.20}$$

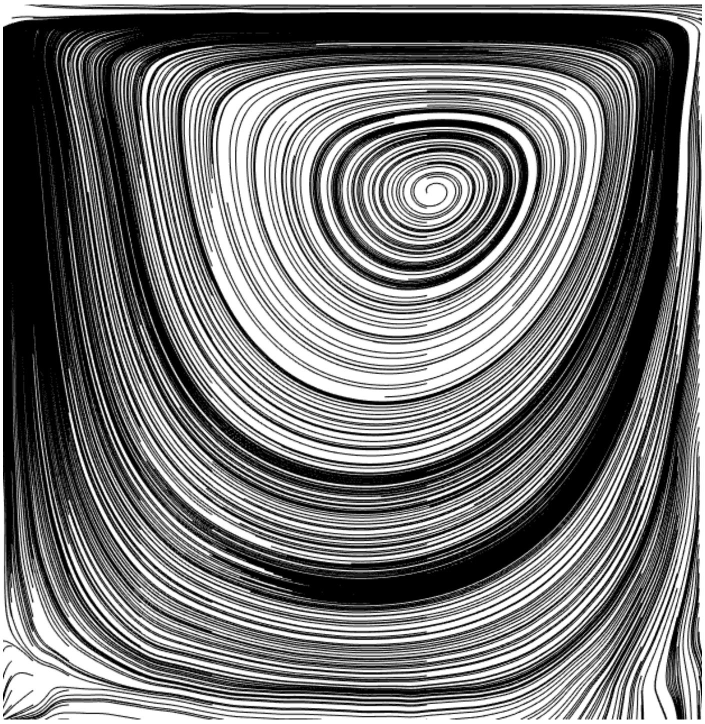


Figure 4.31: Streamlines produced by SPH with mDBC for $Re = 100$.

where $\mathbf{r}_C$ is the position of the corner. The velocity along the normal is then

$$\mathbf{v}_i^n = (\mathbf{v}_i \cdot \hat{\mathbf{n}}_i)\hat{\mathbf{n}}_i \tag{4.21}$$

and the updated velocity is

$$\tilde{\mathbf{v}}_i = \mathbf{v}_i - 2\mathbf{v}_i^n \tag{4.22}$$

Particle loss does not occur when using the release version of mDBC (DualSPHysics 5.0), which extrapolates only density into the boundary whilst using a different method to compute boundary velocity, as described in Section 3.3.4.2. This approach is used for the results shown in this section.

It can be seen in Figure 4.32 that increasing the resolution generates a velocity curve closer to the data of Ghia et al. [73]. To determine the convergence rate for this case, the norm

$$L^2(v_x) = \frac{1}{\max\{\tilde{v_x}\}N}\sqrt{\sum_{i=1}^{N}((v_x)_i - (\tilde{v_x})_i)^2}, \tag{4.23}$$

where $\tilde{v_x}$ is the result given by Ghia et al. [73], is taken over all data points. The error norm of the velocity in Figure 4.33 shows convergence with increasing resolution. These results confirm the ability of this SPH formulation to reproduce lid-driven cavity flow, which shall be further used for mixing of two phases in the cavity.

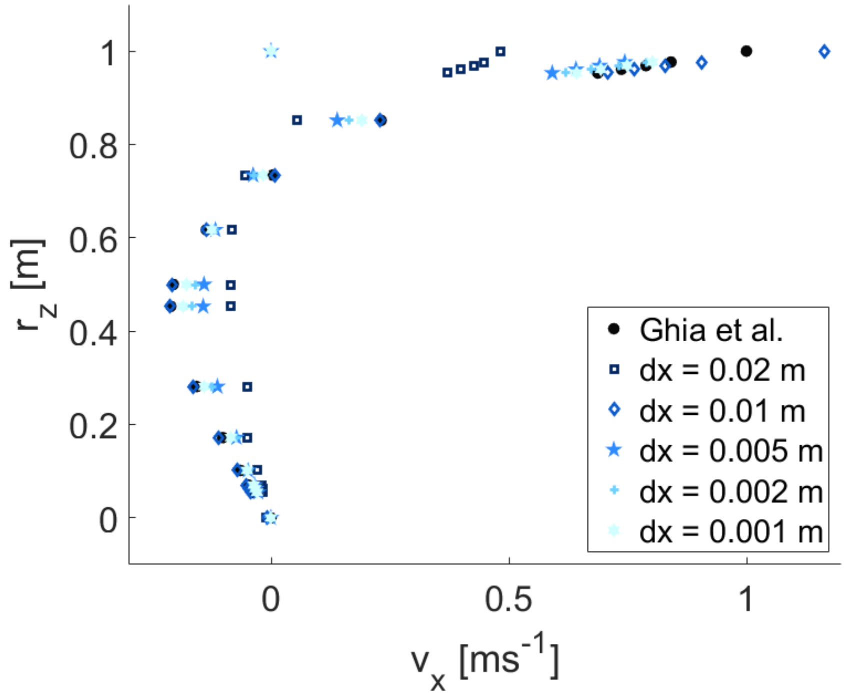


Figure 4.32: Velocity profiles along centre-line for multiple resolutions with mDBC for $Re = 100$.

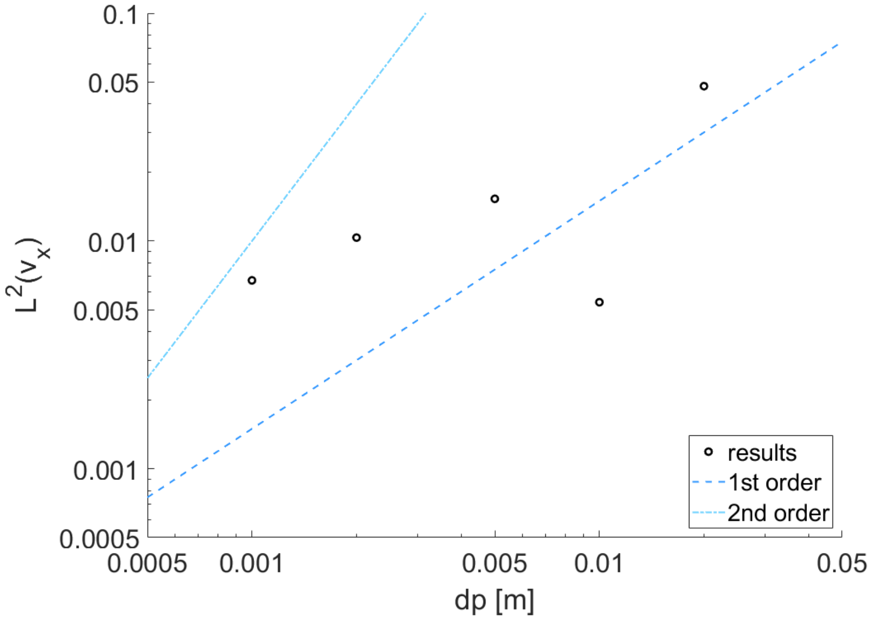


Figure 4.33: L^2 norm of centre-line velocity relative to Ghia et al. [73] results for increasing resolution, demonstrating the convergence rate of the method for the lid-driven cavity case.

4.3.3 Two Phases of Constant Viscosity

The cavity is split horizontally in half into two fluid phases of different constant viscosities with all other parameters the same, as shown in Figure 4.34. A simple multi-phase

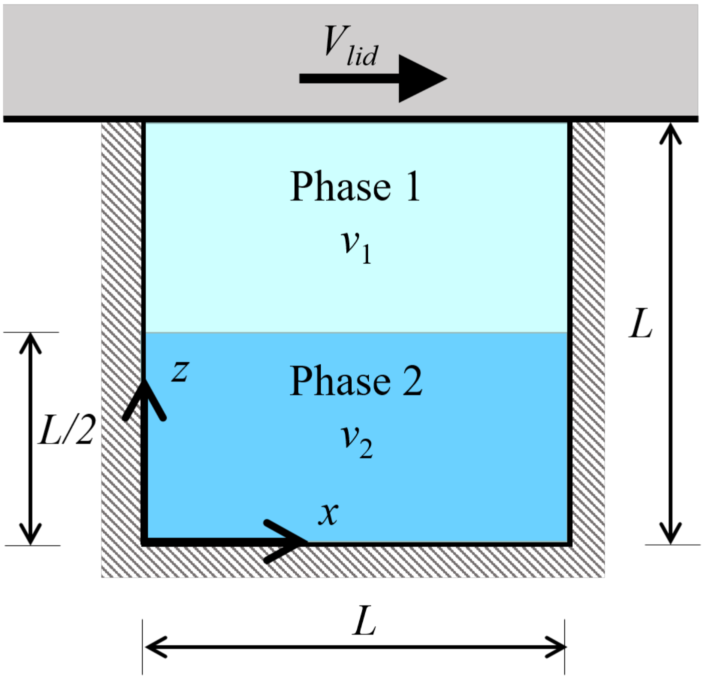


Figure 4.34: Lid-driven cavity with two fluid phases of different viscosity but constant density (adapted from [191]).

model such as this, with only viscosity differing, demonstrates the capacity of SPH to deal with large deformations and a highly deformed interface, before more complex physics is added in Section 4.4 and Chapter 5.

Flow is driven by the lid, as for the single phase case. Here the upper phase has viscosity $\nu_1 = 0.01$ m^2 s^{-1} ($Re = 100$), and the bottom phase $\nu_2 = 0.02$ m^2 s^{-1} ($Re = 50$). The average Reynolds number used in Section 4.2.2 is $Re = 67$. The cavity again has dimension $L = 1$ m with lid velocity $v_{\text{lid}} = 1$ m^2 s^{-1}. Since significant improvements were demonstrated by using mDBC in the previous Section 4.3.2.1, boundary conditions are enforced using the mDBC method for two-phase lid-driven cavity flows. The less viscous fluid moves with greater velocity, whilst the other is more resistant to movement. Once the lid movement is stopped, there is small movement of fluid particles but no separation of phases. This is because there is no difference in density between phases, only viscosity.

For this two phase lid-driven cavity case, since there is no literature for comparison, STAR-CCM+ [208] is used as a benchmark of the current modelling standard for comparison. For multi-phase cases, it uses the VOF method where each mesh cell has fluid properties weighted by the proportion of each phase within it. This results

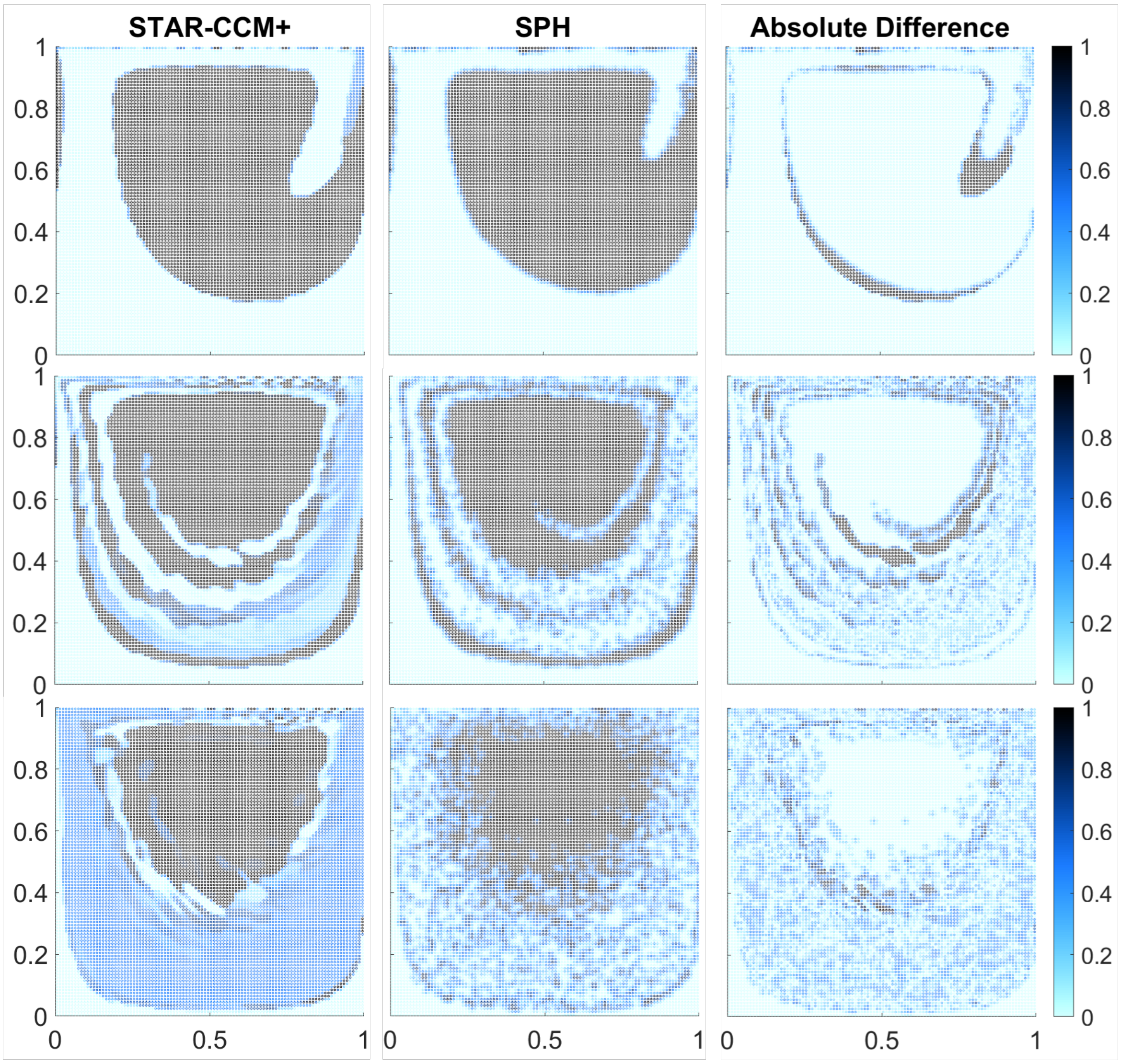


Figure 4.35: Volume fraction from STAR-CCM+, SPH and difference between the two at times 5 s (first row), 20 s (second row), 60 s (third row) for the two-phase lid-driven cavity.

in break up and diffusion of phases, in particular for higher Re, as the position of the interface becomes more difficult to determine. A related case was modelled using FEM by Camesasca et al. [29], involving two fluid phases in a square cavity driven by two moving walls. However, the objective of the work was to demonstrate a mixing measure, rather than accurately simulate the flow.

4.3.4 Mixing Measures

A natural comparison with a VOF method is to plot the volume fraction, calculated using Section 3.6.1. The results for the volume fraction are shown in Figure 4.35, coloured by the volume fraction of phase 1, F^1. Since there are two phases in this case, the volume fraction of phase 2 is $F^2 = 1 - F^1$ at any point. The third column shows the absolute difference in volume fraction between the two methods, where 0 means they are identical at that point. The two approaches produce many qualitative similarities, such as the clear line of phase 1 towards the bottom of the mixing and the shape of unmixed phase 2 after 60 s. Due to the volume fraction approach that does not produce a sharp interface between phases, it is not possible to get complete convergence of the STAR-CCM+ results at the interface, even with a high resolution. The break-up and diffusion of the interface can be seen here around $(x, z) = (0.9, 0.8)$ m at 5 s.

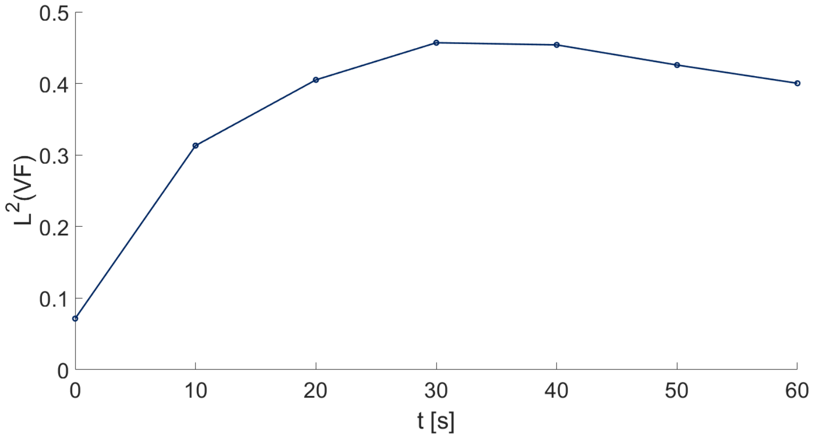


Figure 4.36: L^2 norm of volume fraction difference between SPH and STAR-CCM+ methods with time.

The volume fraction (VF) of the SPH method is compared to STAR-CCM+ quantitatively using the error norm Equation (4.23) in Figure 4.36. The difference initially increases as movement begins, then decreases more gradually after 30 s. This shows the methods tending towards convergence as time increase to 60 s. Therefore, the method developed for this work does not diverge from the benchmark solution as steady state is reached.

SPH is not derived to model individual molecules. Instead, each interpolation point (referred to as an SPH particle) represents a small volume in the continuum. Hence, in the context of the main application, the volume fraction alone cannot determine whether the High Level Waste (HLW) has been fully incorporated into the borosilicate glass, which is key to the vitrification process described in Section 2.2. This is more

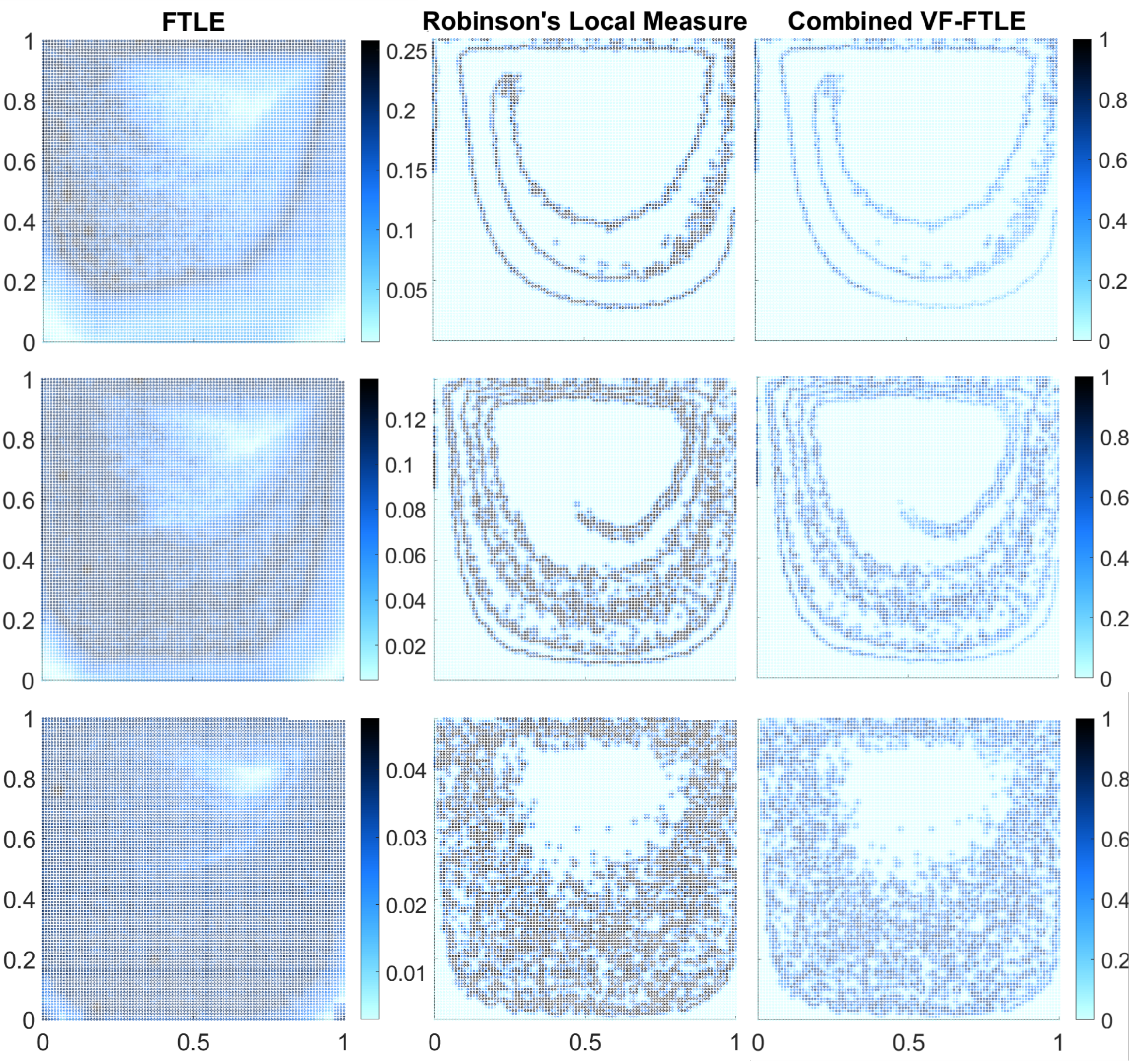


Figure 4.37: FTLE, Robinson's measure and new combined mixing measure at times 5 s (first row), 20 s (second row), 60 s (third row) for the two-phase lid-driven cavity.

likely to be successful if there is significant deformation within and between phases. One way of determining this is by using the finite-time Lyapunov exponent (FTLE), which quantifies the relative movement of SPH particles which start in a neighbourhood of each other.

Figure 4.37 shows the SPH mixing measure results at three different times for the FTLE, Robinson's local mixing measure and the new mixing measure developed in this work. The left-hand column in Figure 4.37 shows the FTLE where the scale is different for each time in order to make the features visible. The Lagrangian approximation for the forward-in-time FTLE in Equation (3.98) is used in post-processing. A greater value of FTLE indicates that SPH particles starting near this point have moved furthest away from each other at the presented time. This is useful for understanding the movement of particles but does not show the mixing between phases relevant for this work. However, it can be seen that little relative movement has occurred at the centre of the main vortex, as well as the smaller corner vertices, suggesting that little mixing between phases has taken place in these regions.

Robinson's mixing measure [188] (Equation (3.108)) in the centre of Figure 4.37 was designed to quantify mixing locally at a point in the domain. The areas of mixing, as well as the regions of unmixed phases, are clear to see. The insight from the FTLE figures of regions of low movement are confirmed to be poorly mixed here. The lid-driven cavity clearly does not produce a homogeneous mixture, at least with phase viscosities as here. This demonstrates that the lid-driven cavity is a poor mixing case, but shows the value of a mixing measure in determining the quality of mixing. For a greater degree of mixing to be obtained, a mechanical stirrer could be added to the middle of the tank to increase circulation through the unmixed regions.

The right hand column of Figure 4.37 shows the new combined VF-FTLE measure, defined in Equation (3.114), which illustrates both the mixing of phases and the movement of SPH particles. Here the volume fraction component must be calculated at SPH particle locations at $t = 0$ so that these values are at the same points as the FTLE values. This new measure is similar in distribution to Robinson's mixing measure, but appears to be smoother. The definition has a clear effect towards the bottom right of the cavity at 60 s, where VF is higher but FTLE is small. Therefore, both phases are present but there is little divergence between particles which begin within each-other's neighbourhood and the resulting mixing measure is low.

While the volume fraction gives information about the distribution of phases at a given time and the FTLE shows how much particles starting near a point have diverged from each other, both the Robinson and VF-FTLE measures display the amount of movement between particles of different phases. In particular, the new VF-FTLE measure can exhibit regions where either both phases are present but particles have remained close to those they were near at $t = 0$, or particles have moved considerably relative to each other but there has been no mixing between phases. For example, around $(x, z) = (0.9, 0.2)$ m at 20 s.

The mixing entropy (Equation (3.111)) is plotted for both methods in Figure 4.38.

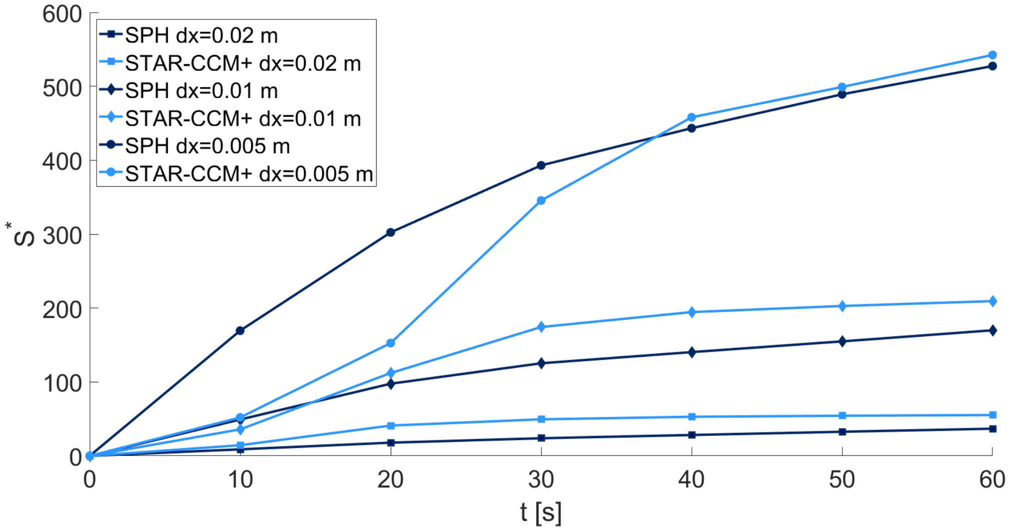


Figure 4.38: Mixing entropy of phase A over time for both methods with different resolutions.

For all resolutions, mixing entropy S^* increases with time, for STAR-CCM+ it increases initially at a different rate than for SPH, but both methods begin to converge after 40 s. For $dx = 0.02$ m especially, S^* appears to be converging towards a constant value as the maximum degree of mixing for the case is reached.

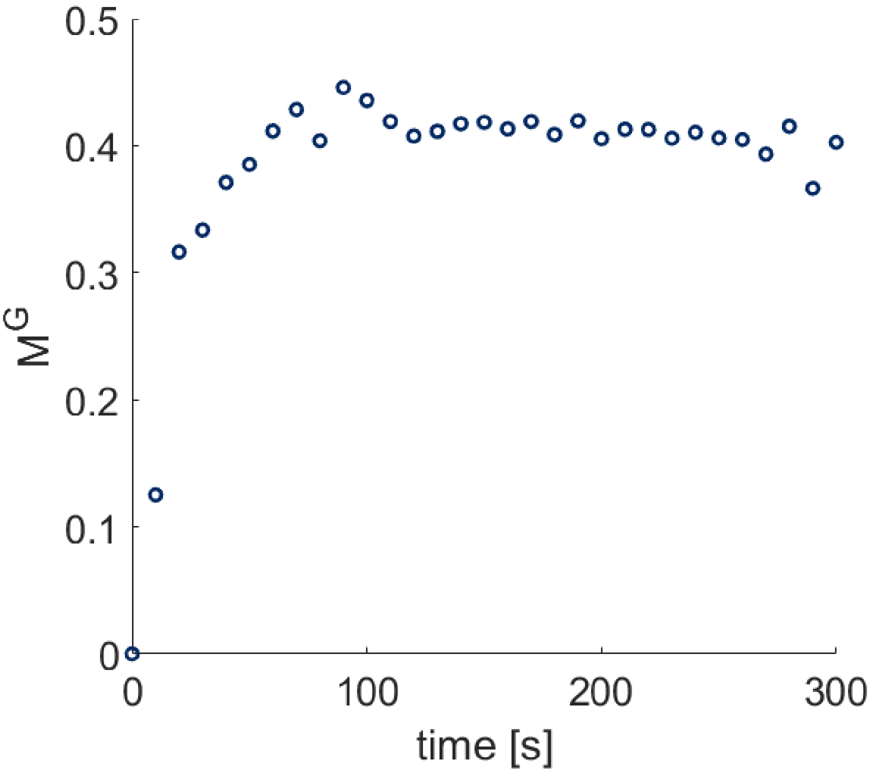


Figure 4.39: Global mixing measure for two-phase lid-driven cavity with time.

An investigation into how the mixing measures change with time was undertaken using the global mixing measure in Equation (3.115), since it gives the most information in terms of overall mixing. In this way, a global assessment of the mixing is obtained. Figure 4.39 shows the evolution of the global mixing measure. At the point the global mixing measure stops changing a steady state has been reached, demonstrating another use for the new measure. The measure is seen to plateau around 120 s, when the maximum amount of mixing has been achieved. This does not, however, mean that the phases are completely mixed. Due to the contribution of the FTLE, the maximum value will only be 1 if all particles have moved the same distance from their original neighbours. In reality, this is almost impossible to happen. For example, the lower M^G at 290 s is due to a lower maximum M at this time.

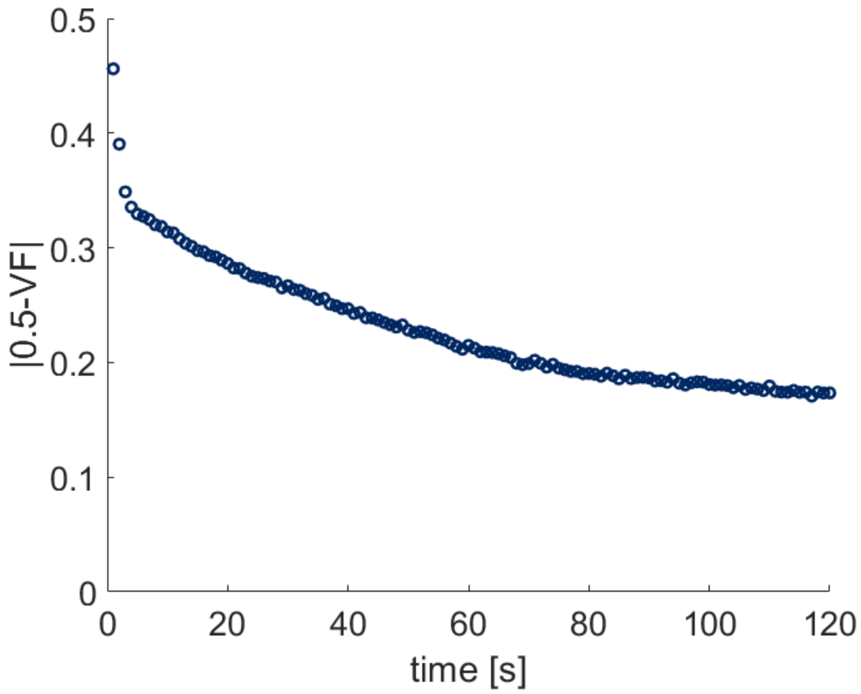


Figure 4.40: Global volume fraction for two-phase lid-driven cavity with time.

A global measure based on the volume fraction is also defined in Equation (3.96) to quantify the degree of mixing at a given time and is shown in Figure 4.40. This measure decreases with time as mixing increases and becomes close to the volume fraction of the whole domain. As time approaches 120 s the measure tends towards a constant value, agreeing with the global mixing measure in Figure 4.39 that the maximum level of mixing is achieved around this time. As seen in Figure 4.37, regions of unmixed phases remain for this case and so at steady state the average difference from global volume fraction is significantly more than zero. These global comparisons, along with the single phase results in Section 4.3.2.1, demonstrate that this SPH model is a good candidate for flow within a cavity generating mixing of viscous fluids.

4.4 Differentially Heated Cavity

The differentially heated cavity case is run to test the addition of the Boussinesq term in the momentum equation, as in Section 3.2.3, as well as a governing equation for temperature, as in Section 3.3.2.3. This demonstrates that natural convection and circulation driven by buoyancy can be modelled successfully.

4.4.1 Heat Conduction Validation

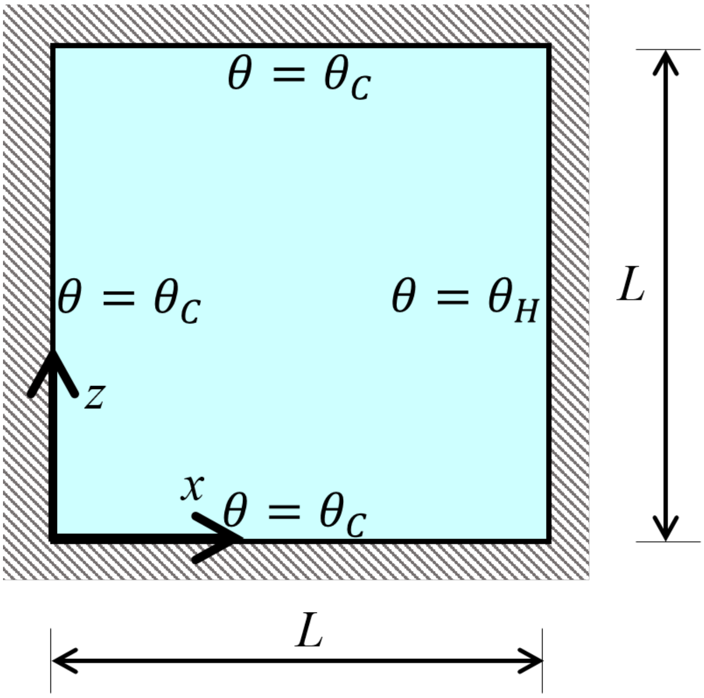


Figure 4.41: Diagram of cavity for heat conduction case.

Firstly, the ability of the added governing equation for temperature (Equation (3.40)) to produce a temperature gradient is demonstrated. The square cavity of width $L = 1$ m in Figure 4.41 is filled by a fluid with constant viscosity. The fluid is discretised with particle spacing $dx = 0.02$ m, giving 50 particles across the cavity. The Boussinesq approximation is not included and gravity is set to zero as $\mathbf{g} = (0, 0)$ $\mathrm{kg\,m\,s^{-2}}$, so the fluid remains approximately stationary. The speed of sound is then manually set to $c = 10\ \mathrm{m\,s^{-1}}$. The right boundary has constant temperature $\theta_H = 100$ K and the other 3 boundaries are at $\theta_C = 0$ K constant. Initially, the fluid has a uniform temperature of θ_C with thermal parameters $C_p = 100\ \mathrm{J\,kg^{-1}\,K^{-1}}$, $k = 2\ \mathrm{W\,m^{-1}\,K^{-1}}$, and has physical parameters of water $\overline{\rho} = 1000\ \mathrm{kg\,m^{-3}}$, $\nu = 10^{-6}\ \mathrm{m^2\,s^{-1}}$. Its temperature is then governed by Equation (3.40). Boundary particles are given the same parameter values, but have constant temperature. Density is extrapolated into the boundary using the mDBC method. The resulting temperature field is shown in Figure 4.42 and Figure 4.43.

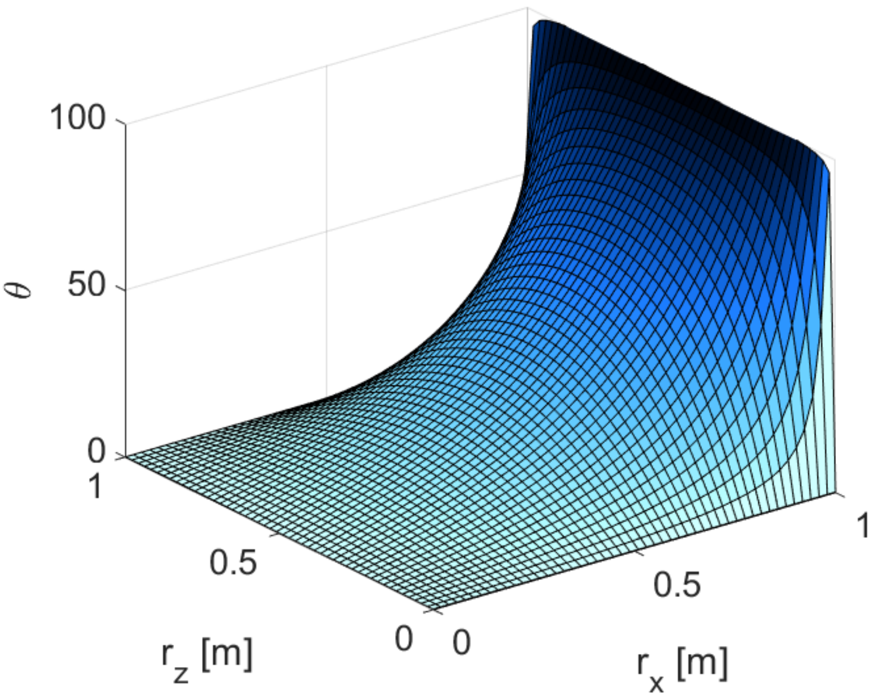


Figure 4.42: Surface of heat conduction case temperature.

An analytical solution is found from the two-dimensional diffusion equation for temperature, Equation (3.12),

$$\frac{\partial \theta}{\partial t} = \frac{k}{C_p \rho} \nabla^2 \theta = \frac{k}{C_p \rho} \left(\frac{\partial^2 \theta}{\partial x^2} + \frac{\partial^2 \theta}{\partial z^2} \right) \tag{4.24}$$

on the domain $x \in [0, L]$, $z \in [0, L]$, with inhomogeneous boundary conditions

$$\theta(0, z, t) = 0;$$

$$\theta(L, z, t) = \theta_H = 100;$$

$$\theta(x, 0, t) = \theta(x, L, t) = \theta_C = 0.$$

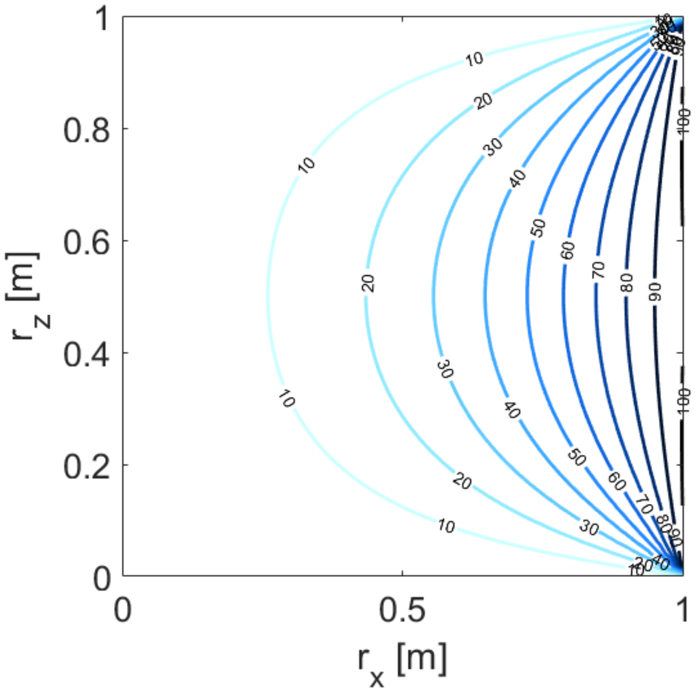


Figure 4.43: Contours of heat conduction case temperature.

Looking for steady state solution for this Dirichlet problem by separation of variables, the global solution is

$$\theta(x,z) = \sum_{n=1}^{\infty} \frac{2\theta_H}{n\pi \sinh(n\pi)} (1-(-1)^n) \sin\left(\frac{n\pi}{L}z\right) \sinh\left(\frac{n\pi}{L}x\right). \tag{4.25}$$

Since for even n, $(1-(-1)^n) = 0$; and for odd n, $(1-(-1)^n) = 2$ the solution can be rewritten as

$$\theta(x,z) = \frac{4\theta_H}{\pi} \sum_{n=1}^{\infty} \frac{1}{(2n-1)\sinh((2n-1)\pi)} \sin\left(\frac{(2n-1)\pi}{L}z\right) \sinh\left(\frac{(2n-1)\pi}{L}x\right). \tag{4.26}$$

This is particularly useful numerically where the solution must be approximated as a finite sum. When computing the solution numerically, and since $\sinh(x)$ is not bounded, a finite sum must be used where N approximates infinity. $N = 100$ was found to be adequate in MATLAB.

It can be seen in Figure 4.44 that the results of this SPH model compare well with the analytical solution found. The error is quantified as

$$L^2(\theta) = \sqrt{\frac{\sum(\theta_{\text{analytical}} - \theta_{\text{SPH}})^2}{\sum \theta^2_{\text{analytical}}}} = 0.102, \tag{4.27}$$

and is approximately constant throughout the fluid domain.

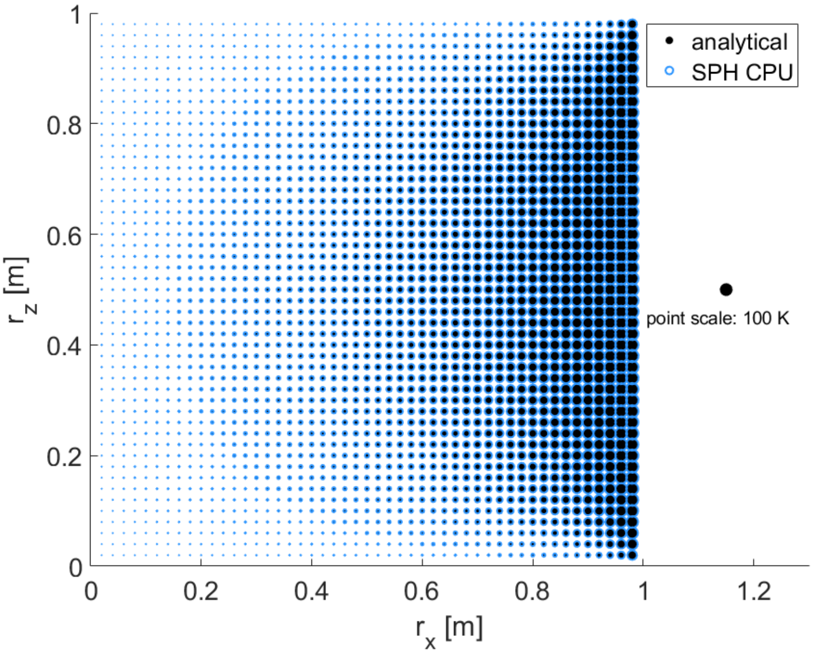


Figure 4.44: Comparison of SPH results with analytical solution (Equation (4.26)) in cavity for heat conduction validation. Temperature is represented by size of circle.

4.4.2 Differentially Heated Cavity

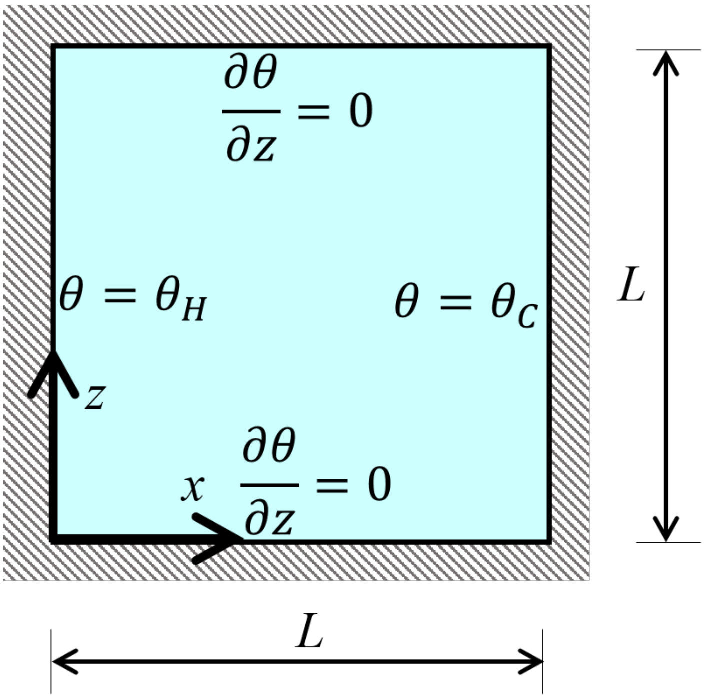

$\frac{\partial \theta}{\partial z} = 0$
$\theta = \theta_H$
$\theta = \theta_C$
L
z
x
$\frac{\partial \theta}{\partial z} = 0$
L

Figure 4.45: Diagram of differentially heated cavity flow.

For buoyancy-driven flow within the cavity, the governing equation for energy

(Equation (3.40)) is employed, along with the Boussinesq approximation in the conservation of momentum (Equation (3.38)). The key non-dimensional parameters for this case are the Prandtl number

$$Pr = \frac{\nu \rho C_p}{k}, \tag{4.28}$$

which gives a ratio of viscous to thermal diffusion, and the Rayleigh number

$$Ra = \frac{g \beta L^3 (\theta_H - \theta_C) C_p}{\nu k}, \tag{4.29}$$

which characterises natural convection in the flow. A higher Rayleigh number indicates that buoyancy forces are more important to the flow than viscous forces.

The geometry is again a two-dimensional square cavity, but with boundary conditions as in Figure 4.45. The literature [49, 111, 244] has the fluid filling the cavity as air at room temperature, with $Pr = 0.71$, and Ra ranging from 10^3 to 10^8 (all within the region of laminar flow [244]). In this work, the fluid is taken to be air at $\theta = 20\,^{\circ}\mathrm{C}$ and $Ra = 10^5$ chosen for comparison with data in the literature. The values used can be seen in Table 4.7. Following Barakos et al. [18], the temperature difference is fixed whilst the cavity width is varied to set Ra. To ensure that the temperature difference between boundaries is large enough to have a significant effect on the flow, and any velocities are orders of magnitude larger than the size of errors, it is chosen to be $\theta_H - \theta_C = 20$ K with $\theta_H = 303$ K. This means that the cavity width must be $L = 0.0386$ m to maintain chosen characteristic values. This case has a Reynolds number of approximately 1.5×10^5.

Fluid is initially stationary and starts with uniform temperature $\theta(t = 0) = 293$ K, which is also set as the reference temperature ($\overline{\theta}$) in the Boussinesq term. Unlike the previous case in Section 4.4.1, a realistic gravitational acceleration is included in the Boussinesq term (Equation (3.8)).

Boundary conditions for density and velocity are imposed using mDBC, as described in Section 3.3.4.2. The boundaries are given the same parameters as the fluid in Table 4.7, but are stationary and not governed by Equation (3.40) so do not need a value for thermal expansion coefficient, β. Instead, left and right boundaries are held at constant temperatures θ_H and θ_C respectively by assigning corresponding boundary particles these temperatures. Top and bottom boundaries are adiabatic, that is

parameter	value
ν	$1.506 \times 10^{-5}\ \mathrm{m^2\,s^{-1}}$
ρ	$1.204\ \mathrm{kg\,m^{-3}}$
C_p	$1.006 \times 10^{3}\ \mathrm{J\,kg^{-1}\,K^{-1}}$
k	$0.02587\ \mathrm{W\,m^{-1}\,K^{-1}}$
β	$3.43 \times 10^{-3}\ \mathrm{K^{-1}}$
g	$-9.81\ \mathrm{m\,s^{-2}}$

Table 4.7: Parameter values for differentially heated cavity flow.

$\frac{\partial \theta}{\partial z} = 0$, and are given the same initial temperature as the fluid. This is first achieved by setting $k = 0\ \mathrm{W\,m^{-1}\,K^{-1}}$, although the higher order approach is used later in this section. All four boundaries have no-slip conditions through $\mathbf{v} = (0, 0)$. Following the timestep constraint of Wan et al. [244],

$$dt \leq \min\left\{ \frac{4Pr}{(|v_x| + |v_z|)^2}, \frac{Pr(dx)^2}{2} \right\} = 1.42 \times 10^{-6}, \tag{4.30}$$

the initial timestep size is chosen to be $dt = 1.5 \times 10^{-6}$ s.

h	$2dx$
c	$12\lvert\mathbf{v}\rvert_{\max}$
CFL number	0.1
Viscous term	Monaghan and Gingold [155] operator with arithmetic mean
Particle shifting type	Full
Shifting coefficient	-2
Density diffusion type	None

Table 4.8: Simulation parameters for successful differentially heated cavity case.

A kernel smoothing length of $h = 2dx$ was found to be required for flow to remain in motion at steady state, demonstrating that the number of neighbouring particles is key to running this case successfully. For smaller h, numerical dissipation dominates and results in the temperature profile diffusing. This results in a reduction of the particle velocities such that fluid stops moving, and all heat transfer is by conduction rather than convection. Simulation parameters used for the successful differentially heated cavity cases are summarised in Table 4.8. Contributions to numerical dissipation come from the timestep, shifting (all particle types with coefficient -2) and the viscous term (no density diffusion is applied). The timestep constraints for viscosity and temperature (Equations (3.63) and (3.66)) are both included and the simulation is run with a Courant-Friedrichs-Lewy (CFL) number of 0.1 and speed of sound $c = 12|\mathbf{v}|_{\max}$. The viscous term is approximated with the Monaghan and Gingold operator, using the arithmetic mean in Equation (3.74).

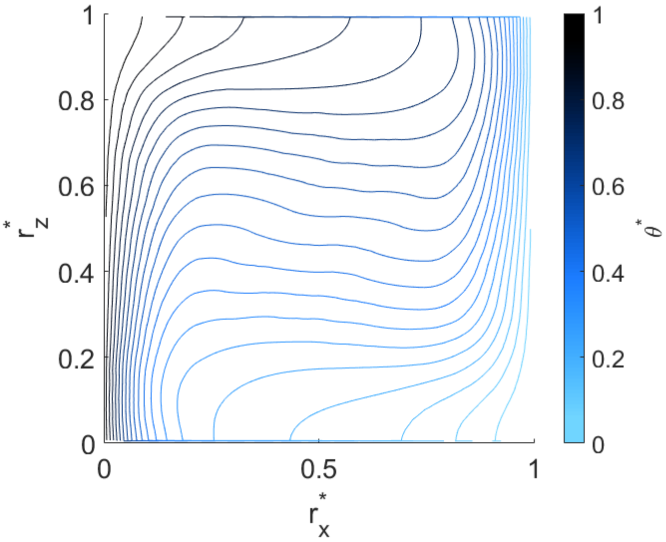


Figure 4.46: Isotherms at $t^* = 0.5$ for differentially heated cavity at $Ra = 10^5$ with new mDBC adiabatic boundary condition.

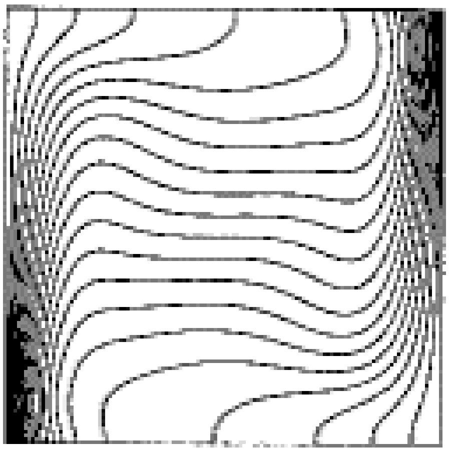


Figure 4.47: Isotherms at $Ra = 10^5$ from Wan et al. [244].

The adiabatic boundary condition can be approximated by setting $k = 0\,\mathrm{W\,m^{-1}\,K^{-1}}$ of the corresponding boundary particles so they do not contribute to the temperature equation of neighbouring fluid particles. However, this approach introduces additional truncation error, since the kernels of fluid particles near the boundary are incomplete when computing a change in temperature. This error means the approach has a low order of accuracy and contributes to the decaying solution. The mDBC presented earlier in Section 3.3.4.4 is modified for the $\frac{\partial \theta}{\partial z}$ boundary, using the same approach as employed for density, to give Equation (3.50). Heat transfer is blocked through the boundary by reflecting values for temperature onto the boundary particles. This is seen to be effective in Figure 4.46, which compares well to the isotherms found by Wan et al. [244] in Figure 4.47. Isotherms are normal to the adiabatic boundaries as they approach, since there is no change in temperature through them. mDBC is also used here for density and velocity, since it provided a significant improvement to the lid-driven cavity case in Section 4.3.2.1.

Results are compared with those of Wan et al. [244], who use a high order discrete singular convolution scheme, for $Ra = 10^5$ at the highest resolution (161×161 mesh points). The SPH model is first run with a particle spacing of $dx = 0.0005$ m, which means there are approximately 77×77 fluid particles. Non-dimensionalised variables, denoted by *, are used for comparison and computed through [244]

$$\begin{aligned}
\mathbf{r}^* &= \frac{\mathbf{r}}{L};\\
\mathbf{v}^* &= \frac{\mathbf{v}L\rho C_P}{k};\\
\theta^* &= \frac{\theta-\theta_C}{\theta_H-\theta_C};\\
t^* &= \frac{tk}{L^2\rho C_p}.
\end{aligned}$$

For $L = 0.0386$ m, the horizontal line through the centre of the cavity lies at $r_z = \frac{L}{2} = 0.0193$ m. Values for temperature and vertical velocity are computed at equally spaced points along this line using the SPH interpolation in Equation (3.22) of output data. Measurements are taken at time $t^* = 0.5$, which corresponds to $t = 34.88$ s, when Wan et al. [244] found the flow has reached a steady state.

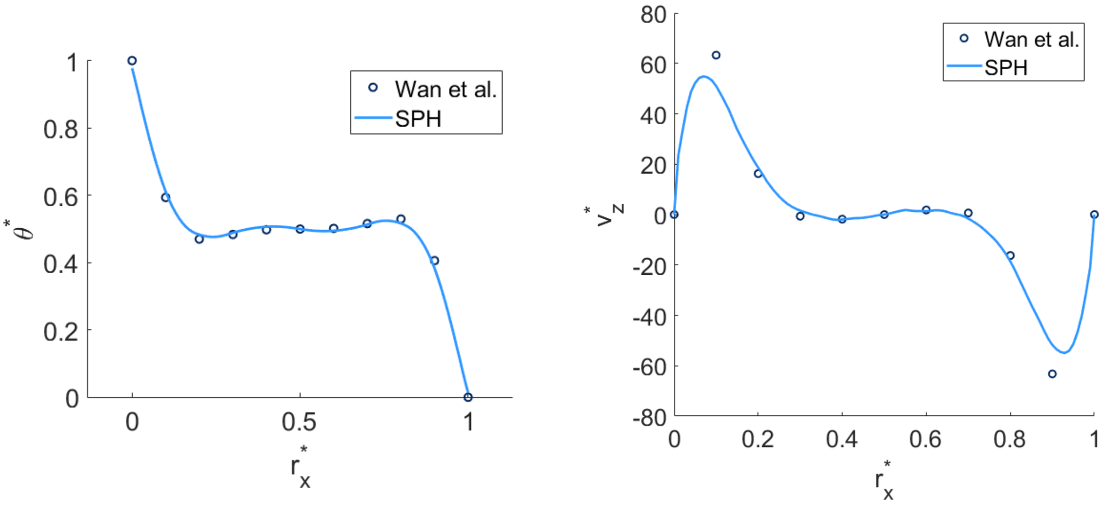


Figure 4.48: Temperature (θ^*) and vertical velocity (v_z^*) along horizontal mid-line for $dx = 0.00025$ m at $t^* = 0.5$.

Results for temperature and vertical velocity component can be seen in Figure 4.48 respectively, interpolated from output data and plotted with the benchmark data of Wan et al. [244]. Due to the different boundary approaches for temperature and velocity, when plotting results boundary particles are included in the SPH summation for temperature, but only fluid particles are included (with a kernel correction) when computing velocity. The condition for no-slip boundaries is achieved in the current version of mDBC by setting $\mathbf{v} = 0 \text{ m s}^{-1}$ for boundary particles. The same approach is used for no-slip DBC and is known to produce poor velocity gradients near walls.

This results in non-zero velocity at the wall when boundary particles are included in the summation, and may contribute to the decrease in flow velocity found for smaller smoothing lengths.

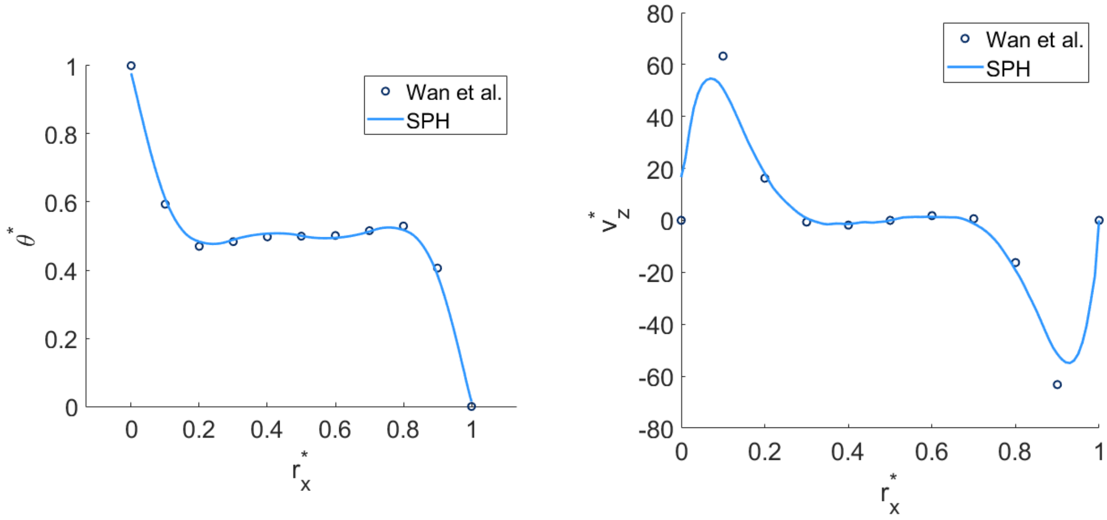


Figure 4.49: Temperature (θ^*) and vertical velocity (v_z^*) along horizontal mid-line for $dx = 0.00025$ m at $t^* = 0.5$ with '$k = 0$' adiabatic boundary condition.

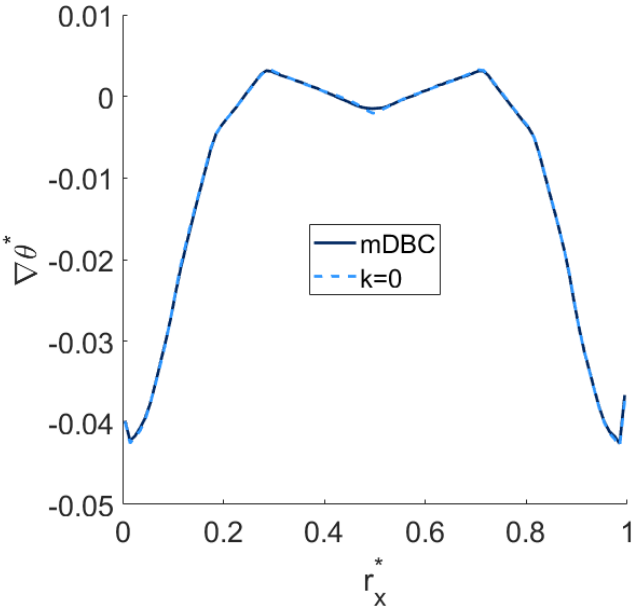


Figure 4.50: Gradient of temperature along horizontal mid-line for $dx = 0.00025$ m at $t^* = 0.5$, with both mDBC and '$k = 0$' adiabatic boundary conditions.

The more simple '$k = 0$' adiabatic boundary method reproduces both temperature and density well in Figure 4.49, demonstrating that the shortcomings of the '$k = 0$' approach do not affect flow in the centre of the cavity. However, these figures do not

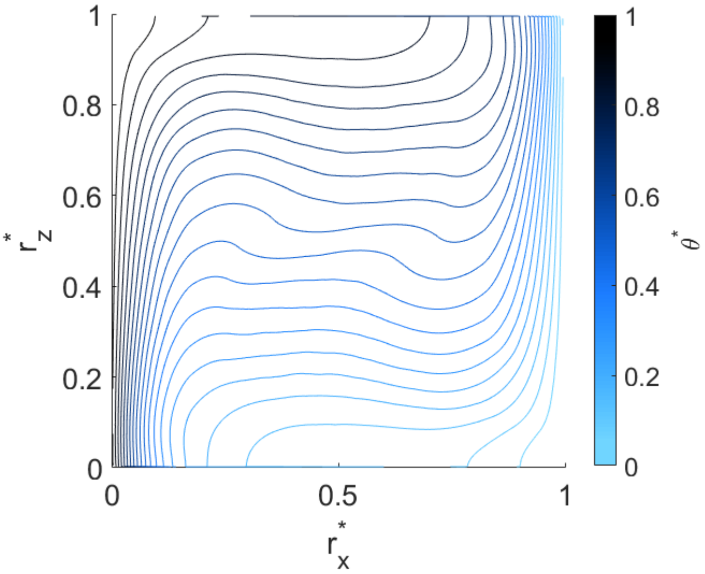


Figure 4.51: Isotherms at $t^* = 0.5$ for differentially heated cavity with '$k = 0$' adiabatic boundary condition.

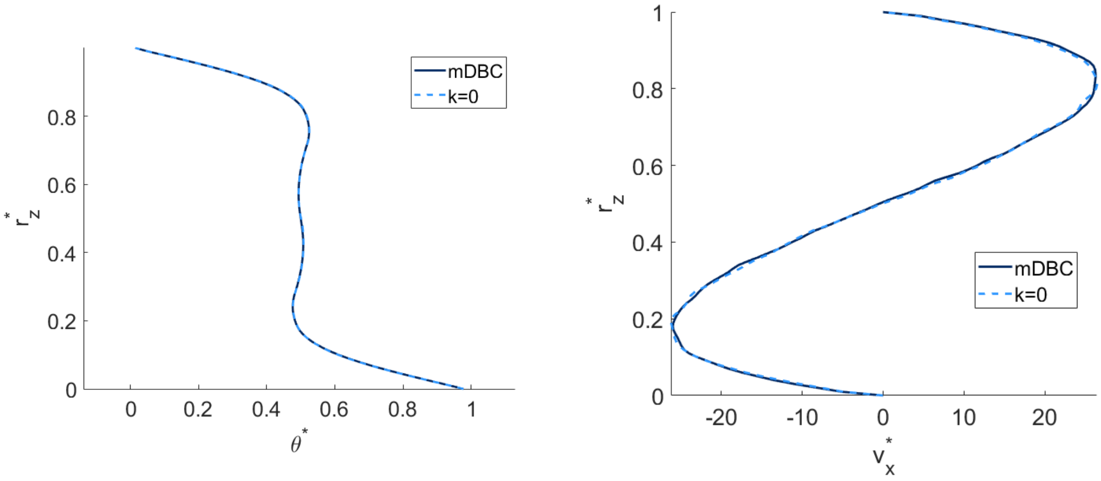


Figure 4.52: Temperature (θ^*) and horizontal velocity (v_x^*) along vertical mid-line for $dx = 0.00025$ m at $t^* = 0.5$.

validate the flow in proximity to the adiabatic boundaries. The temperature gradient across the centre of the cavity in Figure 4.50 is close for both methods, and the isotherms in Figure 4.51 are also close to the benchmark solution of Figure 4.47. However, immediately next to the adiabatic boundaries at the top and bottom, the isotherms are not as close to perpendicular to the line of the boundary as for the mDBC method in Figure 4.46. In addition, isotherms are straighter than the benchmark solution near the top and bottom boundaries but less straight in the vicinity of the centre of the cavity. The temperature and velocity along a vertical centre line are plotted in Figure 4.52 in order to compare adiabatic boundary methods. This figure demonstrates that the differences between the simple '$k = 0$' approach and the higher order mDBC method are minimal for temperature gradient in the bulk of the fluid. Overall, there is little difference in results from the choice of adiabatic boundary method for this case, despite the mDBC method achieving a greater order of accuracy.

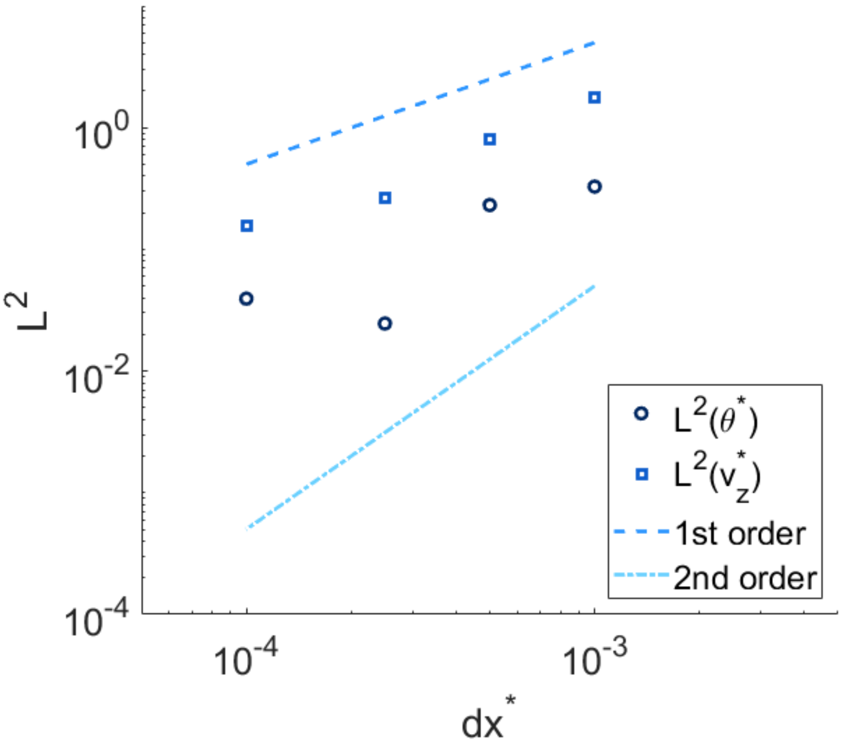


Figure 4.53: Convergence of θ^* and v_z^* with increasing resolution, for $h = 2.5dx$.

A convergence study is undertaken with fixed $h = 2.5dx$ for varying resolution by changing particle spacing dx. Error is computed through the norms over N data points

$$L^2(\theta^*) = \sqrt{\frac{\sum_{i=1}^{N}(\theta^*_{\text{SPH}} - \theta^*_{\text{lit}})^2}{\sum_{i=1}^{N}(\theta^*_{\text{lit}})^2}} \tag{4.31}$$

$$L^2(v_z^*) = \sqrt{\frac{\sum_{i=1}^{N}((v_z^*)_{\text{SPH}} - (v_z^*)_{\text{lit}})^2}{\sum_{i=1}^{N}(v_z^*)^2_{\text{lit}}}} \tag{4.32}$$

where subscript 'lit' denotes the value in the literature. A larger smoothing length is chosen than previously to allow the simulation to run successfully at the coarsest resolution. However, the flow becomes stationary before reaching a steady state for this largest particle spacing. Figure 4.53 shows that the velocity is converging at between first and second order before the rate of convergence begins to decrease. Since the ratio $\frac{h}{dx}$ remains constant for all resolutions, a plateau is found as resolution is increased as the limiting smoothing error is reached [183].

Although the size of the error in temperature is smaller than for velocity, due to the no-slip boundary method, the limit of convergence is reached at a larger particle spacing and the error increases with the finest resolution. The limit is reached sooner since the computation of temperature involves the approximation of a second order gradient, which introduces an extra error. Wan et al. [244] also found that temperature converged faster with time than velocity, in agreement with these results. A coarsest resolution of $dx = 0.0005$ m was found for circulation to develop successfully and be sustained, without dissipation dominating the flow.

4.4.2.1 Three-Dimensional Validation

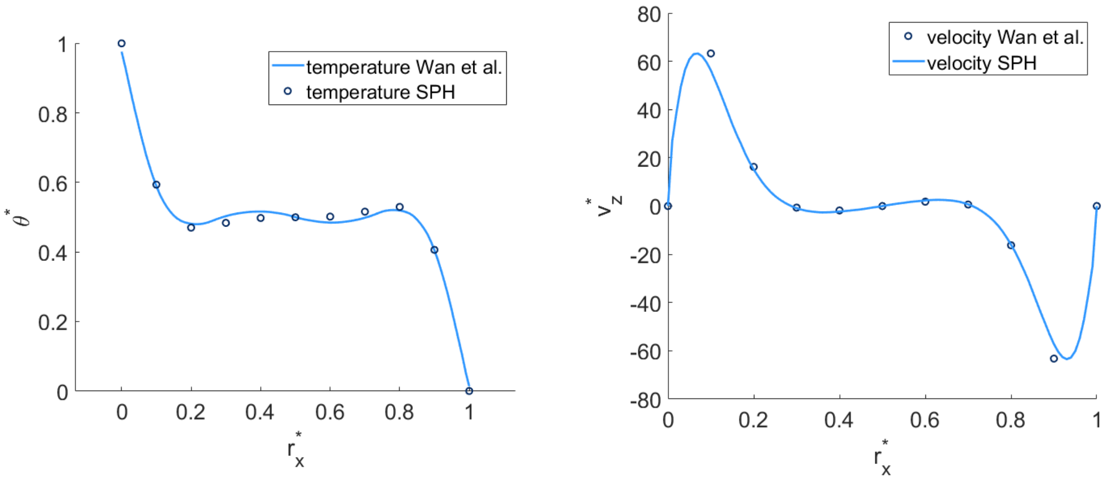


Figure 4.54: Temperature (θ^*, left) and vertical velocity component (v_z^*, right) plots along a horizontal line through the centre of the cavity for the quasi-2D case, compared with the results of Wan et al. [244]. CFL number 0.1, $h = 2.0dx$, 10 particles in periodic dimension.

dimensions	runtime [s]
2	32330
3	443700

Table 4.9: Runtime for two or 3 dimensions (to 4 significant figures) with CFL number 0.1, smoothing length $h = 2.0dx$, particle spacing 0.00025 m for 60 s.

A quasi-2D case is run by extending the domain into the third dimension with periodic boundaries, so that any slice of the $x-z$ plane is equivalent to the previous two dimensional cases. A finer resolution of $dx = 0.00025$ m is used, since the simulation is more sensitive to numerical dissipation with the extra dimension included and does not run successfully for $dx = 0.0005$ m. The case is run as previously, but with 10 particles in the $y-$dimension. This is the minimum required for no kernel to overlap itself through the periodic boundary. The results in Figure 4.54 show the comparison of temperature and velocity with the benchmark literature data.

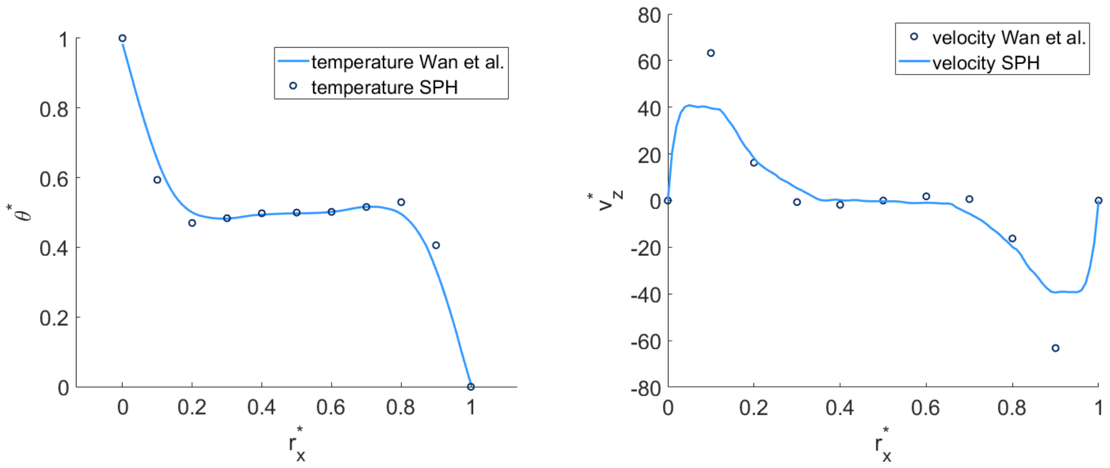


Figure 4.55: Temperature (θ^*, left) and vertical velocity component (v_z^*, right) plots along a horizontal line through the centre of the cavity for the quasi-2D case, compared with the results of Wan et al. [244]. CFL number 0.2, $h = 1.5dx$, 20 particles in periodic dimension.

At least two kernel supports are included across the lateral width of the domain in order to mitigate against periodic effects. Increasing the depth in the third dimension to 20 SPH particles significantly increases runtime by more than 13 times, as can be seen for an example case in Table 4.9. In order to mitigate this, the smoothing length is set to $h = 1.5dx$ to maintain approximately the same number of neighbours as the 2D case, and the CFL number is increased from 0.1 to 0.2. The results in Figure 4.55 are poorer because of these compromises, but this demonstrates the capability of running the case with a smaller smoothing length as a result of the extra dimension increasing particle numbers within the kernel. It is possible that the difference is also due to 3D effects, which accumulate with time. Although results are close for this smaller smoothing length, the exponential part of the temperature curve is not captured.

Thus, it is clear that a large smoothing length, h, is required in three-dimensions, as

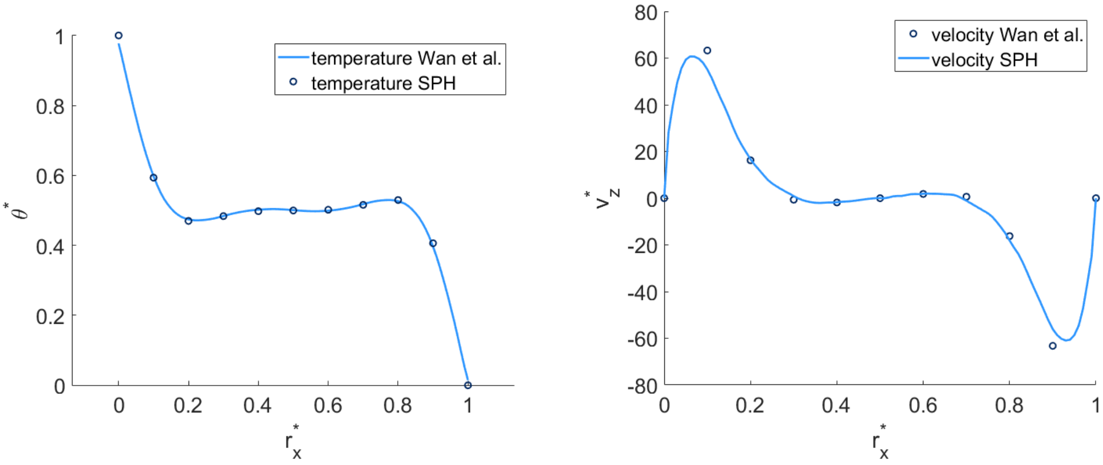


Figure 4.56: Temperature (θ^*, left) and vertical velocity component (v_z^*, right) plots along a horizontal line through the centre of the cavity for the quasi-2D case, compared with the results of Wan et al. [244]. CFL number 0.1, $h = 2.0dx$, 20 particles in periodic dimension.

well as two, in order to get accurate results. The smoothing length $h = 2.0dx$, as was successful for two dimensions, is chosen, but the domain is extended to 20 particles in the periodic dimension. This larger smoothing length gives significantly better results, in Figure 4.56, than the previous h. Increasing the number of neighbours in this way is shown to improve results and get very close to the benchmark solution of Wan et al. [244]. To reduce runtime without impacting accuracy, it may be possible to increase the CFL number, since the simulation is stable.

This case demonstrates the capability of DualSPHysics with the modifications made for this project to run three-dimensional cases, and take advantage of the computational capacity of the graphics processing unit (GPU) code and hardware for these simulations. The advantages of running on a GPU are described in more detail in the next section.

4.5 GPU Profiling and Speed-up

An investigation of the speed-up provided by the GPU capability of the model is investigated, before undertaking a basic profiling of the new and modified Compute Unified Device Architecture (CUDA) kernels to compare with existing kernels.

Table 4.10 shows that for the three-dimensional differentially heated cavity cases in Section 4.4.2.1, using the GPU decreases runtime by over 45 times on the specified

	CPU	GPU
hardware	Intel Core i7-9750H (2.60 GHz)	NVidia GeForce RTX 2080
CPU memory	217.49 MB	83.02 MB
GPU memory	-	208.46 MB
runtime per physical sec.	349277 s	7655 s

Table 4.10: Comparison of CPU and GPU computation for three-dimensional differentially heated cavity case with $dx = 0.00025$ m (537920 particles).

hardware. The CPU used has 6 cores with 12 threads, which are taken advantage of by parallelisation using OpenMP. Meanwhile, CUDA is used to run the code on GPUs, as discussed in Section 3.4. Exactly the same method is implemented for the GPU code as for the CPU code, with all additions and modifications presented in Chapter 3 included in both. There was no difference found in L^2 norm between results from CPU and GPU sides of the code for the temperature-dependent Poiseuille flow in Section 4.2.3 or the heat conduction validation in Section 4.4.1.

kernel	total time (%)	call count	average time (μs)	occupancy (%)	registers per thread
fluid particle interactions	16.32	9151	9185.170	0.00	75
boundary particle interactions	0.58	9151	327.026	0.00	56
compute viscosity	0.02	9151	10.926	0.00	24
compute temperature	0.01	9151	6.724	0.00	16
adiabatic mDBC calculation	17.32	9151	9746.548	0.00	78
fluid particle interactions (original)	21.61	9875	7384.972	0.00	66
mDBC calculation	14.18	4576	15957.892	0.00	100

Table 4.11: GPU profiling metrics for newly implemented CUDA kernels, with existing kernels under the line for comparison.

For the same case, GPU profiling of the new CUDA kernels is undertaken and results shown in Table 4.11. Pre-existing kernels are included for comparison at the bottom of the table. The modified viscous term (Equation (3.70)) and temperature evolution (Equation (3.40)) are computed within the fluid particle interactions kernel. In Table 4.11 it is seen that these additions increase both time and memory requirements for this kernel. However, the percentage of total time is lower. The functions to compute viscosity, through any chosen method from Section 3.5.1, and temperature, where it is not dependent on the temperature evolution equation, make only very small increases to the total time taken. Since the adiabatic mDBC method does not require a gradient to be calculated to compute boundary values, as in Equation (3.50), unlike for density in the original mDBC method in Equation (3.46) the time taken for the adiabatic kernel is shorter than for the mDBC kernel.

Although this work does not focus on code optimisation, the newly implemented CUDA kernels have comparable profiling metrics with existing kernels in DualSPHysics.

4.6 Concluding Remarks

Each of the sections in this chapter exhibits a case which demonstrates an aspect of the physics key to the vitrification model in Chapter 5. In this way, the results shown validate the model which has been developed for this project and prove that the model produces good results for two-phase and temperature dependent viscous flows.

As demonstrated in Section 4.2, the model compares well to the analytical solutions for Poiseuille flow velocity profiles of both one and two phases. The two-phase flows test the addition of multiple phases with different viscosities. It is also able to develop the instabilities that occur at the interface between phases, in line with perturbation theory. Additionally, the growth rate of the interfacial instabilities compares closely with results in the literature. This highlights one of the key advantages of SPH over the VOF method, in following a deforming interface. The introduction of temperature and viscosity variables, as well as the temperature-dependent viscosity functions in Section 3.5.1, are validated through the temperature-dependent Poiseuille flow case. The model is shown to reproduce well velocity along the channel for these cases.

The lid-driven cavity case in Section 4.3 is an example of mechanically driven flow. Although the streamlines appeared reasonable with the basic boundary condition, a more sophisticated approach was demonstrated to have greater success in replicating the velocities of the benchmark data. Going forward, the mDBC method is used for other cases of flow within a cavity. Mixing measures described in Section 3.6 are applied to the two-phase case and their relative merits discussed. In particular, the new combined mixing measure is shown to give additional insight into the deformation within and between phases. Comparisons with the benchmark method confirm that the correct mixing behaviour is captured.

The addition of both the temperature equation (Equation (3.40)) and Boussinesq approximation (Equation (2.11)) are established to be effective for simulating buoyancy-driven flow through the differentially heated cavity case in Section 4.4. The Dirichlet boundary condition for temperature, discussed in Section 3.3.4.4, is found to be adequate for heating. However, the new higher order adiabatic boundary condition, based on the mDBC method, is established to be superior to the more simple method of excluding boundary contributions.

The results in this chapter demonstrate the capability of SPH in modelling temperature dependent viscous mixing applications. In the next chapter, the complete model is utilised in order to simulate complex mixing during vitrification of nuclear waste. The measures of mixing defined in Section 3.6 and investigated in Section 4.3.4 will be used to compare with data and quantify the degree of mixing achieved.

Chapter 5

Application Model: Mixing During Vitrification

5.1 Introduction

This chapter presents the main aim of this project to model mixing during vitrification within the melter. Initial temperature conditions ensure all glass phases are melted, and the internal temperature remains above borosilicate glass transition temperature of 1150 K. Modelling assumptions include the Boussinesq approximation and that there are two separate glass phases at the beginning of the simulation, containing different amounts of High Level Waste (HLW) by wt%. Dynamic viscosity is expected from Section 2.2 to remain within 1-100 Pa s, but shall be determined by the Melt Viscosity Alogirthm (MVA) designed by Miller [144], which uses the viscosity models described in Section 3.5.1. Density can also be computed from the compositions used as inputs for this algorithm. Although there are no direct comparisons for the cases in this chapter to use as validation, the velocity magnitude of 10^{-3} $\mathrm{m\,s^{-1}}$ given by Sundaram et al. [215] will provide a check for reasonable flow behaviour.

A simplified vitrification case, capturing the key physics, is defined. The melter is approximated as a cylindrical tank, completely filled with fluid. Firstly, the effects of heating the curved wall on buoyancy-driven flow are investigated with a single fluid phase with constant density. These cases represent the buoyancy-driven circulation of a homogeneous vitrification mixture. Mixing of phases is investigated by splitting the fluid into two phases of equal size, similar to the lid-driven cavity simulations in Section 4.3.3.

In Section 5.3, the viscosity models introduced in Section 3.5.1 are used to implement temperature-dependent viscosity. A single phase with a typical composition and waste fraction is first simulated to study the flow of a homogeneous mixture. For the final cases in Section 5.3.2, these phases are chosen as one phase containing waste and a second phase representing only molten glass. This approximates the phases present when separation occurs during mixing. The composition and phase parameters are taken from the MVA [144] and corresponding data.

The following cases combine all additional physics to the Smoothed Particle Hydrodynamics (SPH) model, each of which was validated in the previous chapter. Following previous validation, the SPH model simulates separated phases within a melter during vitrification in order to explore the mixing flow. Mixing measures, introduced in Section 3.6, are applied in post-processing to give valuable insight into the state of mixing and mixing process at a given time. With this knowledge, recommendations of the conditions required for optimal mixing, as well as those resulting in a heterogeneous product, can be given.

5.2 Constant Viscosity

At first, the new case is simulated with phases of constant viscosity, similar to the lid-driven cavity and differentially heated cavity cases in Chapter 4. This gives insight into the flow taking place within the cylindrical tank, subject to heating, through buoyancy-driven flow.

5.2.1 Single Phase of Constant Viscosity

Initially, a single phase is considered within the melter. As can be seen in Figure 5.1, the top and bottom boundaries are adiabatic. This is enforced by the new modified dynamic boundary condition (mDBC) adiabatic boundary condition, defined in Section 3.3.4.4, and mDBC are used for other boundary variables. The curved boundary is held at constant temperature θ_H, heating the fluid inside the container. The fluid has a uniform initial temperature of θ_C. In order to apply separate boundary

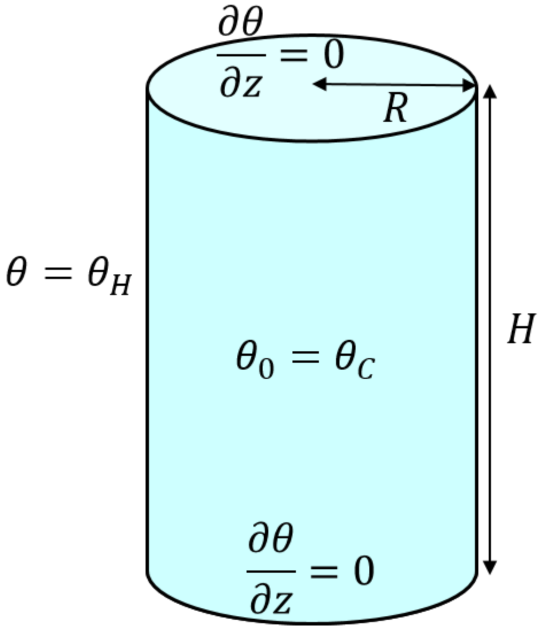
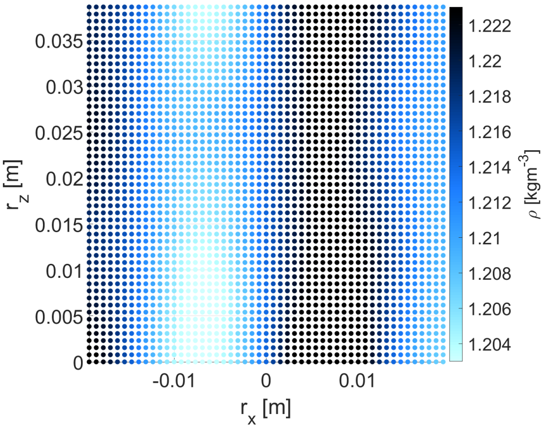

$\frac{\partial\theta}{\partial z} = 0$
R
$\theta = \theta_H$
H
$\theta_0 = \theta_C$
$\frac{\partial\theta}{\partial z} = 0$

Figure 5.1: Thee-dimensional vitrification case with a single fluid phase.

conditions, the boundary phase modifications introduced in Section 3.4.2 are applied. When using mDBC, this requires the normal invert option to be set to true in order to ensure all normals are pointing into the cylindrical cavity. The Freedraw option in GenCase is also used for all geometry components to position particles along the curved geometry of the cylinder. Fluid particles are arranged on concentric circles in cylindrical coordinates, rather than a Cartesian grid, for a uniform distribution inside the tank.

Beginning with the parameters used in Section 4.4.2, where the fluid is air with a constant viscosity of $\nu = 1.506 \times 10^{-5}$ m^2 s^{-1}, and also setting $H = 2R = 0.0386$ m, a similar Rayleigh number ($Ra = 10^5$) is expected. The temperatures are also chosen to be the same, with $\theta_C = 283$ K and $\theta_H = 303$ K. All boundaries have the same thermal properties as the fluid and the initial particle spacing is chosen to be $dx = 0.001$ m.

Density diffusion is included to reduce accumulation of numerical error from pressure fluctuations, and particle shifting has typical coefficient -2 to maintain a uniform particle distribution. The variables at 45 s in planes $x = 0$ and $y = 0$ are displayed in Figures 5.2 to 5.4, where it can be seen that heating is clearly taking place via conduction from the curved wall and convection of the fluid. However, density is not radially symmetric as expected, with velocity behaving similarly. No relevant literature could be found to determine whether this is an appearance of circulation cells due



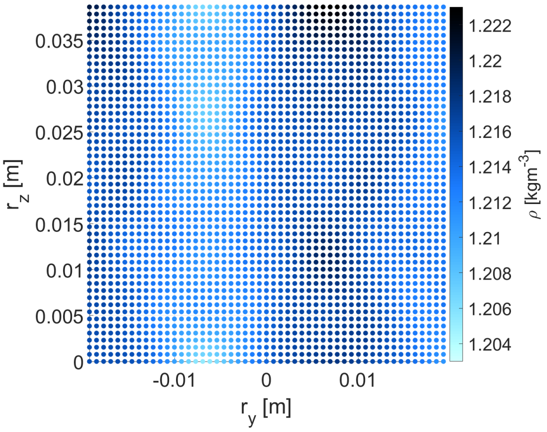
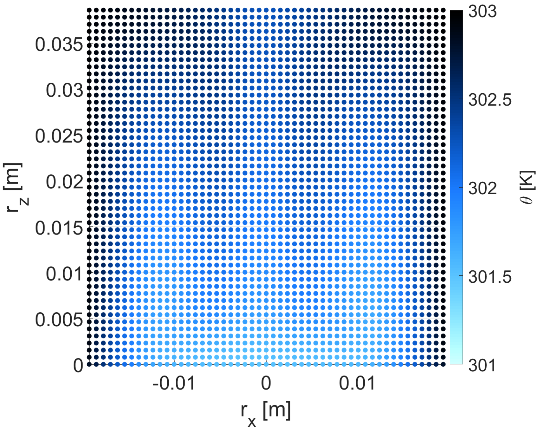


Figure 5.2: Density in $x = 0$ and $y = 0$ planes at 45 s for single phase melter with differentially heated cavity parameters.



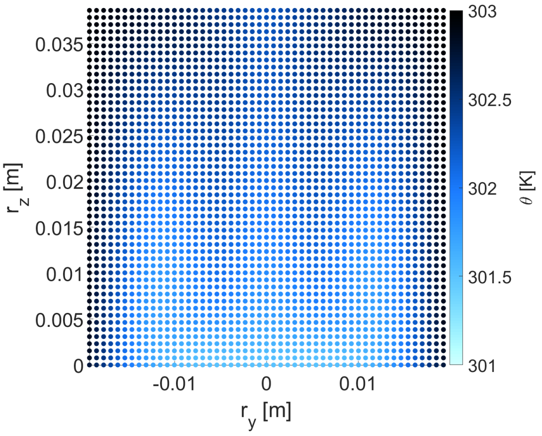
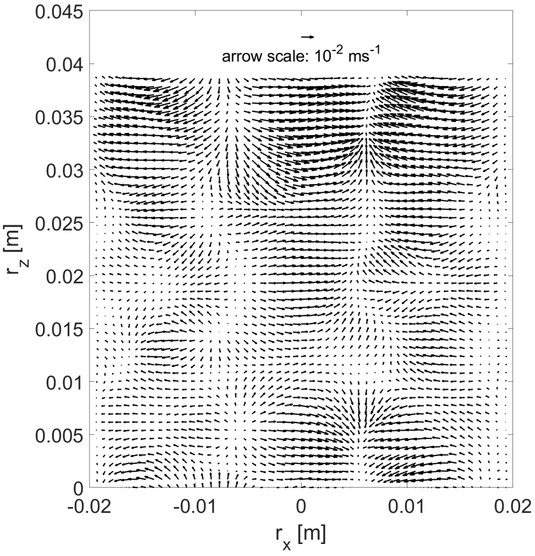


Figure 5.3: Temperature in $x = 0$ and $y = 0$ planes at 45 s for single phase melter with differentially heated cavity parameters.

to a physical instability. Later, around 60 s, particles are lost when set density limits are exceeded.

Increasing the constant viscosity from $\nu = 1.506 \times 10^{-5}$ $\mathrm{m^2\,s^{-1}}$ to $\nu = 10^{-4}$ $\mathrm{m^2\,s^{-1}}$ to avoid the density pattern found previously results in a symmetric density as well as velocities clearly following convective flow, as seen in Figures 5.5 to 5.7. This results in heating throughout the cavity and evidence of recirculation, with all particles having an increased temperature at 45 s. The Reynolds number is 0.4, which is six orders of



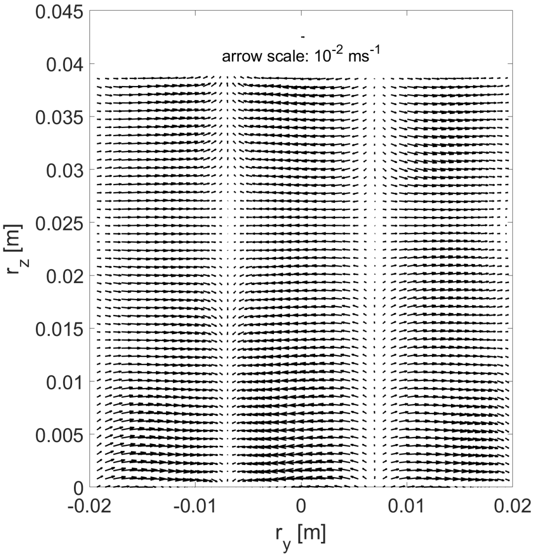


Figure 5.4: Velocity in $x = 0$ and $y = 0$ planes at 45 s for single phase melter with differentially heated cavity parameters.

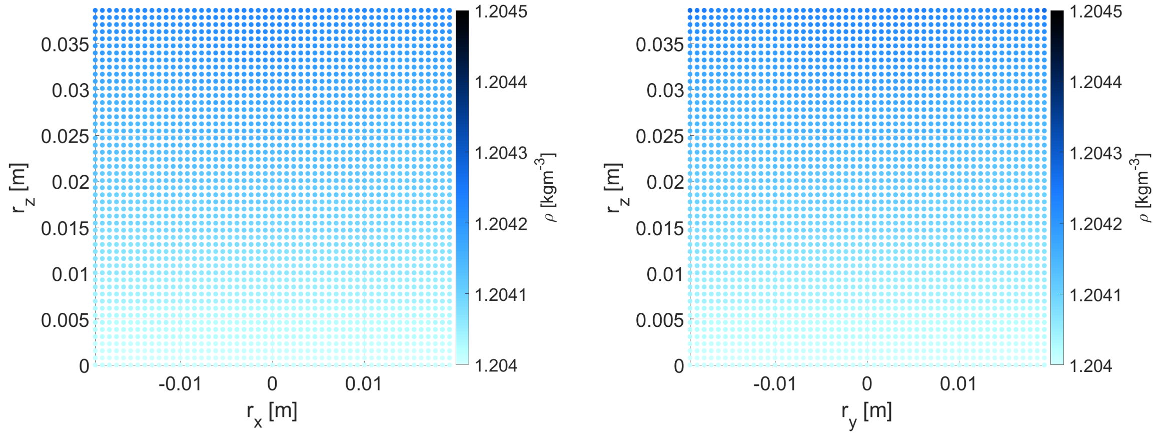


Figure 5.5: Density in $x = 0$ and $y = 0$ planes at 45 s for single phase melter with differentially heated cavity parameters but increased viscosity.

magnitude smaller than that of the differentially heated cavity case in Section 4.4.2 due to greater viscosity and significantly lower velocity. The contribution to acceleration from temperature difference is smaller for this case and decreases with time. This results in a longer time taken for recirculation. *Re* would be increased by increasing the dimensions of the melter to a realistic size, as long as velocities do not reduce significantly.

Particles are steadily lost for this case after around 15 s as they escape through

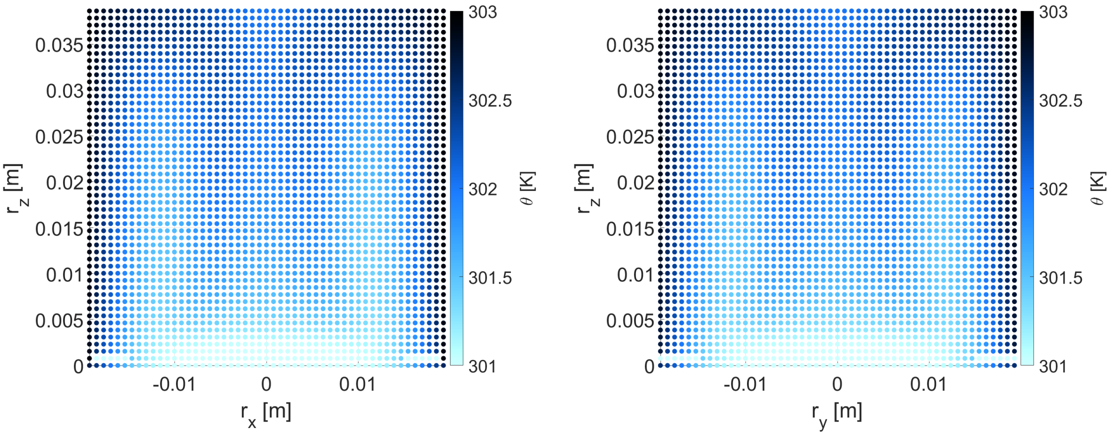


Figure 5.6: Temperature in $x = 0$ and $y = 0$ planes at 45 s for single phase melter with differentially heated cavity parameters but increased viscosity.

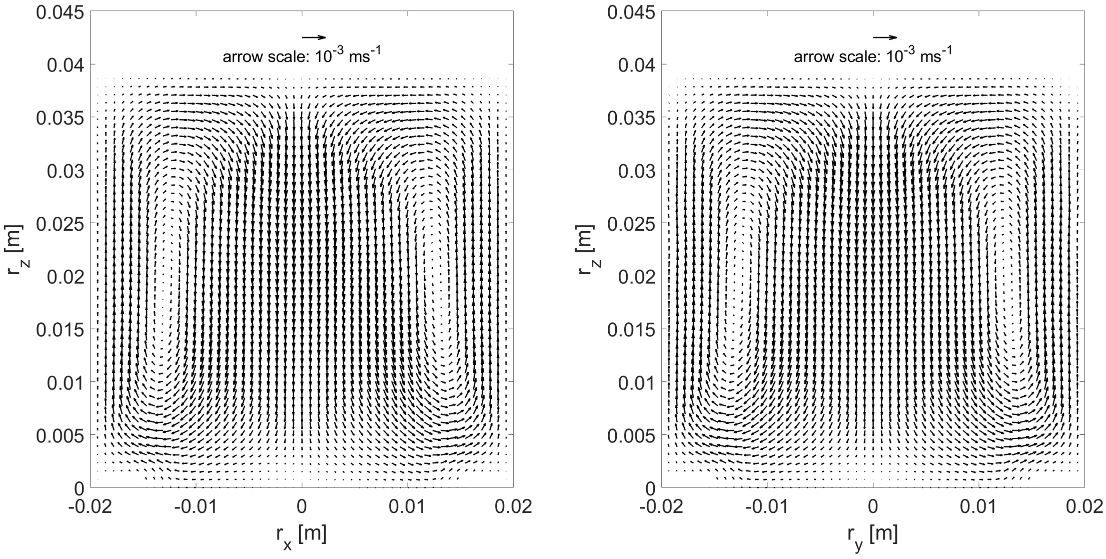


Figure 5.7: Velocity in $x = 0$ and $y = 0$ planes at 45 s for single phase melter with differentially heated cavity parameters but increased viscosity.

the boundaries in the upper half of the cavity. The number of fluid particles penetrating the boundary is reduced when using the dynamic boundary condition (DBC), suggesting a correction needs to be made to the mDBC method to take account of the difference in volume when extrapolating density into a concave boundary. A single phase within the heated cylindrical melter exhibits radially symmetric circulation due to buoyancy-driven flow.

5.2.2 Two Phases of Constant Viscosity

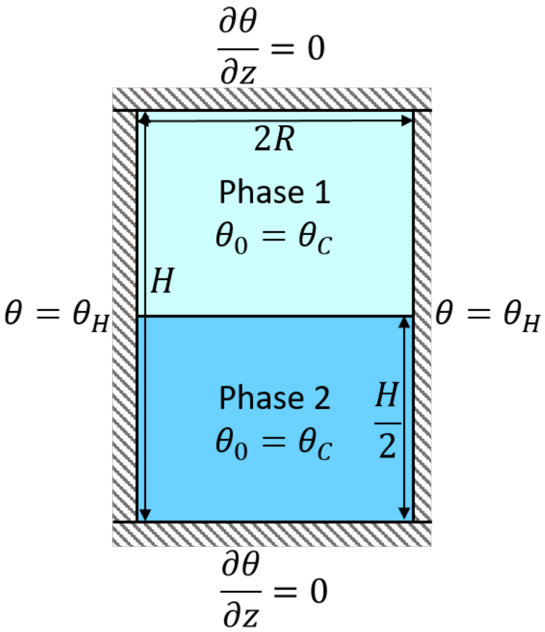


Figure 5.8: Two-dimensional vitrification case with a two fluid phases.

To approximate separated phases, the fluid within the cavity is split horizontally into two phases of equal height. Assuming radial symmetry of flow and variables within the tank, a two-dimensional plane at $y = 0$ is simulated, as shown in Figure 5.8. Starting from the previous case in Section 5.2.1 with dimensions $H = 2R = 0.0386$ m, each phase has a constant viscosity. The top phase is chosen to have $\nu_1 = 2 \times 10^{-4}$ $\mathrm{m^2\,s^{-1}}$ and the bottom $\nu_2 = 10^{-4}$ $\mathrm{m^2\,s^{-1}}$. Following the method for obtaining a Reynolds number for two phases in Section 4.2.2, $Re = 3$ for this case. Running two-dimensional simulations allows a general understanding of a large number of cases to be obtained by taking advantage of the shorter runtime whilst capturing the principle flow behaviour. This is shown in Figure 5.8, where boundary conditions remain as before. These initial conditions demonstrate mixing due to natural convection when separated phases are present.

The results at 300 s in Figure 5.9 show the circulatory flow, as for the single phase case, in particular the circulation caused by heating at both side walls and symmetry of density and temperature. Particles are still lost, but fewer than the three-dimensional case at 7% over 300 s.

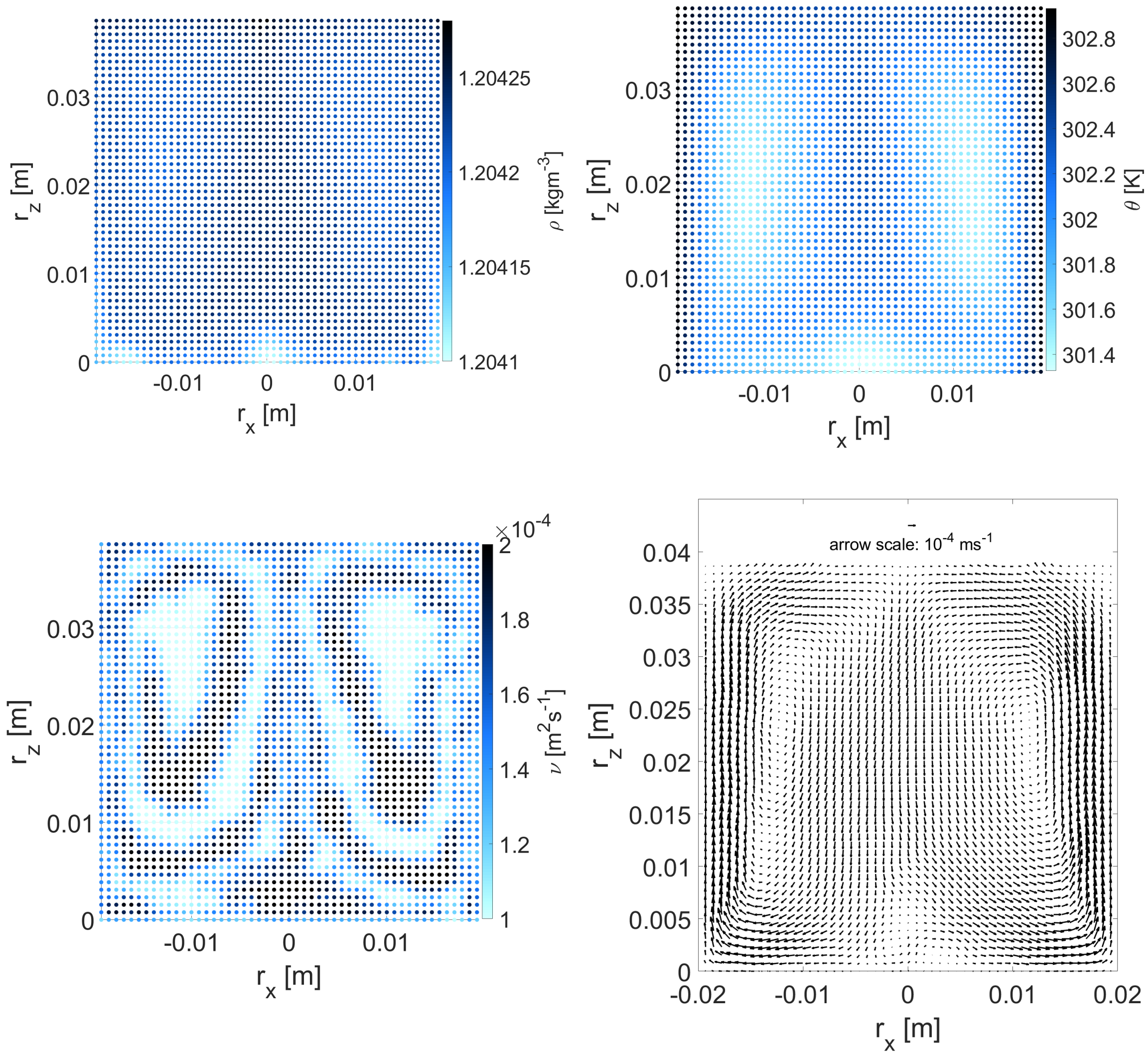


Figure 5.9: Density, temperature, viscosity and velocity at 300 s for two-dimensional two-phase melter with differentially heated cavity dimensions.

The effect of this circulation on mixing between phases is demonstrated in Figure 5.10 with the mixing measures introduced in Section 3.6. The volume fraction (VF) clearly shows that the convection is causing circulation and mixing of phases, although there are still regions of separate phases at 300 s. The finite-time Lyapunov exponent (FTLE) is fairly constant for all times, consisting of the two circulating cells. Robinson's measure shows clearly the interface between phases, whereas the new mixing measure has more variation at the interface depending on where it lies on areas of high circulation.

The three-dimensional case with two constant phases is shown in Figure 5.11, with $\nu_1 = 2 \times 10^{-4}$ m^2 s^{-1}, $\nu_2 = 10^{-4}$ m^2 s^{-1}. Other fluid properties are the same as for the previous single phase case in Section 5.2.1. Both temperature and density are

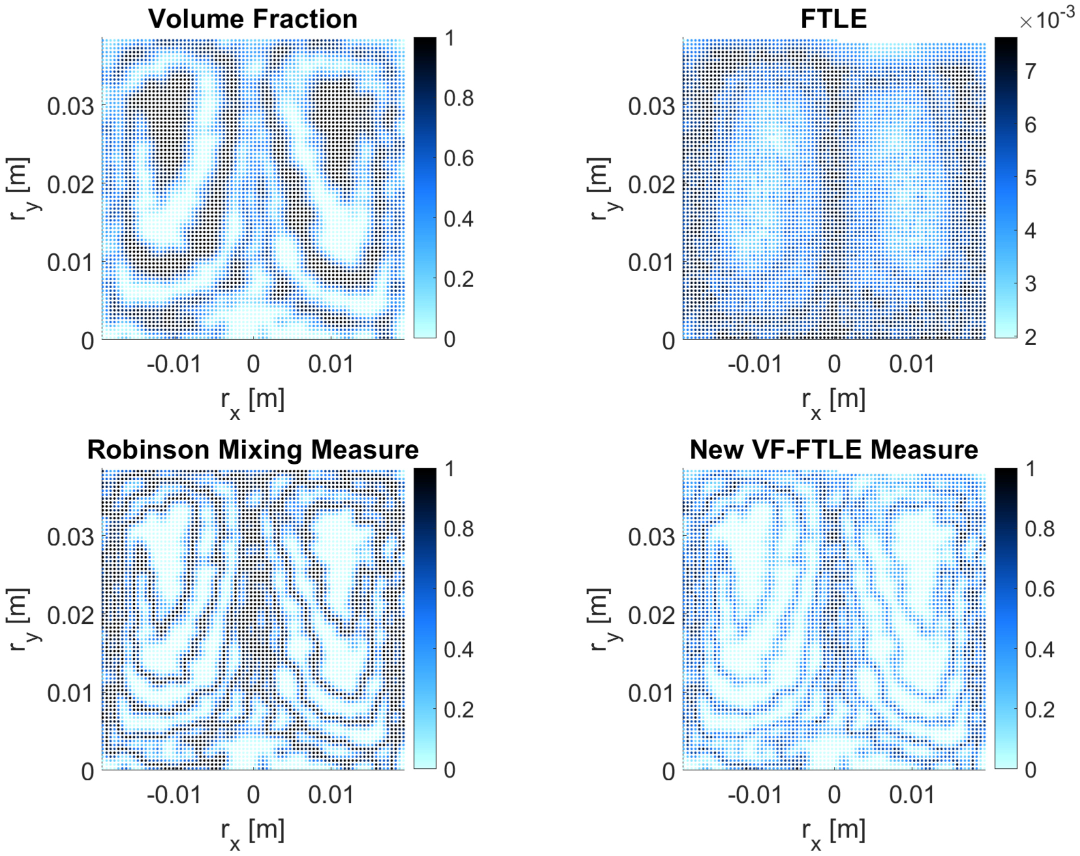


Figure 5.10: Mixing measures at 300 s for two-dimensional two-phase melter with differentially heated cavity dimensions.

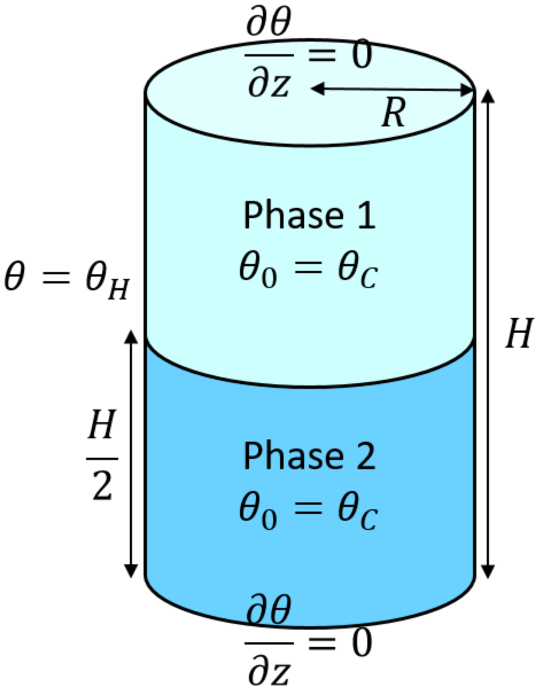


Figure 5.11: Thee-dimensional vitrification case with a two fluid phases.

radially symmetric in Figures 5.12 and 5.13, but temperature has almost reached θ_H everywhere by 60 s. Since viscosity is constant in each phase, Figure 5.14 shows clear movement of and mixing between phases caused by convective flow. This is confirmed by the velocity field in Figure 5.15. The flow behaviour closely resembles the two-dimensional case, justifying the use of two-dimensional simulations in preliminary investigations.

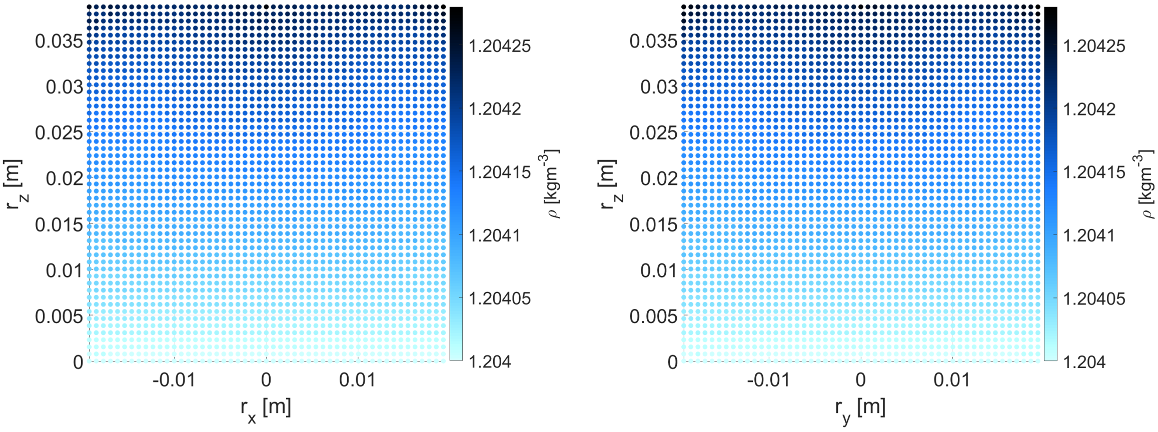


Figure 5.12: Density in $x = 0$ and $y = 0$ planes at 60 s for two-phase melter with differentially heated cavity dimensions.

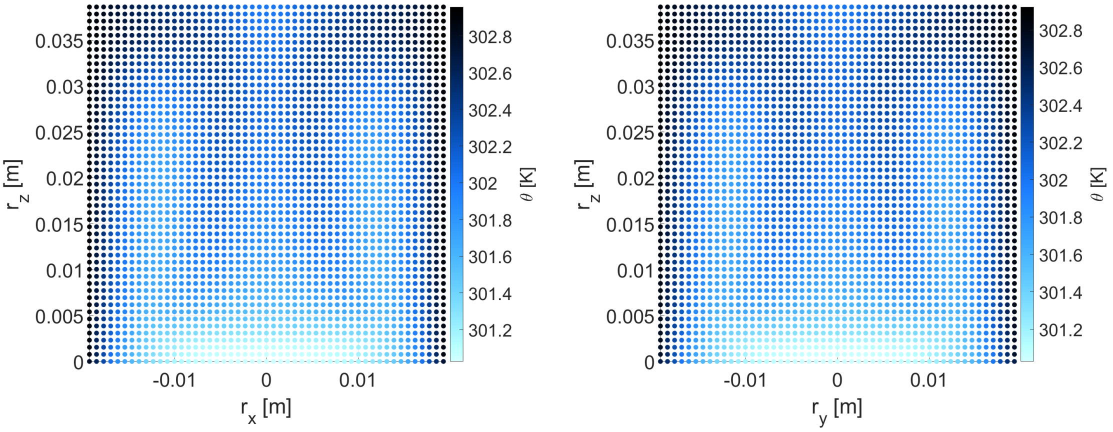


Figure 5.13: Temperature in $x = 0$ and $y = 0$ planes at 60 s for two-phase melter with differentially heated cavity dimensions.

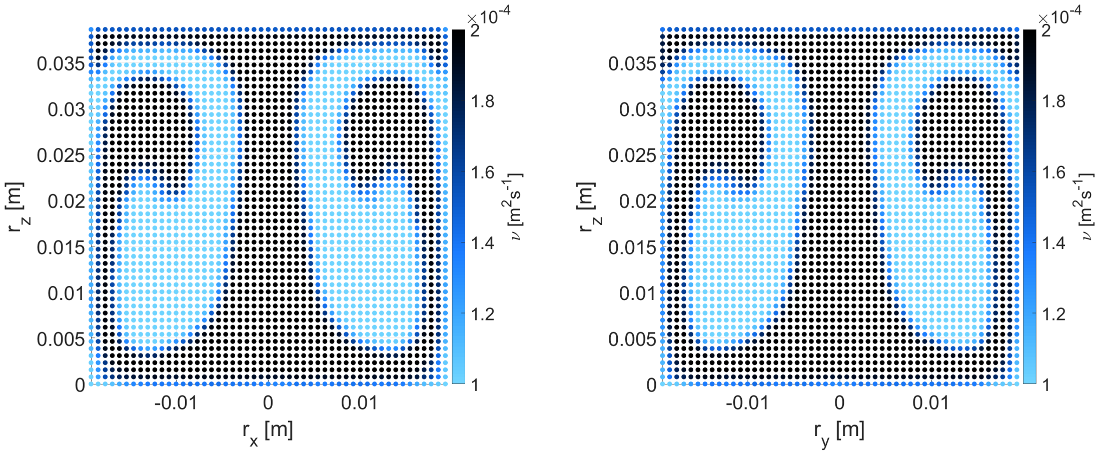


Figure 5.14: Viscosity in $x = 0$ and $y = 0$ planes at 60 s for two-phase melter with differentially heated cavity dimensions.

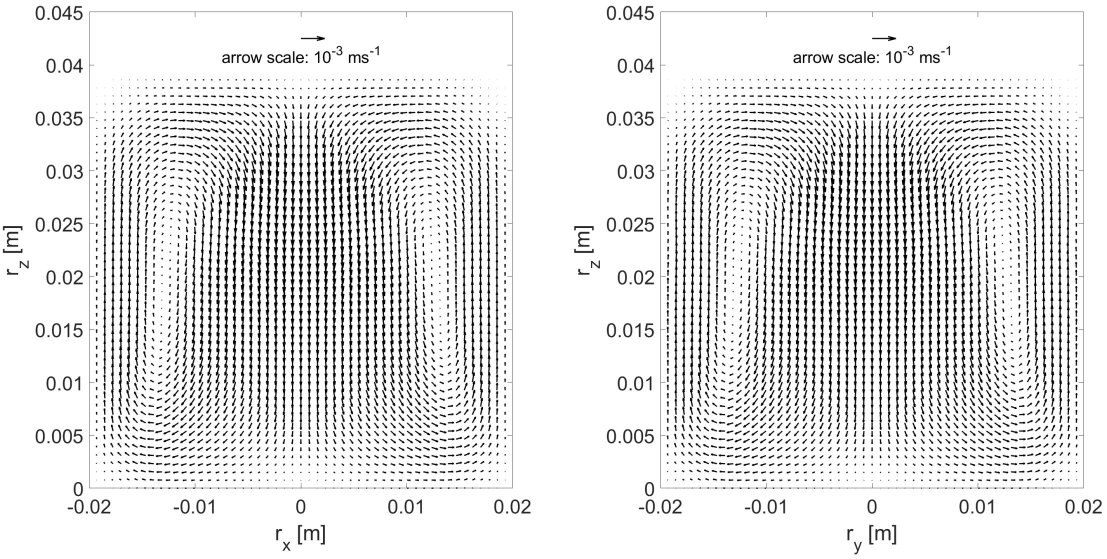


Figure 5.15: Velocity in $x = 0$ and $y = 0$ planes at 60 s for two-phase melter with differentially heated cavity dimensions.

The FTLE in Figure 5.16 shows convective flow and circulation, whilst mixing between phases is seen in the other measures. The interface between phases is highlighted, and large areas of each isolated phase shown. Both mixing entropy (both phases) and the global mixing measure in Figure 5.17 show mixing increasing with time. By 120 s the rate of increase is slowing, but a steady state has not yet been reached.

The previous cases demonstrate the ability to model buoyancy-driven flow within

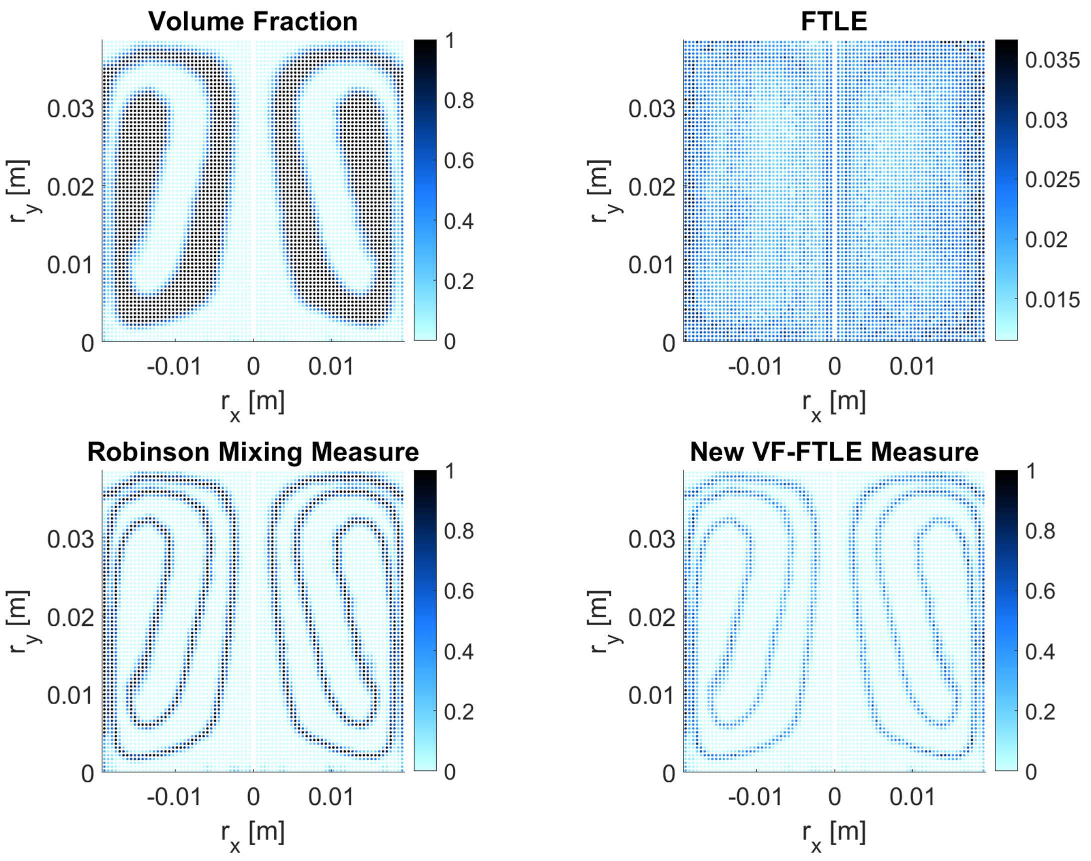


Figure 5.16: Mixing measures at 120 s for three-dimensional two-phase melter with differentially heated cavity dimensions.

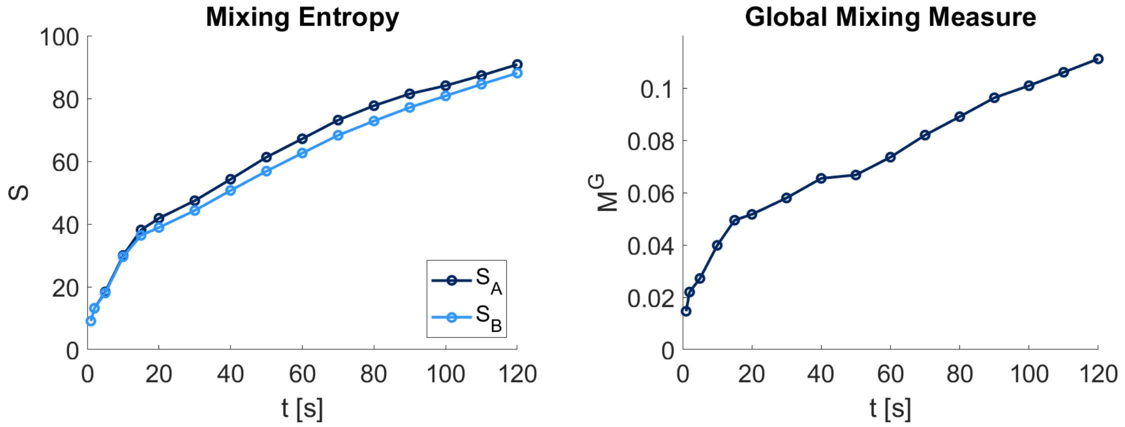


Figure 5.17: Global mixing measures plotted against time for three-dimensional two-phase melter with differentially heated cavity dimensions.

a cylindrical tank, and establish that the recently developed mDBC boundary method is not yet reliable for use inside a cylinder. Instead, DBC are used for the following melter cases.

5.2.3 Modelling a Vitrification Melter

geometry	H	R
i	0.0386	0.0193
ii	1	0.18
iii	0.504	0.252

Table 5.1: Dimensions in metres used for vitrification cases.

The exact geometry of the vitrification melter is not known, but from the 100 l capacity two potential pairs of dimensions are chosen in Table 5.1 as geometries *ii* and *iii*. Geometry *i* is used in the previous sections of this chapter and based on the dimensions for the differentially heated cavity case in Section 4.4. Following the success of these cases, geometry *iii* is also chosen so that $H = 2R$.

parameter	value
$\overline{\rho}$	$3000\ \mathrm{kg\,m^{-3}}$
ν	$6.7 \times 10^{-4}\ \mathrm{m^2\,s^{-1}}$
C_p	$900\ \mathrm{J\,kg^{-1}\,K^{-1}}$
k	$1.14\ \mathrm{W\,m^{-1}\,K^{-1}}$
β	$3.3{\times}10^{-6}\ \mathrm{K^{-1}}$
g	$-9.81\ \mathrm{m\,s^{-2}}$

Table 5.2: Typical material and thermal parameters for borosilicate glass used in vitrification simulations.

Geometry *ii* is run with smoothing length $h = 2.0dx$, which was required for accurate results in Section 4.4. The density and viscosity of the fluid are also changed to be within a reasonable range for borosilicate glass, as in Table 5.2 where thermal parameters are also stated. For the initial cases viscosity is constant. As previously, density diffusion is included, but temperatures are increased to $\theta_C = 1073$ K and $\theta_H = 1323$ K with reference temperature $\overline{\theta} = \theta_H$.

For this case there is no significant heating, and density and velocity are not symmetric, as seen in Figure 5.18. Therefore, the expected circulation due to buoyancy-driven flow does not take place. Around 13 s, large fluctuations in density occur within the fluid and many particles are excluded due to exceeding density limits. This is most likely due to accumulation of a numerical error, which was less significant for the cases with geometry *i*. Increasing the density diffusion coefficient to attempt to dissipate the density variation does not resolve the issue. Increasing the resolution, modifying the shifting coefficient and smoothing length also have no effect.

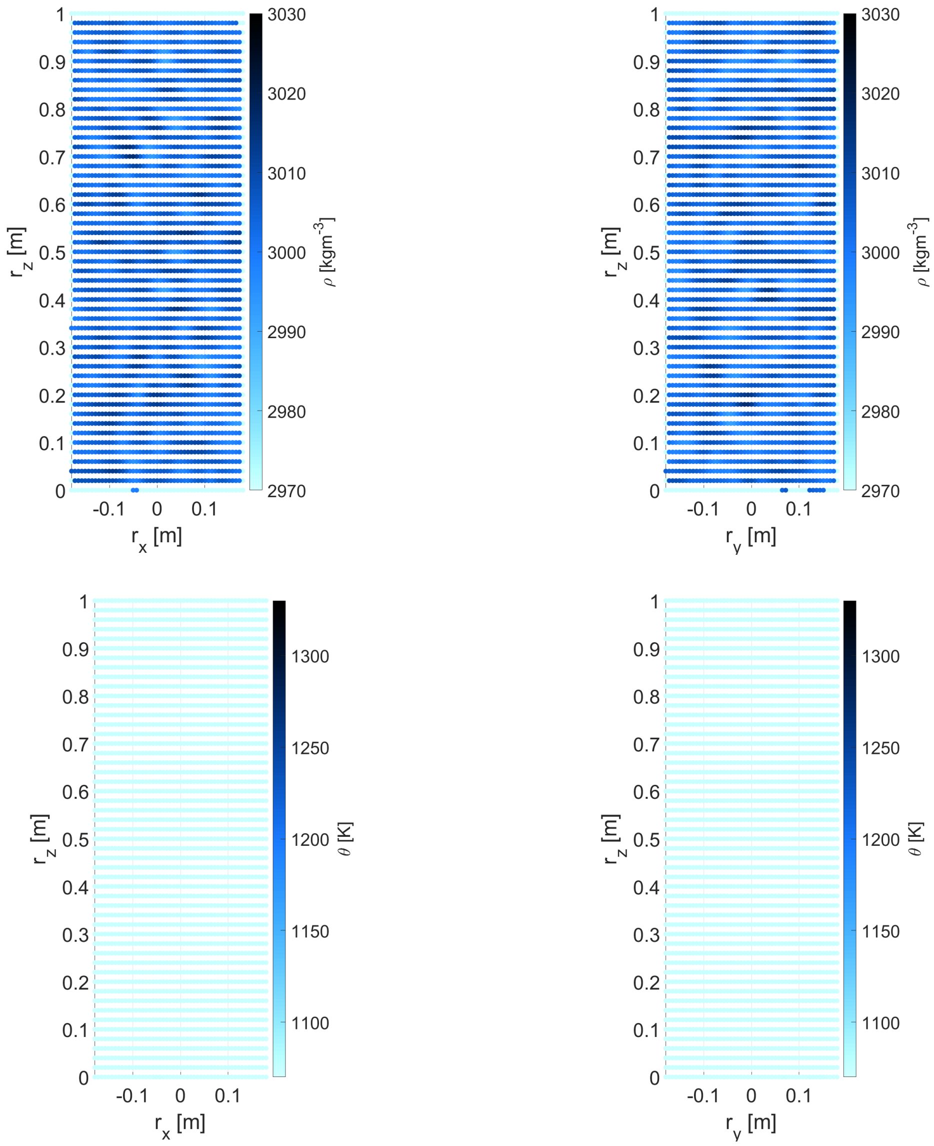


Figure 5.18: Density and temperature in $x = 0$ and $y = 0$ planes at 10 s for single phase melter with geometry *ii*.

Melter geometry *iii* is proposed where $H = 2R$, as for geometry *i* used for the initial vitrification cases. The tank volume is maintained at 100 l, therefore $2\pi R^3 = 0.1$ m^3 and $R = 0.252$ m, $H = 0.504$ m. The single fluid phase has constant viscosity $\nu = 1.6 \times 10^{-3}$ m^2 s^{-1}, density $\rho = 3000$ kg m^{-3} and is run with particle spacing $dx = 0.01$ m and smoothing length $h = 2.0dx$. At 5 s in Figure 5.19, heating is beginning to take place. However, this case also suffers from an instability around 6 s, which causes rapid mixing of particles and then exclusion due to exceeding density limits. Significant variation in density can be seen at 5 s in Figure 5.20. Along with the previous results with geometry *ii*, this suggests that the problem stems from

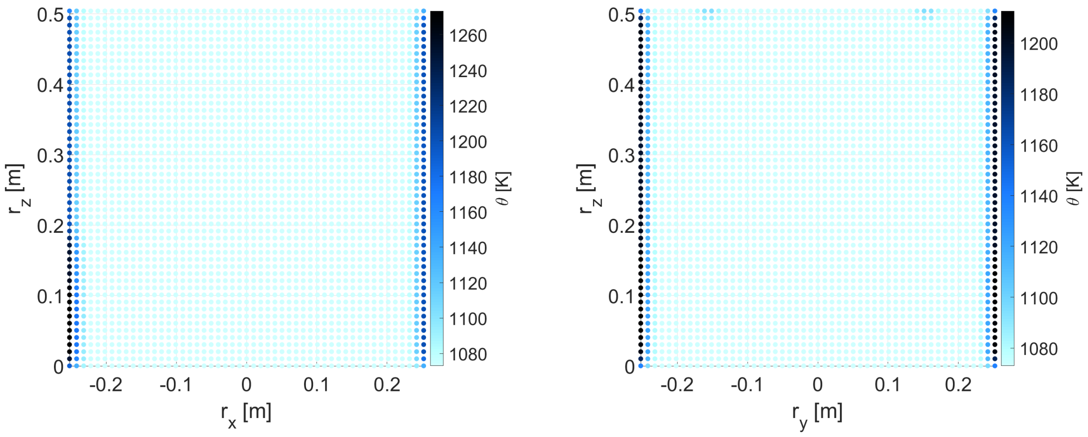


Figure 5.19: Temperature in $x = 0$ and $y = 0$ planes at 5 s for single phase melter with geometry *iii*.

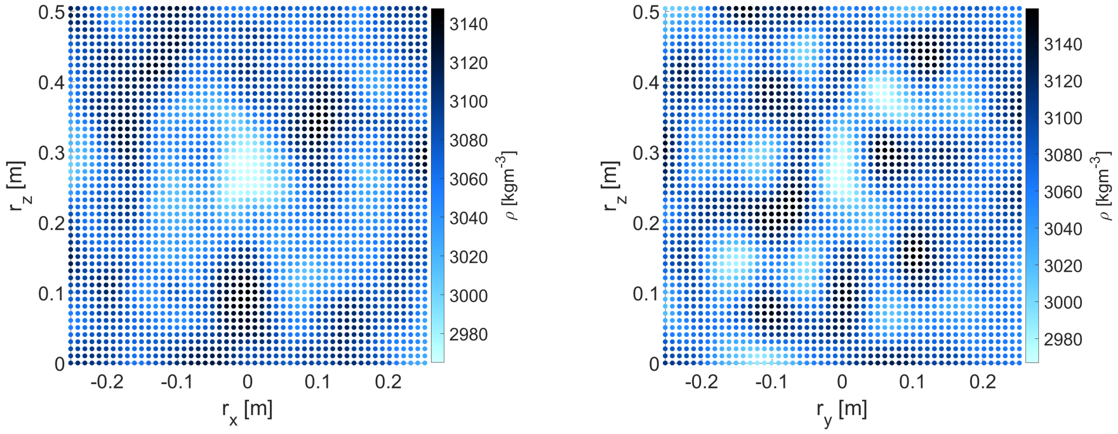


Figure 5.20: Density in $x = 0$ and $y = 0$ planes at 5 s for single phase melter with geometry *iii*.

more complex physics involved in the vitrification melter which the model is unable to capture.

An investigation of the influence of key dimensionless numbers, listed in Table 5.3, is undertaken to establish if there is any link to the instability found. Since the above cases for geometries *ii* and *iii* have little to no movement before the onset of instability, a characteristic velocity is determined through dimensional analysis. A commonly used

geometry	Re	Ra	Pr
i	8.8×10^1	1.8×10^4	4.7×10^0
ii	1.9×10^2	2.9×10^7	1.6×10^3
iii	2.8×10^1	1.5×10^6	3.8×10^3

Table 5.3: Values of dimensionless numbers for vitrification geometries with initial parameters.

form is

$$\overline{v} = \sqrt{2gL\beta(\Delta\theta)}. \tag{5.1}$$

Characteristic velocity is computed in this way for all geometries for consistency.

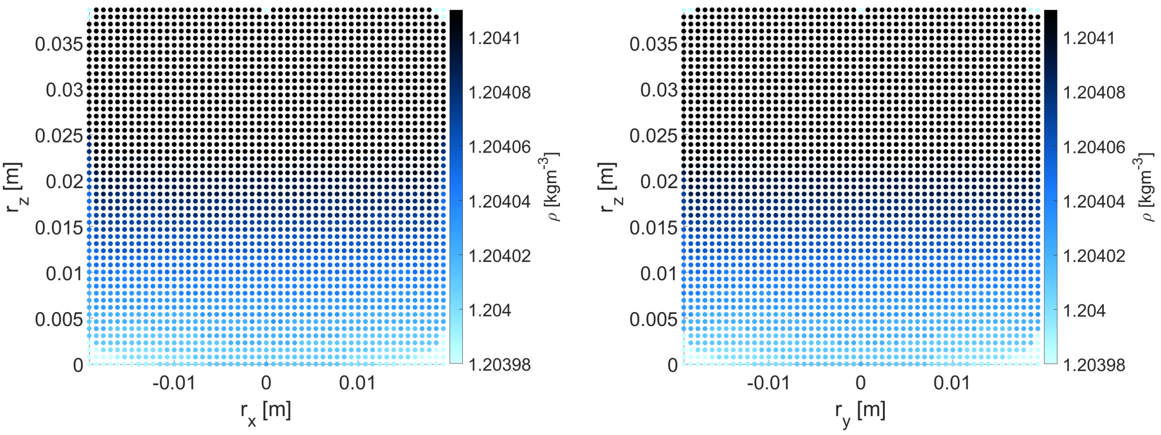


Figure 5.21: Density in $x = 0$ and $y = 0$ planes at 10 s for single phase melter with geometry i and constant $\nu = 3 \times 10^{-4}$ m^2 s^{-1}.

Geometry i is run with similar Reynolds number to geometry iii, by increasing the viscosity by three times to obtain $Re = 29$. Heating takes place and the density is symmetrical as expected in Figures 5.21 and 5.22. Buoyancy-driven circulation also takes place, and there is no sign of the instability seen for the larger geometries, despite the lower Re. On the other hand, increasing the Reynolds number of geometry iii by reducing the viscosity by three times for $Re = 91$, similar to i in Table 5.3, finds the instability occurring again. This shows that the Reynolds number is not critical in the appearance of this instability.

Instead, the initial density of case iii is reduced to $\rho = 30$ kg m^{-3} so that the Rayleigh number is of the same order of magnitude as case i. Here, the instability does not appear and density remains within the prescribed limits. Heating takes

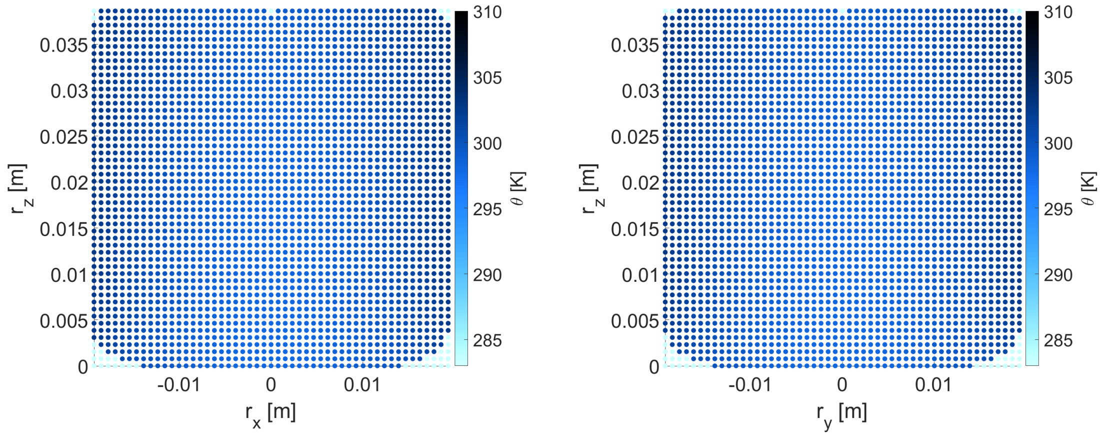


Figure 5.22: Temperature in $x = 0$ and $y = 0$ planes at 10 s for single phase melter with geometry i and constant $\nu = 3 \times 10^{-4}$ $\mathrm{m^2\,s^{-1}}$.

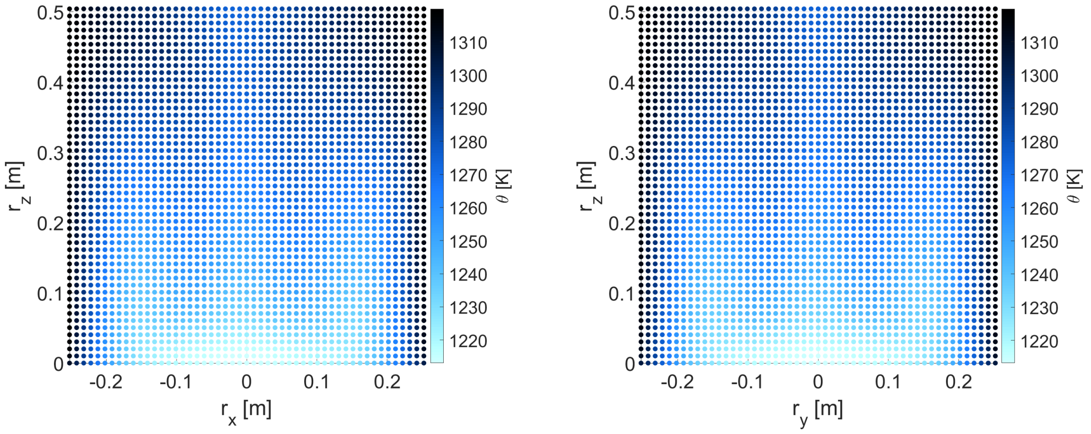


Figure 5.23: Temperature in $x = 0$ and $y = 0$ planes at 300 s for single phase melter with geometry iii and $\rho = 30$ $\mathrm{kg\,m^{-3}}$.

place in Figure 5.23 and circulation can be seen in Figure 5.24, remaining after 300 s. Circulation is shown by the FTLE plotted in Figure 5.25, especially where hotter fluid is rising next to the curved boundary, and the velocities are comparable to those expected from the literature review of order 10^{-3} $\mathrm{m\,s^{-1}}$. These results demonstrate that the instability found is connected to the Reynolds number of the flow. It appears when Ra is much higher, and therefore heat transfer is dominated by convection. Keeping the kinematic viscosity fixed, the Reynolds number remains unchanged.

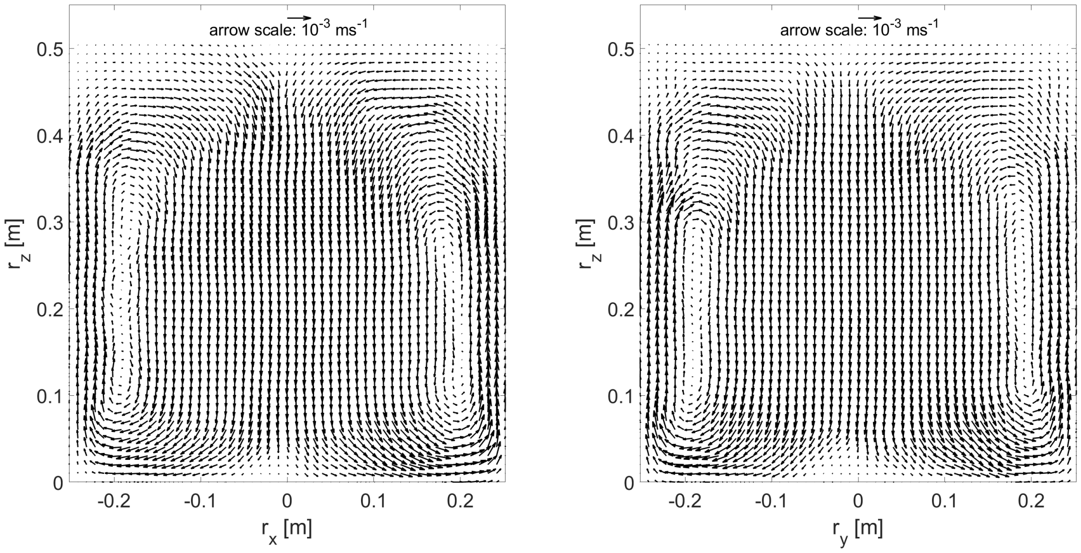


Figure 5.24: Velocity in $x = 0$ and $y = 0$ planes at 300 s for single phase melter with geometry *iii* and $\rho = 30\ \mathrm{kg\,m^{-3}}$.

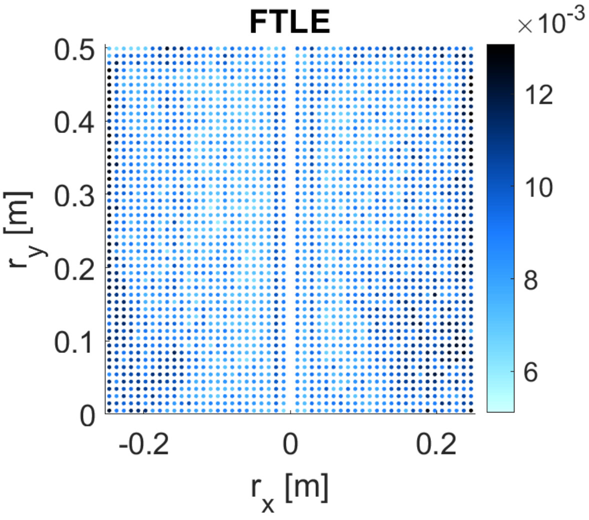


Figure 5.25: FTLE in $y = 0$ plane at 300 s for single phase melter with geometry *iii* and $\rho = 30\ \mathrm{kg\,m^{-3}}$.

The Prandtl number, also listed in Table 5.3, is three orders of magnitude greater for the melter cases than for case *i*. A large Pr results in a thin thermal boundary layer, as seen previously in cases *ii* and *iii*, which is much thinner than the velocity boundary layer. Meanwhile, the cases with geometry *i* have a Prandtl number of order 10, where the thermal and velocity boundary layers are closer in proportion. This suggests that the model is not currently able to capture the physics in a thin thermal boundary layer.

On the other hand, increasing the density of case *i* by two orders of magnitude to

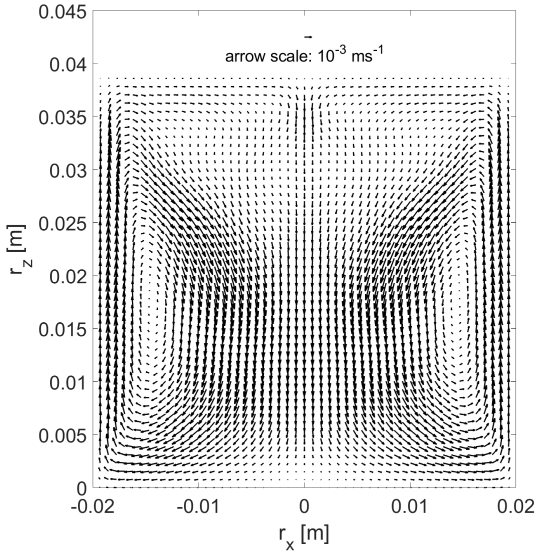


Figure 5.26: Velocity in $x = 0$ and $y = 0$ planes at 20 s for single phase melter with geometry i and $\rho = 120.4\ \mathrm{kg\,m^{-3}}$.

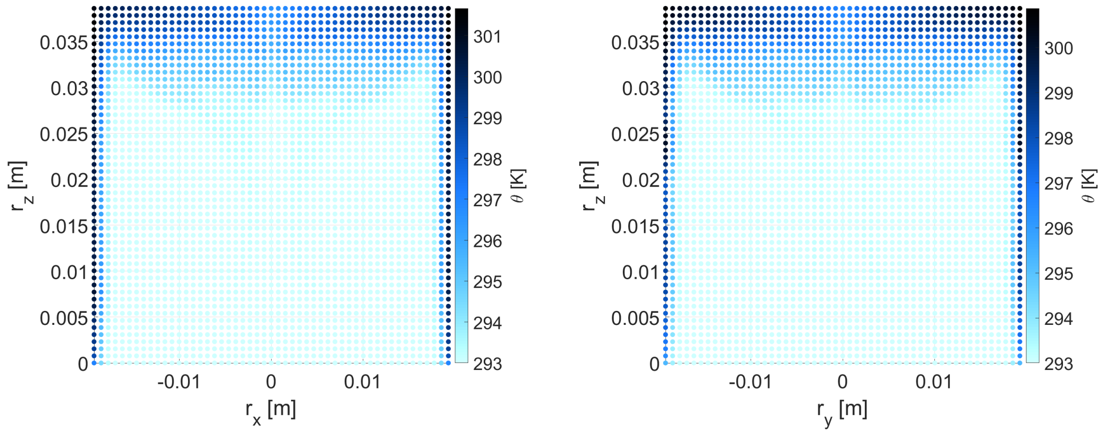


Figure 5.27: Temperature in $x = 0$ and $y = 0$ planes at 20 s for single phase melter with geometry i and $\rho = 120.4\ \mathrm{kg\,m^{-3}}$.

$\rho = 120.4\ \mathrm{kg\,m^{-3}}$ also results in circulatory flow, as seen in Figure 5.26. For this case Ra is of the same order of magnitude as i in Table 5.3, but Pr remains three orders of magnitude smaller. The plot of temperature in Figure 5.27 shows that convection is also taking place, and no instability appears within the first 30 s. This suggests that a high Rayleigh number is not sufficient for the instability to appear, and a high Prandtl number is also required.

To run this model with a higher density, closer to that of borosilicate glass, treatment of a thin thermal boundary layer at high Rayleigh number and corresponding appropriate boundary conditions must be introduced.

5.3 Variable Viscosity

The viscosity models in Section 3.5.1 are now included for viscosities dependent on temperature. Parameters for appropriate phase compositions are found using the melt viscosity algorithm [144].

5.3.1 Single Phase with Temperature-Dependent Viscosity

phase	A	B	C
25wt% HLW	13.93	-4988	-420700

Table 5.4: Parameter values for single phase vitrification case with temperature-dependent viscosity, using the Jenckel3 model (Equation (3.83)).

The cylindrical cavity with geometry *iii* in Figure 5.1 is filled with a single fluid with temperature-dependent viscosity to approximate homogeneous flow within the melter. The Jenckel3 viscosity equation in Equation (3.83) is used with the Vitrification Test Rig (VTR) data for a 25wt% borosilicate glass mixture. The MVA gives parameters in Table 5.4 for input to the fluid phase. The reference viscosity, $\overline{\nu}$, is calculated from the experimental values given for high temperatures to be 6.7×10^{-4} $\mathrm{m^2\,s^{-1}}$.

Geometry *iii* is used with material parameters as in Table 5.2, but density $\rho = 30$ $\mathrm{kg\,m^{-3}}$ as was found to be successful in the previous section. Although this density is considerably lower than found in reality, insight can still be gained into the circulation and mixing whilst avoiding the instability found at higher Rayleigh numbers. Figure 5.28 shows that as the temperature of the fluid changes near the boundaries, the viscosity is also reducing in Figure 5.29 according to the Jenckel3 relationship. However, the viscosity model results in viscosities two to three orders of magnitude greater than the reference viscosity $\overline{\nu} = 6.7 \times 10^{-4}$ $\mathrm{m^2\,s^{-1}}$, since the viscosity computed by Equation (3.83) is proportional to ρ^{-1}. Therefore, little movement and no circulation is seen within the 300 s simulated.

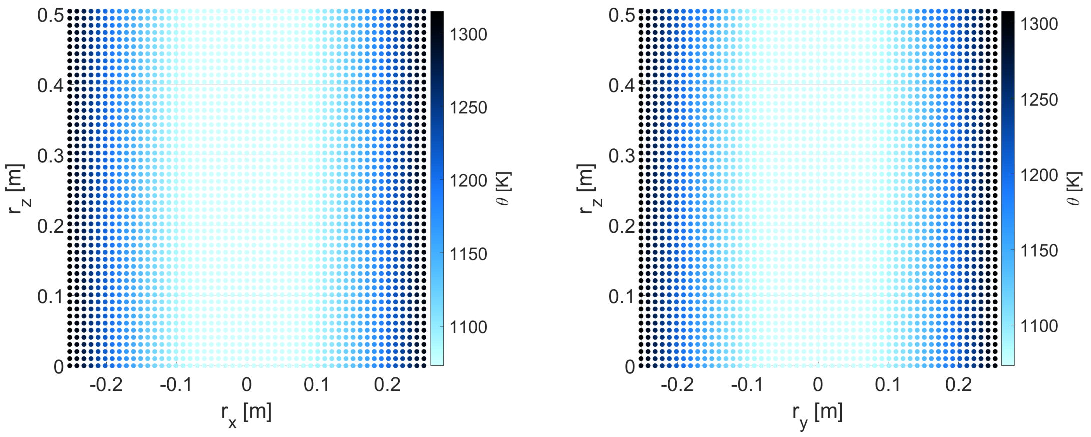


Figure 5.28: Temperature in $x = 0$ and $y = 0$ planes at 120 s for single phase melter with geometry *iii*, $\rho = 30\ \mathrm{kg\,m^{-3}}$ and temperature-dependent viscosity.

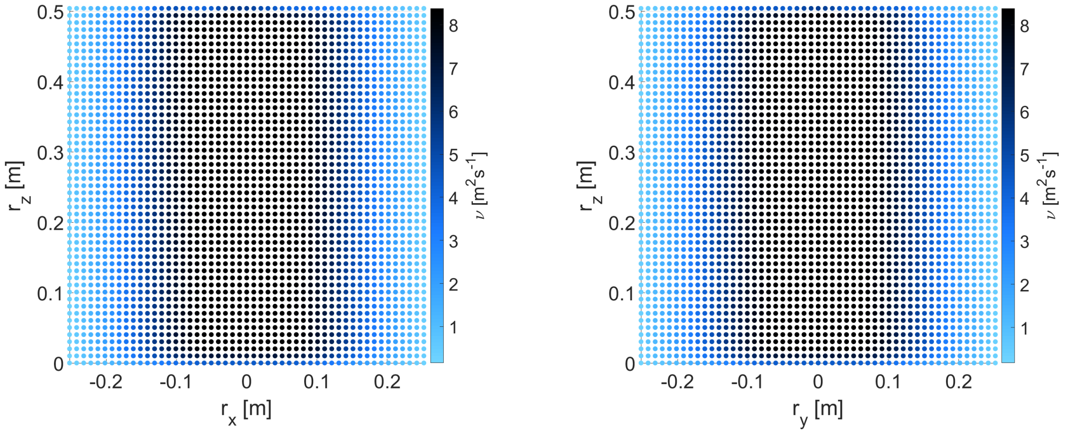


Figure 5.29: Viscosity in $x = 0$ and $y = 0$ planes at 120 s for single phase melter with geometry *iii*, $\rho = 30\ \mathrm{kg\,m^{-3}}$ and temperature-dependent viscosity.

geometry	Re	Ra	Pr
iii	5.7×10^{-3}	3.1×10^{0}	1.9×10^{5}
vitrification	1.2×10^{3}	1.6×10^{9}	1.79×10^{3}

Table 5.5: Values of dimensionless numbers for vitrification geometry *iii* with reduced density.

Furthermore, the increased viscosity results in a much lower Reynolds number, as in Table 5.5, which is no longer comparable to the original case with geometry *iii.* The Prandtl number for this case is also much greater, so again the thermal boundary layer is very thin. Although there are some small variations in density, the previously observed instability does not appear within the 300 s simulated. Therefore, confirming that both high Rayleigh number and high Prandtl number are necessary for the instability to appear.

Compared to the values found for a real melter in Section 2.2.2 of the literature review, and rewritten in Table 5.3, the Reynolds number is six orders of magnitude smaller. Therefore, the proposed case may not capture all the physics of the flow taking place during vitrification. For realistic vitrification flows, the Rayleigh number is significantly higher, as expected with the higher density, but the Prandtl number is lower, so it is unclear whether the instability would appear with these values.

5.3.2 Two Phases with Temperature-Dependent Viscosity

phase	phase number	A	B	C
38wt% HLW	1	13.03	-5715	-536800
glass frit	2	15.83	-3443	-174200

Table 5.6: Parameter values for two phase vitrification case with temperature-dependent viscosity, using the Jenckel3 model (Equation (3.83)).

Next, it is assumed that the mixture inside the melter has separated into one phase of purely borosilicate glass and another glass phase containing all the HLW being vitrified. Therefore, the base glass frit and the Magnox blend with 38wt% HLW from the VTR are chosen as inputs to the MVA, with the Jenckel3 model in Equation (3.83). The resulting parameters for each phase are shown in Table 5.6. Again, a constant initial density of $\rho = 30 \text{ kg m}^{-3}$ is used. The reference viscosity of $\overline{\nu} = 6.7 \times 10^{-4}$ $\text{m}^2\,\text{s}^{-1}$ is used for both phases, so they differ only in the parameters used.

Heating and corresponding change in viscosity adjacent to the curved boundary is seen in Figures 5.30 and 5.31. The difference in viscosity between high and low temperatures is greater in phase 2 (the glass frit), showing the effect of different viscosity parameters in each phase. As in the single phase case in Section 5.3.1, the Reynolds number is low and there is little movement and no circulation within the first few minutes of simulation. There are again very small variations in density, but no appearance of the instability. The global mixing measure in Figure 5.32 confirms

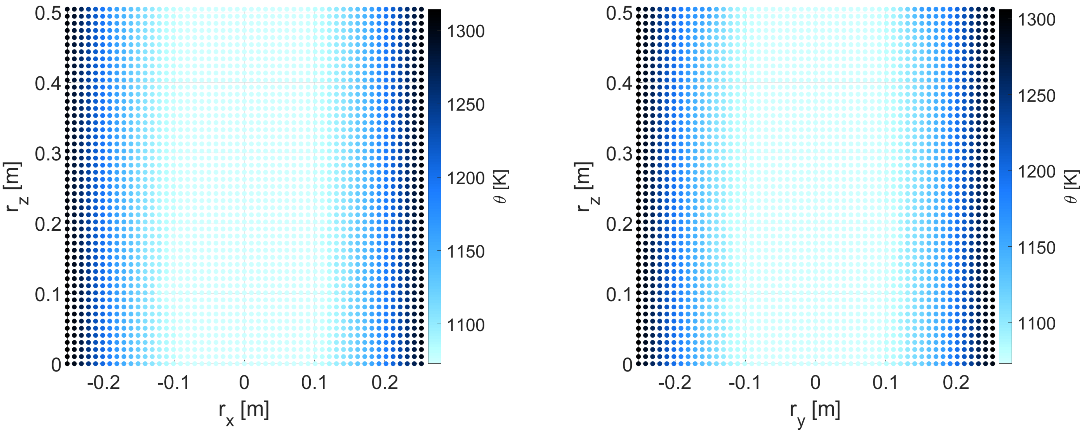


Figure 5.30: Temperature in $x = 0$ and $y = 0$ planes at 60 s for two-phase melter with geometry *iii*, $\rho = 30\ \mathrm{kg\,m^{-3}}$ and temperature-dependent viscosity.

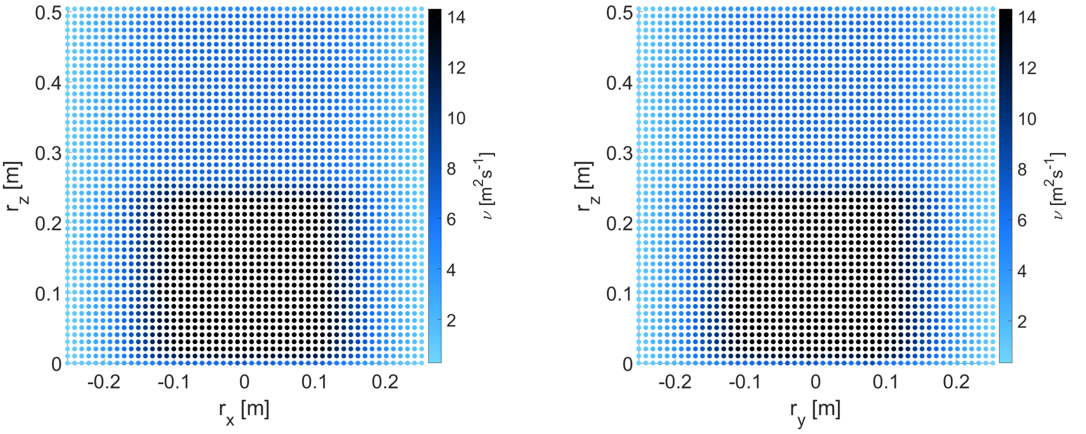


Figure 5.31: Viscosity in $x = 0$ and $y = 0$ planes at 60 s for two-phase melter with geometry *iii*, $\rho = 30\ \mathrm{kg\,m^{-3}}$ and temperature-dependent viscosity.

that overall mixing in the $y = 0$ plane is increasing, but remains at a low value by 300 s.

5.4 Concluding Remarks

The initial cases with geometry *i* demonstrated buoyancy-driven circulation within a cylindrical tank heated at the curved wall. When two phases are present, the flow

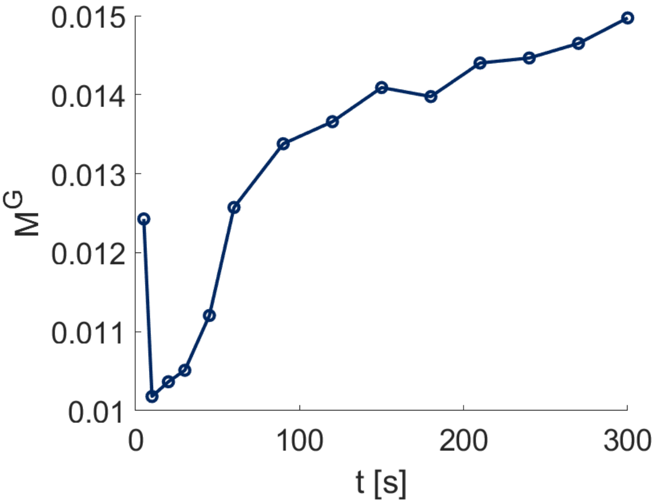


Figure 5.32: Global mixing measure (VF-FTLE) with time for two-phase melter with geometry *iii*, $\rho = 30\ \mathrm{kg\,m^{-3}}$ and temperature-dependent viscosity.

causes gradual mixing between the phases. Local mixing measures show the movement of phases, while the distinctions of the new VF-FTLE are accentuated where the regions of high circulation lie. The global mixing measures give insight into the overall degree of mixing achieved at an instant, as well as the progress towards homogeneity over time.

Further cases with dimensions estimating those of a vitrification melter are also run but, although some initially show heating, an instability occurs. This is determined to be related to thin thermal boundary layers present in these cases and can be avoided by decreasing the Rayleigh number by two orders of magnitude. To simulate these cases with realistic Rayleigh number, further modifications must be made to the model so that it can resolve the physics present in a thin thermal boundary layer at high Prandtl numbers. Further understanding of the cause would be gained by performing related experiments of two-phase mixing in a heated cylindrical cavity, which could then be simulated using this model for comparison.

In the first case there is also loss of particles through layers of boundary particles. It is likely caused by the cylindrical geometry and requires extra consideration of the mDBC extrapolation process for such cases. For later cases, the DBC is used instead, but this method is known to produce inaccurate densities and velocity gradients near boundaries.

When including variable viscosity through the temperature-dependent viscosity equations, the kinematic viscosity depends also on density. Therefore, reducing the Rayleigh number to avoid instability by decreasing density also decreases the Reynolds number of the case significantly. Consequently, the temperature-dependent viscosity cases do not show formation of circulatory flow in the time frames currently simulated.

Chapter 6

Conclusions and Recommendations

6.1 General Conclusions

This thesis aimed to produce a model of mixing of separated phases during the vitrification of High Level Waste (HLW) in order to determine under what criteria these phases are fully mixed or remain separated in the vitrified product.

To do this, a literature review was first conducted in Chapter 2 to understand the vitrification process and why separation occurs, as well as current models for vitrification and the state of the vitrified product. A gap was identified in the modelling of mixing within the melter, with most models focussing on melting within the cold cap or pouring from the melter. Due to the dangerous nature of the materials concerned, many models regarding separation of the vitrified product are empirical and do not simulate the physical processes. Key dimensionless numbers are identified and calculations performed using available parameter values. As a result, it is determined that surface tension is negligible, whilst convection is key for both heat transfer and fluid dynamics. Methods for measuring the state of mixing quantitatively at a given time are also investigated.

To determine the best modelling approach, existing work using Computational Fluid Dynamics (CFD) for mixing and multi-phase flows was reviewed, leading to the choice of Smoothed Particle Hydrodynamics (SPH) as the method for this work. Its mesh-free formulation makes it ideally suited to the mixing and flows taking place during vitrification. In particular, large deformations within the fluid can be modelled without any requirement for re-meshing. SPH can also follow the interface between phases naturally, even as it deforms or becomes inter-connected. Previous work in the literature using SPH has modelled multi-component flows, especially focussing on air and water phases.

The existing open-source software DualSPHysics [55], which has been rigorously validated, is chosen as a basis for the computational model. A number of modifications and additions are made to tailor it to the requirements of the vitrification model, as described in Chapter 3. This is the first time that these functionalities have been introduced together, so that new physics of buoyancy-driven flow of multiple phases driven by heated boundaries can be investigated. All changes are made to both the central processing unit (CPU) and graphics processing unit (GPU) code, so the model can be run on either to take advantage of available hardware.

Firstly, arrays are added for viscosity and temperature variables, and can also be included in output files. In the current release, viscosity is a constant parameter and all cases are isothermal. The viscous term in the governing momentum equation is modified so that the value of the average viscosity of the two interacting SPH particles is used in place of the constant case viscosity. Alternative viscous operators are also considered in Section 3.3.6, taking into account the potential variations in viscosity for the intended model.

A governing equation for temperature is added, allowing a change in temperature to occur where there is a difference in temperature between neighbouring SPH particles. Heating is included by applying a constant temperature to a boundary. To model buoyancy-driven flows, the Boussinesq approximation is adopted and included in the momentum equation calculations. Therefore, the density is not directly dependent on the temperature.

Multiple phases are included by extending the recently released non-Newtonian multi-phase code [68], with viscosity options added so that each phase can have a constant viscosity or follow a temperature-dependent equation with its own set of parameters. Modifications allow these phase options to also be applied to boundary phases. An adiabatic option is developed, based on the modified dynamic boundary condition (mDBC) method, and is included to be applied to boundary phases. All new options can be specified to use in the input xml case file, with examples in Appendix A.

The mixing measures to be investigated and used in this work are outlined in

Section 3.6. A new measure is also defined as a combination of existing measures, which is later demonstrated to provide further insight into the mixing process where existing measures cannot. By combing both volume fraction (VF) and finite-time Lyapunov exponent (FTLE), regions of separated phases can be identified as well as structures of the flow which are causing or inhibiting mixing. The new measure is one of the novel contributions of this thesis.

Each addition to the model is rigorously validated in Chapter 4 with a range of test cases. Where possible, an analytical solution is used. In the cases of the lid-driven cavity (Section 4.3) and differentially heated cavity (Section 4.4), results are compared with well-regarded numerical results from the literature. Where this is missing for the two-phase lid-driven cavity in Section 4.3.3, a benchmark solution is obtained using commercial software. This validation ensures that the model is working as designed and can give reasonable results for vitrification, in order to better understand the mixing process.

Finally, in Chapter 5 a model vitrification case is proposed. Initial cases with constant viscosity show promising circulation driven by buoyancy. However, when parameters closer to those expected for vitrification are used the simulation suffers an instability. This is determined to be related to both the Rayleigh and Prandtl numbers. By reducing the density, single and two-phase cases with constant and temperature-dependent viscosities are run and both the flow and mixing behaviour observed.

The numerical models should be useful to CFD modellers and industrial engineers in fields where vitrification and similar mixing processes take place.

6.2 Detailed Conclusions and Key Results

The objectives listed in Section 1.4 are each met, as described above. Conclusions on the aim of modelling mixing of separated phases during vitrification are given in Section 6.2.4. Here, key results found during each part of this work are highlighted.

6.2.1 Mixing Multiple Viscous Phases and Flow Instability

The capability of modelling two phases of the same density but different viscosity is demonstrated by the two-phase Poiseuille flow in in Section 4.2.2, where velocity along the channel closely matches the analytical solution. This SPH model is also able to correctly predict the onset of long-wavelength instabilities which appear at the interface under certain combinations of parameters. Moreover, the growth rate is accurately replicated and the interface can be seen deforming as the instability progresses. It is not possible to obtain these results using a mesh-based volume-of-fluid (VOF) method, where the interface must be reconstructed. The accuracy of the instability growth rate when compared to an analytical solution implies that the Shao and Lo [205] viscous operator is sufficient for these cases.

6.2.2 Buoyancy-Driven Flow

The effectiveness of the chosen form of the governing equation for temperature is shown by the successful formation of the temperature gradient within a cavity heated at one side in Section 4.4.1. Furthermore, the adoption of the Boussinesq approximation for buoyancy-driven flow produces good results for the differentially heated cavity case in Section 4.4.2. A larger smoothing length was required to increase the accuracy of results, but these remain close to the benchmark results when extended to a pseudo-3D case with periodic boundary. A new method to enforce an adiabatic boundary condition, based on the new mDBC option in DualSPHysics, improved temperature gradients in proximity of the respective boundary.

6.2.3 New Mixing Measure

Each of the mixing measures in Section 3.6 was applied to the case of a two-phase lid-driven cavity in Section 4.3.3. The commonly found single-phase case was also run in Section 4.3.2 as an example of mechanically-driven mixing. Both the VF and FTLE were found to give useful insights into the degree of mixing, whilst neither showed the full picture. Using the VF shows where phases are at an instant of time, whereas the FTLE highlights flow structures where the most movement between SPH particles is taking place.

A new mixing measure was defined which combines both VF and FTLE to gain the advantages of both in one measure. The new measure is found to give close results to the local mixing measure of Robinson [188] is some cases, whilst in others clear differences are seen where the FTLE, and correspondingly relative particle movement, is low. Therefore, the novelty of this measure is to give information about both the state mixing in terms of phase location and relative movement over time in one value. A global mixing measure is also derived from the new VF-FTLE mixing measure which can be plotted against time, allowing observation of the change in degree of mixing. In this way, it can be seen whether mixing is continuing to increase or reaching a steady state where the phases are never fully mixed.

6.2.4 Vitrification Model

Vitrification is modelled in Chapter 5 by defining a simplified case which includes the key physical processes. This is an exploratory application reliant on the previously validated SPH model. It is found that the model can successfully simulate buoyancy-driven circulation within a cylindrical tank with heated walls. This is demonstrated by the plots of temperature, velocity vectors and FTLE. Furthermore, with two phases of constant viscosity the mixing measures show the change in distribution of each phase and the global measure shows the degree of mixing increasing with time.

Increasing the tank to the proposed melter geometry with a realistic density was found to be unstable when the thermal boundary layer is thin. However, cases with a reduced density, and therefore lower Rayleigh and Prandtl numbers, showed the model runs successfully when the thermal boundary layer is closer in thickness to the velocity boundary layer. Additional developments are needed to simulate cases with very thin thermal boundary layers, as well as further investigation into the cause of the instability.

Introducing temperature-dependent viscosity with parameters from the melt viscosity algorithm has a significant effect on the viscosity of the fluid phases where heating has taken place. However, the lower density required for stability results in large kinematic viscosities and small Reynolds numbers. Therefore, circulation is not seen for these cases.

6.2.5 Limitations of the Model and Validation Tests

The instability investigated in detail, which appears in two-phase Poiseuille flow, is considered only in two dimensions. Therefore, the model is not directly validated for an instability occurring in a three-dimensional flow. Understanding of the vitrification case in Chapter 5 and the origin of the instabilities found is limited by the lack of literature on this particular case. No numerical or experimental work could be found on a filled cylindrical tank heated through the curved surface. The case has not currently been run for the lengths of time vitrification takes place in practice.

The SPH formulation limits accuracy in some places. The order of convergence of conventional SPH methods is low and, although it is increased by modifications such as the mDBC, further improvements exist which could be implemented. The low order of convergence means a high resolution is needed to reach a high accuracy solution. The resulting large number of particles increases computation time, particularly for three-dimensional simulations.

The physics included in the model is limited by the time taken to both add to the code and validate sufficiently rigorously during this project. The key physical processes were chosen as priorities, but others may still have some effect of the results. For example, internal heating due to radiation contributes to the overall heating within the melter. Since the model is no longer isothermal, an equation of state including temperature may improve results for the cases where heating is significant.

The idealised vitrification case in Chapter 5 assumes the mixture is already separated into two phases. The phases chosen currently are the best approximations found in the Melt Viscosity Alogirthm (MVA) data, so may not truly represent the composition of the separated phases.

6.3 Recommendations for Future Research

There are a number of different directions further work on this model could take. Some suggestions are listed below.

6.3.1 Three-Dimensional Investigation of the Poiseuille Flow Instability

To validate a three-dimensional instability, the two-phase Poiseuille flow case could be extended to a pseudo-three-dimensional case, as the differentially heated cavity was in Section 4.4.2.1. More thoroughly, a review into whether a three-dimensional case has been previously investigated, either numerically or analytically, should be undertaken. If suitable results are found then they can be used to validate the model, otherwise a related case of two-phase flow could be used for validation.

6.3.2 Improvements to the SPH Formulation

To increase the accuracy of the solution at a reasonable resolution, a high-order formulation should be included. Many high-order methods use an incompressible SPH (ISPH) formulation. The method of Lind and Stansby [126] is developed in ISPH, but is also demonstrated for an Arbitrary-Lagrangian–Eulerian (ALE) approach which may be integrated with the current physics included in this model.

Modifications need to be made in order to accurately model the thin thermal boundary layer present in the vitrification melter cases. Recent work by Hosain et al. [94] using SPH to model heat transfer in industrial processes acknowledges that the thermal boundary layer is not fully resolved by current SPH models. Advanced treatments need to be developed, for example variable resolution in the thermal boundary layer or more accurate temperature boundary conditions [94].

6.3.3 Experimental Investigation of Two-Phase Vitrification in a Thermally Heated Domain

Since the proposed melter case in Chapter 5 is an estimation of the geometry and composition involved in vitrification, and there are no measurements during the process itself, it would be valuable to undertake a series of corresponding experiments. The physical model can then be used to validate two-phase convective flow driven by heating through the curved wall of a cylindrical tank, and any physical phenomena or instabilities present can be better understood.

Measurements of the location of each phase at a series of times could be used for comparisons with the VF results from the model. Additionally, use of a tracer dye would reveal the locations of manifolds of the flow which could be compared with the FTLE. A standard deviation of phase concentration could be plotted against the global measures of the computational data and the rate of change of mixing compared.

6.3.4 Further Development of Vitrification Model

Greater understanding of the source of the instability from experiments would allow longer periods of time to be simulated, closer to the hours taken for vitrification to complete. This knowledge would also determine whether density differences contribute to the instability and loss of SPH particles, or whether different phase densities can then be included, given the difference is small, without changes to the SPH formulation.

Viscometry measurements for the expected separated phase compositions, which are known to be silica-rich and borate-rich respectively, would provide new input data for the melt viscosity algorithm [144]. This would produce more accurate viscosity model parameters for the two phases.

6.3.5 Additions for a More Realistic Vitrification Case

Once the simplified vitrification case is validated experimentally, the model could be extended in different ways. For example, an air phase could be included at the top of the tank either as an actual phase or absence of SPH particles. This would require extra consideration into the densities involved, as the density ratio between glass and air phases is high, even if the density ratio between separated glass phases is small. A dependency of density on temperature could be included, as well as taking into account temperature changes in the equation of state. Although the current equation of state performs well in the validation cases, tailoring it to the material concerned may improve results for the vitrification case.

Alternatively, the model could be developed to include chemical phenomena which are known to be present. The high levels of radioactivity contribute to heating, meaning the waste-containing phase itself could be included as a heat source. A mechanism for phase separation under certain conditions could also be investigated, with the fluid beginning as a single phase. Thus, a homogeneous vitrified product could also be achieved when no phase separation takes place.

The model could also be extended to include additional physical processes involved in vitrification. Glass frit enters the melter as a solid and melts within the cold cap. The effect on the convective flow of the gradual addition of molten glass may influence mixing of any separated phases. Separation of phases and further mixing can also occur during pouring into Vitrification Product Container (VPC)s and subsequent cooling. Pouring of vitrified waste has been investigated previously using numerical models, but the longer timescales involved in cooling would make it difficult to include in the same model.

6.3.6 Extending the New Mixing Measure

Currently, the new mixing measure defined in Section 3.6.5 can be applied to only two phases. Since its benefit of quantifying both phase position and particle movement in a single value would be beneficial in many applications, it would be advantageous to extend the measure to three or more phases. The FTLE component is independent of phase, but the VF component must be modified. This should take into account how the local volume fraction of all phases present compares to its corresponding global fraction.

6.3.7 Further Code Development

In order for internal heating to be included, a heat source term must be added to the governing equation for temperature. The development of a heterogeneous Neumann boundary condition for temperature would also allow heating to be applied through a flux. Similar to the adiabatic boundary method developed in this work, viscosity could also be extrapolated into the boundary using a method based on mDBC.

The code modifications and developments currently made for this model require further validation before being included in a future release of DualSPHysics. Many of the additions are applicable to a wide variety of cases which may be of interest to users of the software. In particular, inclusion of the mixing measures to the post-processing tools would provide insight into many cases both with multi-phase and, with the FTLE, single phase flow.

For three-dimensional simulations, the numbers of particles can be very large for a resolution that captures the significant physical processes. In the pseudo-three-dimensional differentially heated cavity case the smoothing length required for an accurate solution was greater than in two dimensions, increasing computation time as the number of particles included in each summation is greater. Extending the SPH formulation to a multi-GPU implementation would decrease computation time and allow higher resolution simulations to be run.

Bibliography

[1] S. Adami, X. Y. Hu, and N. A. Adams. A new surface-tension formulation for multi-phase SPH using a reproducing divergence approximation. *Journal of Computational Physics*, 229(13):5011–5021, 2010.

[2] S. Adami, X. Y. Hu, and N. A. Adams. A generalized wall boundary condition for smoothed particle hydrodynamics. *Journal of Computational Physics*, 231 (21):7057–7075, 2012. doi: 10.1016/j.jcp.2012.05.005.

[3] Advanced Fuel Cycle Programme. Fuel cycle themes, 2021. URL `https://afcp.nnl.co.uk/fuel-cycle-themes/`. accessed 24-09-2021.

[4] B. J. Alder and T. E. Wainwright. Studies in molecular dynamics. I. general method. *The Journal of Chemical Physics*, 31(2):459–466, 1959.

[5] F. J. Alexander, S. Chen, and D. W. Grunau. Hydrodynamic spinodal decomposition: Growth kinetics and scaling functions. *Physical Review B*, 48(1):634, 1993.

[6] Altair. nanoFluidX. URL `altair.com/altaircfd-capabilities/#sph`. accessed 11-08-2021.

[7] A. Amicarelli, B. Kocak, S. Sibilla, and J. Grabe. A 3D smoothed particle hydrodynamics model for erosional dam-break floods. *International Journal of Computational Fluid Dynamics*, 31(10):413–434, 2017. doi: https://dx.doi.org/10.1080/10618562.2017.1422731.

[8] J. D. Anderson. Computational fluid dynamics: the basics with applications. 1995.

[9] B. Andersson, R. Andersson, L. Hakansson, M. Mortensen, R. Sudiyo, and B. van Wachem. *Computational Fluid Dynamics for Engineers*. Cambridge University Press, 2011. ISBN 9781139093590.

[10] Ansys. LS-DYNA. URL `https://www.lstc.com/products/ls-dyna`. accessed 11-08-2021.

[11] Ansys. Ansys Fluent - CFD software, 2016. URL `http://www.ansys.com/products/fluids/ansys-fluent`. accessed 09-07-2021.

[12] M. Antuono, A. Colagrossi, S. Marrone, and D. Molteni. Free-surface flows solved by means of SPH schemes with numerical diffusive terms. *Computer Physics Communications*, 181(3):532–549, 2010.

[13] N. R. Anturkar, T. C. Papanastasiou, and J. O. Wilkes. Linear stability analysis of multilayer plane Poiseuille flow. *Physics of Fluids A: Fluid Dynamics*, 2(4): 530–541, 1990.

[14] Autorite de Surete Nucleaire. France's answers to questions and comments received from other contracting parties on its second report for the JC. In *Joint Convention on the safety of spent fuel management and on the safety of radioactive waste management*.

[15] J. B. Avalos, M. Antuono, A. Colagrossi, and A. Souto-Iglesias. Shear-viscosity-independent bulk-viscosity term in smoothed particle hydrodynamics. *Physical Review E*, 101(1):013302, 2020.

[16] S. Babnik, V. Erkalvec Zajec, B. Oblak, B. Likozar, and A. Pohar. A review of computational fluid dynamics (CFD) simulations of mixing in the pharmaceutical industry. *Biomedical Journal of Scientific & Technical Research*, 3(27): 20732–20736, 2020.

[17] B. Bai, D. Rao, T. Xu, and P. Chen. SPH-FDM boundary for the analysis of thermal process in homogeneous media with a discontinuous interface. *International Journal of Heat and Mass Transfer*, 117:517–526, 2018.

[18] G. Barakos, E. Mitsoulis, and D. O. Assimacopoulos. Natural convection flow in a square cavity revisited: laminar and turbulent models with wall functions. *International Journal for Numerical Methods in Fluids*, 18(7):695–719, 1994.

[19] L. Barreira. *Lyapunov Exponents.* Springer International Publishing, 2017. ISBN 9783319712611.

[20] M. Basa, N. J. Quinlan, and M. Lastiwka. Robustness and accuracy of SPH formulations for viscous flow. *International Journal for Numerical Methods in Fluids*, 60(10):1127–1148, 2009. doi: 10.1002/fld.1927.

[21] T. Belytschko, Y. Y. Lu, and L. Gu. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37(2):229–256, 1994.

[22] D. Bestion. Applicability of two-phase CFD to nuclear reactor thermalhydraulics and elaboration of best practice guidelines. *Nuclear Engineering and Design*, 253: 311–321, 2012.

[23] G. Bilotta, A. Hérault, A. Cappello, G. Ganci, and C. Del Negro. GPUSPH: a smoothed particle hydrodynamics model for the thermal and rheological evolution of lava flows. *Geological Society, London, Special Publications*, 426(1): 387–408, 2016.

[24] P. A. Bingham and R. J. Hand. Vitrification of toxic wastes: a brief review. *Advances in Applied Ceramics*, 105(1):21–31, 2006.

[25] R. B. Bird, W. E. Stewart, and E. N. Lightfoot. *Transport Phenomena.* John Wiley & Sons, 2007.

[26] J. Bonet and T.-S. L. Lok. Variational and momentum preservation aspects of smooth particle hydrodynamic formulations. *Computer Methods in Applied Mechanics and Engineering*, 180(1-2):97–115, 1999.

[27] J. Bonet Avalos, A. Colagrossi, M. Antuono, and A. Souto-Iglesias. Formulation for an angular momentum conservative SPH bulk viscosity term. In *Proceedings of the 2020 SPHERIC Harbin International Workshop.*

[28] L. Brookshaw. *The Stability of Binary Systems and Rotating Stars*. PhD thesis, Monash University, 1986.

[29] M. Camesasca, M. Kaufman, and I. Manas-Zloczower. Quantifying fluid mixing with the Shannon entropy. *Macromolecular Theory and Simulations*, 15(8):595–607, 2006.

[30] R. B. Canelas, A. M. Ricardo, R. M. L. Ferreira, J. M. Domínguez, and A. J. C. Crespo. Hunting for Lagrangian coherent structures: SPH-LES turbulence simulations with wall-adapting local eddy viscosity (WALE) model. In *Proceedings of the 11th International Smoothed Particle Hydrodynamics European Research Interest Community (SPHERIC) workshop, Munich, Germany*, 2016.

[31] D. Caurant, P. Loiseau, O. Majerus, V. Aubin-Chevaldonnet, I. Bardez, and A. Quintas. *Glasses, Glass-Ceramics and Ceramics for Immobilization of Highly Radioactive Nuclear Wastes*. Nova Science, 2007.

[32] Centroid Lab. neutrino. URL `https://centroidlab.com/neutrinodynamics`. accessed 11-08-2021.

[33] J. L. Cercos-Pita. AQUAgpusph, a new free 3D SPH solver accelerated with OpenCL. *Computer Physics Communications*, 192:295–312, 2015.

[34] S. L. Chang, C. Q. Zhou, and B. Golchert. Eulerian approach for multiphase flow simulation in a glass melter. *Applied Thermal Engineering*, 25(17-18):3083–3103, 2005.

[35] M. E. Charles and L. U. Lilleleht. An experimental investigation of stability and interfacial waves in co-current flow of two liquids. *Journal of Fluid Mechanics*, 22(2):217–224, 1965.

[36] F. Charru and J. Fabre. Long waves at the interface between two viscous fluids. *Physics of Fluids*, 6(3):1223–1235, 1994.

[37] S. Chen and G. D. Doolen. Lattice Boltzmann method for fluid flows. *Annual Review of Fluid Mechanics*, 30(1):329–364, 1998.

[38] P. W. Cleary. Modelling confined multi-material heat and mass flows using SPH. *Applied Mathematical Modelling*, 22(12):981–993, 1998.

[39] P. W. Cleary and J. J. Monaghan. Conduction modelling using smoothed particle hydrodynamics. *Journal of Computational Physics*, 148(1):227–264, 1999.

[40] A. Colagrossi and M. Landrini. Numerical simulation of interfacial flows by smoothed particle hydrodynamics. *Journal of Computational Physics*, 191(2): 448–475, 2003.

[41] A. Colagrossi, B. Bouscasse, and S. Marrone. Energy-decomposition analysis for viscous free-surface flows. *Physical Review E*, 92(5):053003, 2015.

[42] A. Colagrossi, D. Durante, J. B. Avalos, and A. Souto-Iglesias. Discussion of Stokes' hypothesis through the smoothed particle hydrodynamics model. *Physical Review E*, 96(2):023101, 2017.

[43] A. J. C. Crespo, M. Gómez-Gesteira, and R. A. Dalrymple. Boundary conditions generated by dynamic particles in SPH methods. *CMC-Tech Science Press*, 5 (3):173, 2007.

[44] A. J. C. Crespo, J. M. Domínguez, B. D. Rogers, M. Gómez-Gesteira, S. Longshaw, R. Canelas, R. Vacondio, A. Barreiro, and O. García-Feal. DualSPHysics: Open-source parallel CFD solver based on smoothed particle hydrodynamics (SPH). *Computer Physics Communications*, 187:204–216, 2015.

[45] S. J. Cummins and M. Rudman. An SPH projection method. *Journal of Computational Physics*, 152(2):584–607, 1999.

[46] A. K. Das and P. K. Das. Modeling of liquid–vapor phase change using smoothed particle hydrodynamics. *Journal of Computational Physics*, 303:125–145, 2015.

[47] T. F. Dauch, C. Ates, M. C. Keller, G. Chaussonnet, J. Kaden, M. Okraschevski, R. Koch, H.-J. Bauer, T. Rapp, and C. Dachsbacher. Analyzing primary breakup in fuel spray nozzles by means of Lagrangian-coherent structures. In *Proceedings of the 14th SPHERIC International Workshop.*

[48] T. F. Dauch, T. Rapp, G. Chaussonnet, S. Braun, M. C. Keller, J. Kaden, R. Koch, C. Dachsbacher, and H.-J. Bauer. Highly efficient computation of finite-time Lyapunov exponents (FTLE) on GPUs based on three-dimensional SPH datasets. *Computers & Fluids*, 175:129–141, 2018.

[49] G. de Vahl Davis. Natural convection of air in a square cavity: a bench mark numerical solution. *International Journal for Numerical Methods in Fluids*, 3 (3):249–264, 1983.

[50] W. Dehnen and H. Aly. Improving convergence in smoothed particle hydrodynamics simulations without pairing instability. *Monthly Notices of the Royal Astronomical Society*, 425(2):1068–1082, 2012.

[51] Department for Business, Innovation & Skills and Department of Energy & Climate Change. Long-term nuclear energy strategy. URL `https://www.gov.uk/government/publications/long-term-nuclear-energy-strategy`. accessed 09-08-2021.

[52] Department of Energy & Climate Change. Map showing nuclear power stations in the UK. URL `https://www.gov.uk/government/publications/map-of-nuclear-power-stations-in-the-uk`. accessed 09-08-2021.

[53] G. Dhatt. *Finite Element Method*. Numerical Methods Series. ISTE ; Wiley, 2012. ISBN 9781118569764.

[54] J. M. Domínguez, A. J. C. Crespo, M. Gómez-Gesteira, and J. C. Marongiu. Neighbour lists in smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids*, 67(12):2026–2042, 2011.

[55] J. M. Domínguez, G. Fourtakas, C. Altomare, R. B. Canelas, A. Tafuni, O. García-Feal, I. Martínez-Estévez, A. Mokos, R. Vacondio, A. J. C. Crespo, et al. DualSPHysics: from fluid dynamics to multiphysics problems. *Computational Particle Mechanics*, pages 1–29, 2021.

[56] DualSPHysics. Test cases. URL `https://github.com/DualSPHysics/DualSPHysics/wiki/7.-Testcases#71-main-examples`. accessed 22-07-2021.

[57] A. Eitzlmayr, G. Koscher, and J. Khinast. A novel method for modeling of complex wall geometries in smoothed particle hydrodynamics. *Computer Physics Communications*, 185(10):2436–2448, 2014.

[58] A. English, J. M. Domínguez, R. Vacondio, A. J. C. Crespo, P. K. Stansby, S. J. Lind, and M. Gómez-Gesteira. Correction for dynamic boundary conditions. In *Proceedings of the 14th SPHERIC International Workshop*.

[59] A. English, J. M. Domínguez, R. Vacondio, A. J. C. Crespo, P. K. Stansby, S. J. Lind, L. Chiapponi, and M. Gómez-Gesteira. Modified dynamic boundary conditions (mDBC) for general-purpose smoothed particle hydrodynamics (SPH): application to tank sloshing, dam break and fish pass problems. *Computational Particle Mechanics*, pages 1–15, 2021.

[60] N. Erkan, T. Kawakami, H. Madokoro, P. Chai, Y. Ishiwatari, and K. Okamoto. Numerical simulation of droplet deposition onto a liquid film by VOF–MPS hybrid method. *Journal of visualization*, 18(2):381–391, 2015.

[61] R. Fatehi and M. T. Manzari. Error estimation in smoothed particle hydrodynamics and a new scheme for second derivatives. *Computers & Mathematics with Applications*, 61(2):482–498, 2011.

[62] Z.-G. Feng and M. C. Ponton. Smoothed particle method for studying heat and mass transfer between fluid and solid. In *ASME 2014 International Mechanical Engineering Congress and Exposition*, pages V08BT10A018–V08BT10A018. American Society of Mechanical Engineers, 2014.

[63] M. Ferrand, D. R. Laurence, B. D. Rogers, D. Violeau, and C. Kassiotis. Unified semi-analytical wall boundary conditions for inviscid, laminar or turbulent flows in the meshless SPH method. *International Journal for Numerical Methods in Fluids*, 71(4):446–472, 2013.

[64] A. Ferrari, M. Dumbser, E. F. Toro, and A. Armanini. A new 3D parallel SPH scheme for free surface flows. *Computers & Fluids*, 38(6):1203–1217, 2009.

[65] J. H. Ferziger. *Computational Methods for Fluid Dynamics.* Springer, 3rd rev. ed. edition, 2002. ISBN 3540420746.

[66] N. Filipovic, M. Ivanovic, and M. Kojic. A comparative numerical study between dissipative particle dynamics and smoothed particle hydrodynamics when applied to simple unsteady flows in microfluidics. *Microfluidics and Nanofluidics*, 7(2):227–235, 2009.

[67] A. Fluegel. Phase separation in borosilicate and alkali earth silicate glasses. URL `http://glassproperties.com/phase-separation/`. accessed 17-07-2019.

[68] G. Fourtakas and B. D. Rogers. Modelling multi-phase liquid-sediment scour and resuspension induced by rapid flows using smoothed particle hydrodynamics (SPH) accelerated with a graphics processing unit (GPU). *Advances in Water Resources*, 92:186–199, 2016.

[69] G. Fourtakas, J. M. Dominguez, R. Vacondio, and B. D. Rogers. Local uniform stencil (LUST) boundary condition for arbitrary 3-D boundaries in parallel smoothed particle hydrodynamics (SPH) models. *Computers & Fluids*, 190: 346–361, 2019.

[70] G. Fourtakas, R. Vacondio, J. M. Domínguez, and B. D. Rogers. Improved density diffusion term for long duration wave propagation. In *Proceedings of the International SPHERIC Workshop, Harbin, China*, 2020.

[71] Fraunhofer, I. W. M. SimPARTIX. URL `https://www.simpartix.com/`. accessed 11-08-2021.

[72] A. Ghaïtanellis, D. Violeau, M. Ferrand, K. E. K. Abderrezzak, A. Leroy, and A. Joly. A SPH elastic-viscoplastic model for granular flows and bedload transport. *Advances in Water Resources*, 111:156–173, 2018. doi: https://dx.doi.org/10.1016/j.advwatres.2017.11.007.

[73] U. K. N. G. Ghia, K. N. Ghia, and C. T. Shin. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics*, 48(3):387–411, 1982.

[74] R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181(3):375–389, 1977.

[75] A. Goel, J. S. McCloy, R. Pokorny, and A. A. Kruger. Challenges with vitrification of Hanford high-level waste (HLW) to borosilicate glass – an overview. *Journal of Non-Crystalline Solids: X*, 4:100033, 2019.

[76] M. Gómez-Gesteira, B. D. Rogers, R. A. Dalrymple, and A. J. C. Crespo. State-of-the-art of classical SPH for free-surface flows. *Journal of Hydraulic Research*, 48:6–27, 2010. ISSN 0022-1686.

[77] M. Gómez-Gesteira, A. J. C. Crespo, B. D. Rogers, R. A. Dalrymple, J. M. Dominguez, and A. Barreiro. SPHysics–development of a free-surface fluid solver–part 2: Efficiency and test cases. *Computers & Geosciences*, 48:300–307, 2012.

[78] M. Gomez-Gesteira, B. D. Rogers, A. J. C. Crespo, R. A. Dalrymple, M. Narayanaswamy, and J. M. Dominguez. SPHysics–development of a free-surface fluid solver–part 1: Theory and formulations. *Computers & Geosciences*, 48:289–299, 2012.

[79] D. D. Gray and A. Giorgini. The validity of the boussinesq approximation for liquids and gases. *International Journal of Heat and Mass Transfer*, 19(5):545–551, 1976.

[80] N. Grenier, M. Antuono, A. Colagrossi, D. Le Touzé, and B. Alessandrini. An Hamiltonian interface SPH formulation for multi-fluid and free surface flows. *Journal of Computational Physics*, 228(22):8380–8393, 2009.

[81] N. Gribble and R. Short. The vitrification test rig: a research and development tool for high level waste vitrification. *NNL Science*, 2:3–11, 2014.

[82] M. Griebel, T. Dornseifer, and T. Neunhoeffer. *Numerical simulation in fluid dynamics: a practical introduction.* SIAM, 1998.

[83] D. Grunau, S. Chen, and K. Eggert. A lattice Boltzmann model for multiphase fluid flows. *Physics of Fluids A: Fluid Dynamics*, 5(10):2557–2562, 1993.

[84] D. Gu, Z. Liu, J. Li, Z. Xie, C. Tao, and Y. Wang. Intensification of chaotic mixing in a stirred tank with a punched rigid-flexible impeller and a chaotic

motor. *Chemical Engineering and Processing: Process Intensification*, 122:1–9, 2017.

[85] A. K. Gunstensen, D. H. Rothman, S. Zaleski, and G. Zanetti. Lattice Boltzmann model of immiscible fluids. *Physical Review A*, 43(8):4320, 1991.

[86] X. Guo, S. Gin, P. Lei, T. Yao, H. Liu, D. K. Schreiber, D. Ngo, G. Viswanathan, T. Li, S. H. Kim, et al. Self-accelerated corrosion of nuclear waste forms at material interfaces. *Nature materials*, 19(3):310–316, 2020.

[87] Z. Han, B. Su, Y. Li, W. Wang, W. Wang, J. Huang, and G. Chen. Numerical simulation of debris-flow behavior based on the SPH method incorporating the Herschel-Bulkley-Papanastasiou rheology model. *Engineering Geology*, 255:26–36, 2019. doi: https://dx.doi.org/10.1016/j.enggeo.2019.04.013.

[88] A. B. Harker and J. F. Flintoff. Crystalline-phase formation in hot isostatic pressing of nuclear waste ceramics with high zirconia content. *Journal of the American Ceramic Society*, 68(3):159–165, 1985.

[89] F. H. Harlow. PIC and its progeny. *Computer Physics Communications*, 48(1): 1–10, 1988.

[90] M. T. Harrison. Vitrification of high level waste in the UK. *Procedia Materials Science*, 7:10–15, 2014.

[91] S. Hinson. New nuclear power. URL `https://researchbriefings.files.parliament.uk/documents/CBP-8176/CBP-8176.pdf`. accessed 09-08-2021.

[92] A. P. Hooper. The stability of two superposed viscous fluids in a channel. *Physics of Fluids A: Fluid Dynamics*, 1(7):1133–1142, 1989.

[93] A. P. Hooper and W. G. C. Boyd. Shear-flow instability at the interface between two viscous fluids. *Journal of Fluid Mechanics*, 128:507–528, 1983.

[94] M. L. Hosain, J. M. Domínguez, R. B. Fdhila, and K. Kyprianidis. Smoothed particle hydrodynamics modeling of industrial processes involving heat transfer. *Applied Energy*, 252:113441, 2019.

[95] P. Hrma. Towards optimization of nuclear waste glass: Constraints, property models, and waste loading. Technical report, Pacific Northwest Lab., 1994.

[96] P. Hrma, B. M. Arrigoni, and M. J. Schweiger. Viscosity of many-component glasses. *Journal of Non-Crystalline Solids*, 355(14-15):891–902, 2009.

[97] P. Hrma, A. A. Kruger, and R. Pokorny. Nuclear waste vitrification efficiency: cold cap reactions. *Journal of Non-Crystalline Solids*, 358(24):3559–3562, 2012.

[98] X. Y. Hu and N. A. Adams. A multi-phase SPH method for macroscopic and mesoscopic flows. *Journal of Computational Physics*, 213(2):844–861, 2006.

[99] IMPETUS. Afea solver. URL `https://www.impetus.no/`. accessed 11-08-2021.

[100] International Atomic Energy Agency. *Classification of Radioactive Waste*, 2009.

[101] R. Issa, E. S. Lee, D. Violeau, and D. R. Laurence. Incompressible separated flows simulations with the smoothed particle hydrodynamics gridless method. *International Journal for Numerical Methods in Fluids*, 47(10-11):1101–1106, 2005.

[102] M. Jacoby. As nuclear waste piles up, scientists seek the best long-term storage solutions. *Chemical & Engineering News*, 98(12), 2020.

[103] P. F. James. Liquid-phase separation in glass-forming systems. *Journal of Materials Science*, 10(10):1802–1825, 1975. ISSN 1573-4803. doi: 10.1007/BF00554944.

[104] C. M. Jantzen. First principles process-product models for vitrification of nuclear waste: Relationship of glass composition to glass viscosity, resistivity, liquidus temperature, and durability. Technical report, Westinghouse Savannah River Co., 1991.

[105] C. M. Jantzen. Impact of phase separation on waste glass durability. Technical report, Savannah River Site, 1999.

[106] C. M. Jantzen. Using polymerization, glass structure, and quasicrystalline theory to produce high level radioactive borosilicate glass remotely: a 20+ year legacy. *Journal of the South Carolina Academy of Science*, 15(1):4, 2017.

[107] G. Jimeno, Y. C. Lee, and X.-W. Ni. Smoothed particle hydrodynamics–a new approach for modeling flow in oscillatory baffled reactors. *Computers & Chemical Engineering*, 124:14–27, 2019. doi: https://dx.doi.org/10.1016/j.compchemeng.2019.02.003.

[108] T. Jin, J. Chun, D. R. Dixon, D. Kim, J. V. Crum, C. C. Bonham, B. J. VanderVeer, C. P. Rodriguez, B. L. Weese, and M. J. Schweiger. Melter feed viscosity during conversion to glass: Comparison between low-activity waste and high-level waste feeds. *Journal of the American Ceramic Society*, 101(5): 1880–1891, 2018.

[109] J. C. Joubert, D. N. Wilke, N. Govender, P. Pizette, U. Tuzun, and N.-E. Abriak. 3D gradient corrected SPH for fully resolved particle–fluid interactions. *Applied Mathematical Modelling*, 78:816–840, 2020.

[110] T. G. Kang, S. Ahn, S. H. Chung, S. T. Chung, Y. S. Kwon, S. J. Park, and R. M. German. 9 - modeling and simulation of metal injection molding (MIM). In D. F. Heaney, editor, *Handbook of Metal Injection Molding*, Woodhead Publishing Series in Metals and Surface Engineering, pages 197–236e. Woodhead Publishing, 2012. ISBN 978-0-85709-066-9. doi: 10.1533/9780857096234.2.197.

[111] P.-H. Kao and R.-J. Yang. Simulating oscillatory flows in Rayleigh–Benard convection using the lattice Boltzmann method. *International Journal of Heat and Mass Transfer*, 50(17-18):3315–3328, 2007.

[112] T. W. Kao and C. Park. Experimental investigations of the stability of channel flows. part 2. two-layered co-current flow in a rectangular channel. *Journal of Fluid Mechanics*, 52(3):401–423, 1972.

[113] C. P. Kaushik, R. K. Mishra, P. Sengupta, A. Kumar, D. Das, G. B. Kale, and K. Raj. Barium borosilicate glass–a potential matrix for immobilization of sulfate bearing high-level radioactive liquid waste. *Journal of Nuclear Materials*, 358(2-3):129–138, 2006.

[114] O. M. Knio, H. N. Najm, and R. G. Ghanem. A stochastic projection method

for fluid flow: I. basic formulation. *Journal of Computational Physics*, 173(2): 481–511, 2001.

[115] S. Koshizuka and Y. Oka. Moving-particle semi-implicit method for fragmentation of incompressible fluid. *Nuclear Science and Engineering*, 123(3):421–434, 1996.

[116] N. Kreidl. Phase separation in glasses. *Journal of Non-Crystalline Solids*, 129 (1):1–11, 1991. ISSN 0022-3093. doi: 10.1016/0022-3093(91)90074-G.

[117] R. F. Kunz, D. A. Boger, D. R. Stinebring, T. S. Chyczewski, J. W. Lindau, H. J. Gibeling, S. Venkateswaran, and T. R. Govindan. A preconditioned Navier–Stokes method for two-phase flows with application to cavitation prediction. *Computers & Fluids*, 29(8):849–875, 2000.

[118] H. A. Kusch and J. M. Ottino. Experiments on mixing in continuous chaotic flows. *Journal of Fluid Mechanics*, 236:319–348, 1992.

[119] J. Kwon and J. J. Monaghan. A novel SPH method for sedimentation in a turbulent fluid. *Journal of Computational Physics*, 300:520–532, 2015.

[120] J. Kwon and J. J. Monaghan. Sedimentation in homogeneous and inhomogeneous fluids using SPH. *International Journal of Multiphase Flow*, 72:155–164, 2015.

[121] W. E. Langlois and M. O. Deville. *Slow viscous flow*, volume 173436. Springer, 1964.

[122] E.-S. Lee, C. Moulinec, R. Xu, D. Violeau, D. Laurence, and P. Stansby. Comparisons of weakly compressible and truly incompressible algorithms for the SPH mesh free particle method. *Journal of Computational Physics*, 227(18):8417–8436, 2008.

[123] B. Leimkuhler and C. Matthews. *Molecular Dynamics.* Springer, 2016.

[124] A. Leroy, D. Violeau, M. Ferrand, and A. Joly. Buoyancy modelling with incompressible SPH for laminar and turbulent flows. *International Journal for Numerical Methods in Fluids*, 78(8):455–474, 2015.

[125] G. Li, J. Gao, P. Wen, Q. Zhao, J. Wang, J. Yan, and A. Yamaji. A review on MPS method developments and applications in nuclear engineering. *Computer Methods in Applied Mechanics and Engineering*, 367:113166, 2020.

[126] S. J. Lind and P. K. Stansby. High-order Eulerian incompressible smoothed particle hydrodynamics with transition to Lagrangian free-surface motion. *Journal of Computational Physics*, 326:290–311, 2016.

[127] S. J. Lind, R. Xu, P. K. Stansby, and B. D. Rogers. Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves. *Journal of Computational Physics*, 231(4):1499–1523, 2012.

[128] S. J. Lind, P. K. Stansby, and B. D. Rogers. Incompressible–compressible flows with a transient discontinuous interface using smoothed particle hydrodynamics (SPH). *Journal of Computational Physics*, 309:129–147, 2016.

[129] G.-R. Liu. *Meshfree Methods: Moving Beyond the Finite Element Method.* CRC press, 2009.

[130] G.-R. Liu and M. B. Liu. *Smoothed Particle Hydrodynamics: a Meshfree Particle Method.* World Scientific, 2003.

[131] M. B. Liu and G.-R. Liu. Restoring particle consistency in smoothed particle hydrodynamics. *Applied Numerical Mathematics*, 56(1):19–36, 2006.

[132] M. B. Liu and G. R. Liu. Smoothed particle hydrodynamics (SPH): an overview and recent developments. *Archives of Computational Methods in Engineering*, 17(1):25–76, 2010.

[133] E. Y. M. Lo and S. Shao. Simulation of near-shore solitary wave mechanics by an incompressible SPH method. *Applied Ocean Research*, 24(5):275–286, 2002.

[134] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, 82:1013–1024, 1977.

[135] F. Macia, M. Antuono, L. M. González, and A. Colagrossi. Theoretical analysis of the no-slip boundary condition enforcement in SPH methods. *Progress of Theoretical Physics*, 125(6):1091–1121, 2011.

[136] F. Macia, L. M. González, J. L. Cercos-Pita, and A. Souto-Iglesias. A boundary integral SPH formulation: consistency and applications to ISPH and WCSPH. *Progress of Theoretical Physics*, 128(3):439–462, 2012.

[137] S. Manenti, S. Sibilla, M. Gallati, G. Agate, and R. Guandalini. SPH simulation of sediment flushing induced by a rapid water flow. *Journal of Hydraulic Engineering*, 138(3):272–284, 2011.

[138] S. Manenti, S. Sibilla, M. Gallati, G. Agate, and R. Guandalini. SPH simulation of sediment flushing induced by a rapid water flow. *Journal of Hydraulic Engineering*, 138(3):272–284, 2012. doi: https://dx.doi.org/10.3390/w11091875.

[139] S. Marrone, M. Antuono, A. Colagrossi, G. Colicchio, D. Le Touzé, and G. Graziani. δ-SPH model for simulating violent impact flows. *Computer Methods in Applied Mechanics and Engineering*, 200(13-16):1526–1542, 2011.

[140] G. Mathew, I. Mezić, and L. Petzold. A multiscale measure for mixing. *Physica D: Nonlinear Phenomena*, 211(1-2):23–46, 2005.

[141] T. Matsunaga, K. Shibata, K. Murotani, and S. Koshizuka. Hybrid grid-particle method for fluid mixing simulation. *Computational Particle Mechanics*, 2(3): 233–246, 2015.

[142] A. Mayrhofer, B. D. Rogers, D. Violeau, and M. Ferrand. Investigation of wall bounded flows using SPH and the unified semi-analytical wall boundary conditions. *Computer Physics Communications*, 184(11):2515–2527, 2013.

[143] R. Miesen and B. J. Boersma. Hydrodynamic stability of a sheared liquid film. *Journal of Fluid Mechanics*, 301:175–202, 1995.

[144] J. Miller. *Modelling Melt Viscosity for Nuclear Waste Glass.* PhD thesis, University of Sheffield, 2014.

[145] A. Mokos, B. D. Rogers, P. K. Stansby, and J. M. Domínguez. Multi-phase SPH modelling of violent hydrodynamics on GPUs. *Computer Physics Communications*, 196:304–316, 2015.

[146] A. Mokos, B. D. Rogers, and P. K. Stansby. A multi-phase particle shifting algorithm for SPH simulations of violent hydrodynamics with a large number of particles. *Journal of Hydraulic Research*, 55(2):143–162, 2017.

[147] D. Molteni and A. Colagrossi. A simple procedure to improve the pressure evaluation in hydrodynamic context using the SPH. *Computer Physics Communications*, 180(6):861–872, 2009.

[148] J. J. Monaghan. Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30(1):543–574, 1992. ISSN 0066-4146.

[149] J. J. Monaghan. Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30(1):543–574, 1992.

[150] J. J. Monaghan. Simulating free surface flows with SPH. *Journal of Computational Physics*, 110(2):399–406, 1994. ISSN 0021-9991.

[151] J. J. Monaghan. Simulating free surface flows with SPH. *Journal of Computational Physics*, 110(2):399–406, 1994.

[152] J. J. Monaghan. Implicit SPH drag and dusty gas dynamics. *Journal of Computational Physics*, 138(2):801–820, 1997.

[153] J. J. Monaghan. SPH without a tensile instability. *Journal of Computational Physics*, 159(2):290–311, 2000.

[154] J. J. Monaghan. Smoothed particle hydrodynamics. *Reports on Progress in Physics*, 68(8):1703–1759, 08 2005. ISSN 0034-4885.

[155] J. J. Monaghan and R. A. Gingold. Shock simulation by the particle method sph. *Journal of Computational Physics*, 52(2):374–389, 1983.

[156] J. J. Monaghan and A. Kocharyan. SPH simulation of multi-phase flow. *Computer Physics Communications*, 87(1-2):225–235, 1995.

[157] J. J. Monaghan and A. Kos. Solitary waves on a Cretan beach. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 125(3):145–155, 1999.

[158] G. Montante, M. Moštěk, M. Jahoda, and F. Magelli. CFD simulations and experimental validation of homogenisation curves and mixing time in stirred Newtonian and pseudoplastic liquids. *Chemical Engineering Science*, 60(8-9): 2427–2437, 2005.

[159] G. W. Morey. Phase equilibrium relationships determining glass compositions. *Industrial & Engineering Chemistry*, 25(7):742–748, 1933. doi: 10.1021/ie50283a009.

[160] J. P. Morris. Simulating surface tension with smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids*, 33(3):333–353, 2000.

[161] J. P. Morris, P. J. Fox, and Y. Zhu. Modeling low Reynolds number incompressible flows using SPH. *Journal of Computational Physics*, 136(1):214–226, 1997.

[162] F. D. Murnaghan. The compressibility of media under extreme pressures. *Proceedings of the National Academy of Sciences*, 30(9):244–247, 1944.

[163] B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: diffuse approximation and diffuse elements. *Computational Mechanics*, 10(5): 307–318, 1992.

[164] R. M. Nestor, M. Basa, M. Lastiwka, and N. J. Quinlan. Extension of the finite volume particle method to viscous flow. *Journal of Computational Physics*, 228 (5):1733–1749, 2009.

[165] Nextflow Software. Nextflow. URL `https://www.nextflowsoftware.com/`. accessed 11-08-2021.

[166] K.-C. Ng, Y.-H. Hwang, T. W. Sheu, and C.-H. Yu. Moving particle level-set (MPLS) method for incompressible multiphase flow computation. *Computer Physics Communications*, 196:317–334, 2015.

[167] K. C. Ng, Y. L. Ng, T. W. H. Sheu, and A. Alexiadis. Assessment of smoothed particle hydrodynamics (SPH) models for predicting wall heat transfer rate at complex boundary. *Engineering Analysis with Boundary Elements*, 111:195–205, 2020.

[168] H. T. Nguyen, T. A. Do, and B. Cosson. Numerical simulation of submerged flow bridge scour under dam-break flow using multi-phase SPH method. *Mathematical Biosciences and Engineering*, 16(5):5395–5418, 2019. doi: https://dx.doi.org/10.3934/mbe.2019269.

[169] T. Norton and D.-W. Sun. Computational fluid dynamics (CFD)–an effective and efficient design and analysis tool for the food industry: a review. *Trends in Food Science & Technology*, 17(11):600–620, 2006.

[170] Nuclear Decomissioning Authority. Uk radioactive waste inventory. URL `https://ukinventory.nda.gov.uk/about-radioactive-waste/what-is-radioactivity/what-are-the-main-waste-categories/`. accessed 24-06-2021.

[171] Nuclear Power. Reynolds number. URL `https://www.nuclear-power.net/nuclear-engineering/fluid-dynamics/reynolds-number/`. accessed 17-07-2019.

[172] Y. Ohtsuki, S. Sakida, Y. Benino, and T. Nanba. Phase separation of borosilicate glass containing phosphorus. *IOP Conference Series: Materials Science and Engineering*, 18(11):112022, 2011.

[173] M. I. Ojovan. On alteration rate renewal stage of nuclear waste glass corrosion. *MRS Advances*, 5(3):111–120, 2020.

[174] M. I. Ojovan and W. E. Lee. Glassy wasteforms for nuclear waste immobilization. *Metallurgical and Materials Transactions A*, 42(4):837–851, 2011.

[175] A. N. Parshikov, S. A. Medin, I. I. Loukashenko, and V. A. Milekhin. Improvements in SPH method by means of interparticle contact algorithm and analysis of perforation tests at moderate projectile velocities. *International Journal of Impact Engineering*, 24(8):779–796, 2000.

[176] E. L. Paul, V. A. Atiemo-Obeng, and S. M. Kresta. *Handbook of industrial mixing: science and practice*. John Wiley & Sons, 2004.

[177] A. Pinarbasi and A. Liakopoulos. The effect of variable viscosity on the interfacial stability of two-layer Poiseuille flow. *Physics of Fluids*, 7(6):1318–1324, 1995.

[178] A. B. Pippard. *Elements of Classical Thermodynamics: for Advanced Students of Physics*. Cambridge University Press, 1964.

[179] R. Pokorny and P. Hrma. Model for the conversion of nuclear waste melter feed to glass. *Journal of Nuclear Materials*, 445(1-3):190–199, 2014.

[180] M. C. Potter and E. Graber. Stability of plane Poiseuille flow with heat transfer. *The Physics of Fluids*, 15(3):387–391, 1972.

[181] D. J. Price, J. Wurster, T. S. Tricco, C. Nixon, S. Toupin, A. Pettitt, C. Chan, D. Mentiplay, G. Laibe, and S. Glover. Phantom: A smoothed particle hydrodynamics and magnetohydrodynamics code for astrophysics. *Publications of the Astronomical Society of Australia*, 35, 2018.

[182] J. H. Pu, Y. Huang, S. Shao, and K. Hussain. Three-Gorges Dam fine sediment pollutant transport: turbulence SPH model simulation of multi-fluid flows. *Journal of Applied Fluid Mechanics*, 9:1–10, 2016.

[183] N. J. Quinlan, M. Basa, and M. Lastiwka. Truncation error in mesh-free particle methods. *International Journal for Numerical Methods in Engineering*, 66(13): 2064–2085, 2006.

[184] P. W. Randles and L. D. Libersky. Smoothed particle hydrodynamics: some recent improvements and applications. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):375–408, 1996.

[185] L. Rayleigh. On the stability, or instability, of certain fluid motions. *Proceedings of the London Mathematical Society*, 1(1):57–72, 1879.

[186] G. Reece, B. D. Rogers, S. Lind, and G. Fourtakas. New instability and mixing simulations using SPH and a novel mixing measure. *Journal of Hydrodynamics*, 32(4):684–698, 2020.

[187] N. Riemer and M. West. Quantifying aerosol mixing state with entropy and diversity measures. *Atmospheric Chemistry and Physics*, 13(22):11423–11439, 2013.

[188] M. Robinson. *Turbulence and Viscous Mixing using Smoothed Particle Hydrodynamics*. PhD thesis, Monash Universtiy, 2009.

[189] M. Robinson and P. W. Cleary. The influence of cam geometry and operating conditions on chaotic mixing of viscous fluids in a twin cam mixer. *AIChE journal*, 57(3):581–598, 2011.

[190] M. Robinson, P. Cleary, and J. Monaghan. Analysis of mixing in a twin cam mixer using smoothed particle hydrodynamics. *AIChE journal*, 54(8):1987–1998, 2008.

[191] B. D. Rogers. Test 3. URL `https://spheric-sph.org/tests/test-3`. accessed 10-07-2019.

[192] R. Rook, M. Yildiz, and S. Dost. Modeling transient heat transfer using SPH and implicit time integration. *Numerical Heat Transfer, Part B: Fundamentals*, 51(1):1–23, 2007.

[193] L. Rosenhead. Introduction-the second coefficient of viscosity: a brief review of fundamentals. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 226(1164):1–6, 1954.

[194] G. Roth and S. Weisenburger. Vitrification of high-level liquid waste: glass chemistry, process chemistry and process technology. *Nuclear Engineering and Design*, 202(2-3):197–207, 2000.

[195] C. F. Rowlatt and S. J. Lind. Bubble collapse near a fluid-fluid interface using the spectral element marker particle method with applications in bioengineering. *International Journal of Multiphase Flow*, 90:118–143, 2017.

[196] F. Sadlo and R. Peikert. Visualizing lagrangian coherent structures and comparison to vector field topology. In *Topology-Based Methods in Visualization II*, pages 15–29. Springer, 2009.

[197] R. Scardovelli and S. Zaleski. Direct numerical simulation of free-surface and interfacial flow. *Annual review of fluid mechanics*, 31(1):567–603, 1999.

[198] P. Schäfer and H. Herwig. Stability of plane Poiseuille flow with temperature dependent viscosity. *International Journal of Heat and Mass Transfer*, 36(9): 2441–2448, 1993.

[199] M. Schaller, P. Gonnet, P. W. Draper, A. B. G. Chalk, R. G. Bower, J. Willis, and L. Hausammann. SWIFT: SPH with inter-dependent fine-grained tasking, 2018.

[200] H. F. Schwaiger. An implicit corrected SPH formulation for thermal diffusion with linear free surface boundary conditions. *International Journal for Numerical Methods in Engineering*, 75(6):647–671, 2008.

[201] S. C. Shadden, F. Lekien, and J. E. Marsden. Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows. *Physica D: Nonlinear Phenomena*, 212(3-4):271–304, 2005.

[202] M. S. Shadloo, A. Zainali, M. Yildiz, and A. Suleman. A robust weakly compressible SPH method and its comparison with an incompressible SPH. *International Journal for Numerical Methods in Engineering*, 89(8):939–956, 2012.

[203] R. Shamsoddini, M. Sefid, and R. Fatehi. Lagrangian simulation and analysis of the micromixing phenomena in a cylindrical paddle mixer using a modified weakly compressible smoothed particle hydrodynamics method. *Asia-Pacific Journal of Chemical Engineering*, 10(1):112–124, 2015.

[204] X. Shan and H. Chen. Lattice Boltzmann model for simulating flows with multiple phases and components. *Physical review E*, 47(3):1815, 1993.

[205] S. Shao. Incompressible smoothed particle hydrodynamics simulation of multifluid flows. *International Journal for Numerical Methods in Fluids*, 69(11): 1715–1735, 2012.

[206] R. Short. Phase separation and crystallisation in UK HLW vitrified products. *Procedia Materials Science*, 7:93–100, 2014.

[207] A. Shrivastava. *1 - Introduction to Plastics Engineering*. Plastics Design Library. William Andrew Publishing, 2018. ISBN 978-0-323-39500-7. doi: https://doi.org/10.1016/B978-0-323-39500-7.00001-0. URL `https://www.sciencedirect.com/science/article/pii/B9780323395007000010`.

[208] Siemens. *STAR-CCM+ 13.02 User Guide*.

[209] M. A. E. Sikarudi and A. H. Nikseresht. Neumann and Robin boundary conditions for heat conduction modeling using smoothed particle hydrodynamics. *Computer Physics Communications*, 198:1–11, 2016.

[210] A. Skillen, S. Lind, P. K. Stansby, and B. D. Rogers. Incompressible smoothed particle hydrodynamics (SPH) with reduced temporal noise and generalised Fickian smoothing applied to body–water slam and efficient wave–body interaction. *Computer Methods in Applied Mechanics and Engineering*, 265:163–173, 2013.

[211] A. Smolianski. *Numerical Modeling of Two-Fluid Interfacial Flows*. University of Jyväskylä, 2001.

[212] V. Springel. The cosmological simulation code GADGET-2. *Monthly Notices of the Royal Astronomical Society*, 364(4):1105–1134, 2005.

[213] P. N. Sun, A. Colagrossi, S. Marrone, and A. M. Zhang. Detection of Lagrangian coherent structures in the SPH framework. *Computer Methods in Applied Mechanics and Engineering*, 305:849–868, 2016.

[214] X. Sun, M. Sakai, K. Shibata, Y. Tochigi, and H. Fujiwara. Numerical modeling on the discharged fluid flow from a glass melter by a Lagrangian approach. *Nuclear Engineering and Design*, 248:14–21, 2012.

[215] S. K. Sundaram, P. P. Woskov, and W. E. Daniel. Millimeter wave diagnostics of materials and melts. In *Conference Digest of the Joint 29th International Conference on Infrared and Millimeter Waves and 12th International Conference on Terahertz Electronics*, pages 799–800. IEEE, 2004.

[216] P. D. Swanson and J. M. Ottino. A comparative computational and experimental study of chaotic mixing of viscous fluids. *Journal of Fluid Mechanics*, 213:227–249, 1990.

[217] M. R. Swift, W. R. Osborn, and J. M. Yeomans. Lattice Boltzmann simulation of non-ideal fluids. *Physical Review Letters*, 75(5):830, 1995.

[218] K. Szewc, J. Pozorski, and A. Taniere. Modeling of natural convection with smoothed particle hydrodynamics: non-Boussinesq formulation. *International Journal of Heat and Mass Transfer*, 54(23-24):4807–4816, 2011.

[219] K. Szewc, J. Pozorski, and J.-P. Minier. Analysis of the incompressibility constraint in the smoothed particle hydrodynamics method. *International Journal for Numerical Methods in Engineering*, 92(4):343–369, 2012.

[220] K. Szewc, A. Taniere, J. Pozorski, and J.-P. Minier. A study on application of smoothed particle hydrodynamics to multi-phase flows. *International Journal of Nonlinear Sciences and Numerical Simulation*, 13(6):383–395, 2012.

[221] K. Szewc, J. Pozorski, and J.-P. Minier. Spurious interface fragmentation in multiphase SPH. *International Journal for Numerical Methods in Engineering*, 103(9):625–649, 2015.

[222] H. Takeda, S. M. Miyama, and M. Sekiya. Numerical simulation of viscous flow by smoothed particle hydrodynamics. *Progress of Theoretical Physics*, 92(5): 939–960, 1994.

[223] H. Tang and L. C. Wrobel. Modelling the interfacial flow of two immiscible liquids in mixing processes. *International Journal of Engineering Science*, 43 (15-16):1234–1256, 2005.

[224] H. Tang, L. C. Wrobel, and Z. Fan. Tracking of immiscible interfaces in multiple-material mixing processes. *Computational Materials Science*, 29(1):103–118, 2004.

[225] A. Tartakovsky and P. Meakin. Modeling of surface tension and contact angles with smoothed particle hydrodynamics. *Physical Review E*, 72(2):026301, 2005.

[226] A. M. Tartakovsky and A. Panchenko. Pairwise force smoothed particle hydrodynamics model for multiphase flow: surface tension and contact line dynamics. *Journal of Computational Physics*, 305:1119–1146, 2016.

[227] A. M. Tartakovsky, K. F. Ferris, and P. Meakin. Lagrangian particle model for multiphase flows. *Computer Physics Communications*, 180(10):1874–1881, 2009.

[228] J.-L. Thiffeault. Using multiscale norms to quantify mixing and transport. *Nonlinearity*, 25(2):R1, 2012.

[229] M. Tomozawa, V. McGahay, and J. M. Hyde. Phase separation of glasses. *Journal of Non-Crystalline Solids*, 123(1):197–207, 1990. ISSN 0022-3093. doi: 10.1016/0022-3093(90)90785-K. XVth International Congress on Glass.

[230] B. Tóth and K. Szabó. Flow structure detection with smoothed particle hydrodynamics. In D. Violeau, A. Hérault, and A. Joly, editors, *Proceedings of the 9th International SPHERIC Workshop.*

[231] I. Tovena, T. Advocat, D. Ghaleb, E. Vernaz, and F. Larche. Thermodynamic and structural models compared with the initial dissolution rates of "SON" glass samples. *MRS Online Proceedings Library Archive*, 333, 1993.

[232] C. Ulrich, M. Leonardi, and T. Rung. Multi-physics SPH simulation of complex marine-engineering hydrodynamic problems. *Ocean Engineering*, 64:109–121, 2013. doi: https://dx.doi.org/10.1016/j.oceaneng.2013.02.007.

[233] D. R. Unger, F. J. Muzzio, and R. S. Brodkey. Experimental and numerical characterization of viscous flow and mixing in an impinging jet contactor. *The Canadian Journal of Chemical Engineering*, 76(3):546–555, 1998.

[234] R. Vacondio, C. Altomare, M. De Leffe, X. Hu, D. Le Touzé, S. Lind, J.-C. Marongiu, S. Marrone, B. D. Rogers, and A. Souto-Iglesias. Grand challenges for smoothed particle hydrodynamics numerical schemes. *Computational Particle Mechanics*, 8(3):575–588, 2021.

[235] P. Valluri, L. Ó. Náraigh, H. Ding, and P. D. M. Spelt. Linear and nonlinear spatio-temporal instability in laminar two-layer flows. *Journal of Fluid Mechanics*, 656:458–480, 2010.

[236] S. Venkateswaran, J. W. Lindau, R. F. Kunz, and C. L. Merkle. Computation of multiphase mixture flows with compressibility effects. *Journal of Computational Physics*, 180(1):54–77, 2002.

[237] L. Verlet. Computer "experiments" on classical fluids. I. thermodynamical properties of Lennard-Jones molecules. *Physical Review*, 159(1):98, 1967.

[238] H. K. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics: the Finite Volume Method*. Pearson Education, 2007.

[239] J. D. Vienna. Nuclear waste vitrification in the United States: recent developments and future options. *International Journal of Applied Glass Science*, 1(3): 309–321, 2010.

[240] D. Violeau. *Fluid Mechanics and the SPH Method: Theory and Applications*. Oxford University Press, 2012.

[241] R. Viskanta. Review of three-dimensional mathematical modeling of glass melting. *Journal of Non-Crystalline Solids*, 177:347–362, 1994.

[242] H. Wakabayashi, H. Yamanaka, R. Terai, and S. Fukumoto. Effect of MoO_3 addition on phase separation in borosilicate glasses. *Yogyo Kyokai-Shi*, 95(5): 486–493, 1987.

[243] D. P. Wall and S. K. Wilson. The linear stability of channel flow of fluid with temperature-dependent viscosity. In *Progress in Industrial Mathematics at ECMI 96*, pages 416–423. Springer, 1997.

[244] D. C. Wan, B. S. V. Patnaik, and G. W. Wei. A new benchmark quality solution for the buoyancy-driven cavity by discrete singular convolution. *Numerical Heat Transfer: Part B: Fundamentals*, 40(3):199–228, 2001.

[245] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4 (1):389–396, 1995.

[246] L. Wieth, K. Kelemen, S. Braun, R. Koch, H.-J. Bauer, and H. P. Schuchmann. Smoothed particle hydrodynamics (SPH) simulation of a high-pressure homogenization process. *Microfluidics and Nanofluidics*, 20(2):42, 2016.

[247] C. Witz, D. Treffer, T. Hardiman, and J. Khinast. Local gas holdup simulation and validation of industrial-scale aerated bioreactors. *Chemical Engineering Science*, 152:636–648, 2016.

[248] Wolfram Research. Rayleigh number. URL `http://scienceworld.wolfram.com/physics/RayleighNumber.html`. accessed 17-07-2019.

[249] World Nuclear Association. Nuclear power in the United Kingdom. URL `http://www.world-nuclear.org/information-library/country-profiles/countries-t-z/united-kingdom.aspx`. accessed 17-06-2018.

[250] World Nuclear News. Reprocessing ceases at UK's Thorp plant. URL `https://www.world-nuclear-news.org/Articles/Reprocessing-ceases-at-UKs-Thorp-plant`. accessed 24-09-2021.

[251] A. Xenakis, S. Lind, P. K. Stansby, and B. Rogers. An incompressible SPH scheme with improved pressure predictions for free-surface generalized Newtonian flows. *Journal of Non-Newtonian Fluid Mechanics*, 218, 2015. doi: 10.1016/j.jnnfm.2015.01.006.

[252] A. M. Xenakis, S. J. Lind, P. K. Stansby, and B. D. Rogers. Landslides and tsunamis predicted by incompressible smoothed particle hydrodynamics (SPH) with application to the 1958 Lituya Bay event and idealized experiment. *Proceedings of the Royal Society A*, 473(2199):20160674, 2017.

[253] R. Xu, P. Stansby, and D. Laurence. Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach. *Journal of computational Physics*, 228(18):6703–6725, 2009.

[254] G. Q. Yang, B. Du, and L. S. Fan. Bubble formation and dynamics in gas–liquid–solid fluidization – a review. *Chemical Engineering Science*, 62(1-2):2–27, 2007.

[255] X. Yang and S.-C. Kong. Numerical study of natural convection in a horizontal concentric annulus using smoothed particle hydrodynamics. *Engineering Analysis with Boundary Elements*, 102:11–20, 2019.

[256] T. Ye and Y. Li. A comparative review of smoothed particle hydrodynamics, dissipative particle dynamics and smoothed dissipative particle dynamics. *International Journal of Computational Methods*, 15(08):1850083, 2018.

[257] S. G. Yiantsios and B. G. Higgins. Linear stability of plane Poiseuille flow of two superposed fluids. *The Physics of Fluids*, 31(11):3225–3238, 1988.

[258] C.-S. Yih. Instability due to viscosity stratification. *Journal of Fluid Mechanics*, 27(2):337–352, 1967.

[259] J. Zhu. Summary of dimensionless numbers of fluid mechanics and heat transfer. URL `http://jingweizhu.weebly.com/course-note.html`. accessed 17-07-2019.

[260] E. H. Zubeldia, G. Fourtakas, B. D. Rogers, and M. M. Farias. Multi-phase SPH model for simulation of erosion and scouring by means of the shields and Drucker–Prager criteria. *Advances in Water Resources*, 117:98–114, 2018.

Appendix A

Example xml Case Files

A.1 Two-Phase Poiseuille Flow with Constant Viscosities

```
<case>
  <casedef>
    <constantsdef>
      <gravity x="0.1" y="0" z="0" comment="Gravitational␣acceleration" units_comment
    ="m/s^2" />
      <rhop0 value="1000" comment="Reference␣density␣of␣the␣fluid" units_comment="kg/
    m^3" />
      <hswl value="0" auto="true" comment="Maximum␣still␣water␣level␣to␣calculate␣
    speedofsound␣using␣coefsound" units_comment="metres␣(m)" />
      <gamma value="7" comment="Polytropic␣constant␣for␣water␣used␣in␣the␣state␣
    equation" />
      <speedsystem value="0" auto="true" comment="Maximum␣system␣speed␣(by␣default␣
    the␣dam-break␣propagation␣is␣used)" />
      <coefsound value="1" comment="Coefficient␣to␣multiply␣speedsystem" />
      <speedsound value="10" auto="false" comment="Speed␣of␣sound␣to␣use␣in␣the␣
    simulation␣(by␣default␣speedofsound=coefsound*speedsystem)" />
      <coefh value="1.0" comment="Coefficient␣to␣calculate␣the␣smoothing␣length␣(h=
    coefh*sqrt(3*dp^2)␣in␣3D)" />
      <cflnumber value="0.1" comment="Coefficient␣to␣multiply␣dt" />
    </constantsdef>
    <mkconfig boundcount="240" fluidcount="9"/>
    <geometry>
      <definition dp="0.01">
        <pointmin x="-0.1" y="0" z="-0.6" />
        <pointmax x="2.1" y="0" z="0.6" />
      </definition>
      <commands>
    <mainlist>
      <setshapemode>dp | bound</setshapemode>
      <setdrawmode mode=”full” />
      <setmkfluid mk=”0” /> <!——Phase0——>
      <drawbox>
        <boxfill>solid</boxfill>
        <point x=”0” y=”-2” z=”-0.5” />
        <size x=”2” y=”2” z=”0.5” />
      </drawbox>
      <setmkfluid mk=”1” /> <!——Phase1——>
      <drawbox>
        <boxfill>solid</boxfill>
        <point x=”0” y=”-2” z=”0.0” />
        <size x=”2” y=”2” z=”0.5” />
      </drawbox>

      <setmkbound mk=”1” />
      <drawbox>
        <boxfill>solid</boxfill>
        <point x=”0” y=”-2” z=”-0.54” />
        <size x=”2” y=”2” z=”0.04” />
      </drawbox>
      <drawbox>
        <boxfill>solid</boxfill>
        <point x=”0” y=”-2” z=”0.5” />
        <size x=”2” y=”2” z=”0.04” />
      </drawbox>
      <shapeout file=”Box” />
    </mainlist>
  </commands>
 </geometry>
</casedef>
<execution>
 <special>
  <vphases>
    <phase phasetype=”0” mkfluid=”0”>
      <rhop value=”1000” comment=”Density”/>
      <visco value=”0.1” comment=”Kinematic␣viscosity(m2/s)” />
      <op1 value=”0” comment=”constant␣␣viscosity”/>
    </phase>
    <phase phasetype=”0” mkfluid=”1”>
      <rhop value=”1000” comment=”Density” />
      <visco value=”0.05” comment=”Kinematic␣viscosity” />
      <op1 value=”0” comment=”constant␣␣viscosity”/>
    </phase>
    <phase phasetype=”1” mkbound=”1”>
      <rhop value=”1000” comment=”Density” />
      <visco value=”0.1” comment=”Kinematic␣viscosity” />
      <op1 value=”0” comment=”constant␣␣viscosity”/>
    </phase>
  </vphases>
</special>
<parameters>
  <parameter key=”SavePosDouble” value=”0” comment=”Saves␣particle␣position␣using
␣double␣precision␣(default=0)” />
  <parameter key=”Boundary” value=”1” comment=”Boundary␣method␣1:DBC,␣2:mDBC” />

  <parameter key=”StepAlgorithm” value=”2” comment=”Step␣Algorithm␣1:Verlet,␣2
:Symplectic␣(default=1)” />
  <parameter key=”VerletSteps” value=”40” comment=”Verlet␣only:␣Number␣of␣steps␣
to␣apply␣Euler␣timestepping␣(default=40)” />
  <parameter key=”Kernel” value=”2” comment=”Interaction␣Kernel␣1:Cubic␣Spline,␣2
:Wendland␣(default=2)” />
  %Choice of reology treatment, velocity gradient calculation and viscosity
treatment
  <parameter key=”RheologyTreatment” value=”1” comment=”Reology␣formulation␣1
:Single−phase␣classic,␣2:␣Single␣and␣multi−phase” />
  <parameter key=”VelocityGradientType” value=”1” comment=”Velocity␣gradient␣
formulation␣1:FDA,␣2:SPH” />
  <parameter key=”ViscoTreatment” value=”2” comment=”Viscosity␣formulation␣1
:Artificial,␣2:Laminar+SPS,␣3:Constitutive␣␣eq.” />
  <parameter key=”ViscPhases” value=”2”/>
  %Wall boundary viscosity or/and artificial viscosity if ViscoTreatment is 1
:Artificial
  <parameter key=”Visco” value=”0.1” comment=”Viscosity␣value” /> % Note alpha
can depend on the resolution when using artificial viscosity
  <parameter key=”ViscoBoundFactor” value=”1” comment=”Multiply␣viscosity␣value␣
with␣boundary␣(default=1)” />
  <parameter key=”DensityDT” value=”3” comment=”Density␣Diffusion␣Term␣0:None,␣1
:Molteni,␣2:Fourtakas,␣3:Fourtakas(full)␣(default=0)” />
  <parameter key=”DensityDTvalue” value=”0.1” comment=”DDT␣value␣(default=0.1)” /
>
  <parameter key=”Shifting” value=”3” comment=”Shifting␣mode␣0:None,␣1:Ignore␣
bound,␣2:Ignore␣fixed,␣3:Full␣(default=0)” />
  <parameter key=”ShiftCoef” value=”−10” comment=”Coefficient␣for␣shifting␣
computation␣(default=−2)” />
  <parameter key=”ShiftTFS” value=”1.5” comment=”Threshold␣to␣detect␣free␣surface
.␣Typically␣1.5␣for␣2D␣and␣2.75␣for␣3D␣(default=0)” />
  <parameter key=”RigidAlgorithm” value=”1” comment=”Rigid␣Algorithm␣0:collision−
free,␣1:SPH,␣2:DEM,␣3:Chrono␣(default=1)” />
  <parameter key=”FtPause” value=”0.0” comment=”Time␣to␣freeze␣the␣floatings␣at␣
simulation␣start␣(warmup)␣(default=0)” units_comment=”seconds” />
  <parameter key=”CoefDtMin” value=”0.05” comment=”Coefficient␣to␣calculate␣
minimum␣time␣step␣dtmin=coefdtmin*h/speedsound␣(default=0.05)” />
  <parameter key=”RelaxationDt” value=”1.0” comment=”Relaxation␣parameter␣for␣the
␣viscous␣time␣step␣restricition(default=0.2)” />
  <parameter key=”DtIni” value=”0” comment=”Initial␣time␣step.␣Use␣0␣to␣defult␣
use␣(default=h/speedsound)” units_comment=”seconds” />
```

```
      <parameter key=”DtMin” value=”0” comment=”Minimum␣time␣step.␣Use␣0␣to␣defult␣
    use␣(default=coefdtmin*h/speedsound)” units_comment=”seconds” />
      <parameter key=”DtFixed” value=”0” comment=”Fixed␣Dt␣value.␣Use␣0␣to␣disable␣(
    default=disabled)” units_comment=”seconds” />
      <parameter key=”DtFixedFile” value=”NONE” comment=”Dt␣values␣are␣loaded␣from␣
    file.␣Use␣NONE␣to␣disable␣(default=disabled)” units_comment=”milliseconds␣(ms)” /
    >
      <parameter key=”DtAllParticles” value=”0” comment=”Velocity␣of␣particles␣used␣
    to␣calculate␣DT.␣1:All,␣0:Only␣fluid/floating␣(default=0)” />
      <parameter key=”TimeMax” value=”20.0” comment=”Time␣of␣simulation”
    units_comment=”seconds” />
      <parameter key=”TimeOut” value=”0.05” comment=”Time␣out␣data” units_comment=”
    seconds” />
      <parameter key=”PartsOutMax” value=”1” comment=”%/100␣of␣fluid␣particles␣
    allowed␣to␣be␣excluded␣from␣domain␣(default=1)” units_comment=”decimal” />
      <parameter key=”RhopOutMin” value=”500” comment=”Minimum␣rhop␣valid␣(default
    =700)” units_comment=”kg/m^3” />
      <parameter key=”RhopOutMax” value=”3000” comment=”Maximum␣rhop␣valid␣(default
    =1300)” units_comment=”kg/m^3” />
      <parameter key=”XPeriodicIncZ” value=”0” comment=”Increase␣of␣Z␣with␣periodic␣
    BC” />
    </parameters>
  </execution>
</case>
```

A.2 Two-Phase Vitrification Melter with Temperature-Dependent Viscosities

```
<case>
  <casedef>
    <constantsdef>
      <gravity x=”0.0” y=”0” z=”−9.81” comment=”Gravitational␣acceleration”
    units_comment=”m/s^2” />
      <rhop0 value=”30” comment=”Reference␣density␣of␣the␣fluid” units_comment=”kg/m
    ^3” />
      <hswl value=”0” auto=”true” comment=”Maximum␣still␣water␣level␣to␣calculate␣
    speedofsound␣using␣coefsound” units_comment=”metres␣(m)” />
      <gamma value=”7” comment=”Polytropic␣constant␣for␣water␣used␣in␣the␣state␣
    equation” />
      <speedsystem value=”0” auto=”true” comment=”Maximum␣system␣speed␣(by␣default␣
    the␣dam−break␣propagation␣is␣used)” />
      <coefsound value=”12” comment=”Coefficient␣to␣multiply␣speedsystem” />
      <speedsound value=”10” auto=”true” comment=”Speed␣of␣sound␣to␣use␣in␣the␣
    simulation␣(by␣default␣speedofsound=coefsound*speedsystem)” />
      <hdp value=”2.0” comment=”Coefficient␣to␣calculate␣the␣smoothing␣length␣(hdp=h/
    dp)” />
  <cflnumber value="0.1" comment="Coefficient␣to␣multiply␣dt" />
</constantsdef>
<mkconfig boundcount="240" fluidcount="9"/>
<geometry>
  <definition dp="0.01">
    <pointmin x="-0.355" y="-0.355" z="-0.105" />
    <pointmax x="0.35" y="0.35" z="1.0" />
  </definition>
  <commands>
    <mainlist>
      <!-- real boundary interface at a distance of dp/2 (actual geometry) -->
      <setshapemode>actual | bound</setshapemode>

      <setmkbound mk="0"/>
      <setdrawmode mode="face"/>
      <setfrdrawmode auto="true"/>
      <setnormalinvert invert="true"/>
      <drawcylinder radius="0.252" mask="0">
        <point x="0.0" y="0.0" z="0.0"/>
        <point x="0.0" y="0.0" z="0.504"/>
      </drawcylinder>

      <matrixreset />
      <!-- Save geometry file and remove particles -->
      <shapeout file="hdp" />
      <setmkvoid />
      <redraw />

      <!-- particle generation -->
      <setmkbound mk="1" /> <!--curved bound-->
      <setdrawmode mode="face"/>
      <drawcylinder radius="0.257" mask="1␣|␣2">
        <point x="0.0" y="0.0" z="0.005"/>
        <point x="0.0" y="0.0" z="0.499"/>
        <layers vdp="0,1,2,3,4"/>
      </drawcylinder>
      <setdrawmode mode="full"/>
      <setmkbound mk="2"/>  <!--bottom-->
      <drawcylinder radius="0.307" mask="0">
        <point x="0.0" y="0.0" z="-0.055"/>
        <point x="0.0" y="0.0" z="-0.005"/>
      </drawcylinder>
      <setmkbound mk="3"/>  <!--top-->
      <drawcylinder radius="0.307" mask="0">
        <point x="0.0" y="0.0" z="0.509"/>
        <point x="0.0" y="0.0" z="0.559"/>
      </drawcylinder>

      <setdrawmode mode="full"/>
```

```
          <setmkfluid mk="0" /> <!--bottom fluid-->
          <drawcylinder radius="0.247" mask="0">
          <point x="0.0" y="0.0" z="0.005"/>
          <point x="0.0" y="0.0" z="0.252"/>
          </drawcylinder>
          <setmkfluid mk="1" /> <!--top fluid-->
          <drawcylinder radius="0.247" mask="0">
          <point x="0.0" y="0.0" z="0.252"/>
          <point x="0.0" y="0.0" z="0.499"/>
          </drawcylinder>
          <setfrdrawmode auto="false"/>

          <newvar tempH="1050+273" tempC="800+273" /> <!--define temperatures-->
        </mainlist>
      </commands>
    </geometry>
    <normals>
      <distanceh value="2.0" comment="Smooth␣factor,␣it␣uses␣H*distanceh␣(default=2)"
     />
      <geometryfile file="[CaseName]_hdp_Actual.vtk" comment="Name␣of␣file␣with␣
    boundary␣geometry" />
      <svshapes value="1" comment="(default=0)"/>
    </normals>
  </casedef>
  <execution>
    <special>
      <vphases> %Defines phase parameters
        <phase phasetype="0" mkfluid="0"> <!--fluid0 bottom glass-->
          <rhop value="30" comment="Density␣of␣the␣phase"/>
          <visco value="0.00067" comment="Kinematic␣viscosity␣(or␣consistency␣index)␣
    value␣for␣phase␣(m2/s)"/>
          <temp0 value="#tempC" comment="initial␣temperature"/>
          <cp value="900" comment="specific␣heat"/>
          <k value="1.14" comment="thermal␣conductivity"/>
          <Tref value="#tempH" comment="reference␣temperature"/>
          <beta value="0.0000033" comment="coefficient␣of␣thermal␣expansion"/>
          <op1 value="2" comment="viscosity␣option␣1:␣0-constant;␣1-Wall\&Wilson␣
    equations;␣2-melt␣viscosity␣algorithm"/>
          <op2 value="3"/>     <!-- Jenckel3 -->
          <cst1 value="15.83"/>
          <cst2 value="-3443"/>
          <cst3 value="-174200"/>
          <tempop value="1" comment="temperature␣option:␣0-constant;␣1-temperature␣
    governing␣equation;␣2-temperature␣field"/>
        </phase>
        <phase phasetype="0" mkfluid="1"> <!--fluid1 top inc. waste-->
          <rhop value="30" comment="Density␣of␣the␣phase"/>
          <visco value="0.00067" comment="Kinematic␣viscosity␣(or␣consistency␣index)␣
    value␣for␣phase␣(m2/s)"/>
```

```
      <temp0 value="#tempC" comment="initial␣temperature"/>
      <cp value="900" comment="specific␣heat"/>
      <k value="1.14" comment="thermal␣conductivity"/>
      <Tref value="#tempH" comment="reference␣temperature"/>
      <beta value="0.0000033" comment="coefficient␣of␣thermal␣expansion"/>
      <op1 value="2" comment="viscosity␣option␣1:␣0−constant;␣1−Wall\&Wilson␣
equations;␣2−melt␣viscosity␣algorithm"/>
      <op2 value="3"/>    <!−− Jenckel3 −−>
      <cst1 value="13.03"/>
      <cst2 value="−5715"/>
      <cst3 value="−536800"/>
      <tempop value="1" comment="temperature␣option:␣0−constant;␣1−temperature␣
governing␣equation;␣2−temperature␣field"/>
    </phase>
    <phase phasetype="1" mkbound="0"> <!−−actual boundary surface for mDBC−−>
      <rhop value="30" comment="Density␣of␣the␣phase"/>
      <visco value="0.00067" comment="Kinematic␣viscosity␣(or␣consistency␣index)␣
value␣for␣phase␣(m2/s)"/>
      <temp0 value="#tempH" comment="initial␣temperature"/>
      <cp value="900" comment="specific␣heat"/>
      <k value="1.14" comment="thermal␣conductivity"/>
      <op1 value="0" comment="viscosity␣option␣1:␣0−constant;␣1−Wall\&Wilson␣
equations;␣2−melt␣viscosity␣algorithm"/>
      <tempop value="0" comment="temperature␣option:␣0−constant;␣1−temperature␣
governing␣equation;␣2−temperature␣field;␣3−adiabatic(mDBC)"/>
    </phase>
    <phase phasetype="1" mkbound="1"> <!−−curved boundary−−>
      <rhop value="30" comment="Density␣of␣the␣phase"/>
      <visco value="0.00067" comment="Kinematic␣viscosity␣(or␣consistency␣index)␣
value␣for␣phase␣(m2/s)"/>
      <temp0 value="#tempH" comment="initial␣temperature"/>
      <cp value="900" comment="specific␣heat"/>
      <k value="1.14" comment="thermal␣conductivity"/>
      <op1 value="0" comment="viscosity␣option␣1:␣0−constant;␣1−Wall\&Wilson␣
equations;␣2−melt␣viscosity␣algorithm"/>
      <tempop value="0" comment="temperature␣option:␣0−constant;␣1−temperature␣
governing␣equation;␣2−temperature␣field;␣3−adiabatic(mDBC)"/>
    </phase>
    <phase phasetype="1" mkbound="2"> <!−−bottom−−>
      <rhop value="30" comment="Density␣of␣the␣phase"/>
      <visco value="0.00067" comment="Kinematic␣viscosity␣(or␣consistency␣index)␣
value␣for␣phase␣(m2/s)"/>
      <temp0 value="#tempC" comment="initial␣temperature"/>
      <cp value="900" comment="specific␣heat"/>
      <k value="1.14" comment="thermal␣conductivity"/>
      <op1 value="0" comment="viscosity␣option␣1:␣0−constant;␣1−Wall\&Wilson␣
equations;␣2−melt␣viscosity␣algorithm"/>
      <tempop value="3" comment="temperature␣option:␣0−constant;␣1−temperature␣
governing␣equation;␣2−temperature␣field;␣3−adiabatic(mDBC)"/>
```

```
    </phase>
    <phase phasetype=”1” mkbound=”3”> <!−−top−−>
      <rhop value=”30” comment=”Density␣of␣the␣phase”/>
      <visco value=”0.00067” comment=”Kinematic␣viscosity␣(or␣consistency␣index)␣
value␣for␣phase␣(m2/s)”/>
      <temp0 value=”#tempC” comment=”initial␣temperature”/>
      <cp value=”900” comment=”specific␣heat”/>
      <k value=”1.14” comment=”thermal␣conductivity”/>
      <op1 value=”0” comment=”viscosity␣option␣1:␣0−constant;␣1−Wall\&Wilson␣
equations;␣2−melt␣viscosity␣algorithm”/>
      <tempop value=”3” comment=”temperature␣option:␣0−constant;␣1−temperature␣
governing␣equation;␣2−temperature␣field;␣3−adiabatic(mDBC)”/>
    </phase>
  </vphases>
</special>
<parameters>
  <parameter key=”SavePosDouble” value=”0” comment=”Saves␣particle␣position␣using
␣double␣precision␣(default=0)” />
  <parameter key=”Boundary” value=”2” comment=”Boundary␣method␣1:DBC,␣2:mDBC” />

  <parameter key=”StepAlgorithm” value=”2” comment=”Step␣Algorithm␣1:Verlet,␣2
:Symplectic␣(default=1)” />
  <parameter key=”VerletSteps” value=”40” comment=”Verlet␣only:␣Number␣of␣steps␣
to␣apply␣Euler␣timestepping␣(default=40)” />
  <parameter key=”Kernel” value=”2” comment=”Interaction␣Kernel␣1:Cubic␣Spline,␣2
:Wendland␣(default=2)” />
  %Choice of reology treatment, velocity gradient calculation and viscosity
treatment
  <parameter key=”RheologyTreatment” value=”1” comment=”Reology␣formulation␣1
:Single−phase␣classic,␣2:␣Single␣and␣multi−phase” />
  <parameter key=”VelocityGradientType” value=”1” comment=”Velocity␣gradient␣
formulation␣1:FDA,␣2:SPH” />
  <parameter key=”ViscoTreatment” value=”2” comment=”Viscosity␣formulation␣1
:Artificial,␣2:Laminar+SPS,␣3:Constitutive␣␣eq.” />
  %ViscPhases options
  <parameter key=”ViscPhases” value=”2”/>
  %Wall boundary viscosity or/and artificial viscosity if ViscoTreatment is 1
:Artificial
  <parameter key=”Visco” value=”0.00067” comment=”Viscosity␣value” /> % Note
alpha can depend on the resolution when using artificial viscosity
  <parameter key=”ViscoBoundFactor” value=”1” comment=”Multiply␣viscosity␣value␣
with␣boundary␣(default=1)” />
  <parameter key=”DensityDT” value=”3” comment=”Density␣Diffusion␣Term␣0:None,␣1
:Molteni,␣2:Fourtakas,␣3:Fourtakas(full)␣(default=0)” />
  <parameter key=”DensityDTvalue” value=”1.0” comment=”DDT␣value␣(default=0.1)” /
>
  <parameter key=”Shifting” value=”3” comment=”Shifting␣mode␣0:None,␣1:Ignore␣
bound,␣2:Ignore␣fixed,␣3:Full␣(default=0)” />
```

```
      <parameter key="ShiftCoef" value="-2" comment="Coefficient␣for␣shifting␣
    computation␣(default=-2)" />
      <parameter key="ShiftTFS" value="1.5" comment="Threshold␣to␣detect␣free␣surface
    .␣Typically␣1.5␣for␣2D␣and␣2.75␣for␣3D␣(default=0)" />
      <parameter key="RigidAlgorithm" value="1" comment="Rigid␣Algorithm␣0:collision-
    free,␣1:SPH,␣2:DEM,␣3:Chrono␣(default=1)" />
      <parameter key="FtPause" value="0.0" comment="Time␣to␣freeze␣the␣floatings␣at␣
    simulation␣start␣(warmup)␣(default=0)" units_comment="seconds" />
      <parameter key="CoefDtMin" value="0.05" comment="Coefficient␣to␣calculate␣
    minimum␣time␣step␣dtmin=coefdtmin*h/speedsound␣(default=0.05)" />
      <parameter key="RelaxationDt" value="1.0" comment="Relaxation␣parameter␣for␣the
    ␣viscous␣time␣step␣restricition(default=0.2)" />
      <parameter key="DtIni" value="0.0000015" comment="Initial␣time␣step.␣Use␣0␣to␣
    defult␣use␣(default=h/speedsound)" units_comment="seconds" />
      <parameter key="DtMin" value="0.00000015" comment="Minimum␣time␣step.␣Use␣0␣to␣
    defult␣use␣(default=coefdtmin*h/speedsound)" units_comment="seconds" />
      <parameter key="DtFixed" value="0" comment="Fixed␣Dt␣value.␣Use␣0␣to␣disable␣(
    default=disabled)" units_comment="seconds" />
      <parameter key="DtFixedFile" value="NONE" comment="Dt␣values␣are␣loaded␣from␣
    file.␣Use␣NONE␣to␣disable␣(default=disabled)" units_comment="milliseconds␣(ms)" /
    >
      <parameter key="DtAllParticles" value="0" comment="Velocity␣of␣particles␣used␣
    to␣calculate␣DT.␣1:All,␣0:Only␣fluid/floating␣(default=0)" />
      <parameter key="TimeMax" value="300" comment="Time␣of␣simulation" units_comment
    ="seconds" />
      <parameter key="TimeOut" value="0.25" comment="Time␣out␣data" units_comment="
    seconds" />
      <parameter key="PartsOutMax" value="1" comment="%/100␣of␣fluid␣particles␣
    allowed␣to␣be␣excluded␣from␣domain␣(default=1)" units_comment="decimal" />
      <parameter key="RhopOutMin" value="10" comment="Minimum␣rhop␣valid␣(default
    =700)" units_comment="kg/m^3" />
      <parameter key="RhopOutMax" value="50" comment="Maximum␣rhop␣valid␣(default
    =1300)" units_comment="kg/m^3" />
    </parameters>
  </execution>
</case>
```