The Feasibility of Modelling Liquid Phase Diffusion Growth of $\text{Si}_x\text{Ge}_{1-x}$ with Smoothed Particle Hydrodynamics

by

Rusty Allen Rook
B.Eng., University of Victoria, 2003

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Mechanical Engineering.

© RUSTY ALLEN ROOK, 2006
University of Victoria

All rights reserved. This thesis may not be reproduced in whole or in part, by photocopy or other means, without permission of the author.
Supervisory Committee

Dr. Sadik Dost
Supervisor (Dept. of Mech. Eng.)

Dr. Henning Struchtrup
Departmental Member (Dept. of Mech. Eng.)

Dr. Frank van Veggel
Outside Member (Dept. of Chemistry)

Dr. Boualem Khoudier
External Examiner (Dept. of Mathematics & Statistics)
Abstract

Mathematical and numerical modelling of silicon germanium (Si$_x$Ge$_{1-x}$) crystal growth by liquid phase diffusion (LPD) was carried out in the Crystal Growth Laboratory at the University of Victoria. The crystal growth process was modelled using a relatively new, meshless, Lagrangian technique called smoothed particle hydrodynamics (SPH). The purpose of this modelling was to determine the suitability of the SPH method for simulating general crystal growth processes. Crystal growth simulations in the literature use a mesh-dependent, Eulerian, finite volume method, which requires complicated treatment of mesh adaptation at the crystal solidification surface. In contrast, the crystal solidification process can be modelled with relative ease using SPH, as this approach is automatically adaptable, and does not require a mesh.

Several benchmark problems were also solved using SPH in order to investigate the problematic aspects of SPH simulation. It was found that for simulations involving flows with low Reynolds numbers, SPH performed well even with relatively low particle number densities when compared with analytical and finite difference solutions. A simulation time study was also
conducted for the two-dimensional driven cavity problem, in order to determine the simulation time as a function of particle resolution. This result showed that simulation times in the SPH method increase almost exponentially as the number of SPH particles is increased. The drastic increase in simulation times is due to the particle averaging procedure involving particle \textquotedblleft neighbor\textquotedblright
determination, in addition to the necessary reduction in time-step for a stable solution.

The results of the LPD simulation using SPH were compared to the previously obtained solution found using the finite volume method. Although the thermal field computed by SPH agreed well with the finite volume results, the SPH velocity field broke down when domains having 738 and 2754 particles were used. The SPH concentration field was simulated correctly when the fluid flow was neglected. After an extensive, systematic review of the SPH coding, it was determined that the SPH method requires several thousands of particles to produce an accurate crystal growth simulation, where realistic Reynolds numbers are involved. Unfortunately, such a large number of particles requires unreasonable simulation times based on available computational power, and therefore SPH does not seem feasible for crystal growth simulations at this time. Future advances in computational power may render the SPH method comparable to the mesh-dependent approaches in terms of computational simulation times. At that point in time, the SPH approach would have significant advantages over mesh-dependent techniques when simulating crystal growth.
# Table of Contents

Abstract i

List of Figures vii

List of Tables x

Acknowledgements xi

Nomenclature xiii

1 Introduction 1

1.1 Motivation and Focus of this Research .................. 1

1.2 Applications of Si$_x$Ge$_{1-x}$ Crystals .................. 2

1.2.1 Photodetectors ........................................ 2

1.2.2 Solar Cells ............................................ 3

1.2.3 Heterojunction Bipolar Transistors (HBT) .............. 3

1.3 Thermodynamic Phase Equilibrium of the Binary Si-Ge System 3

1.4 Crystal Growth by Liquid Phase Diffusion (LPD) ........... 6
TABLE OF CONTENTS

1.5 Thesis Outline ................................................. 9

2 Smoothed Particle Hydrodynamics .................................. 10
  2.1 Mathematical Preliminaries .................................. 10
    2.1.1 Even and Odd Functions .................................. 12
    2.1.2 Multi-Dimensional Taylor Series Expansions ............ 12
    2.1.3 The Dirac-delta Function .................................. 13
  2.2 Smoothed Particle Hydrodynamics (SPH) ......................... 15
    2.2.1 The Kernel Smoothing Function .......................... 17
    2.2.2 The Smoothing Length .................................... 23
    2.2.3 Spatial Derivatives in SPH .............................. 25
    2.2.4 Particle Approximations in SPH ......................... 35
    2.2.5 Boundary Conditions in SPH ............................. 39
    2.2.6 Corrective SPH .................................. 41
    2.2.7 Neighbor Searching in SPH ............................ 45
    2.2.8 Quasi-Incompressible SPH ............................... 46
    2.2.9 Time Integration in SPH ............................... 48
    2.2.10 Projective SPH ...................................... 50
    2.2.11 Advantages and Shortcomings of SPH for Crystal Growth Simulations ......................................... 55

3 SPH Simulation Test Problems .................................... 58
TABLE OF CONTENTS

3.1 The Basic Equations of Thermomechanics of Continuous Media 59

3.2 2D Steady-State Heat Conduction with Isothermal Boundary Conditions .......................... 61

3.3 2D Transient Heat Conduction with Isothermal Boundary Conditions ................................ 64

3.4 2D Transient Heat Conduction with Convective and Adiabatic Boundary Conditions ............. 67

3.5 1D Poiseuille Flow with No-Slip and Periodic Boundary Conditions ................................. 73

3.6 1D Couette Flow with No-Slip and Periodic Boundary Conditions ..................................... 76

3.7 2D Flow Field Over a Cavity with No-Slip Boundary Conditions ..................................... 79

4 Governing Equations for Crystal Growth by LPD 89

4.1 Governing Equations for the Liquid Phase ................................................................. 91

4.2 Governing Equations for the Solid Phase ......................................................................... 92

4.3 PSPH Governing Equations for the Liquid and Solid Phases .......................................... 92

4.4 Boundary, Initial, and Interface Conditions .................................................................... 95

4.5 SPH Boundary, Initial, and Interface Conditions ............................................................ 97

4.6 Physical Parameters of Si$_x$Ge$_{1-x}$ ................................................................. 99

5 SPH Simulation of LPD Crystal Growth 101

5.1 SPH Coding for LPD Crystal Growth Simulation ......................................................... 104
TABLE OF CONTENTS

5.2 LPD PSPH Simulation Results .......................... 106
5.3 Code Troubleshooting ................................. 110
  5.3.1 Function LPDInitializeParticlesEven.m ........... 110
  5.3.2 Function LPDBoxNeighborFinder.m ................. 110
  5.3.3 Function SecondKernelDerivative.m, KernelGradLPD.m, and KernelGradReMesh.m ................. 111
  5.3.4 Function LPDReMeshPSPH.m ....................... 111
  5.3.5 Function LPDConcentration.m ...................... 112
  5.3.6 Function LPDPPESolverPreConjGM.m ................. 113
  5.3.7 Function LPDVelocity.m .......................... 114

6 Conclusions ........................................... 117
  6.1 Contributions ........................................ 118
  6.2 Future Work ........................................ 118

References ............................................. 119

Appendix A: Covariant Derivatives ..................... 124
List of Figures

Figure 1.1  Si-Ge Binary Phase Diagram .................. 4
Figure 1.2  Representative Si-Ge Phase Diagram .......... 5
Figure 1.3  LPD Crystal Growth Apparatus ................ 6
Figure 1.4  Constitutional Supercooling Diagram .......... 8
Figure 2.1  Dirac-delta Function in 1D .................... 14
Figure 2.2  SPH Illustration in a 2D Domain ............... 16
Figure 2.3  Kernel Approximation for the Dirac-delta Function ... 19
Figure 2.4  2D Cubic Spline Kernel of Equation (2.12) .... 21
Figure 2.5  Illustrative Domain of Particles ............... 23
Figure 2.6  Scatter and Gather Averaging Approaches ....... 23
Figure 2.7  Kernel Function Truncated by a Domain Boundary ... 39
Figure 2.8  2D SPH Cartesian $(x,y)$ Domain ............... 40
Figure 2.9  Box Approach for Neighbor Sorting ............... 46
Figure 2.10 PSPH Concept ........................................ 51

Figure 3.1  SPH Test Problem Domain for $N = 1600$ Particles ... 58
Figure 3.2 2D Steady-State Thermal Problem Domain ............. 61
Figure 3.3 Temperature Profile from Equation (3.9) ............. 62
Figure 3.4 SPH Isothermal Contours for \( h = 0.25 \) cm ......... 63
Figure 3.5 SPH Isothermal Contours for \( h = 0.50 \) cm .......... 64
Figure 3.6 2D Transient Thermal Problem Domain ............. 65
Figure 3.7 Temperature Profiles from Equation (3.11) .......... 66
Figure 3.8 Isothermal Contours from Equations (3.11) and (3.12) . 67
Figure 3.9 Second 2D Transient Thermal Problem Domain ....... 68
Figure 3.10 Temperature Profiles from Equation (3.13) .......... 69
Figure 3.11 Explicit Isothermal Contours ....................... 71
Figure 3.12 Crank-Nicolson Isothermal Contours ................. 72
Figure 3.13 1D Poiseuille Flow Problem Domain ............... 73
Figure 3.14 SPH Poiseuille Flow Velocity Profile for \( N = 1600 \) ... 75
Figure 3.15 1D Couette Flow Problem Domain .................. 77
Figure 3.16 SPH Couette Flow Velocity Profile for \( N = 400 \) ..... 78
Figure 3.17 2D Flow Cavity Problem Domain .................... 79
Figure 3.18 SPH Flow Cavity Velocity Plot for \( N = 441 \) Particles . 81
Figure 3.19 SPH Flow Cavity \( v_y \) at \( y = H/2 \) for \( N = 441 \) Particles . 82
Figure 3.20 SPH Flow Cavity \( v_x \) at \( x = L/2 \) for \( N = 441 \) Particles . 82
Figure 3.21 SPH Flow Cavity Velocity Plot for \( N = 3721 \) Particles . 83
LIST OF FIGURES

Figure 3.22 SPH Flow Cavity $v_y$ at $y = H/2$ for $N = 3721$ Particles 84

Figure 3.23 SPH Flow Cavity $v_x$ at $x = L/2$ for $N = 3721$ Particles 84

Figure 3.24 PSPH Flow Cavity Velocity Plot for $N = 3721$ Particles 86

Figure 3.25 PSPH Flow Cavity $v_y$ at $y = H/2$ 87

Figure 3.26 PSPH Flow Cavity $v_x$ at $x = L/2$ 87

Figure 3.27 Cavity Simulation Times Versus Number of Particles $N$ 88

Figure 4.1 Computational Domain of the LPD Crystal Growth System 89

Figure 5.1 SPH Domains for LPD Simulation 102

Figure 5.2 Finite Volume Domain for LPD Simulation 103

Figure 5.3 PSPH Algorithm for LPD Simulations 105

Figure 5.4 Coarse SPH and FV Temperature Field at $t = 0.5$ Hour 108

Figure 5.5 Fine SPH and FV Temperature Field at $t = 0.5$ Hour 109

Figure 5.6 Coarse SPH and FV Concentration Field at $t = 0.5$ Hour 113

Figure 5.7 Coarse SPH and FV Flow Field at $t = 0.5$ Hour 114
List of Tables

Table 1.1  Characteristic LPD System Dimensions ............ 7

Table 4.1  Physical Properties of LPD Materials ............. 99

Table 4.2  Physical Properties of Quartz at $T = 1300$ K .... 100

Table 5.1  LPD Finite Volume Simulation Mesh Sizes ......... 102

Table 5.2  LPD PSPH Simulation MATLAB Subroutines ........ 104
Acknowledgements

I would like to express my sincerest thanks to Dr. Sadik Dost for his support and guidance throughout this work. It was a privilege to have a supervisor who was so encouraging as my project progressed and eventually developed into what it is. Without his ongoing support, this work could not have been completed. I would also like to thank my beautiful wife Angela for her continuous support, which has made completion of this work a reality. Her help with figures and graphics was especially appreciated, and has greatly contributed to the quality of illustrations within this thesis. I also would like to especially thank my fellow graduate students, and workers in the Crystal Growth Lab. In particular, I would like to express thanks to Dr. Mehmet Yildiz, Mr. Brian Lent, Dr. Yongcai Liu, and Mr. Neil Armour for sharing their knowledge of crystal growth with me.
Dedicated to my loving wife,

*Angela*

and to my parents, *Roger & Geraldine Rook.*
# Nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_{ij}$</td>
<td>Summation parameter in implicit time integration</td>
<td></td>
</tr>
<tr>
<td>$a$</td>
<td>Lattice constant</td>
<td>m</td>
</tr>
<tr>
<td>$a_g$</td>
<td>Acceleration due to gravity</td>
<td>m/s$^2$</td>
</tr>
<tr>
<td>$a_n$</td>
<td>Dimensional kernel coefficient</td>
<td></td>
</tr>
<tr>
<td>$C$</td>
<td>Silicon species concentration</td>
<td>kg/kg</td>
</tr>
<tr>
<td>$C_0$</td>
<td>Reference silicon concentration</td>
<td>kg/kg</td>
</tr>
<tr>
<td>$C_{CFL}$</td>
<td>Courant-Friedrichs-Lewy constant</td>
<td></td>
</tr>
<tr>
<td>$C_{L,eq}$</td>
<td>Liquid phase equilibrium mass fraction of silicon</td>
<td>kg/kg</td>
</tr>
<tr>
<td>$C_{S,eq}$</td>
<td>Solid phase equilibrium mass fraction of silicon</td>
<td>kg/kg</td>
</tr>
<tr>
<td>$c$</td>
<td>Arbitrary constant</td>
<td></td>
</tr>
<tr>
<td>$c_i$</td>
<td>Speed of sound for particle $i$</td>
<td>m/s</td>
</tr>
<tr>
<td>$c_p$</td>
<td>Specific heat capacity</td>
<td>J/kg K</td>
</tr>
<tr>
<td>$c_{p,L}$</td>
<td>Specific heat capacity of the liquid solution</td>
<td>J/kg K</td>
</tr>
<tr>
<td>$c_{p,S}$</td>
<td>Specific heat capacity of the solid phase</td>
<td>J/kg K</td>
</tr>
<tr>
<td>$D_L$</td>
<td>Diffusion coefficient of silicon</td>
<td>m$^2$/s</td>
</tr>
<tr>
<td>$(\hat{e}^1, \hat{e}^2, \ldots, \hat{e}^n)$</td>
<td>$n$-dimensional contravariant unit basis vectors</td>
<td></td>
</tr>
<tr>
<td>$(\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_n)$</td>
<td>$n$-dimensional covariant unit basis vectors</td>
<td></td>
</tr>
</tbody>
</table>
### NOMENCLATURE

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f(x)$, $g(x)$</td>
<td>Arbitrary functions of one variable</td>
<td></td>
</tr>
<tr>
<td>$\tilde{f}(\tilde{r})$, $f^\alpha(x^\beta)$</td>
<td>Arbitrary vector field function</td>
<td></td>
</tr>
<tr>
<td>$\tilde{f}^\alpha(x^\beta)$</td>
<td>Arbitrary intermediate vector field function</td>
<td></td>
</tr>
<tr>
<td>$f(x^\beta)$</td>
<td>Arbitrary scalar field function</td>
<td></td>
</tr>
<tr>
<td>$f_1$, $f_2$</td>
<td>Arbitrary field variables</td>
<td></td>
</tr>
<tr>
<td>$F_B^\alpha$</td>
<td>Body force vector due to gravity per unit mass</td>
<td>N/kg</td>
</tr>
<tr>
<td>$F_{ij}^\alpha$</td>
<td>Repulsive boundary force per unit mass</td>
<td>N/kg</td>
</tr>
<tr>
<td>$F_0$</td>
<td>Reference force</td>
<td>N/kg</td>
</tr>
<tr>
<td>$g_{\alpha\beta}$</td>
<td>Fundamental metric tensor</td>
<td></td>
</tr>
<tr>
<td>$g$</td>
<td>Determinant of the metric tensor $g =</td>
<td>g_{\alpha\beta}</td>
</tr>
<tr>
<td>$H$</td>
<td>Characteristic system length</td>
<td>m</td>
</tr>
<tr>
<td>$H_L$</td>
<td>Height of solution</td>
<td>m</td>
</tr>
<tr>
<td>$h$</td>
<td>Constant smoothing length</td>
<td>m</td>
</tr>
<tr>
<td>$h_{ij}$</td>
<td>Variable smoothing length for particles $i$ and $j$</td>
<td>m</td>
</tr>
<tr>
<td>$h_{ij,\text{min}}$</td>
<td>Minimum smoothing length for all $ij$ pairs</td>
<td>m</td>
</tr>
<tr>
<td>$\mathcal{H}$</td>
<td>Convective heat transfer coefficient</td>
<td>W/m²K</td>
</tr>
<tr>
<td>$\mathcal{H}$</td>
<td>Ratio $\mathcal{H}/\kappa$</td>
<td>1/m</td>
</tr>
<tr>
<td>$\tilde{\mathcal{H}}$</td>
<td>Modified heat transfer coefficient</td>
<td>W/m²K</td>
</tr>
<tr>
<td>$L$</td>
<td>Characteristic system length</td>
<td>m</td>
</tr>
<tr>
<td>$L_0$</td>
<td>Reference system length</td>
<td>m</td>
</tr>
<tr>
<td>$m$</td>
<td>Time-step index</td>
<td></td>
</tr>
<tr>
<td>$m_i$</td>
<td>Mass of particle $i$</td>
<td>kg</td>
</tr>
</tbody>
</table>
NOMENCLATURE

\( N \) \hspace{1em} \text{Total number of particles}

\( N_{\text{opt}} \) \hspace{1em} \text{Optimum number of neighboring particles}

\( n \) \hspace{1em} \text{Number of variables, or dimension}

\( n_{\alpha} \) \hspace{1em} \text{Covariant unit surface normal vector}

\( \text{Pr} \) \hspace{1em} \text{Prandtl number}

\( p \) \hspace{1em} \text{Absolute dynamic pressure} \hspace{1em} \text{N/m}^2

\( \tilde{p} \) \hspace{1em} \text{Total pressure} \hspace{1em} \text{N/m}^2

\( q^\alpha \) \hspace{1em} \text{Heat flux vector} \hspace{1em} \text{W/m}^2

\( R \) \hspace{1em} \text{Outside radius of growth cell} \hspace{1em} \text{m}

\( r_0 \) \hspace{1em} \text{Cutoff radius} \hspace{1em} \text{m}

\( \text{Re} \) \hspace{1em} \text{Reynolds number}

\( (r, \theta, z) \) \hspace{1em} \text{3-dimensional cylindrical coordinates}

\( (r, \theta, \phi) \) \hspace{1em} \text{3-dimensional spherical coordinates}

\( S \) \hspace{1em} \text{Surface bounding the spatial domain } \Omega \hspace{1em} \text{m}^2

\( T^\alpha_{\gamma\eta\ldots\lambda} \) \hspace{1em} \text{Arbitrary tensor}

\( T \) \hspace{1em} \text{Absolute temperature} \hspace{1em} \text{K}

\( T_f(z) \) \hspace{1em} \text{Ambient inside furnace temperature} \hspace{1em} \text{K}

\( T_0 \) \hspace{1em} \text{Absolute reference temperature} \hspace{1em} \text{K}

\( T_i \) \hspace{1em} \text{Absolute temperature of particle } i \hspace{1em} \text{K}

\( T_0, T_1, T_2 \) \hspace{1em} \text{Initial or boundary temperatures} \hspace{1em} \text{K}

\( t \) \hspace{1em} \text{Time coordinate} \hspace{1em} \text{s}

\( t_{\text{max}} \) \hspace{1em} \text{End time for a simulation} \hspace{1em} \text{s}
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_a$</td>
<td>Covariant unit surface tangent vector</td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>Time interval</td>
</tr>
<tr>
<td>$u_y$</td>
<td>Crystal growth velocity</td>
</tr>
<tr>
<td>$\Delta V_i$</td>
<td>Volume of particle $i$</td>
</tr>
<tr>
<td>$v^\alpha$</td>
<td>Fluid velocity</td>
</tr>
<tr>
<td>$\bar{v}^\alpha$</td>
<td>Intermediate fluid velocity</td>
</tr>
<tr>
<td>$v_x, v_y$</td>
<td>Fluid velocity in the $x$ and $y$-directions</td>
</tr>
<tr>
<td>$v_r, v_z$</td>
<td>Fluid velocity in the $r$ and $z$-directions</td>
</tr>
<tr>
<td>$\bar{v}_r, \bar{v}_z$</td>
<td>Intermediate fluid velocity in the $r$ and $z$-directions</td>
</tr>
<tr>
<td>$v_0$</td>
<td>Reference fluid velocity</td>
</tr>
<tr>
<td>$W(x^\beta - x_0^\beta, h)$</td>
<td>SPH kernel function</td>
</tr>
<tr>
<td>$W(x_{ij}, h_{ij})$</td>
<td>SPH kernel function</td>
</tr>
<tr>
<td>$x$</td>
<td>Atomic percent silicon</td>
</tr>
<tr>
<td>$x^\alpha$</td>
<td>Position vector</td>
</tr>
<tr>
<td>$x_0^\alpha$</td>
<td>Initial position vector</td>
</tr>
<tr>
<td>$x_i^\alpha$</td>
<td>Position vector of particle $i$</td>
</tr>
<tr>
<td>$x_{ij}^\alpha$</td>
<td>Distance vector between particles $i$ and $j$</td>
</tr>
<tr>
<td>$x_{ij}$</td>
<td>Magnitude of $x_{ij}^\alpha$</td>
</tr>
<tr>
<td>$\Delta x$</td>
<td>Initial particle spacing</td>
</tr>
<tr>
<td>$(x^1, x^2, \ldots, x^n)$</td>
<td>$n$-dimensional orthogonal curvilinear coordinates</td>
</tr>
<tr>
<td>$(x, y, z)$</td>
<td>3-dimensional Cartesian coordinates</td>
</tr>
<tr>
<td>$\alpha_D$</td>
<td>Thermal diffusivity</td>
</tr>
</tbody>
</table>
**NOMENCLATURE**

\( \alpha_{DL} \) | Thermal diffusivity of the liquid solution | \( m^2/s \)
---|---|---
\( \alpha_{DS} \) | Thermal diffusivity of the solid phase | \( m^2/s \)
\( \beta_C \) | Solutal expansion coefficient | \( 1/\% \text{ Wt.Si} \)
\( \beta_T \) | Thermal expansion coefficient | \( 1/K \)
\( \beta_k, \gamma_l \) | Transcendental roots for indices \( k \) and \( l \)
\( \delta(x - x_0) \) | One-dimensional Dirac-delta function
\( \delta^n(x^\beta - x_0^\beta) \) | \( n \)-dimensional Dirac-delta function
\( \delta^\beta_\beta \) | Kronecker delta
\( \epsilon \) | Small parameter, or energy density
\( \kappa \) | Thermal conductivity | \( \text{W/m K} \)
\( \kappa_L \) | Thermal conductivity of the liquid solution | \( \text{W/m K} \)
\( \kappa_S \) | Thermal conductivity of the solid crystal | \( \text{W/m K} \)
\( \mu \) | Dynamic viscosity | \( \text{kg/m s} \)
\( \nu \) | Kinematic viscosity | \( m^2/s \)
\( \rho \) | Mass density | \( \text{kg/m}^3 \)
\( \Delta \rho \) | Mass density difference | \( \text{kg/m}^3 \)
\( \rho_0 \) | Reference mass density | \( \text{kg/m}^3 \)
\( \rho_i \) | Mass density of particle \( i \) | \( \text{kg/m}^3 \)
\( \rho_L \) | Liquid phase density | \( \text{kg/m}^3 \)
\( \rho_S \) | Solid phase density | \( \text{kg/m}^3 \)
\( \sigma^{\alpha\beta} \) | Stress tensor | \( \text{N/m}^2 \)
\( \Omega \) | Domain of \( n \)-dimensional space | \( m^n \)
Subscripts and Superscripts

0 Property at a reference value
\( \alpha, \beta, \gamma, \zeta, \eta \) Coordinate component indices
\( \lambda, \mu, \nu, \sigma, \psi \) Coordinate component indices
\( B \) Volumetric body force property
\( C \) Solutal property
\( CFL \) Courant-Friedrichs-Lewy
\( D \) Diffusive property
\( eq \) Property at thermodynamic equilibrium
\( f \) Furnace property
\( g \) Growth property
\( i \) Property of particle \( i \)
\( ij \) Difference of properties between particles \( i \) and \( j \)
\( j \) Property of particle \( j \)
\( k \) Index of summation, or integer constant
\( L \) Liquid phase property
\( l \) Index of summation
\( m \) Time-step index
\( \text{max} \) Maximum value
\( \text{opt} \) Optimum value
\( S \) Solid phase property
\( T \) Thermal property
Differential Operators

\[ \frac{d(\_)}{dt} \quad \text{Total time derivative; Material derivative; Lagrangian derivative} \]

\[ \frac{\partial(\_)}{\partial x^\alpha} \quad \text{Partial derivative with respect to vector } x^\alpha \]

\[ \frac{\partial(\_)}{\partial x_i^\alpha} \quad \text{Partial derivative with respect to vector } x_i^\alpha \]

\[ \frac{\partial(\_)}{\partial x_{ij}} \quad \text{Partial derivative with respect to the scalar magnitude } x_{ij} \]

\[ \frac{\partial^2(\_)}{\partial x_i^\alpha \partial x_i^\beta} \quad \text{Second partial derivative with respect to vectors } x_i^\alpha \text{ and } x_i^\beta \]

\[ (\_);\alpha \quad \text{Covariant derivative with respect to } x^\alpha \]

\[ (\_);\alpha_i \quad \text{Covariant derivative with respect to } x_i^\alpha \]

\[ (\_);\alpha_\beta \quad \text{Second covariant derivative with respect to } x^\alpha \text{ and } x^\beta \]

\[ (\_);\alpha_i\beta_i \quad \text{Second covariant derivative with respect to } x_i^\alpha \text{ and } x_i^\beta \]
1 Introduction

1.1 Motivation and Focus of this Research

Semiconducting crystals are used in many electronics applications. Single crystals are grown using a variety of methods, each having advantages and drawbacks. One promising type of semiconductor is silicon-germanium $\text{Si}_x\text{Ge}_{1-x}$ where $x$ is the atomic percent concentration of silicon.

Single crystals of $\text{Si}_x\text{Ge}_{1-x}$ have been successfully grown [1] using a technique called liquid phase diffusion (LPD). These experiments produced compositionally graded crystals having a compositional range of $0.06 < x < 0.08$.

To try and understand the growth process, a mathematical model based on continuum assumptions and the thermomechanical balance laws can be formulated [1]. These types of models are typically simulated using a mesh-dependent, finite volume discretization. However, modelling the crystal growth interface becomes extremely complicated using a mesh-dependent technique, as the mesh must be fitted to adapt to the contours of the growth surface. Therefore, alternative simulation approaches to crystal growth modelling are being investigated.

Smoothed particle hydrodynamics (SPH) is a relatively new, meshfree alternative approach to simulating physical problems. It is relatively easy to code, and handles free surfaces and interfacial boundaries naturally. However, this method has not yet been successfully applied to crystal growth simulations.
1. INTRODUCTION

In crystal growth applications [2, 3, 4], the system must be simulated as it evolves over a period of several minutes or hours. A review of SPH literature reveals that SPH has not yet been applied to simulate crystal growth. Successful implementation would be a significant contribution to the new field of SPH, as well as an advance in the field of crystal growth. It is the purpose of this thesis to determine whether SPH can be used to model crystal growth. To this end, the LPD growth process is simulated for \( \text{Si}_x\text{Ge}_{1-x} \) using SPH, and the results compared with the known solution based on the current thermomechanical model. The current model uses a mesh-dependent, finite volume approach with an adaptive mesh [5, 6] and is quite difficult to implement.

1.2 Applications of \( \text{Si}_x\text{Ge}_{1-x} \) Crystals

By specifying various values of \( x \) in the \( \text{Si}_x\text{Ge}_{1-x} \) system, the compound semiconductor electronic and optical properties can be customized to the needs of a particular device application. Current applications of \( \text{Si}_x\text{Ge}_{1-x} \) include photodetectors, solar cells, and heterojunction bipolar transistors (HBT).

1.2.1 Photodetectors

Photodetectors operate by converting optical signals into electrical ones. They use the process of absorption, by which a photon is absorbed by the semiconductor, increasing the energy of the material. Maximum absorption efficiencies have been reported for compositions \( 0.10 < x < 0.15 \) [7] at wavelengths of 1.1 to 1.3 \( \mu \text{m} \). As well, \( \text{Si}_x\text{Ge}_{1-x} \) photodetectors used in infrared wavelength applications have reported quantum efficiencies of 40%, where \( 0.4 < x < 0.6 \) [8].
1.2.2 Solar Cells

Solar cells, like photodetectors, convert energy in the form of light into electrical energy. However, solar cells also deliver this power to an external circuit for use. $\text{Si}_x\text{Ge}_{1-x}$ solar cells are a promising alternative to pure silicon because of a higher sensitivity in the long wavelength detector region. Recent studies have examined $\text{Si}_x\text{Ge}_{1-x}$ for $0.885 < x < 0.965$ [9].

1.2.3 Heterojunction Bipolar Transistors (HBT)

Heterojunction bipolar transistors are used in integrated circuits. It has been reported [10, 11] that high current gains have been obtained using $0.74 < x < 0.75$. As well, high germanium content alloys of $\text{Si}_x\text{Ge}_{1-x}$ have been shown [12] to have higher carrier velocities than pure silicon, which allows for better transistor characteristics.

Before discussing the LPD growth process in detail, the necessary principles of thermodynamic phase equilibrium that describe the relationships between solid and liquid phases of the binary Si-Ge system are reviewed.

1.3 Thermodynamic Phase Equilibrium of the Binary Si-Ge System

The phase diagram for the binary Si-Ge system is given below in Figure 1.1 [13] for $x = 0$ (pure germanium) to $x = 1$ (pure silicon). It is seen that for pure germanium, the melting temperature is 938.3°C (1211.45 K). Pure silicon, on the other hand, melts at 1414°C (1687.15 K).

In general, phase diagrams are a function of temperature and pressure. Usually, data is presented at standard atmospheric pressure, so that the phase diagram is only a function of temperature. Technically speaking, the phase
Figure 1.1: Si-Ge Binary Phase Diagram

diagram will differ as the pressure changes. However, in LPD crystal growth, system pressures are very close to atmospheric pressure, which allows use of the Si-Ge phase diagram given in Figure 1.1. There are three distinct regions of the phase diagram – a liquid region, a solid region, and an intermediate region. The top line of the binary phase diagram is called the liquidus curve, above which the material is a liquid. The bottom curve is called the solidus curve, below which the material solidifies. The separation between liquidus and solidus curves is called the miscibility gap. It is apparent that for a general species concentration x, the temperature at which the compound solidifies from the liquid state is different than the temperature which the same compound liquifies from the solid state. Figure 1.2 can be used to illustrate how the thermodynamic phase diagram for a binary system is interpreted.
We assume that at some initial silicon concentration $x_0$ the mixture is a liquid, and therefore at a temperature above the liquidus curve. At this silicon concentration, we define the vertical line $L_1$. The binary liquid is subsequently cooled to temperature $T_1$, at which time the binary mixture will just begin to solidify. Since the mixture is now at $T_1$, the silicon concentration in the solid crystal $x_{S1}$ is found by drawing a vertical line down from the intersection of the line of constant $T_1$ and the solidus curve. By observation from the phase diagram, the concentration of silicon in the solid crystal is greater than the initial silicon concentration in the liquid melt, $x_{S1} > x_0$. Therefore, silicon is consumed preferentially from the liquid melt as the mixture solidifies. In other words, since silicon has a higher melting temperature compared to germanium, it will solidify quicker than the germanium as the liquid melt cools. As the system temperature is decreased further to $T_2$, the silicon concentration in the liquid melt drops to $x_{L2}$, with a concentration of $x_{S2}$ in the solidified crystal. We observe that $x_{S1} > x_{S2}$, which means that as a crystal solidifies, the atomic percentage of silicon will decrease in the direction of the crystal growth. Note that as the solidification process pro-
1. INTRODUCTION

gresses, the silicon concentration decreases in the liquid melt and produces a compositionally graded solid crystal.

1.4 Crystal Growth by Liquid Phase Diffusion (LPD)

The LPD crystal growth system consists of:

• a three-zone furnace

• a quartz crucible that contains all raw materials, which are solids at room temperature

• a quartz ampoule that houses the crucible, along with a quartz pedestal that supports the crucible

• a sealing cap, and

• two quartz rods that hold insulating material around the crucible.

The LPD system is illustrated below in Figure 1.3 [1].

Figure 1.3: LPD Crystal Growth Apparatus
Initially, the LPD growth zone contains three layers of solid material, a silicon source material, a germanium solvent material, and a single crystal of germanium seed material. This single crystal germanium seed is also referred to as the substrate. As the furnace is turned on, a temperature gradient is established along the vertical direction, where the temperature increases with height. This temperature gradient is held constant as the crystal growth progresses. As the furnace temperature reaches the growth temperature, the germanium solvent material completely liquifies, whereas the silicon source material does not, since silicon has a higher melting temperature than germanium. Due to proper selection of the temperature gradient, the germanium substrate will only partially melt. The characteristic system dimensions for the growth zone are given below in Table 1.1.

<table>
<thead>
<tr>
<th>System Dimension</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Height of Si Source Material</td>
<td>3 mm</td>
</tr>
<tr>
<td>Si-Ge Liquid Solution Height</td>
<td>27 mm</td>
</tr>
<tr>
<td>Initial Height of Ge Seed Material</td>
<td>10 mm</td>
</tr>
<tr>
<td>Seed Crystal Diameter</td>
<td>25 mm</td>
</tr>
<tr>
<td>Quartz (Ampoule and Crucible) Wall Thickness</td>
<td>4 mm</td>
</tr>
<tr>
<td>Total Growth Zone Height</td>
<td>40 mm</td>
</tr>
</tbody>
</table>

At the growth temperature, the system will try to maintain thermodynamic phase equilibrium. This fact requires that silicon be dissolved into the germanium solvent. As silicon is dissolved, the Si-Ge solution is formed, where \( x \) is dependent on the temperature across the interface between the silicon source and the binary solution. With the transport of silicon species into the solution towards the germanium substrate, a supersaturated solution is formed in the vicinity just ahead of the substrate-solution interface. Due to equilibrium requirements, the supersaturated solution is constitutionally supercooled, producing solidification of a crystal of a certain silicon concentration as required by the Si-Ge phase diagram [1]. Constitutional super-
cooling refers to the cooled zone ahead of the solid/liquid interface caused by a change in composition in accordance with the phase diagram.

**Figure 1.4** illustrates the concept of constitutional supercooling, where $z$ is a distance associated with the growth of the crystal, and $T_L$ is the liquidus curve in the vicinity of the growth front.

![Constitutional Supercooling Diagram](image)

**Figure 1.4: Constitutional Supercooling Diagram**

As explained previously, as solidification occurs in the Si-Ge system, the Si concentration in the solid crystal $x_S$ is different from the concentration in the liquid melt $x_L$. This difference in concentration produces a variation in the liquidus temperature (the temperature below which solidification begins) as a function of position $z$ ahead of the growth front. If a temperature gradient $T_f$ is applied to the system such that

$$\left.\frac{\partial T_f}{\partial z}\right|_{z=0} < \left.\frac{\partial T_L}{\partial z}\right|_{z=0}$$

constitutional supercooling will occur, since the actual temperature of the system is less than the liquidus temperature in the vicinity of the growth interface, and the system will try to maintain thermal equilibrium. This mechanism of constitutional supercooling leads to solidification and crystal growth in LPD.
1.5 Thesis Outline

Following a review of the necessary mathematics, Chapter 2 introduces the SPH technique. This chapter provides all necessary mathematical derivations, and lays out the assumptions used in the SPH approximations. As well, a Crank-Nicolson implicit time-stepping approach is reviewed. Chapter 3 presents six benchmark problem simulations carried out using SPH. Chapter 4 outlines the governing equations for Si$_x$Ge$_{1-x}$ crystal growth by the LPD method, including the necessary initial and boundary conditions. Chapter 5 presents and compares the results of the SPH and finite volume simulations for the temperature, flow, and concentration profiles for two-dimensional crystal growth. Chapter 6 concludes the thesis with a summary of how SPH can be used to successfully simulate crystal growth problems, as well as some recommendations for future work. Finally, Appendix A gives a mathematical overview of covariant differentiation.
2 Smoothed Particle Hydrodynamics

Before deriving the governing equations used to simulate the growth of Si$_x$Ge$_{1-x}$ crystals, some background must be established to justify the approximations used in the smoothed particle technique. Hence, we begin with the requisite mathematical review.

2.1 Mathematical Preliminaries

For clarity, it is worthwhile taking a moment to explicitly state the notational conventions that will be used throughout this thesis. All vector quantities will be written using the index notation, with greek indices ($\alpha, \beta, \gamma, \zeta, \eta, \lambda, \mu, \nu, \sigma, \psi$) denoting the vector or tensor components exclusively. It is convention in the SPH literature to use latin indices to denote particles, and greek indices to denote tensor components. These components will always be written in contravariant form as superscripts. For example, an $n$-dimensional position vector $\mathbf{x}$ will be written with contravariant components as

$$\mathbf{x} \rightarrow x^\alpha \quad \text{for} \quad \alpha = 1, 2, \ldots, n$$

As well, throughout this thesis the Einstein summation convention is employed, where any repeated index is summed over the range of the index. For example, the quantity $g_{\alpha \beta} x^\alpha$ is understood to expand as

$$g_{\alpha \beta} x^\alpha = g_{13} x^1 + g_{23} x^2 + \cdots + g_{n3} x^n.$$
If an index is not to be summed, the index (or repeated indices) will be placed in brackets.

Latin indices \((i, j)\) will be used to denote particles, and will always be placed as subscripts. For example, the \(n\)-dimensional position vector of particle \(i\) is denoted

\[
\vec{x}_i \rightarrow x^\alpha_i \quad \text{for} \quad \alpha = 1, 2, \ldots, n
\]

Note that all vectors will be written in contravariant form. However, in practical instances, we work with so-called “physical components”. In general, for any \(n\)-dimensional orthogonal coordinate system having fundamental metric tensor \(g_{\alpha\beta} = 0\) for \(\alpha \neq \beta\), a physical component can be determined by taking the contravariant component and multiplying by the appropriate term in the metric [14]. The physical components \(x(\alpha)\) are related to contravariant components \(x^\alpha\) through the metric tensor as

\[
x(1) = x^1 \sqrt{g_{11}} \quad x(2) = x^2 \sqrt{g_{22}} \quad \cdots \quad x(n) = x^{(n)} \sqrt{g_{(nn)}}
\]

For a higher-order vector \(T^{\alpha\beta\cdots\gamma}_{\zeta\eta\cdots\lambda}\) (namely a tensor) the physical components \(T(\alpha\beta\cdots\gamma\zeta\eta\cdots\lambda)\) are given by

\[
T(\alpha\beta\cdots\gamma\zeta\eta\cdots\lambda) = T^{(\alpha)(\beta)\cdots(\gamma)}_{(\zeta)(\eta)\cdots(\lambda)} \sqrt{g_{(\alpha\alpha)}} \sqrt{g_{(\beta\beta)}} \cdots \sqrt{g_{(\gamma\gamma)}} \sqrt{g_{(\zeta\zeta)}} \sqrt{g_{(\eta\eta)}} \cdots \sqrt{g_{(\lambda\lambda)}}
\]

(2.1)

where the brackets surrounding the indices remind us that there are to be no summations. In Cartesian coordinates, because \(g_{11} = g_{22} = \cdots = g_{(nn)} = 1\), the covariant, contravariant, and physical components are all identical. In three-dimensional Cartesian coordinates, the metric is simply the identity tensor

\[
g_{\alpha\beta}^{\text{Cartesian}} = \begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix}.
\]

In cylindrical \((r, \theta, z)\) coordinates, the metric has the form

\[
g_{\alpha\beta}^{\text{cylindrical}} = \begin{pmatrix}
1 & 0 & 0 \\
0 & r^2 & 0 \\
0 & 0 & 1
\end{pmatrix}.
\]
2.1.1 Even and Odd Functions

A function \( f(x) \) is called even if it satisfies the condition

\[
f(x) = f(-x).
\]

Graphically speaking, even functions are symmetric about the \( f(x) \) axis. For example, it is easily verified that the function \( f(x) = x^2 \) is an even function.

A function \( g(x) \) is odd if it satisfies

\[
g(-x) = -g(x).
\]

Examples of odd functions are \( g(x) = 1/x \) and \( g(x) = x^3 \). It is well understood that for any odd function \( g(x) \), an integration of the function over all space is equal to zero, that is,

\[
\int_{-\infty}^{+\infty} g(x) dx = 0 \quad \text{if} \quad g(-x) = -g(x).
\]

Also, recall that even and odd functions exhibit the following properties:

- the product of an even and an odd function is an odd function
- the derivative of an even function is odd
- the derivative of an odd function is even.

2.1.2 Multi-Dimensional Taylor Series Expansions

All continuous functions can be approximated using a Taylor series expansion. In a single dimension, a function \( f(x) \) of one variable \( x \) can be approximated by taking a sufficient number of terms in the following series

\[
f(x) = \sum_{k=0}^{\infty} \left( \frac{(x - x_0)^k}{k!} \frac{d^k f(x)}{dx^k} \right)_{x=x_0}
\]

\[
= f(x_0) + (x - x_0) \frac{df(x)}{dx}_{x=x_0} + \frac{(x - x_0)^2}{2} \frac{d^2 f(x)}{dx^2}_{x=x_0} + \cdots
\]
where \( x_0 \) is the center point of the expansion and \( k \) is the index of summation. The Taylor series can also be extended to higher-dimensional vector field functions. In general, a vector function \( \bar{f}(\bar{r}) \) of the position vector \( \bar{r} = x^1 \hat{e}_1 + x^2 \hat{e}_2 + \cdots + x^n \hat{e}_n \) can be represented by the following Taylor series, expanded around the vector location \( \bar{r}_0 \) as

\[
\bar{f}(\bar{r}) = \sum_{k=0}^{\infty} \left[ \frac{1}{k!} \left( (\bar{r} - \bar{r}_0) \cdot \nabla_{\bar{r}} \right)^k \bar{f}(\bar{r}) \right]_{\bar{r} = \bar{r}_0}.
\] (2.2)

Now, using the component notation, a vector function \( f^\alpha(x^1, x^2, \ldots, x^n) = f^\alpha(x^\beta) \) of the position vector \( x^\beta \) can be represented by the following Taylor series, expanded around the vector location \( x_0^\beta \) as

\[
f^\alpha(x^\beta) = \sum_{k=0}^{\infty} \left[ \frac{1}{k!} [(x^\gamma - x_0^\gamma)(x^\beta)]^k f^\alpha(x^\beta) \right]_{x^\beta = x_0^\beta}
\] (2.3)

where the \( (x^\gamma)_{\gamma} \) operator denotes covariant differentiation with respect to coordinate \( x^\gamma \) and we are taking advantage of the summation convention for the repeated index \( \gamma \). Covariant derivatives are briefly reviewed in Appendix A. If we expand Equation (2.3) we obtain explicitly,

\[
f^\alpha(x^\beta) = f^\alpha(x_0^\beta) + (x^\gamma - x_0^\gamma)f^\alpha_{\gamma\alpha}(x_0^\beta)
+ \frac{x^\lambda - x_0^\lambda}{2} [(x^\gamma - x_0^\gamma)f^\alpha_{\gamma\alpha}(x_0^\beta)]_{,\lambda} \bigg|_{x^\beta = x_0^\beta}
+ \cdots
= f^\alpha(x_0^\beta) + (x^\gamma - x_0^\gamma)f^\alpha_{\gamma\alpha}(x_0^\beta)
+ \frac{(x^\gamma - x_0^\gamma)(x^\lambda - x_0^\lambda)}{2} f^\alpha_{\gamma\lambda}(x_0^\beta) \bigg|_{x^\beta = x_0^\beta}
+ \cdots
\] (2.4)

since the difference vector \( x^\gamma - x_0^\gamma \) is a constant and can be pulled out of the covariant derivative expression.

### 2.1.3 The Dirac-delta Function

The one-dimensional Dirac-delta function can be written as

\[
\delta(x - x_0) = \lim_{\epsilon \to 0} \begin{cases} 
0 & \text{when } x < x_0 - \epsilon \\
1/2\epsilon & \text{when } x_0 - \epsilon \leq x \leq x_0 + \epsilon \\
0 & \text{when } x > x_0 + \epsilon
\end{cases}
\]
2. SMOOTHED PARTICLE HYDRODYNAMICS

where \( \epsilon \) is a small parameter, and \( x_0 \) is the point in space where the impulse occurs. The Dirac-delta function is an even function, since \( \delta(x - x_0) = \delta(x_0 - x) \). One of the interesting properties of this function is that it satisfies the identity

\[
f(x_0) = \int_{-\infty}^{+\infty} f(x)\delta(x - x_0)\,dx = \lim_{\epsilon \to 0} \int_{x_0-\epsilon}^{x_0+\epsilon} f(x)\delta(x - x_0)\,dx
\]
as well as the special case for the constant function \( f(x) = 1 \) where

\[
\int_{-\infty}^{+\infty} \delta(x - x_0)\,dx = 1.
\]

The one-dimensional Dirac-delta function is illustrated below in Figure 2.1.

![Figure 2.1: Dirac-delta Function in 1D](image)

In higher dimensions, for an \( n \)-dimensional general curvilinear coordinate system having coordinates \( (x^1, x^2, \ldots, x^n) \), the Dirac-delta function with impulse at coordinate point \( (x_0^1, x_0^2, \ldots, x_0^n) \) can be generally expressed as

\[
\delta^n(x^\alpha - x_0^\alpha) = \frac{1}{\sqrt{g}} \delta(x^1 - x_0^1)\delta(x^2 - x_0^2) \cdots \delta(x^n - x_0^n)
\]

where \( g = |g_{\alpha\beta}| \) is the determinant of the fundamental metric tensor for the curvilinear system of coordinates. In cylindrical \((r, \theta, z)\) coordinates, \( g = r^2 \)
so that $\sqrt{g} = r$. In spherical $(r, \theta, \phi)$ coordinates, $g = r^4 \sin^2 \theta$ so that $\sqrt{g} = r^2 \sin \theta$. Recall that in $n$-dimensions, we still have the Dirac-delta identity

$$\int \cdots \int_\Omega \delta^n (x^\beta - x_0^\beta) d\Omega = 1,$$

when the integration is carried out throughout the space $\Omega$. This is consistent with the understanding that the differential volume element for an arbitrary orthogonal coordinate system $(x^1, x^2, \ldots, x^n)$ is written as

$$d\Omega = \sqrt{g} dx^1 dx^2 \cdots dx^n.$$

The corresponding $n$-dimensional Dirac-delta identity of interest is then

$$f^\alpha (x_0^\beta) = \int \cdots \int_\Omega f^\alpha (x^\beta) \delta^n (x^\beta - x_0^\beta) d\Omega.$$

As well, since the Dirac-delta function is an even function, it is also correct to write the above identity as

$$f^\alpha (x_0^\beta) = \int \cdots \int_\Omega f^\alpha (x^\beta) \delta^n (x_0^\beta - x^\beta) d\Omega. \quad (2.5)$$

The identity in Equation (2.5) is an exact mathematical relationship, and is the starting point for the smoothed particle hydrodynamics (SPH) technique.

### 2.2 Smoothed Particle Hydrodynamics (SPH)

Smoothed particle hydrodynamics is an adaptive, meshfree, Lagrangian numerical approximation technique used for modelling physical problems. Unlike Eulerian computational techniques such as the finite volume and finite element methods, SPH does not require a grid, as it is formulated within a particle (Lagrangian) framework. Each "particle" in the domain can be associated with one discrete physical object, or it may represent a macroscopic part of the continuum [15]. The continuum is represented by an ensemble of
particles each carrying mass, momentum, and other hydrodynamic properties (such as temperature, pressure, internal energy, and entropy).

Although originally proposed in the context of cosmological simulations [16, 17] SPH has become increasingly generalized to handle many types of fluid and solid mechanics problems [18, 19, 20, 21]. SPH advantages include relatively easy modelling of complex material surface behaviour, as well as simple implementation of more complicated physics, such as multiple phases, compressibility, solidification, and electromagnetics [22]. In addition, SPH has the advantage of immediate tracking of a material interface due to its Lagrangian nature, as is required in crystal growth simulations; with mesh-dependent techniques such as the finite volume approach, additional mathematics must be introduced to handle an adaptive mesh for a moving interface.

The SPH method involves computing particle properties $f$ using a smoothing, kernel distribution function to account for the effects of surrounding particles. **Figure 2.2** illustrates the basic SPH concept [23] for a particle in a two-dimensional domain $(x^1, x^2)$ at location $x_0^3$.

![Figure 2.2: SPH Illustration in a 2D Domain](image-url)
It is assumed that the properties characteristic of the particle of interest are influenced by all other particles in the global domain. However, one approximation of SPH is to include only the effects of nearby particles, within a smoothing radius $2h$. The length $h$ defines the support domain of the particle of interest. The support domain represents a localized domain over which the kernel will be nonzero. Therefore, the kernel $W(x_0^\beta - x_j^\beta, h)$ is a function of the distance between the particle of interest and a neighboring particle $j$, as well as the smoothing length $h$.

As previously suggested, we begin with Equation (2.5), which tells us that any vector function of the three-dimensional position vector $x_0^\beta$ can be represented as an integral in terms of the Dirac-delta function using the identity

$$f^\alpha(x_0^\beta) = \iiint_{\Omega} f^\alpha(x^\beta) \delta^3(x_0^\beta - x^\beta)d\Omega.$$  

This equation is exact. The fundamental approximation of SPH is to replace the Dirac-delta function with a representative kernel function.

### 2.2.1 The Kernel Smoothing Function

In SPH, the Dirac-delta function is approximated by a kernel distribution function, written as

$$W(x_0^\beta - x^\beta, h)$$

where the position $x_0^\beta$ defines the center point of the kernel function. Since we now have an approximation, we can write a definition of the fundamental SPH approximation (in three-dimensions)

$$f^\alpha(x_0^\beta) \approx \left< f^\alpha(x_0^\beta) \right> = \iiint_{\Omega} f^\alpha(x^\beta)W(x_0^\beta - x^\beta, h)d\Omega \quad (2.6)$$

where $\left< f^\alpha(x_0^\beta) \right>$ is the kernel approximation of the field $f^\alpha(x_0^\beta)$ at location $x_0^\beta$. Recall that integration is carried out over position coordinates referencing
where $d\Omega = \sqrt{g}dx^1dx^2dx^3 = \sqrt{g}dx^\beta$. If the kernel was replaced by the Dirac-delta function, the above equation would be exact.

The kernel approximation of the Dirac-delta function is not arbitrary. There are several conditions that the kernel must satisfy. The first condition is that as the smoothing length $h$ approaches zero, the kernel should approach the Dirac-delta function. That is, in $n$-dimensional space,

$$\lim_{h \to 0} W(x_0^\beta - x^\beta, h) = \delta^n(x_0^\beta - x^\beta).$$

As well, by substituting the Taylor series expansion Equation (2.4) into Equation (2.6) we obtain the expansion:

$$\left\langle f^\alpha(x_0^\beta) \right\rangle = \iiint_\Omega \left[ f^\alpha(x_0^\beta) + (x^\gamma - x_0^\gamma)f^{\alpha}_{,\gamma}(x_0^\beta) + \cdots \right] W(x_0^\beta - x^\beta, h)d\Omega$$

$$= f^\alpha(x_0^\beta) \iiint_\Omega W(x_0^\beta - x^\beta, h)d\Omega$$

$$+ f^{\alpha}_{,\gamma}(x_0^\beta) \iiint_\Omega (x^\gamma - x_0^\gamma)W(x_0^\beta - x^\beta, h)d\Omega$$

$$+ \frac{f^{\alpha}_{,\gamma\lambda}(x_0^\beta)}{2} \iiint_\Omega (x^\gamma - x_0^\gamma)(x^\lambda - x_0^\lambda)W(x_0^\beta - x^\beta, h)d\Omega + O(h^3)$$

where $O(h^3)$ indicates that the written approximation is third-order accurate in the smoothing length $h$. Technically speaking, a given accuracy is $O(x_0^\beta - x^\beta)$, but since by definition $\|x_0^\beta - x^\beta\| \leq 2h$, it is common to write that a given approximation is $O(h)$. From Equation (2.7) it is immediately observed that for $(k+1)$-order accuracy, the kernel function should satisfy the normalization condition

$$\iiint_\Omega W(x_0^\beta - x^\beta, h)d\Omega = 1$$

as well as the higher-moment consistency conditions

$$\iiint_\Omega (x_0^\gamma - x^\gamma)^lW(x_0^\beta - x^\beta, h)d\Omega = 0, \quad \text{for} \quad 0 < l \leq k.$$
That is, substituting the above conditions into Equation (2.7) produces

\[ \left\langle f^\alpha(x_0^\beta) \right\rangle = f^\alpha(x_0^\beta) + \mathcal{O}(h^{k+1}). \]

The final condition the kernel should satisfy is that it is a symmetric, or even function. For interior particles within the domain, all integrals with odd multiples of \((x^\gamma - x_0^\gamma)\) on the right hand side of Equation (2.8) are identically zero; an odd function \((x^\gamma - x_0^\gamma)\) multiplied by an even function \(W(x_0^\beta - x_\beta, h)\) is an odd function, which, when integrated over all space is zero. We are therefore left with the result

\[ \left\langle f^\alpha(x_0^\beta) \right\rangle = f^\alpha(x_0^\beta) + \mathcal{O}(h^2) \]

for interior particles.

Several variations on the choice of kernel have been proposed, including a Gaussian kernel distribution, spline distributions, and others. Stability, accuracy and speed of the SPH simulation depends greatly on the choice of the smoothing kernel distribution, and the smoothing length \(h\). An illustrative kernel approximation for a two-dimensional domain is shown below in Figure 2.3.

![Kernel Approximation for the Dirac-delta Function](image-url)
Kernels should also exhibit compact support, which means that the kernel function goes to zero beyond the smoothing distance $2h$. For example, the two-dimensional SPH Gaussian kernel

$$W(x_{ij}, h) = \frac{1}{\pi h^2} e^{-\frac{x_{ij}^2}{h^2}}$$

does not have the property of compact support, since the exponential nature of the function prevents the kernel from ever quite reaching zero. Note that we have used the $(\cdot)_{ij}$ difference notation

$$x_{ij} \equiv \left\| x_i^\beta - x_j^\beta \right\|$$

$$= \sqrt{g_{\alpha\beta}(x_i^\alpha - x_j^\alpha)(x_i^\beta - x_j^\beta)}$$

$$= \sqrt{g_{\alpha\beta}x_i^\alpha x_j^\beta}$$

$$= \sqrt{g_{11}x_{ij}^{x_1^1} x_{ij}^{x_1^1} + g_{22}x_{ij}^{x_1^2} x_{ij}^{x_1^2} + \cdots + g_{nn}x_{ij}^{x_1^n} x_{ij}^{x_1^n}} \quad (2.11)$$

since kernel functions are radially symmetric, implying that only the magnitude of the distance between the particles has an influence on the value of the kernel. Unlike the Gaussian kernel, the cubic spline kernel

$$W(x_{ij}, h) = a_n \begin{cases} \frac{3}{2} - \left(\frac{x_{ij}}{h}\right)^2 + \frac{1}{2} \left(\frac{x_{ij}}{h}\right)^3 & \text{if } 0 \leq x_{ij} < h \\ \frac{1}{6} \left(2 - \frac{x_{ij}}{h}\right)^3 & \text{if } h \leq x_{ij} < 2h \\ 0 & \text{if } x_{ij} \geq 2h \end{cases} \quad (2.12)$$

is compactly supported. In Equation (2.12), $n$ is the dimension of the space, and the coefficient $a_n$ is dependent on the smoothing length $h$ and has values

$$a_1 = \frac{1}{h} \quad a_2 = \frac{15}{7\pi h^2} \quad a_3 = \frac{3}{2\pi h^3}$$

in one, two, and three-dimensions, respectively. This kernel distribution is used quite often in the literature, and will be used in all simulations throughout this thesis, except for simulations involving fluid flow. The cubic spline kernel distribution of Equation (2.12) is illustrated below for a two-dimensional $(x_{ij}^1, x_{ij}^2)$ domain in Figure 2.4.
For fluid flow simulations at low Reynolds numbers, it has been found that the cubic spline kernel of Equation (2.12) produces inaccurate pressure and velocity fields, whereas the higher-order quintic spline kernel remains stable [24]. The quintic spline kernel is given as [25]

$$W(x_{ij}, h) = \frac{7}{478\pi h^2} \begin{cases} 
(3 - \frac{x_{ij}}{h})^5 & \text{if } 0 \leq x_{ij} < h \\
-6 \left(2 - \frac{x_{ij}}{h}\right)^5 & \text{if } h \leq x_{ij} < 2h \\
15 \left(1 - \frac{x_{ij}}{h}\right)^5 & \text{if } 2h \leq x_{ij} < 3h \\
0 & \text{if } x_{ij} \geq 3h 
\end{cases}$$

in two dimensions, and will be used to simulate fluid flow in this thesis.

In order to determine the gradient of the kernel, we consider the case of two particles \(i\) and \(j\) having kernel \(W(x_{ij}, h)\). In \(n\)-dimensional orthogonal curvilinear coordinates \((x^1, x^2, \ldots, x^n)\), we have that the gradient of the kernel with respect to particle \(i\) coordinates in terms of contravariant basis vectors
\( \hat{e}^1, \hat{e}^2, \ldots, \hat{e}^n \) is
\[
\frac{\partial W(x_{ij}, h)}{\partial x_i^\alpha} \hat{e}^\alpha = \frac{\partial W(x_{ij}, h)}{\partial x_i^1} \hat{e}^1 + \frac{\partial W(x_{ij}, h)}{\partial x_i^2} \hat{e}^2 + \ldots + \frac{\partial W(x_{ij}, h)}{\partial x_i^n} \hat{e}^n \tag{2.14}
\]
By the chain rule, in the \( x_i^1 \)-direction we have that
\[
\frac{\partial W(x_{ij}, h)}{\partial x_i^1} = \frac{\partial x_{ij}}{\partial x_i^1} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} + \frac{\partial h}{\partial x_i^1} \frac{\partial W(x_{ij}, h)}{\partial h}.
\]
Therefore, using Equation (2.11) in the above equation we obtain
\[
\frac{\partial W(x_{ij}, h)}{\partial x_i^1} = 2g_{11}x_{ij}^1 \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} + \frac{\partial h}{\partial x_i^1} \frac{\partial W(x_{ij}, h)}{\partial h} = \frac{g_{11}x_{ij}^1}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} + \frac{\partial h}{\partial x_i^1} \frac{\partial W(x_{ij}, h)}{\partial h}
\]
Analogous results are obtained in the \( x_i^2, x_i^3, \ldots, x_i^n \)-directions, producing the following relation for the components of Equation (2.14)
\[
\frac{\partial W(x_{ij}, h)}{\partial x_i^\alpha} = \frac{g_{\alpha\beta}x_{ij}^\beta}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} + \frac{\partial h}{\partial x_i^\alpha} \frac{\partial W(x_{ij}, h)}{\partial h} \tag{2.15}
\]
since in an orthogonal coordinate system \( g_{\alpha\beta} = 0 \) for \( \alpha \neq \beta \). Performing the identical procedure with differentiation with respect to \( j \) coordinates, it will be shown that
\[
\frac{\partial W(x_{ij}, h)}{\partial x_j^\alpha} = -\frac{g_{\alpha\beta}x_{ij}^\beta}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} + \frac{\partial h}{\partial x_j^\alpha} \frac{\partial W(x_{ij}, h)}{\partial h}. \tag{2.16}
\]
If the smoothing length is constant, we obtain the important result
\[
\frac{\partial W(x_{ij}, h)}{\partial x_i^\alpha} = -\frac{\partial W(x_{ij}, h)}{\partial x_j^\alpha}
\]
or alternatively,
\[
W_{i,\alpha}(x_{ij}, h) = -W_{j,\alpha}(x_{ij}, h) \tag{2.17}
\]
since the covariant derivative of a scalar field reduces to the partial derivative of the field. The second derivative of the kernel \( W_{i,\alpha\beta}(x_{ij}, h) \) can be derived for a constant smoothing length as
\[
\frac{\partial^2 W(x_{ij}, h)}{\partial x_i^\alpha \partial x_i^\beta} = \frac{g_{\alpha\gamma}g_{\beta\delta}x_{ij}^\gamma x_{ij}^\delta}{x_{ij}x_{ij}} \left[ \frac{\partial^2 W(x_{ij}, h)}{\partial x_i^\gamma \partial x_i^\delta} - \frac{1}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} \right] + \frac{g_{\alpha\beta}}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}}. \tag{2.18}
\]
2.2.2 The Smoothing Length

The spatial resolution of SPH is limited by the smoothing length scale, \( h \). In order to optimize the use of the Lagrangian approach in SPH, we should allow a variable \( h \), depending on local number densities of particles. Namely, in high particle density regions of the domain, \( h \) should be small; in low particle density regions of the domain, \( h \) should be large. This scenario is illustrated in Figure 2.5.

![Figure 2.5: Illustrative Domain of Particles](image)

In order to average the hydrodynamic quantities in SPH, the smoothing length \( h \) must be given. There are two different interpretations of local averaging in SPH, the scatter approach and the gather approach, illustrated below in Figure 2.6.

![Figure 2.6: Scatter and Gather Averaging Approaches](image)
The scatter process has particle $i$ collecting contributions from particles $j$, $k$, and $m$, as the smoothing lengths of these particles “scatter” onto location $i$. In the particle averaging summations, the kernel for particles $i$ and $j$ is of the form $W(x_i^\beta - x_j^\beta, h_j)$. The alternative method is for particle $i$ to “gather” the contributions from all particles whose centers fall within the smoothing length $h_i$. In these particle averaging summations, the kernel for particles $i$ and $j$ is of the form $W(x_i^\beta - x_j^\beta, h_i)$. In the special case of a constant smoothing length $h_i = h_j = h$ for all particles $i$ and $j$, the scatter and gather approaches are equivalent. However, for a variable smoothing length, it is possible to violate Newton’s third law. For example, it would be possible for a particle $j$ to put a force on particle $i$, and not experience an equal and opposite reaction force from particle $i$ on particle $j$. To ensure that Newton’s third law is not violated, we make the smoothing length substitution

$$h \rightarrow h_{ij} = \frac{h_i + h_j}{2} \quad (2.19)$$

for all particle pairs $i$ and $j$. This notation will be used interchangeably with $h$ for the purpose of the following derivations. A common way of evolving the smoothing length is according to the divergence of velocity

$$\frac{dh_{ij}}{dt} = \frac{h_{ij}}{n} \left(v_i^\alpha - v_j^\alpha\right)_\alpha \quad (2.20)$$

where $v_i^\alpha$ is the velocity of particle $i$, and $n$ is the number of spatial dimensions. Another simple way of updating $h_{ij}$ in two dimensions is by using [26]

$$h_{ij}^{m+1} = \frac{h_{ij}^m}{2} \left(1 + \sqrt{\frac{N_{\text{opt}}}{N^m}}\right) \quad (2.21)$$

where $m$ is a time-step, $N_{\text{opt}}$ is the optimum number of neighbors within a $2h_{ij}$ radius of the particle of interest, and $N^m$ is the number of particles within $2h_{ij}$ at the previous time-step. Alternative approaches to modification of the smoothing length exist [27], and there is as yet no general consensus on the best approach. In this thesis, unless noted otherwise, Equation (2.20) is used to update the smoothing length, and the gather approach is employed.
2.2.3 Spatial Derivatives in SPH

Since SPH is an approximation technique, there are several different ways of approximating spatial derivatives. In this section, we determine SPH approximations for the gradient of a scalar function, the divergence of a vector function, and the Laplacian of a scalar function.

**SPH Gradient**

Consider a scalar function $f(x^\beta)$ of position $x^\beta$. The gradient of this function is denoted by

$$\nabla f \rightarrow f_{,\alpha}(x^\beta)$$

in component notation. In order to determine the SPH approximation for the gradient of this function, we substitute the quantity $f_{,\alpha}(x^\beta)$ into Equation (2.6) to produce

$$\left\langle f_{,\alpha_0}(x_0^\beta) \right\rangle = \iiint_{\Omega} f_{,\alpha}(x^\beta) W(x_0^\beta - x^\beta, h) d\Omega$$

where the covariant differentiations take place referencing $x^\beta$ coordinates using the $( )_{,\alpha}$ operator, and $x_0^\beta$ coordinates using the $( )_{,\alpha_0}$ operator. Notice that both functions $f(x_0^\beta)$ and $f(x^\beta)$ are functions of position, but they reference different particles; the particle referenced at $x_0^\beta$ is the particle we wish to approximate, and for each one of these particles, an integration will be carried out over coordinates $x_\beta$ so that $d\Omega = \sqrt{g} dx^\beta$. Understand that for any vector or scalar function $f$, the covariant derivatives $f_{,\alpha}$ and $f_{,\alpha_0}$ referencing both coordinates are equivalent – they are both derivatives with respect to spatial position. We distinguish derivatives with respect to coordinates $x^\beta$ and $x_0^\beta$ because these are not equivalent in terms of the gradient of the kernel $W(x_0^\beta - x^\beta, h)$ since it is a function of the difference in position between two particles located at $x_0^\beta$ and $x^\beta$, respectively. Mathematically speaking, the gradient of a function is defined as $f_{,\alpha}(x^\beta)$ when $f = f(x^\beta)$, and $f_{,\alpha_0}$ when
\( f = f(x^\beta_0) \); however, these gradients are identical functions, since it does not matter whether we define the position coordinate as \( x^\beta \) or \( x^\beta_0 \).

Now, using the product rule of calculus, we know that

\[
\left[ f(x^\beta)W(x^\beta_0 - x^\beta, h) \right]_{x^\alpha} = f(x^\beta)W(x^\beta_0 - x^\beta, h) + f(x^\beta)W_{x^\alpha}(x^\beta_0 - x^\beta, h)
\]

or upon rearranging

\[
f_{x^\alpha}(x^\beta_0 - x^\beta, h) = \left[ f(x^\beta)W(x^\beta_0 - x^\beta, h) \right]_{x^\alpha} - f(x^\beta)W_{x^\alpha}(x^\beta_0 - x^\beta, h)
\]

so that

\[
\left\langle f_{x^\alpha_0}(x^\beta_0) \right\rangle = \iiint_{\Omega} \left( f(x^\beta)W(x^\beta_0 - x^\beta, h) \right)_{x^\alpha} d\Omega
\]

\[= \iiint_{\Omega} f(x^\beta)W_{x^\alpha}(x^\beta_0 - x^\beta, h) d\Omega
\]

\[= \iiint_{S} f(x^\beta)W(x^\beta_0 - x^\beta, h)n_{\alpha} dS
\]

\[= \iiint_{\Omega} f(x^\beta)W_{x^\alpha}(x^\beta_0 - x^\beta, h) d\Omega.
\]

In the above equation, we have used the divergence theorem, where \( n_{\alpha} \) is a unit normal vector to the surface \( S \) which bounds the volume \( \Omega \). The bounding surface will, of course, be at the spatial extents of the volume domain. Therefore, the surface integral on the right hand side of the equation will go to zero for all interior particles, since the kernel \( W(x^\beta_0 - x^\beta, h) \to 0 \) at the spatial extents. That is, the kernel exhibits compact support, and equals zero at the spatial extents of the domain for all sufficiently interior particles since the kernel equals zero outside of the support domain \( 2h \). However, near the domain boundary, this result is incorrect, since for \( \| x^\gamma_0 - x^\gamma \| \leq 2h \), we have that \( W(x^\gamma_0 - x^\gamma, h) \neq 0 \). Therefore, for interior particles we can set the surface integral to zero, leaving

\[
\left\langle f_{x^\alpha_0}(x^\beta_0) \right\rangle = -\iiint_{\Omega} f(x^\beta)W_{x^\alpha}(x^\beta_0 - x^\beta, h) d\Omega.
\] (2.22)
By using the chain rule of calculus, we have shown from Equation (2.17) that

\[ W_\alpha(x_0^\beta - x^\beta, h) = -W_\alpha(x_0^\beta - x^\beta, h). \quad (2.23) \]

Therefore, combining the results of Equations (2.23) and (2.22) we obtain the SPH gradient of a function as the integral

\[ \left\langle f_\alpha(x_0^\beta) \right\rangle = \iiint_\Omega f(x^\gamma)W_\alpha(x_0^\beta - x^\beta, h)d\Omega \quad (2.24) \]

This result is especially significant, since it states that the spatial derivatives of particle properties can be found simply by differentiation of the kernel. To determine the error associated with the approximation of Equation (2.24), we can substitute the Taylor series expansion for \( f(x^\gamma) \) using Equation (2.4) into Equation (2.24) to obtain

\[
\left\langle f_\alpha(x_0^\beta) \right\rangle = \iiint_\Omega f(x^\gamma)W_\alpha(x_0^\beta - x^\beta, h)d\Omega \\
= \iiint_\Omega \left[ f(x_0^\beta) + (x_\gamma - x_0^\gamma)f_\gamma(x_0^\beta) + \cdots \right] W_\alpha(x_0^\beta - x^\beta, h)d\Omega \\
= f(x_0^\beta)\iiint_\Omega W_\alpha(x_0^\beta - x^\beta, h)d\Omega \\
+ f_\gamma(x_0^\beta)\iiint_\Omega (x_\gamma - x_0^\gamma)W_\alpha(x_0^\beta - x^\beta, h)d\Omega + \mathcal{O}(h^2) \quad (2.25)
\]

Now since the first integral term on the right hand side is an odd function integrated over all space for interior particles, we have that this integral is identically zero, producing

\[
\left\langle f_\alpha(x_0^\beta) \right\rangle = f_\gamma(x_0^\beta)\iiint_\Omega (x_\gamma - x_0^\gamma)W_\alpha(x_0^\beta - x^\beta, h)d\Omega + \mathcal{O}(h^2). \quad (2.26)
\]

The integral term can be evaluated using the product rule

\[
(x_\gamma - x_0^\gamma)W_\alpha(x_0^\beta - x^\beta, h) = \left[ (x_\gamma - x_0^\gamma)W(x_0^\beta - x^\beta, h) \right]_{\alpha} - \left[ (x_\gamma - x_0^\gamma)_{\alpha}W(x_0^\beta - x^\beta, h) \right].
\]
which, when integrated over $d\Omega = \sqrt{g}dx^\beta$ coordinates produces
\[
\iiint_\Omega (x^\gamma - x_0^\gamma) W_{\alpha_0}(x_0^\beta - x^\beta, h) d\Omega = \iiint_\Omega \left[ (x^\gamma - x_0^\gamma)W(x_0^\beta - x^\beta, h) \right]_{\alpha_0} d\Omega \\
- \iiint_\Omega (x^\gamma - x_0^\gamma)_{,\alpha_0} W(x_0^\beta - x^\beta, h) d\Omega
\]

Now using the divergence theorem, we may rewrite the first integral on the right hand side of the above equation in terms of a surface integral, giving
\[
\iiint_\Omega (x^\gamma - x_0^\gamma) W_{\alpha_0}(x_0^\beta - x^\beta, h) d\Omega = \iint_S (x^\gamma - x_0^\gamma)W(x_0^\beta - x^\beta, h)n_{\alpha_0}dS \\
- \iiint_{\Omega} (x^\gamma - x_0^\gamma)_{,\alpha_0} W(x_0^\beta - x^\beta, h) d\Omega \\
= - \iiint_{\Omega} (-\delta_{\alpha_0}^\gamma) W(x_0^\beta - x^\beta, h) d\Omega \\
= \delta_{\alpha_0}^\gamma
\] (2.27)

where we have used the fact that the kernel satisfies the normalization condition, and goes to zero at the spatial extent of the integration for interior particles. Here, the Kronecker delta $\delta_{\alpha_0}^\gamma$ is defined by
\[
\delta_{\alpha_0}^\gamma = \begin{cases} 
1 & \text{when } \alpha_0 = \gamma \\
0 & \text{when } \alpha_0 \neq \gamma.
\end{cases}
\]

Using Equation (2.27) in Equation (2.26) yields
\[
\left< f_{,\alpha_0}(x_0^\beta) \right> = f_{,\alpha_0}(x_0^\beta) + O(h^2)
\]

which indicates that the SPH approximation to the gradient of a scalar function is at least second-order accurate in the smoothing length $h$ for interior particles. The truncated error term $O(h^2)$ depends on the consistency conditions of Equation (2.10) being satisfied, as well as the higher-order derivatives of $f(x_0^\beta)$.

**SPH Divergence**

To determine the SPH approximation for the divergence of a vector function, let $f^\alpha(x_0^\beta)$ be a differentiable vector function. In component notation, we
have the divergence of a vector function written as

\[ \nabla \cdot \mathbf{f} \rightarrow f^\alpha_{;\alpha}(x^\beta). \]

We begin by substituting \( f^\alpha(x^\beta) \) with the quantity \( f^\alpha_{;\alpha_0}(x^\beta_0) \) in Equation (2.6), producing

\[ \left\langle f^\alpha_{;\alpha_0}(x^\beta_0) \right\rangle = \iiint_\Omega f^\alpha_{;\alpha}(x^\beta)W(x^\gamma_0 - x^\gamma, h)d\Omega. \quad (2.28) \]

Once again, note that the differentiation is with respect to different positional coordinates; \( (\ldots)_{;\alpha_0} \) references \( x^\beta_0 \) coordinates for the particle of interest, and \( (\ldots)_{;\alpha} \) operates with respect to the varying position integrated over all \( x^\beta \) coordinates. We now recall the product rule identity

\[ [f^\alpha(x^\beta)W(x^\gamma_0 - x^\gamma, h)]_{;\alpha} = f^\alpha_{;\alpha}(x^\beta)W(x^\gamma_0 - x^\gamma, h) + f^\alpha(x^\beta)W_{;\alpha}(x^\gamma_0 - x^\gamma, h) \]

which we rearrange as

\[ f^\alpha_{;\alpha}(x^\beta)W(x^\gamma_0 - x^\gamma, h) = [f^\alpha(x^\beta)W(x^\gamma_0 - x^\gamma, h)]_{;\alpha} - f^\alpha(x^\beta)W_{;\alpha}(x^\gamma_0 - x^\gamma, h) \]

Substitution of the above equation into Equation (2.28) produces:

\[ \left\langle f^\alpha_{;\alpha_0}(x^\beta_0) \right\rangle = \iiint_\Omega f^\alpha_{;\alpha}(x^\beta)W(x^\gamma_0 - x^\gamma, h)d\Omega \]

\[ = \iiint_\Omega [f^\alpha(x^\beta)W(x^\gamma_0 - x^\gamma, h)]_{;\alpha} d\Omega \]

\[ - \iiint_\Omega f^\alpha(x^\beta)W_{;\alpha}(x^\gamma_0 - x^\gamma, h)d\Omega \]

We can now employ the divergence theorem to the first integral on the right hand side of the above equation, so that

\[ \left\langle f^\alpha_{;\alpha_0}(x^\beta_0) \right\rangle = \iint_S [f^\alpha(x^\beta)W(x^\gamma_0 - x^\gamma, h)] n_\alpha dS \]

\[ - \iiint_\Omega f^\alpha(x^\beta)W_{;\alpha}(x^\gamma_0 - x^\gamma, h)d\Omega \]
where $n_\alpha$ is the unit surface normal vector, and we are now integrating over the surface $dS$ bounding the volume $d\Omega$. However, since we are again integrating over the entire domain, the surface integral will disappear for interior particles. This produces the relation
\begin{equation}
\left\langle f^\alpha_{\alpha_0}(x^0_0) \right\rangle = -\iiint_\Omega f^\alpha(x^3)W_{\alpha_0}(x^\gamma_0 - x^\gamma,h)d\Omega
\end{equation}
which holds for all interior particles. Now for convenience we would like to convert the $(\ )_{\alpha_0}$ operator on the right hand side of Equation (2.29) into one operating with respect to $x^0_0$ coordinates. Using the result of Equation (2.23), we obtain the SPH approximation for the divergence of a vector field as
\begin{equation}
\left\langle f^\alpha_{\alpha_0}(x^0_0) \right\rangle = \iiint_\Omega f^\alpha(x^3)W_{\alpha_0}(x^\gamma_0 - x^\gamma,h)d\Omega
\end{equation}

**SPH Second Derivatives and the Laplacian**

Another operator that comes up frequently in thermomechanical equations is the Laplacian operator ($\nabla^2$). The tensorial expression for the Laplacian operator is
\[ \nabla^2 f(x^\beta) \rightarrow g^{\gamma\lambda}f_{\gamma\lambda}(x^\beta) \]
However, in order to derive the SPH Laplacian [28], it is convenient to first construct a more general second-order derivative approximation. To determine this derivative, we begin by rewriting the Taylor series expansion of $f^\sigma(x^0_i)$ using Equation (2.4) so that
\[ f^\sigma(x^0_i) - f^\sigma(x^0_j) = (x^\gamma_i - x^\gamma_j)f^\sigma(x^\lambda_i) \]
\[ + \frac{(x^\gamma_i - x^\gamma_j)(x^\lambda_i - x^\lambda_j)}{2} f_{\gamma\lambda}(x^\beta) \bigg|_{x^\beta = x^\beta_i} + \ldots \]
or in the more concise $(\ )_{ij}$ difference notation
\[ f^\sigma(x^0_i) - f^\sigma(x^0_j) = x^\gamma_{ij}f^\sigma(x^0_j) + \frac{x^\gamma_{ij}x^\lambda_{ij}}{2} f_{\gamma\lambda}(x^\beta) \bigg|_{x^\beta = x^\beta_i} + \ldots \]
where all derivatives are with respect to $x_i^\alpha$ coordinates. We multiply the above Taylor expansion with the term

$$\frac{x_i^\alpha}{g_{\mu\nu} x^\mu_{ij} x^\nu_{ij}} \partial W(x_i^\beta, h) = \frac{x_i^\alpha}{g_{\mu\nu} x^\mu_{ij} x^\nu_{ij}} W_\zeta(x_i^\beta, h)$$

producing

$$\left[ f^\sigma (x_i^\beta) - f^\sigma (x_i^\beta) \right] x_i^\alpha \frac{x_i^\alpha}{g_{\mu\nu} x^\mu_{ij} x^\nu_{ij}} W_\zeta(x_i^\beta, h) = \frac{x_i^\gamma x_i^\alpha}{g_{\mu\nu} x^\mu_{ij} x^\nu_{ij}} W_\zeta(x_i^\beta, h) f^\sigma (x_i^\beta)$$

$$+ \frac{x_i^\gamma x_i^\lambda x_i^\alpha}{2g_{\mu\nu} x^\mu_{ij} x^\nu_{ij}} W_\zeta(x_i^\beta, h) f_\gamma^\lambda (x_i^\beta) \bigg|_{x_i^\beta = x_i^\beta} + \cdots$$

We then integrate over all three-dimensional space in $d\Omega = \sqrt{g} dx_i^\alpha$ coordinates which yields, after noting that the first integral term has an integrand with an odd function, and therefore goes to zero when integrated over all space,

$$\int\int\int_{\Omega} \left[ f^\sigma (x_i^\beta) - f^\sigma (x_i^\beta) \right] x_i^\alpha \frac{x_i^\alpha}{g_{\mu\nu} x^\mu_{ij} x^\nu_{ij}} W_\zeta(x_i^\beta, h) d\Omega = \frac{f^\sigma_\gamma (x_i^\beta)}{2} \int\int\int_{\Omega} \frac{x_i^\gamma x_i^\lambda x_i^\alpha}{g_{\mu\nu} x^\mu_{ij} x^\nu_{ij}} W_\zeta(x_i^\beta, h) d\Omega + \cdots$$

(2.31)

Similar to before, we expand the integrand on the right hand side using the product rule while neglecting all higher-order terms in the expansion, so that

$$\frac{x_i^\gamma x_i^\lambda x_i^\alpha}{g_{\mu\nu} x^\mu_{ij} x^\nu_{ij}} W_\zeta(x_i^\beta, h) = \left( \frac{x_i^\gamma x_i^\lambda x_i^\alpha}{g_{\mu\nu} x^\mu_{ij} x^\nu_{ij}} W(x_i^\beta, h) \right)_{\zeta} - W(x_i^\beta, h) \left( \frac{x_i^\gamma x_i^\lambda x_i^\alpha}{g_{\mu\nu} x^\mu_{ij} x^\nu_{ij}} \right)_{\zeta}$$

which produces

$$\int\int\int_{\Omega} \frac{x_i^\gamma x_i^\lambda x_i^\alpha}{g_{\mu\nu} x^\mu_{ij} x^\nu_{ij}} W_\zeta(x_i^\beta, h) d\Omega = \int\int\int_{\Omega} \left( \frac{x_i^\gamma x_i^\lambda x_i^\alpha}{g_{\mu\nu} x^\mu_{ij} x^\nu_{ij}} W(x_i^\beta, h) \right)_{\zeta} d\Omega$$

$$- \int\int\int_{\Omega} W(x_i^\beta, h) \left( \frac{x_i^\gamma x_i^\lambda x_i^\alpha}{g_{\mu\nu} x^\mu_{ij} x^\nu_{ij}} \right)_{\zeta} d\Omega$$

for the integral on the right hand side of Equation (2.31). Using the divergence theorem for the first integral on the right hand side of the above
equation gives
\[
\iiint_{\Omega} \frac{x_{ij}^\gamma x_{ij}^\lambda x_{ij}^\alpha}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} W_{\xi}(x_{ij}^\beta, h) d\Omega = \iint_{S} \left( \frac{x_{ij}^\gamma x_{ij}^\lambda x_{ij}^\alpha}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} W(x_{ij}^\beta, h) \right) n_{\xi} dS \\
- \iiint_{\Omega} W(x_{ij}^\beta, h) \left( \frac{x_{ij}^\gamma x_{ij}^\lambda x_{ij}^\alpha}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} \right)_{\xi} d\Omega.
\]

Once again, we have the surface integral vanishing for all interior particles since the kernel goes to zero at the spatial extents, leaving
\[
\iiint_{\Omega} \frac{x_{ij}^\gamma x_{ij}^\lambda x_{ij}^\alpha}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} W_{\xi}(x_{ij}^\beta, h) d\Omega = - \iiint_{\Omega} W(x_{ij}^\beta, h) \left( \frac{x_{ij}^\gamma x_{ij}^\lambda x_{ij}^\alpha}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} \right)_{\xi} d\Omega. \quad (2.32)
\]

Expansion of the covariant derivative in the integrand of Equation (2.32) yields
\[
\left( \frac{x_{ij}^\gamma x_{ij}^\lambda x_{ij}^\alpha}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} \right)_{\xi} = \frac{1}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} \left( x_{ij}^\gamma x_{ij}^\lambda x_{ij}^\alpha \right)_{\xi} + x_{ij}^\gamma x_{ij}^\lambda x_{ij}^\alpha \frac{1}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} \left( \frac{1}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} \right)_{\xi}
\]
\[
= \frac{x_{ij}^\gamma x_{ij}^\lambda x_{ij}^\alpha}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} + x_{ij}^\gamma x_{ij}^\lambda x_{ij}^\alpha \frac{1}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} \left( g_{\sigma\nu} x_{ij}^\sigma x_{ij}^\nu \right)_{\xi} - x_{ij}^\gamma x_{ij}^\lambda x_{ij}^\alpha \frac{1}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} g_{\sigma\nu} x_{ij}^\sigma x_{ij}^\nu
\]
\[
= \frac{\delta_{\xi} x_{ij}^\gamma x_{ij}^\lambda x_{ij}^\alpha + x_{ij}^\lambda x_{ij}^\lambda x_{ij}^\alpha + x_{ij}^\gamma x_{ij}^\lambda x_{ij}^\alpha}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} - \frac{x_{ij}^\gamma x_{ij}^\lambda x_{ij}^\alpha g_{\sigma\nu} x_{ij}^\sigma x_{ij}^\nu}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} g_{\sigma\nu} x_{ij}^\sigma x_{ij}^\nu
\]
\[
= \frac{\delta_{\xi} x_{ij}^\gamma x_{ij}^\lambda x_{ij}^\alpha + x_{ij}^\lambda x_{ij}^\lambda x_{ij}^\alpha + x_{ij}^\gamma x_{ij}^\lambda x_{ij}^\alpha}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} - \frac{x_{ij}^\gamma x_{ij}^\lambda x_{ij}^\alpha g_{\sigma\nu} x_{ij}^\sigma x_{ij}^\nu}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} g_{\sigma\nu} x_{ij}^\sigma x_{ij}^\nu
\]
\[
= \delta_{\xi} x_{ij}^\gamma x_{ij}^\lambda x_{ij}^\alpha + x_{ij}^\lambda x_{ij}^\lambda x_{ij}^\alpha + x_{ij}^\gamma x_{ij}^\lambda x_{ij}^\alpha
\]
\[
\frac{x_{ij}^\gamma x_{ij}^\lambda x_{ij}^\alpha}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} - \frac{x_{ij}^\gamma x_{ij}^\lambda x_{ij}^\alpha g_{\sigma\nu} x_{ij}^\sigma x_{ij}^\nu}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} g_{\sigma\nu} x_{ij}^\sigma x_{ij}^\nu
\] \quad (2.33)

Now using Equation (2.33) in Equation (2.32) gives the result
\[
\iint_{\Omega} \frac{x_{ij}^\gamma x_{ij}^\lambda x_{ij}^\alpha}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} W_{\xi} d\Omega = \left( \frac{2 x_{ij}^\gamma x_{ij}^\lambda x_{ij}^\alpha g_{\sigma\nu} x_{ij}^\sigma}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} - \frac{\delta_{\xi} x_{ij}^\gamma x_{ij}^\lambda x_{ij}^\alpha + x_{ij}^\lambda x_{ij}^\lambda x_{ij}^\alpha + x_{ij}^\gamma x_{ij}^\lambda x_{ij}^\alpha}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} \right) W d\Omega. \quad (2.34)
\]

Now recall that an isotropic tensor is one that has the same components in all rotated coordinate systems. The fourth-order tensor in the above equation must be an isotropic tensor, since the integrand is an even function. Therefore, a fourth-order isotropic tensor can be written in terms of constants $c_1$, ...
2. SMOOTHED PARTICLE HYDRODYNAMICS

\( c_2, \) and \( c_3, \) the Kronecker delta, and the metric tensor as

\[
a^{\gamma\lambda\alpha}_\zeta = c_1 g^{\gamma\lambda} \delta^\alpha_\zeta + c_2 g^{\gamma\alpha} \delta^\lambda_\zeta + c_3 g^{\lambda\alpha} \delta^\gamma_\zeta
\]

where \( a^{\gamma\lambda\alpha}_\zeta \) is an arbitrary fourth-order isotropic tensor. Since the fourth-order tensor in Equation (2.34) is symmetric in all indices, the constants must all be equal to one another, so that \( c_1 = c_2 = c_3 = c. \) We then have from Equation (2.34) that

\[
\iint_{\Omega} \left( \frac{2x_i^j \partial \frac{\partial x_i^j}{\partial x_i^j} x_i^\alpha g_{ij} x_j^\alpha}{g_{\mu\nu} x_i^\mu x_i^\nu g_{\beta\eta} x_i^\beta x_i^\eta} - \delta_i^\alpha \delta_\zeta \right) W d\Omega = c \left( g^{\gamma\lambda} \delta^\alpha_\zeta + g^{\gamma\alpha} \delta^\lambda_\zeta + g^{\lambda\alpha} \delta^\gamma_\zeta \right)
\]

from the appropriate representation of the fourth-order isotropic tensor. The constant \( c \) can be found from inner multiplication of the above equation with the metric \( g_{\gamma\lambda} \) and contraction on indices \( \alpha \) and \( \zeta \) to produce, after some algebra,

\[
g_{\gamma\lambda} \iint_{\Omega} \left( \frac{2x_i^j \partial \frac{\partial x_i^j}{\partial x_i^j} x_i^\alpha g_{ij} x_j^\alpha}{g_{\mu\nu} x_i^\mu x_i^\nu g_{\beta\eta} x_i^\beta x_i^\eta} - \delta_i^\alpha \delta_\zeta \right) W d\Omega = c \left( \delta_i^\alpha \delta_\zeta + 2 \delta^\alpha_\zeta \right)
\]

\[
g_{\gamma\lambda} \iint_{\Omega} \left( \frac{x_i^j \partial \frac{\partial x_i^j}{\partial x_i^j} x_i^\alpha g_{ij} x_j^\alpha}{g_{\mu\nu} x_i^\mu x_i^\nu g_{\beta\eta} x_i^\beta x_i^\eta} - \delta_i^\alpha \delta_\zeta \right) W d\Omega = c \left( \delta_i^\alpha \delta_\zeta + 2 \delta^\alpha_\zeta \right)
\]

\[
- \iint_{\Omega} \left( \frac{g_{\gamma\lambda} \partial \frac{\partial x_i^j}{\partial x_i^j} x_i^\alpha}{g_{\mu\nu} x_i^\mu x_i^\nu} \right) W d\Omega = c \left( \delta_i^\alpha \delta_\zeta + 2 \delta^\alpha_\zeta \right)
\]

which, since the kernel is normalized, yields \( c = -1/5 \) in three dimensions, where \( \delta^\alpha_\zeta = 3. \) In two-dimensional domains, we have that \( c = -1/4 \) since \( \delta^\alpha_\zeta = 2. \) Substituting \( c = -1/5 \) into Equation (2.35), and using the results of Equations (2.34), and (2.31), we obtain

\[
\iint_{\Omega} \left[ \frac{f^\sigma (x_i^\beta) - f^\sigma (x_j^\beta)}{g_{\mu\nu} x_i^\mu x_i^\nu} \right] x_i^\alpha W_{ij} (x_i^\beta, h) d\Omega = - \frac{f^\sigma \partial \frac{\partial x_j^\beta}{\partial x_j^\beta}}{10} \left( g^{\gamma\lambda} \delta^\alpha_\zeta + g^{\gamma\alpha} \delta^\lambda_\zeta + g^{\lambda\alpha} \delta^\gamma_\zeta \right)
\]
which can be slightly simplified to

$$
\iint \int_\Omega \frac{\left[ f^\sigma(x_j^\beta) - f^\sigma(x_i^\beta) \right]}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} W_\zeta(x_{ij}^\beta, h) d\Omega = \frac{f^\sigma(x_j^\beta)}{10} \left( g^{\gamma\lambda} \delta^\zeta_{\beta} + 2g^{\gamma\alpha} \delta^\zeta_{\beta} \right) .
$$

Equation (2.36)

Contracting indices $\alpha$ and $\zeta$ in Equation (2.36) produces the SPH Laplacian of a field

$$
\left< g^{\gamma\lambda} f^{\sigma}_{\gamma\lambda}(x_j^\beta) \right> = 2 \iint \int_\Omega \frac{\left[ f^\sigma(x_j^\beta) - f^\sigma(x_i^\beta) \right]}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} W_\alpha(x_{ij}^\beta, h) d\Omega .
$$

Equation (2.37)

Now if we use the result of Equation (2.37) in Equation (2.36), we obtain

$$
\iint \int_\Omega \frac{\left[ f^\sigma(x_j^\beta) - f^\sigma(x_i^\beta) \right]}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} W_\zeta(x_{ij}^\beta, h) d\Omega - \frac{g^{\gamma\lambda} f^{\sigma}_{\gamma\lambda}(x_j^\beta) \delta^\zeta_{\beta}}{10} = \frac{\gamma^\alpha f^{\sigma}_{\gamma\lambda}(x_j^\beta) \delta^\lambda_{\beta}}{5}
$$

$$
\iint \int_\Omega \frac{f^\sigma(x_j^\beta) - f^\sigma(x_i^\beta)}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} \left( x_{ij}^\beta W_\zeta - \frac{x_{ij}^\beta W_\gamma \delta^\zeta_{\beta}}{5} \right) d\Omega = \frac{g^{\gamma\alpha} f^{\sigma}_{\gamma\zeta}(x_j^\beta) \delta^\lambda_{\beta}}{5}
$$

Equation (2.38)

Expanding the covariant derivatives of the kernel in Equation (2.38), we obtain, in three dimensions,

$$
\left< g^{\gamma\alpha} f^{\sigma}_{\gamma\zeta}(x_j^\beta) \right> = \iint \int_\Omega \frac{\left[ f^\sigma(x_j^\beta) - f^\sigma(x_i^\beta) \right]}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} \left( 5g_{\gamma\zeta} x_{ij}^\gamma x_{ij}^\alpha - g_{\gamma\zeta} x_{ij}^\gamma x_{ij}^\alpha \delta^\zeta_{\beta} \right) \frac{\partial W}{\partial x_{ij}^\beta} d\Omega
$$

Equation (2.39)

where we have used the result of Equation (2.15) for a constant smoothing length,

$$
W_\alpha = \frac{g_{\alpha\beta} x_{ij}^\beta \partial W(x_{ij}^\beta, h)}{x_{ij}^\beta} .
$$

Equation (2.39) is the general second-order SPH derivative for a vector function in three dimensions, and reduces to Equation (2.37) when $\alpha = \zeta$. In two-dimensional problems, it is easily verified that

$$
\left< g^{\gamma\alpha} f^{\sigma}_{\gamma\zeta}(x_j^\beta) \right> = \iint \int_\Omega \frac{\left[ f^\sigma(x_j^\beta) - f^\sigma(x_i^\beta) \right]}{g_{\mu\nu} x_{ij}^\mu x_{ij}^\nu} \left( 4g_{\gamma\zeta} x_{ij}^\gamma x_{ij}^\alpha - g_{\gamma\zeta} x_{ij}^\gamma x_{ij}^\alpha \delta^\zeta_{\beta} \right) \frac{\partial W}{\partial x_{ij}^\beta} d\Omega
$$

Equation (2.40)
Finally, to be consistent with the other results of this section, Equations (2.24) and (2.30), we convert the Laplacian Equation (2.37) back to $x^\beta$ and $x_0^\beta$ notation, producing

$$\left\langle g^{\gamma\lambda} f_{\gamma\lambda}(x_0^\beta) \right\rangle = 2\int\int\int_{\Omega} \left[ \frac{f(x_0^\beta) - f(x^\beta)}{g_{\mu\nu}(x^\mu - x_0^\mu)(x^\nu - x_0^\nu)} \right] W_{\gamma\nu}(x_0^\beta - x^\beta, h) d\Omega$$

(2.41)

Note that all covariant derivatives in the above equations are with respect to $x_i^\beta$ coordinates.

### 2.2.4 Particle Approximations in SPH

It can be shown [15] that the SPH averaging formulation developed in the previous sections is a linear operation, having the following important properties for two arbitrary field variables $f_1$ and $f_2$ and a constant $c$:

$$\left\langle f_1 + f_2 \right\rangle = \left\langle f_1 \right\rangle + \left\langle f_2 \right\rangle$$

$$\left\langle cf_1 \right\rangle = c \left\langle f_1 \right\rangle$$

The SPH approximations are also commutative, so that

$$\left\langle f_1 + f_2 \right\rangle = \left\langle f_2 + f_1 \right\rangle$$

$$\left\langle f_1 f_2 \right\rangle = \left\langle f_2 f_1 \right\rangle .$$

These properties can be used to derive SPH approximations for complex governing equations.

Now if we recall Equation (2.6) in three dimensions, we had that

$$\left\langle f^\alpha(x_0^\beta) \right\rangle = \int\int\int_{\Omega} f^\alpha(x^\beta) W(x_0^\beta - x^\beta, h) d\Omega$$

for a continuous distribution. If, however, we recognize that this integration will be carried out over all $N$ discrete particles within the domain, we can
reformulate the above equation for the particle of interest \((i)\) as

\[
\left< f^\alpha(x_i^\beta) \right> = \sum_{j=1}^{N} f^\alpha(x_j^\beta)W(x_i^\beta - x_j^\beta, h_{ij})\Delta V_j
\]

where \(\Delta V_j\) is the representative volume of particle \(j\). Usually, the particle volume is expressed in terms of its averaged mass \(m_j\) and density \(\rho_j\). As well, kernels exhibit radial symmetry, which means that only the magnitude of the vector \(x_{ij}^\beta\) is of concern. We then write

\[
\left< f^\alpha(x_i^\beta) \right> = \sum_{j=1}^{N} \frac{m_j}{\rho_j} f^\alpha(x_j^\beta)W(x_{ij}, h_{ij})
\] (2.42)

as the SPH approximation of a vector field at particle \(i\) in terms of all other interacting particles \(j\). As well, the SPH approximation for a scalar field function \(f(x^\beta)\) can be written as

\[
\left< f(x_i^\beta) \right> = \sum_{j=1}^{N} \frac{m_j}{\rho_j} f(x_j^\beta)W(x_{ij}, h_{ij})
\] (2.43)

From the results of Equation (2.24), it is observed that in SPH, spatial differentiation of a function is approximated via differentiation of the kernel \(W(x_{ij}, h_{ij})\) only. This is one of the reasons that SPH is popular, since it eliminates the need for a mesh to compute spatial derivatives. There are several ways in which spatial derivatives can be approximated, however, certain valid approximations are practically deficient in that they lack \(i-j\) particle pair symmetry. For example, using Equation (2.24), we could obtain the discrete approximation for the gradient of a scalar field as

\[
\left< f_{,\alpha}(x_i^\beta) \right> = \sum_{j=1}^{N} \frac{m_j}{\rho_j} f(x_j^\beta)W_{,\alpha}(x_{ij}, h_{ij})
\]

where the covariant derivative \((\ )_{,\alpha}\) operates referencing particle \(i\) coordinates \(x_i^\beta\). However, this approximation will not be symmetric for \(i-j\) particle pairs in general. Therefore, we will use SPH derivative approximations that
maintain particle symmetry. In order to determine the SPH approximation to the gradient of a scalar function $f(x^\beta)$, we begin with the identity

$$\rho_i f_{i\alpha}(x^\alpha) = \left[ \rho f(x^\beta) \right]_{i\alpha} - f(x^\alpha) \rho_{i\alpha}$$

which has an SPH equivalent

$$\rho_i \left\langle f_{i\alpha}(x^\alpha_i) \right\rangle = \left\langle \left[ \rho_i f(x^\beta_i) \right]_{i\alpha} \right\rangle - f(x^\alpha_i) \rho_{i\alpha}$$

Recalling that differentiation operates only on the kernel in the SPH method, using Equation (2.24) we obtain the discrete equation

$$\rho_i \left\langle f_{i\alpha}(x^\beta_i) \right\rangle = \sum_{j=1}^{N} \frac{m_j}{\rho_j} \rho_j f(x^\beta_j) W_{i\alpha}(x_{ij}, h_{ij}) - f(x^\alpha_i) \sum_{j=1}^{N} \frac{m_j}{\rho_j} \rho_j W_{i\alpha}(x_{ij}, h_{ij})$$

$$= \sum_{j=1}^{N} m_j \left[ f(x^\beta_j) - f(x^\beta_i) \right] W_{i\alpha}(x_{ij}, h_{ij})$$

which produces the discrete SPH particle approximation for the gradient of a scalar function as

$$\left\langle f_{i\alpha}(x^\beta_i) \right\rangle = \sum_{j=1}^{N} \frac{m_j}{\rho_i} \left[ f(x^\beta_j) - f(x^\beta_i) \right] W_{i\alpha}(x_{ij}, h_{ij})$$

(2.44)

An alternative SPH gradient approximation found often in the literature begins with the identity

$$\left[ \frac{f(x^\beta)}{\rho} \right]_{i\alpha} = \frac{1}{\rho_i} f_{i\alpha}(x^\beta) - \frac{f(x^\beta)}{\rho^2} \rho_{i\alpha}$$

so that

$$\frac{1}{\rho_i} f_{i\alpha}(x^\beta) = \left[ \frac{f(x^\beta)}{\rho} \right]_{i\alpha} + \frac{f(x^\beta)}{\rho^2} \rho_{i\alpha}$$

which has an SPH equivalent

$$\frac{1}{\rho_i} \left\langle f_{i\alpha}(x^\beta_i) \right\rangle = \left\langle \left[ \frac{f(x^\beta_i)}{\rho_i} \right]_{i\alpha} \right\rangle + \frac{f(x^\beta_i)}{\rho_{i\alpha}} \rho_{i\alpha}$$

Now once again we use the result that differentiation of a function is approximated in SPH by differentiation of the kernel multiplied by a particle
summation, which produces

\[
\frac{1}{\rho_i} \left< f^\alpha(x^\beta_i) \right> = \sum_{j=1}^{N} \frac{m_j}{\rho_j} \frac{f(x^\beta_j)}{\rho_j^2} W_\alpha(x_{ij}, h_{ij}) + \frac{f(x^\beta_i)}{\rho_i^2} \sum_{j=1}^{N} \frac{m_j}{\rho_j} \rho_j W_\alpha(x_{ij}, h_{ij})
\]

\[
= \sum_{j=1}^{N} m_j \left[ \frac{f(x^\beta_j)}{\rho_j^2} + \frac{f(x^\beta_i)}{\rho_i^2} \right] W_\alpha(x_{ij}, h_{ij}).
\]

The above equation is then simplified to give an alternative, symmetric approximation for the SPH gradient of a scalar function as

\[
\left< f^\alpha(x^\beta_i) \right> = \rho_i \sum_{j=1}^{N} m_j \left[ \frac{f(x^\beta_j)}{\rho_j^2} + \frac{f(x^\beta_i)}{\rho_i^2} \right] W_\alpha(x_{ij}, h_{ij})
\]

(2.45)

The particle approximation of the divergence of a vector field in SPH is found by using Equation (2.30) and the identity

\[
[\rho f^\alpha(x^\beta)]_\alpha = \rho f^\alpha(x^\beta) + \rho f^{\alpha}(x^\beta)
\]

to obtain

\[
\left< f^\alpha_{\alpha}(x^\beta_i) \right> = \rho_i \sum_{j=1}^{N} \frac{m_j}{\rho_i} \left[ f^\alpha(x^\beta_j) - f^\alpha(x^\beta_i) \right] W_\alpha(x_{ij}, h_{ij})
\]

(2.46)

where the covariant derivative \( (\_)_\alpha \) operates referencing particle \( i \) coordinates \( x^\beta_i \).

The Laplacian operation presents special difficulties in SPH. For example, it is known that intuitive SPH approximations for second derivatives are very sensitive to particle disorder [29]. In addition, an explicit time integration method can quickly lead to an unstable integration if large local gradients are encountered (due to noise). To circumvent these difficulties, we use an approximation for the Laplacian that incorporates only the first derivative of the kernel, as derived in §2.2.3. Using Equation (2.41), the discrete form of the SPH approximation for the Laplacian of a scalar function is written as

\[
\left< g^{\alpha\gamma} f_{\alpha\gamma}(x^\beta_i) \right> = 2 \sum_{j=1}^{N} \frac{m_j}{\rho_j} \frac{f(x^\beta_i) - f(x^\beta_j)}{x_{ij}} \frac{\partial W(x_{ij}, h_{ij})}{\partial x_{ij}}
\]

(2.47)
The discrete second-order derivative of a vector function can be written using Equation (2.39). Therefore, we obtain directly the more general threedimensional SPH second derivative

\[
\left\langle g^{\alpha\kappa} f_{\kappa\lambda}^{\alpha}(x_i^\beta) \right\rangle = \sum_{j=1}^{N} \frac{m_j}{\rho_j} \frac{f^{\sigma}(x_i^\beta) - f^{\sigma}(x_j^\beta)}{x_{ij}} \left( 5g_{\gamma\zeta} x_{ij}^{\gamma} x_{ij}^{\zeta} - \delta_{\zeta}^{\alpha} \right) \frac{\partial W(x_{ij}, h_{ij})}{\partial x_{ij}}
\]

(2.48)

Note that by contracting Equation (2.48) on indices \( \alpha \) and \( \zeta \), we obtain Equation (2.47).

### 2.2.5 Boundary Conditions in SPH

Recall that the SPH approximations derived in the previous sections were valid for all interior particles. However, for particles close to the domain boundary, the kernel will be truncated by the boundary, so that at the boundary \( W(x_{ij}, h) \neq 0 \). For a one-dimensional domain, this situation is illustrated below in Figure 2.7, where a physical boundary is located at \( x_{ij} = -0.5 \).

![Figure 2.7: Kernel Function Truncated by a Domain Boundary](image)

Therefore, the application of boundary conditions is problematic in the SPH technique, since SPH no longer produces the \( O(h^2) \) accuracy. In fact, it can be shown that for particles near a boundary, the kernel truncation produces
no better than first-order accuracy, which is unacceptable by numerical simulation standards. In order to find the value of a field property at a boundary, only interior particles will contribute to the SPH approximation since there are only particles on one side of the boundary. **Figure 2.8** illustrates a typical two-dimensional Cartesian \((x, y)\) SPH domain for \(N = 1600\) particles, where \(\Delta x\) represents the initial particle spacing in the \(x\)-direction.

**Figure 2.8: 2D SPH Cartesian \((x, y)\) Domain**

There are three general classes of particles that are identified in **Figure 2.8**:

1. far-boundary interior particles (denoted with \(\circ\) symbols),
2. near-boundary interior particles (denoted with \(\bullet\) symbols),
3. boundary particles (denoted with \(\ast\) symbols), and

*Far-boundary* particles are all interior particles that do not experience kernel truncation errors in the particle averaging summations. *Near-boundary* particles are those interior particles falling within the length \(2h\) of the domain boundary. It is for these particles that the standard SPH equations are technically incorrect, since the kernel is truncated at the domain boundary.
2. SMOOTHED PARTICLE HYDRODYNAMICS

*Boundary particles* are used to produce a repulsive boundary force to prevent particle penetration which is calculated using a Lennard-Jones type pairwise relation

\[
F_{ij}^a = \begin{cases} \frac{v_0^2 x_{ij}^5}{g_{ij} x_{ij}^5} \left[ \left( \frac{r_0}{x_{ij}} \right)^{12} - \left( \frac{r_0}{x_{ij}} \right)^6 \right] & \text{if } x_{ij} \leq r_0 \\ 0 & \text{if } x_{ij} > r_0 \end{cases}
\]  \quad (2.49)

where \( r_0 \) is the cutoff distance, which is selected as the initial particle spacing, and \( v_0 \) represents the largest particle velocity in the system. In Equation (2.49) particle \( i \) represents an interior particle that feels a repulsive force from boundary particle \( j \). The physical parameters and position of boundary particles do not evolve as the simulation progresses.

It has also been found from several SPH test problems that if a proper smoothing length and number of particles within the domain are chosen, neglecting the kernel truncation errors introduced does not necessarily effect the accuracy of the SPH solution. These test problems involved only static SPH particles. Therefore, throughout this thesis all errors resulting from kernel truncation at the boundary are neglected for thermal conduction problems. For problems involving fluid flow and particle motion, the errors resulting from kernel truncation cannot be ignored, and a boundary correction to the SPH approximations should be incorporated. One approach to boundary treatment is called *corrective smoothed particle hydrodynamics* (CSPH) [30].

### 2.2.6 Corrective SPH

CSPH involves a straightforward modification to the standard SPH approximations for hydrodynamic properties [30]. These corrected equations must include corrections to the standard SPH field approximation Equation (2.43), the SPH gradient Equation (2.44), the SPH divergence Equation (2.46), and the SPH Laplacian Equation (2.47). These corrective equations should sim-
plify to the standard SPH approximations for far-boundary interior particles.

**CSPH Field Approximation**

To derive the CSPH equation for the fundamental SPH approximation, we begin with the Taylor series expansion, Equation (2.4)

\[
f(x_i^\beta) = f(x_j^\beta) + x_{ij}^\gamma f_{\gamma}(x_j^\beta) + \frac{x_{ij}^\gamma x_{ij}^\lambda}{2} f_{\gamma\lambda}(x_j^\beta) \bigg|_{x_i^\beta=x_j^\beta} + \cdots
\]

where all covariant derivatives are with respect to particle \(i\) coordinates. By multiplying through with the kernel \(W(x_{ij}, h)\) and integrating over all space, we obtain

\[
\int \int \int_{\Omega} f(x_i^\beta) W(x_{ij}, h) d\Omega = f(x_j^\beta) \int \int \int_{\Omega} W(x_{ij}, h) d\Omega \\
+ f_{\gamma}(x_j^\beta) \int \int \int_{\Omega} x_{ij}^\gamma W(x_{ij}, h) d\Omega + \mathcal{O}(h^2)
\]

up to second-order. However, the last integral on the right hand side is identically zero for far-boundary interior particles, since the kernel is an even function and \(x_{ij}^\gamma\) is an odd function. For near-boundary particles, this integral will not, in general, be zero since the kernel is no longer an even function when it is truncated by the boundary. In either case, if the last integral on the right hand side is neglected, we obtain the result

\[
\int \int \int_{\Omega} f(x_i^\beta) W(x_{ij}, h) d\Omega = f(x_j^\beta) \int \int \int_{\Omega} W(x_{ij}, h) d\Omega
\]

or

\[
f(x_j^\beta) = \frac{\int \int \int_{\Omega} f(x_i^\beta) W(x_{ij}, h) d\Omega}{\int \int \int_{\Omega} W(x_{ij}, h) d\Omega} \tag{2.50}
\]

up to second-order accuracy for far-boundary particles, and first-order accuracy for near-boundary particles. Note that for all far-boundary interior particles, the denominator in Equation (2.50) is unity, making it equivalent to
Equation (2.6). By interchanging indices $i$ and $j$ and writing Equation (2.50) in discrete form, we obtain the fundamental CSPH approximation

$$
\left< f(x_i^\beta) \right> = \frac{\sum_{j=1}^{N} \frac{m_j}{\rho_j} f(x_j^\beta) W(x_{ij}, h)}{\sum_{j=1}^{N} \frac{m_j}{\rho_j} W(x_{ij}, h)}.
$$

(2.51)

**CSPH Gradient**

The CSPH gradient also begins with the Taylor expansion of Equation (2.4) multiplied through by the quantity $W_{\gamma}(x_{ij}, h)$ and integrated over all space producing

$$
\iint_{\Omega} f(x_i^\beta) W_{\lambda}(x_{ij}, h) d\Omega = \int \int_{\Omega} W_{\lambda}(x_{ij}, h) d\Omega
+f_{\gamma}(x_j^\beta) \int \int_{\Omega} x_{ij}^\gamma W_{\lambda}(x_{ij}, h) d\Omega + O(h^2).
$$

By isolating the gradient term on the right hand side, we obtain

$$
\iint_{\Omega} \left[ f(x_i^\beta) - f(x_j^\beta) \right] W_{\lambda}(x_{ij}, h) d\Omega = f_{\gamma}(x_j^\beta) \int \int_{\Omega} x_{ij}^\gamma W_{\lambda}(x_{ij}, h) d\Omega
$$

(2.52)

up to second-order accuracy. Unfortunately, due to the component summations for repeated indices, Equation (2.52) cannot be solved explicitly for the gradient term. In the discrete form, again by interchanging particles $i$ and $j$, we obtain the CSPH gradient

$$
\left< f_{\alpha}(x_i^\beta) \right> \sum_{j=1}^{N} \frac{m_j}{\rho_j} x_{ij}^\alpha W_{\lambda}(x_{ij}, h) = \sum_{j=1}^{N} \frac{m_j}{\rho_j} \left[ f(x_i^\beta) - f(x_j^\beta) \right] W_{\lambda}(x_{ij}, h).
$$

(2.53)

**CSPH Divergence**

The CSPH divergence is found by multiplying the vector Taylor expansion of Equation (2.4) by the term $W_{\lambda}(x_{ij}, h)$ and integrating over all space to
give
\[
\iint_{\Omega} f^\alpha(x^\beta_i) W_\lambda(x_{ij}, h) d\Omega = f^\alpha(x^\beta_j) \iint_{\Omega} W_\lambda(x_{ij}, h) d\Omega
\]
\[+ f^\alpha(x^\beta_j) \iint_{\Omega} x^\gamma_{ij} W_\lambda(x_{ij}, h) d\Omega + O(h^2)
\]
which produces
\[
\iint_{\Omega} \left[ f^\alpha(x^\beta_i) - f^\alpha(x^\beta_j) \right] W_\lambda(x_{ij}, h) d\Omega = f^\alpha(x^\beta_j) \iint_{\Omega} x^\gamma_{ij} W_\lambda(x_{ij}, h) d\Omega
\]
up to second-order accuracy. Upon interchanging particle indices \(i\) and \(j\), we obtain the discrete form of the CSPH vector gradient
\[
\left\langle f^\alpha_\gamma(x^\beta_i) \right\rangle \sum_{j=1}^N \frac{m_j}{\rho_j} x^\lambda_{ij} W_\lambda(x_{ij}, h) = \sum_{j=1}^N \frac{m_j}{\rho_j} \left[ f^\alpha(x^\beta_i) - f^\alpha(x^\beta_j) \right] W_\lambda(x_{ij}, h).
\]
Matrix inversion can be used to solve for this vector gradient \(\left\langle f^\alpha_\gamma(x^\beta_i) \right\rangle\) which can then be contracted to obtain the divergence.

**CSPH Laplacian**

The CSPH Laplacian can be determined by multiplying the Taylor series expansion by the second derivative of the kernel and integrating over all of space to obtain, up to second-order terms,
\[
\iint_{\Omega} \left[ f(x^\beta_i) - f(x^\beta_j) \right] W_{\mu\nu}(x_{ij}, h) d\Omega = f_\gamma(x^\beta_j) \iint_{\Omega} x^\lambda_{ij} W_{\mu\nu} d\Omega
\]
\[+ \frac{f_\gamma\lambda(x^\beta_j)}{2} \iint_{\Omega} x^\mu_{ij} x^\nu_{ij} W_{\mu\nu} d\Omega.
\]
If the first integral on the right hand side is taken to be zero, due to the odd function in the integrand, we are left with
\[
\iint_{\Omega} \left[ f(x^\beta_i) - f(x^\beta_j) \right] W_{\mu\nu}(x_{ij}, h) d\Omega = \frac{f_\gamma\lambda(x^\beta_j)}{2} \iint_{\Omega} x^\mu_{ij} x^\nu_{ij} W_{\mu\nu} d\Omega
\]
which, upon interchanging indices $i$ and $j$ and transforming to a discrete form becomes

$$
\left\langle f_{\gamma \lambda}(x_i^\alpha) \right\rangle \sum_{j=1}^{N} \frac{m_j}{\rho_j} x_j^\gamma x_j^\lambda W_{\mu \nu}(x_{ij}, h) = 2 \sum_{j=1}^{N} \frac{m_j}{\rho_j} [f(x_j^\gamma) - f(x_i^\gamma)] W_{\mu \nu}(x_{ij}, h).
$$

(2.56)

Equation (2.56) can be solved using matrix inversion to obtain the corrected Laplacian. As well, if the smoothing length is constant, Equation (2.18) can be used for the second derivative of the kernel. Equations (2.51), (2.53), (2.55), and (2.56) partially correct for the kernel truncation at a boundary. However, in discretized form, these corrections are second-order accurate only for far-boundary interior particles. Near the boundaries where the kernel is truncated, the correction is still only first-order accurate, once the transition from continuous to discrete equations is made. More accurate corrective procedures exist, and involve either kernel corrections or simultaneous solution of function derivatives for each particle $i$, such as in the modified smoothed particle hydrodynamics (MSPH) technique [31].

### 2.2.7 Neighbor Searching in SPH

There are several known searching algorithms that will find and store neighboring particles. Recall that neighbor particles are those particles $j$ that satisfy the condition

$$
x_i^\alpha - x_j^\alpha \leq 2h
$$

(2.57)

for a given particle $i$.

The most direct approach for finding particle neighbors is to cycle through all particles, and check whether Equation (2.57) is satisfied or not, storing the results. However, this algorithm searches all $N$ particles for each of the $N$ particles $i$. Therefore, this type of search procedure is of the order $N \times N$ in terms of computational searching effort required.
A more efficient approach is the "box-sorting" algorithm, which is known to be of order $N \log N$. The box-sorting procedure is illustrated in Figure 2.9. This algorithm divides the domain into an ordered number of boxes, with side dimensions equal to $2h$ in length. Each of the $N$ particles $i$ is then catalogued by which box it is located in. Since the box side dimensions are chosen to be $2h$, a neighbor $j$ of particle $i$ must be located in one of the adjacent boxes to the box containing particle $i$. Therefore, instead of searching all $N$ particles, one must search a much smaller group of particles to check the condition of Equation (2.57). Due to the much smaller computational expense, all simulations in this thesis used the box-sorting procedure. Note that for the quintic kernel of Equation (2.13), the box length dimensions must be $3h$, and the neighbor condition is dependent on the value $3h$ to be substituted into Equation (2.57) in place of the value $2h$. As well, the box-sorting algorithm was made adaptive, accommodating a variable smoothing length $h$.

![Figure 2.9: Box Approach for Neighbor Sorting](image)

### 2.2.8 Quasi-Incompressible SPH

In most numerical solution approaches, including finite volume, finite element, and SPH techniques, a major task in the solution algorithm is to determine the pressure field used in the linear momentum balance. If the
fluid to be simulated is compressible, the pressure can be computed directly from an appropriate equation of state. However, for incompressible fluids, the pressure must be assumed in order to compute the velocity field, which is subsequently used in the mass conservation equation; the correct velocity field must satisfy mass conservation. The assumed pressure is then corrected accordingly, until it produces the correct velocity field satisfying the conservation of mass. However, even theoretically incompressible flows are slightly compressible. Therefore, a popular approach is SPH is to use a quasi-incompressible assumption, which allows the pressure field to be computed from an artificially imposed state equation. For flows involving a free surface, the most popular SPH state equation for the dynamic pressure $p$ is

$$p = p_0 \left( \frac{\rho}{\rho_0} \right)^7 - 1$$

where $p_0$ and $\rho_0$ are reference values for the pressure and density, respectively. Notice that with this state equation, a small variation in density produces a relatively large change in pressure. For internal flows, an appropriate SPH state equation for a particle $i$ is

$$p_i = c_i^2 \rho_i.$$  \hfill (2.58)

The speed of sound $c_i$ for each particle must be chosen carefully to ensure that the fluid is very closely incompressible. For incompressible flows having a low Reynolds number,

$$Re \equiv \frac{v_0 L_0}{\nu}$$  \hfill (2.59)

it has been suggested [24] that the sound speed $c_i$ be chosen such that

$$c_i^2 \approx \max \left( \frac{v_0^2}{\Delta \rho / \rho_0}, \frac{\nu v_0}{L_0 \Delta \rho / \rho_0}, \frac{F_0 L_0}{\Delta \rho / \rho_0} \right)$$  \hfill (2.60)

where $v_0$, $L_0$, $\rho_0$, and $F_0$ are a reference velocity, length, density, and force per unit mass, respectively, and $\nu$ is the fluid kinematic viscosity. The quantity $\Delta \rho / \rho_0$ represents the relative incompressibility, which should be limited to
3%, so that $\Delta \rho / \rho_0 \approx 0.03$. The sound speed estimated from the results of Equation (2.60) will have a direct effect on the permissible time-step in a given simulation. More importantly, it provides direct computation of the dynamic pressure for each particle, once the particle density is known using Equation (2.58).

Recent reports [32] in the literature have indicated that the rules of thumb given by Equation (2.60) are not adequate for some problems. In certain cases, such as in the case of separated flows, Equation (2.60) should be as much as $10^3$ greater for the flow to be stable. That is, a higher $c^2$ should be selected to prevent the particle accumulation and depletion encountered in some problem geometries. Therefore, for such problems, the estimation

$$c_i^2 \approx 10^3 \times \max \left( \frac{v_0^2}{\Delta \rho / \rho_0}, \frac{\nu v_0}{L_0 \Delta \rho / \rho_0}, \frac{F_0 L_0}{\Delta \rho / \rho_0} \right)$$

(2.61)

is used as an alternative to Equation (2.60).

### 2.2.9 Time Integration in SPH

In order to increment the time-steps in the quasi-incompressible SPH algorithm, the “leapfrog” method [33] can be employed. This technique is an explicit time integration scheme, and is relatively simple to implement. However, since this is an explicit approach, the algorithm stability is controlled by the Courant-Friedrichs-Lewy (CFL) condition, where the recommended time-step is

$$\Delta t \leq C_{\text{CFL}} \frac{h_{ij,\text{min}}}{\max (c_i + \| v_i^2 \|)}$$

(2.62)

where $h_{ij,\text{min}}$ is the minimum smoothing length, $C_{\text{CFL}}$ is a constant satisfying $0 < C_{\text{CFL}} \leq 1$ (typically around 0.25) and $c_i$ is the speed of sound for particle $i$. The CFL limitation on the time-step ensures that the maximum rate of physical propagation is less than the maximum rate of numerical propagation. Therefore, the time-step must be proportional to the smallest spatial particle
resolution in SPH, which is the minimum smoothing length, $h_{ij,\text{min}}$. The CFL condition is a major drawback of using an explicit time-step approach, as $\Delta t$ is usually required to be extremely small for the quasi-incompressible SPH. As well, Monaghan and Cleary have reported \cite{34} that for pure conduction problems, the time-step in an explicit scheme must satisfy

$$\Delta t \leq 0.15 \frac{h_{ij,\text{min}}^2}{\alpha_D}$$

(2.63)

for a stable solution, where $\alpha_D = \kappa / \rho c_p$ is the thermal diffusivity.

Crystal growth processes such as LPD must be simulated for several minutes or hours. Because explicit time-stepping approaches impose a strict limitation on the maximum allowable time increment $\Delta t$, another time-stepping solution method should be used that remains stable for larger values of $\Delta t$. It is well known that implicit time-stepping approaches, such as the Crank-Nicolson approach, are unconditionally stable with respect to choice of $\Delta t$. However, if $\Delta t$ is too large, the simulation results will become inaccurate.

The Crank-Nicolson approach can be understood by considering a simple transient equation of the form

$$\frac{dT(x,t)}{dt} = cT(x,t)$$

where $c$ is an arbitrary constant, and $T = T(x,t)$ is some scalar field dependent on spatial and temporal coordinates in a convenient coordinate system. In order to solve this equation numerically, the above equation is discretized as

$$\frac{T^{m+1} - T^m}{\Delta t} = cT^m$$

where we have chosen to evaluate the right hand side of the equation at time-step $m$, making this approach an explicit approach. Throughout this thesis, superscripts will be used to indicate the time-step $m$. The above equation is explicit because it can be solved directly for $T^{m+1}$ as long as we know the value of $T^m$. Explicit schemes are straightforward to implement,
2. SMOOTHED PARTICLE HYDRODYNAMICS

but quickly become unstable if \( \Delta t \) is chosen to be too large. An implicit approach, however, evaluates the right hand side at the end of the time-step (at least partly) such that

\[
\frac{T^{m+1} - T^m}{\Delta t} = cT^{m+1}.
\]

Although implicit approaches described in the same way as the above equation are unconditionally stable, they are only first-order accurate in \( \Delta t \). The implicit Crank-Nicolson approach, however, is second-order accurate in the time-step. This approach takes an average value of \( T \) between times \( m \) and \( m+1 \) for the right hand side, to obtain

\[
\frac{T^{m+1} - T^m}{\Delta t} = c \left( \frac{T^{m+1}}{2} + \frac{T^m}{2} \right).
\]

Due to a higher accuracy, the Crank-Nicolson implicit approach will be used throughout the present simulations, unless noted otherwise.

Although the Crank-Nicolson approach is unconditionally stable, that does not mean that the time-step \( \Delta t \) is arbitrary. For accurate simulation results, it is known that the limiting time-step is twice that of the explicit time-stepping methods [35]. Therefore, for pure conduction problems, the limiting time-step for Crank-Nicolson time integration is

\[
\Delta t \leq 0.3 \frac{h^2_{x,\min}}{\alpha_D}
\]  

from Equation (2.63).

2.2.10 Projective SPH

Projective SPH (PSPH) is an alternative approach to quasi-incompressible SPH as a method for determination of the pressure field. It is based on the principle of Hodge decomposition, which states that any vector field \( \tilde{f}^\alpha \) can be broken into a divergence-free part \( f^\alpha \) plus the gradient of an appropriate
scalar potential $\phi$ so that

$$\tilde{f}^\alpha = f^\alpha + g^{\alpha\beta} \phi_{;\beta}$$  \hspace{1cm} (2.65)

with $f^\alpha_{;\alpha} = 0$. By differentiation and contraction using the $(\ )_{;\alpha}$ operator on the above equation, we obtain

$$\tilde{f}^\alpha_{;\alpha} = g^{\alpha\beta} \phi_{;\beta;\alpha}$$  \hspace{1cm} (2.66)

since by definition $f^\alpha_{;\alpha} = 0$ and also $g^{\alpha\beta} \equiv 0$. Equation (2.66) is the well-known Poisson equation. Furthermore, the two terms on the right hand side of Equation (2.65) are orthogonal vectors in an appropriate inner product space, satisfying the condition $f^\alpha \phi_{;\alpha} = 0$ as illustrated below in Figure 2.10.

![Figure 2.10: PSPH Concept](image)

The projection method begins by ignoring the pressure gradient in the momentum balance equation as reviewed in §3.1. An intermediate velocity field $\tilde{v}^\alpha$ is then computed which will not, in general, satisfy mass conservation. However, this incorrect velocity field can be projected onto a divergence-free space after solving a pressure Poisson equation, from which the divergence-free part of the velocity field $v^\alpha$ can be extracted. This divergence-free velocity part will, by definition, satisfy the mass conservation equation identically.
Specifically, the linear momentum balance in an orthogonal coordinate system in a Lagrangian reference frame is

\[
\frac{dv^\alpha}{dt} = -\frac{1}{\rho} g^{\alpha\beta} p_{;\beta} + \frac{\mu}{\rho} g^{\gamma\lambda} v^\alpha_{;\gamma\lambda} + F^\alpha_B
\]  

(2.67)

where \( v^\alpha \) is the fluid velocity, \( p \) is the pressure field, \( \rho \) is the fluid density, \( \mu \) is the dynamic viscosity, and \( F^\alpha_B \) is a body force per unit mass. Since the pressure term is troublesome, we instead solve the equation

\[
\frac{d\bar{v}^\alpha}{dt} = \frac{\mu}{\rho} g^{\gamma\lambda} v^\alpha_{;\gamma\lambda} + F^\alpha_B
\]  

(2.68)

for the intermediate acceleration field \( d\bar{v}^\alpha/dt \). This field is incorrect in that it has neglected the contribution of the pressure field, and will not, in general, be incompressible. However, we also know that from mass conservation an incompressible fluid is divergence-free with respect to velocity, so that

\[
v^\alpha_{;\alpha} = 0.
\]  

(2.69)

Therefore, if we substitute Equation (2.68) into Equation (2.67)

\[
\frac{dv^\alpha}{dt} = -\frac{1}{\rho} g^{\alpha\beta} p_{;\beta} + \frac{d\bar{v}^\alpha}{dt}
\]  

(2.70)

and take the divergence of the above equation, using Equation (2.69) we obtain

\[
0 = -\frac{1}{\rho} g^{\alpha\beta} p_{;\alpha\beta} + \frac{d\bar{v}^\alpha_{;\alpha}}{dt}.
\]

The above equation is then written as

\[
\frac{1}{\rho} g^{\alpha\beta} p_{;\alpha\beta} = \frac{d\bar{v}^\alpha_{;\alpha}}{dt}
\]  

(2.71)

and is known as the pressure Poisson equation. Once Equation (2.71) is used to solve for the pressure field, we can use Equation (2.70) to determine the correct, incompressible acceleration field \( dv^\alpha/dt \).

The general algorithm for PSPH is outlined below, using SPH approximation equations derived in §2.2.4.
1. Determine an estimate $\bar{x}_i^\alpha$ for all particle locations, given the previous particle positions $x_i^{\alpha,m}$ and the previous correct velocity field $v_i^{\alpha,m}$ as

$$\bar{x}_i^\alpha = x_i^{\alpha,m} + v_i^{\alpha,m} \Delta t$$

2. Find an intermediate velocity field $\tilde{v}_i^\alpha$ by omitting the pressure gradient term in the momentum balance so that at time-step $m+1$ we compute

$$\tilde{v}_i^\alpha = v_i^{\alpha,m} + \frac{2\mu_i \Delta t}{\rho_i} \sum_{j=1}^{N} \frac{m_j}{\rho_j} \frac{v_j^{\alpha,m} \partial W(x_{ij}, h)}{x_{ij}}$$

3. Determine the pressure field $p_i$ to enforce incompressibility by solving the pressure Poisson SPH equation [36] given $\tilde{v}_i^\alpha$

$$\sum_{j=1}^{N} \frac{2m_j}{\rho_j} \frac{p_{ij}}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} = \frac{1}{\Delta t} \sum_{j=1}^{N} \frac{m_j}{\rho_j} \left( \frac{\tilde{v}_i^\alpha + \tilde{v}_j^\alpha}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} \right)$$

4. Use the Hodge decomposition principle to solve for the true velocity field $v_i^{\alpha,m+1}$ once $p_i$ is known

$$v_i^{\alpha,m+1} = \tilde{v}_i^\alpha - \Delta t \sum_{j=1}^{N} m_j \left( \frac{p_i}{\rho_i} + \frac{p_j}{\rho_j} \right) \frac{x_{ij}^\alpha \partial W(x_{ij}, h)}{x_{ij}}$$

5. With the correct velocity field for time-step $m+1$, advect all particles to their new positions $x_i^{\alpha,m+1}$ using an average of the previous and current particle velocities

$$x_i^{\alpha,m+1} = x_i^{\alpha,m} + \left( \frac{v_i^{\alpha,m} + v_i^{\alpha,m+1}}{2} \right) \Delta t.$$ 

In step 2, incompressibility is enforced using the condition

$$\left( \frac{g^{\alpha\beta}}{\rho} p_{,\beta} \right)_{,\alpha} = \frac{\tilde{v}_i^\alpha}{\Delta t}$$

which comes from taking the divergence of the Navier-Stokes equations after neglecting the viscous terms. Note that $\tilde{v}_i^\alpha$ is not zero, as this velocity field is not divergence free. It is this nonzero portion that must be removed in step 3.
from the intermediate velocity field $\tilde{v}^\alpha$ to produce the correct incompressible velocity field $v^\alpha$. The above equation is the pressure Poisson equation

$$\frac{g^{\alpha\beta}}{\rho} p_{,\beta\alpha} = \frac{\tilde{v}^\alpha}{\Delta t}$$

which can be converted to SPH form using Equation (2.47) and the divergence form of Equation (2.45) for the Laplacian and divergence operations, respectively, producing the result

$$\frac{1}{\rho_i} \sum_{j=1}^{N} 2m_j \frac{p_{ij}}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} = \frac{\rho_i}{\Delta t} \sum_{j=1}^{N} m_j \left( \frac{\tilde{v}^\alpha}{\rho_i^2} + \frac{\tilde{v}^\alpha}{\rho_j^2} \right) g_{\alpha\beta} x_{ij}^\beta \frac{\partial W(x_{ij}, h)}{\partial x_{ij}}.$$

Upon noting that the density is constant ($\rho_i = \rho_j$) for an incompressible fluid, we obtain the simplified form of the PSPH pressure Poisson equation

$$\sum_{j=1}^{N} 2m_j \frac{p_{ij}}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} = \frac{1}{\Delta t} \sum_{j=1}^{N} m_j \left( \tilde{v}_i^\alpha + \tilde{v}_j^\alpha \right) g_{\alpha\beta} x_{ij}^\beta \frac{\partial W(x_{ij}, h)}{\partial x_{ij}}$$

as given in step 2 of the above algorithm.

During step 2, Neumann boundary conditions are implemented for the pressure field, whereby from Equation (2.44) for a boundary particle $i$

$$\left< \frac{\partial p_i(0, y, t)}{\partial x} \right> = \sum_{j=1}^{N} (p_j - p_i) \frac{\partial W(x_{ij}, h)}{\partial x_i} = 0$$

$$\left< \frac{\partial p_i(L, y, t)}{\partial x} \right> = \sum_{j=1}^{N} (p_j - p_i) \frac{\partial W(x_{ij}, h)}{\partial x_i} = 0$$

$$\left< \frac{\partial p_i(x, 0, t)}{\partial y} \right> = \sum_{j=1}^{N} (p_j - p_i) \frac{\partial W(x_{ij}, h)}{\partial y_i} = 0$$

and

$$\left< \frac{\partial p_i(x, H, t)}{\partial y} \right> = \sum_{j=1}^{N} (p_j - p_i) \frac{\partial W(x_{ij}, h)}{\partial y_i} = 0$$

for a rectangular Cartesian domain $0 \leq x \leq L$ and $0 \leq y \leq H$. In the above boundary equations, the particle masses and densities have been assumed constant, which will be the case for incompressible flow.
A main advantage of using PSPH is the elimination of the speed of sound parameter in the time-step conditions. Much larger time-steps can be used in PSPH, at the computational expense of having to solve the pressure Poisson equation at each time-step. The CFL condition for PSPH becomes

$$\Delta t \leq 0.25 \frac{h_{ij, \text{min}}}{\max \|v_i^p\|}$$

where we take the CFL constant as $C_{\text{CFL}} = 0.25$. It has been reported [36] that PSPH is also more accurate and efficient than the quasi-incompressible approach for flows involving moderate Reynolds numbers.

### 2.2.11 Advantages and Shortcomings of SPH for Crystal Growth Simulations

A comprehensive literature review on SPH has been carried out. It is found that SPH can be applied and generalized to a wide variety of problems [34, 37, 38, 39, 40] including free-surface, and moving interface problems. It is meshfree and Lagrangian in nature. There are no constraints imposed on either the geometry of the system, or how far the system may evolve from the initial state. It is therefore an "adaptive" technique, where changes in density and other flow properties are automatically accounted for without mesh refinement or other complicated techniques necessary when using other computational methods. The method can also be extended to handle problems in solid mechanics [41, 42]. With respect to crystal growth, SPH simulation has definite advantages, such as:

1. It has the built-in capability of tracking the movement of the crystal growth interface. With current finite volume and finite element techniques, this procedure is extremely cumbersome and computationally expensive as the mesh must be fully adaptable.
2. The particle nature of SPH is well-suited to modelling crystal growth from the vapour phase, whereas the finite volume and finite element methods would again be difficult to implement due to their spatial reference frames.

However, there are still a few deficiencies with SPH. For example, there are difficulties in modelling physical boundaries and enforcing boundary conditions. One solution to this shortcoming involves the use of Lennard-Jones type force that repels near-boundary particles, effectively simulating a wall. In addition, there are different ways to form SPH equations, and a consistent approach has not gained public consensus, as SPH has not yet been developed within a mathematically rigorous framework. For example, it has not yet been stated in the literature how to generally determine the minimum number of particles necessary to produce an accurate simulation for a given problem. As well, not all SPH approximations preserve balances of mass, energy, and momentum. However, careful choices of which SPH approximation to use can preserve these balances. In addition, highly irregular particle distributions can cause the numerical algorithms to break down [26].

In terms of crystal growth simulation specifically, the major shortcomings of SPH include:

1. Particle neighbor lists must be stored in the numerical procedure, which increases computer simulation times. The arrays used to store the indices for neighboring particles must be dynamic arrays, able to grow and shrink as necessary.

2. The time scales required to simulate crystal growth by most methods is several hours. It is unclear whether the SPH method will be able to compete in terms of total computational times with the finite volume and finite element approaches.
These shortcomings should not be insurmountable, and crystal growth simulation using SPH is therefore very promising.
3 SPH Simulation Test Problems

In order to verify the quality of the SPH coding, three thermal conduction test problems were solved and compared with known analytical solutions. As well, three simulations of fluid flow were conducted, and compared with analytical and finite difference solutions. All problems were set in a two-dimensional Cartesian domain $0 \leq x \leq L$ and $0 \leq y \leq H$, where $L = H = 10$ cm was selected, unless noted otherwise. This $(x, y)$ problem domain was modelled with SPH using $N = 1600$ (a grid of $40 \times 40$) particles shown below in Figure 3.1. Boundary particles are designated with the * symbol, whereas interior particles use the o symbol.

![Figure 3.1: SPH Test Problem Domain for N = 1600 Particles](image)
All coding was written in MATLAB, since this platform has several advantages, such as:

1. A user-friendly m-file editor which is relatively simple to use and test portions of code.

2. Several useful built-in functions for determining such things as the lengths of vectors, important for SPH neighbor determination and storage.

3. Easy-to-use graphical plotting functions and quick post-processing of data.

4. An excellent collection of coding support files.

These advantages made the preliminary coding of the SPH simulations in MATLAB an attractive option.

3.1 The Basic Equations of Thermomechanics of Continuous Media

The basic equations of continuum mechanics for a fluid in a Lagrangian frame of reference are the conservation of mass

\[
\frac{d\rho}{dt} = -\rho v^\alpha_{;\alpha} \tag{3.1}
\]

the balance of linear momentum

\[
\frac{d\sigma}{dt} = \frac{1}{\rho} \sigma^{\alpha\beta}_{;\beta} + F^\alpha_B \tag{3.2}
\]

and the balance of energy

\[
\rho \frac{dc}{dt} = -q^\alpha_{;\alpha} + \rho h + g_{\alpha\gamma} \sigma^{\alpha\beta}_{;\beta} v^\gamma_{;\beta} \tag{3.3}
\]
where \( \rho \) is the fluid density, \( \mathbf{u} \) is the fluid velocity, \( \sigma^{\alpha\beta} \) is the stress tensor, \( F^\beta_B \) is a body force per unit mass, \( \epsilon \) is the internal energy density per unit mass, \( q^\alpha \) is the heat flux vector, and \( h \) is the heat energy density per unit mass per unit time. However, these general equations alone do not constitute a closed system of equations. Therefore, additional constitutive equations must be constructed from logical deductions, based on the type of continuum under analysis. The constitutive equations used in this thesis are Fourier’s law

\[
q^\alpha = -\kappa g^{\alpha\beta} T_\beta
\]

where \( T \) is the absolute temperature, and the stress tensor for a Newtonian fluid

\[
\sigma^{\alpha\beta} = -\rho g^{\alpha\beta} + \mu \left( g^{\alpha\lambda} v_\lambda^\beta + g^{\beta\lambda} v_\lambda^\alpha \right)
\]

where \( \rho \) is the absolute pressure and \( \mu \) is the dynamic fluid viscosity. As well, since there is no heat generation in the materials considered, we have that \( h = 0 \). In this case, we also state that the change in internal energy density \( \epsilon \) is related to the material’s ability to store heat by raising its temperature so that for an incompressible fluid, where the constant-volume specific heat \( c_v \) equals the constant-pressure specific heat capacity \( (c_v = c_p) \)

\[
\frac{d\epsilon}{dt} \approx c_p \frac{dT}{dt}.
\]

It then follows that for an incompressible Newtonian fluid where \( \sigma^{\alpha\beta} = -g^{\alpha\beta} \rho \partial_\beta + \mu g^{\beta\lambda} v_\lambda^{\alpha} \) the two-dimensional conservation of mass and linear momentum balance equations in Cartesian \((x, y)\) coordinates become

\[
\begin{align*}
\frac{d\rho}{dt} &= -\rho \left( \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} \right) \quad (3.4) \\
\frac{dv_x}{dt} &= -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x} + \mu \left( \frac{\partial^2 v_x}{\partial x^2} + \frac{\partial^2 v_x}{\partial y^2} \right) + F^x_B \quad (3.5) \\
\frac{dv_y}{dt} &= -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial y} + \mu \left( \frac{\partial^2 v_y}{\partial x^2} + \frac{\partial^2 v_y}{\partial y^2} \right) + F^y_B. \quad (3.6)
\end{align*}
\]
where all vector components are Cartesian physical components. For two-dimensional thermal conduction in Cartesian coordinates, the governing equation for the evolution of the temperature field \( T = T(x, y, t) \) is obtained from the balance of energy, Equation (3.3), which takes the following form upon substitution of constitutive relations for a solid

\[
\frac{dT}{dt} = \alpha_D \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right)
\]  

(3.7)

where \( \alpha_D = \kappa/\rho c_p \) is the (constant) thermal diffusivity, \( c_p \) is the specific heat, \( \kappa \) is the thermal conductivity, and \( \rho \) is the material density.

### 3.2 2D Steady-State Heat Conduction with Isothermal Boundary Conditions

The first test problem was to determine the steady-state temperature profile \( T(x, y) \) subject to the boundary conditions

\[
T(x, 0) = T(0, y) = T(L, y) = T_1 \quad \text{and} \quad T(x, H) = T_2
\]

as illustrated in the schematic below in Figure 3.2.

![Figure 3.2: 2D Steady-State Thermal Problem Domain](image-url)
At steady-state, \( dT/dt = 0 \) and Equation (3.7) reduces to

\[
\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0.
\]

(3.8)

By separating variables, and applying the given boundary conditions, an analytical solution to the steady-state temperature field \( T(x, y) \) can be determined [43] and given in terms of the infinite series

\[
T(x, y) = T_1 + \frac{4(T_2 - T_1)}{\pi} \sum_{k=1,3,5,\ldots}^{\infty} \frac{\sin(k\pi x/L) \sinh(k\pi y/L)}{k \sinh(k\pi H/L)}
\]

(3.9)

The temperature profile is presented below in Figure 3.3, where \( T_1 = 100 \) and \( T_2 = 300 \) degrees Celsius were taken for the isothermal boundary temperatures, respectively.

![Figure 3.3: Temperature Profile from Equation (3.9)](image)

Equation (3.8) is approximated in SPH using Equation (2.47) to yield

\[
\langle g^{\alpha\beta} T_{(\alpha\beta)} \rangle = 2 \sum_{\substack{j=1 \atop j \neq i}}^{N} \frac{m_i}{\rho_j} \frac{(T_i - T_j)}{x_{ij}} \frac{\partial W(x_{ij}, h_{ij})}{\partial x_{ij}} = 0
\]

producing

\[
\sum_{\substack{j=1 \atop j \neq i}}^{N} \frac{T_i - T_j}{x_{ij}} \frac{\partial W(x_{ij}, h_{ij})}{\partial x_{ij}} = 0
\]

(3.10)
since the mass and density for each particle is the same (for equally spaced particles that do not move for a solid material). Boundary conditions were enforced by requiring that all boundary particles remain at temperature $T_1$ or $T_2$, respectively.

A comparison between the analytical, SPH, and finite difference solutions is illustrated below in Figure 3.4. The SPH isothermal contours are given by dotted lines, which cannot be distinguished from the solid lines representing the analytical solution contours. The dashed line represents the finite difference solution, which was computed using a mesh size of $40 \times 40$.

![Figure 3.4: SPH Isothermal Contours for $h = 0.25$ cm](image)

Note that the SPH solution is quite accurate for a smoothing length $h = 0.25$ cm. Over the domain, the mean percent difference between the analytical and SPH solutions was only 0.03%. During the simulations, it was found that proper selection of the smoothing length $h$ was critical in producing the correct results. Figure 3.5 shows the SPH simulation for $N = 1600$ and
$h = 0.5$ cm. Notice the reduction in solution accuracy as the smoothing length was increased. Recall that as $h \to 0$ the kernel goes to the Dirac-$\delta$ function which is exact. Therefore, a given solution accuracy depends not only on the number of particles used to discretize the domain, but also on the smoothing length. Also note that the points where the temperature is discontinuous (i.e. at the top corners) are subject to the Gibbs phenomenon.

![Figure 3.5: SPH Isothermal Contours for $h = 0.50$ cm](image)

### 3.3 2D Transient Heat Conduction with Isothermal Boundary Conditions

For the second test, a transient conduction problem was investigated. The problem was to model the temperature profile $T(x, y, t)$ of the cooling of the two-dimensional domain $0 \leq x \leq L$ and $0 \leq y \leq H$ at select times $t$. The
plate was subjected to the boundary conditions

\[ T(x, 0, t) = T(x, H, t) = T(0, y, t) = T(L, y, t) = T_1 \]

and the initial condition

\[ T(x, y, 0) = T_0 \]

where \( T_0 \) is a uniform initial temperature throughout the domain. This domain is illustrated in the schematic below in Figure 3.6.

![Figure 3.6: 2D Transient Thermal Problem Domain](image)

Equation (3.7) governs the heat transfer for the system. By separating variables, and applying the given boundary and initial conditions, an analytical solution to the temperature field \( T(x, y, t) \) can be determined and given in terms of the double infinite series

\[
T(x, y, t) = \frac{16T_0}{\pi^2} \sum_{k=1,3,\ldots}^{\infty} \sum_{l=1,3,\ldots}^{\infty} \exp\left[-\alpha_D \pi^2 \left( \frac{k^2}{L^2} + \frac{l^2}{H^2} \right) t \right] \sin\left( \frac{k\pi x}{L} \right) \sin\left( \frac{l\pi y}{H} \right)
\]

(3.11)

The evolution of the temperature profile according to Equation (3.11) is plotted below in Figure 3.7 for times \( t = 0.25, t = 1, t = 4, \) and \( t = 8 \) seconds, respectively. Note that in this simulation, parameters having values \( \alpha_D = 1 \text{ cm}^2/\text{s}, T_0 = 100 \) and \( T_1 = 0 \) degrees Celcius were taken.
Equation (3.7) was approximated with SPH using Equation (2.47) and an appropriate *explicit* discrete time-stepping procedure as

\[
\frac{T_i^{m+1} - T_i^m}{\Delta t} = 2\alpha_{Di} \sum_{j=1, j\neq i}^N \frac{m_j (T_i^m - T_j^m)}{\rho_j \partial x_{ij}} \partial W(x_{ij}, h_{ij})
\]

where \(m\) denotes the time-step. Rearranging the above equation, we have that

\[
T_i^{m+1} = T_i^m + 2\alpha_{Di}\Delta t \sum_{j=1, j\neq i}^N \frac{m_j (T_i^m - T_j^m)}{\rho_j \partial x_{ij}} \partial W(x_{ij}, h_{ij})
\]

(3.12)

which was the SPH equation used to solve for the temperature distribution at the discrete time-step \(m + 1\). In three-dimensional simulations, the quantity \(m_j/\rho_j\) represents the particle volume. However, for the two-dimensional problems considered, this quantity is interpreted as a characteristic particle area. Since the particles are equally spaced, the particle area is found by dividing the domain area \((LH)\) by the number of particles. For the chosen domain of \(N = 1600\) particles, we obtain \(m_j/\rho_j \cong 0.0625\) cm\(^2\), for \(L = H = 10\) cm. From Equation (2.63), the time-step for an explicit scheme is restricted
to $\Delta t \leq 0.009$ s to ensure an accurate solution, since we have chosen $h = 0.25$ cm and $\alpha_D = 1$ cm$^2$/s.

Figure 3.8 illustrates the isothermal plots at times $t = 0.25$, $t = 1$, $t = 4$, and $t = 8$ seconds obtained from Equations (3.11) and (3.12) where $T_0 = 100$ and $T_1 = 0$ degrees Celsius. Again, the SPH solution is represented by the dotted lines, and the analytical series solution by the solid lines. The SPH solution correlates very well with the known analytical solution for a smoothing length of $h = 0.25$ cm, $N = 1600$ particles, and $\Delta t = 0.009$ s.

![Isothermal Contours](image)

Figure 3.8: Isothermal Contours from Equations (3.11) and (3.12)

### 3.4 2D Transient Heat Conduction with Convective and Adiabatic Boundary Conditions

The third test problem involved the transient heat conduction of a two-dimensional Cartesian domain with more complicated boundary conditions.
Once again, the rectangular domain \( 0 \leq x \leq L \) and \( 0 \leq y \leq H \) was heated to an initial uniform temperature of \( T(x, y, 0) = T_0 \) with two insulated boundaries
\[
\frac{\partial T(0, y, t)}{\partial x} = 0 \quad \frac{\partial T(x, 0, t)}{\partial y} = 0,
\]
and two boundaries that dissipate heat by convection to a medium at zero temperature
\[
\frac{\partial T(L, y, t)}{\partial x} = -\frac{\mathcal{H}}{\kappa} T(L, y, t) \quad \frac{\partial T(x, H, t)}{\partial y} = -\frac{\mathcal{H}}{\kappa} T(x, H, t)
\]
where \( \mathcal{H} \) is the convective heat transfer coefficient. These boundary conditions are illustrated below in Figure 3.9.

![Figure 3.9: Second 2D Transient Thermal Problem Domain](image)

As with the previous test problems, the governing equation of heat transfer is given by Equation (3.7). Using the above mixed boundary conditions, the analytical solution to this problem can be shown to be [44]
\[
T(x, y, t) = 4T_0 \sum_{k=1}^{\infty} \sum_{l=1}^{\infty} \frac{\mathcal{H}^2 \exp \left[ -\alpha_D \left( \beta_k^2 + \gamma_l^2 \right) t \right] \cos(\beta_k x) \cos(\gamma_l y)}{[L(\beta_k^2 + \mathcal{H}^2) + \mathcal{H}] [H(\gamma_l^2 + \mathcal{H}^2) + \mathcal{H}] \cos(\beta_k L) \cos(\gamma_l H)}
\]
(3.13)

where \( \mathcal{H} = \mathcal{H}/\kappa \), and \( \beta_k \) and \( \gamma_l \) are the positive roots to the transcendental equations
\[
\beta_k \tan(\beta_k L) = \frac{\mathcal{H}}{\kappa} \quad \text{and} \quad \gamma_l \tan(\gamma_l H) = \frac{\mathcal{H}}{\kappa}.
\]
In the following simulation, we take $\alpha_D = 1 \text{ cm}^2/\text{s}$, $\kappa = 1 \text{ W/cm K}$ and $\mathcal{H} = 0.1 \text{ W/cm}^2\text{K}$ for the diffusivity, conductivity and convective heat transfer coefficient, respectively, as well as $T_0 = 100$ degrees Celcius.

The evolution of the temperature profile according to Equation (3.13) is plotted below in Figure 3.10 for times $t = 0.5$, $t = 1$, $t = 4$, and $t = 8$ seconds, respectively.

![Temperature Profiles from Equation (3.13)](image)

As before, the SPH governing heat transfer equation is expressed explicitly by Equation (3.12) as

$$T_i^{m+1} = T_i^m + 2\alpha_D\Delta t \sum_{j=1, j\neq i}^N \frac{m_j}{\rho_j} \frac{(T_i^m - T_j^m)}{x_{ij}} \frac{\partial W(x_{ij}, h_{ij})}{\partial x_{ij}}$$

where the error introduced by the kernel truncation of near-boundary interior particles has been neglected, as was done previously. The explicit scheme limited time-step was chosen as $\Delta t = 0.02 \text{ s}$. The SPH boundary conditions were written using Equation (2.44) for the boundary temperature gradients,
where a factor of 2 was added to balance the one-sided nature of the gradient at the boundary. Written explicitly for each boundary surface in Cartesian coordinates, we obtain the boundary conditions that must be satisfied for all time as

\[ \left\langle \frac{\partial T_i(0, y, t)}{\partial x} \right\rangle = \sum_{j = 1}^{N} \frac{m_j}{\rho_i} (T_j - T_i) \frac{\partial W(x_{ij}, h_{ij})}{\partial x_i} = 0 \]

\[ \left\langle \frac{\partial T_i(x, 0, t)}{\partial y} \right\rangle = \sum_{j = 1}^{N} \frac{m_j}{\rho_i} (T_j - T_i) \frac{\partial W(x_{ij}, h_{ij})}{\partial y_i} = 0 \]

and

\[ \left\langle \frac{\partial T_i(L, y, t)}{\partial x} \right\rangle = \sum_{j = 1}^{N} \frac{m_j}{\rho_i} (T_j - T_i) \frac{\partial W(x_{ij}, h_{ij})}{\partial x_i} = -\frac{\mathcal{H}}{\kappa} T_i(L, y, t) \]

\[ \left\langle \frac{\partial T_i(x, H, t)}{\partial y} \right\rangle = \sum_{j = 1}^{N} \frac{m_j}{\rho_i} (T_j - T_i) \frac{\partial W(x_{ij}, h_{ij})}{\partial y_i} = -\frac{\mathcal{H}}{\kappa} T_i(x, H, t) \]

where in this case \( i \) is the boundary particle of interest. The results of the analytical and SPH temperature contours at times \( t = 0.5, t = 1, t = 4, \) and \( t = 8 \) seconds are given below in Figure 3.11 for an explicit time discretization. In addition, a simulation was carried out that used \( N = 625 \) particles \((25 \times 25)\) along with \( h = 0.42 \) cm and \( m_j/\rho_j = 0.16 \) cm\(^2\). This simulation produced very similar results given by Figure 3.11, confirming that the number of particles can be decreased, as long as \( h \) is modified appropriately to produce correct results. The rule of thumb used is that \( h \) is set equal to the initial particle spacing, which in this case is \( h = L/(25 - 1) \cong 0.42 \) cm where \( L = 10 \) cm is the domain length, and 25 is the number of particles along the \( x \)-axis. This simulation to \( t = 8 \) seconds took just over 4 minutes of CPU time on a 64-bit AMD 3500+ processor. Other simulations for \( N = 100 \) and \( N = 9 \) were carried out, producing a complete breakdown in simulation accuracy. Therefore, the number of particles used must be sufficient to obtain the desired solution resolution, as expected. It was also found that a reduction in time-step did nothing to improve SPH solution accuracy. For this test
problem, it was found that the maximum allowable time-step before solution breakdown occurred was $\Delta t \approx 0.026$ s for the explicit Equation (3.12).

![Isothermal Contours at various times](image)

**Figure 3.11: Explicit Isothermal Contours**

To solve this test problem using an implicit technique, the Crank-Nicolson scheme was employed, described by

$$T_i^{m+1} = T_i^m + \alpha_{Di} \Delta t \sum_{j \neq i}^N \frac{m_j (T_i^{m+1} - T_j^{m+1})}{\rho_j x_{ij}} \frac{\partial W(x_{ij}, h_{ij})}{\partial x_{ij}}$$

$$+ \alpha_{Di} \Delta t \sum_{j = 1}^N \frac{m_j (T_i^m - T_j^m)}{\rho_j x_{ij}} \frac{\partial W(x_{ij}, h_{ij})}{\partial x_{ij}}$$

(3.14)

which is also known to be unconditionally stable. Instead of systematically solving for each temperature $T_i^{m+1}$ as was done in the explicit scheme, with the implicit approach we must now solve a system of equations. This system
of equations is rewritten and solved in the form

\[ T_{i}^{m+1} \left[ 1 - \sum_{j=1 \atop j \neq i}^{N} \frac{A_{ij}}{2} \right] + \sum_{j=1 \atop j \neq i}^{N} \frac{A_{ij}}{2} T_{j}^{m+1} = T_{i}^{m} \left[ 1 + \sum_{j=1 \atop j \neq i}^{N} \frac{A_{ij}}{2} \right] - \sum_{j=1 \atop j \neq i}^{N} \frac{A_{ij}}{2} T_{j}^{m} \]

where \( A_{ij} \) is a dimensionless parameter characterizing the relationship between particles \( i \) and \( j \) defined by

\[ A_{ij} \equiv 2\alpha_{ij} \Delta t \sum_{m} \frac{1}{\rho_{j}} \frac{1}{x_{ij}} \frac{\partial W(x_{ij}, h_{ij})}{\partial x_{ij}}. \]  

(3.15)

Here, at each time-step \( m + 1 \) a matrix must be inverted.

Using the implicit Crank-Nicolson approach, the SPH temperature contours were found, and are given along with the analytical solution in Figure 3.12, where a time-step of \( \Delta t = 0.5 \) s was used.

![Isothermal Contours](image)

**Figure 3.12:** Crank-Nicolson Isothermal Contours
3.5 1D Poiseuille Flow with No-Slip and Periodic Boundary Conditions

The first fluid flow test problem involved a simulation of one-dimensional, incompressible Poiseuille flow. Poiseuille flow consists of two infinitely long stationary plates, parallel to the $x$-axis, with a fluid in between them. These plates are separated by distance $H$, where $y = 0$ is the position of the bottom plate, and $y = H$ the position of the top plate. At time $t = 0$ a body force $F_B^x$ is applied to the fluid acting in the $x$-direction, producing a pressure gradient and causing fluid motion $v_x = v_x(y, t)$. The no-slip boundary conditions are employed at the plate/fluid interfaces $y = 0$ and $y = H$, so that

$$v_x(0, t) = v_x(H, t) = 0.$$  

The initial condition for all interior particles is taken as $v_x(y, 0) = 0$. Figure 3.13 illustrates the problem domain.

![Figure 3.13: 1D Poiseuille Flow Problem Domain](image)

The governing flow equation is the one-dimensional balance of linear momentum, Equation (3.5) where $\bar{p}$ is the total pressure, including both the static and dynamic pressures. However, since this is one-dimensional flow, where there is no variation of $v_x$ in the $x$-direction, we have Equation (3.5) reducing to

$$\frac{dv_x}{dt} = -\frac{1}{\rho} \frac{\partial p}{\partial x} - F_B^x + \frac{\mu}{\rho} \frac{\partial^2 v_x}{\partial y^2}$$  \hspace{1cm} (3.16)
where now a body force $F_B^x$ is used to model the hydrostatic part of the pressure gradient, accounting for the minus sign, and $p$ represents the dynamic pressure. The solution to the transient velocity profile for the incompressible Poiseuille flow governed by Equation (3.16) can be found as [24]

$$v_x(y, t) = \frac{4pF_B^x H^2}{\mu \pi^3} \sum_{k=0}^{\infty} \frac{1}{(2k + 1)^3} \sin \left( \frac{(2k + 1)\pi y}{H} \right) \exp \left[ -\frac{(2k + 1)^2 \mu \pi^2 t}{\rho H^2} \right] + \frac{\rho F_B^x}{2\mu} y(y - H)$$

(3.17)

which takes on a parabolic velocity profile as $t \to \infty$. The quasi-incompressible SPH version of Equation (3.16) using an implicit Crank-Nicolson time integration approach and Equations (2.44), (2.48), and (2.58) is

$$v_x^{m+1}(x_i^j) = v_x^m(x_i^j) + \frac{c_s^2 \Delta t}{2(\rho_i^{m+1})^2} \sum_{j=1}^{N} m_j x_{ij}^m \rho_{ij}^m \frac{1}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} - \Delta t F_B^x$$

$$+ \frac{\mu_i \Delta t}{\rho_i^{m+1}} \sum_{j=1}^{N} m_j \left[ v_x^{m+1}(x_i^j) - v_x^{m+1}(x_j^j) \right] \frac{1}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}}$$

$$+ \frac{\mu_i \Delta t}{\rho_i^{m}} \sum_{j=1}^{N} m_j \left[ v_x^m(x_i^j) - v_x^m(x_j^j) \right] \frac{1}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}}$$

(3.18)

where $x_{ij}^{m+1}$ and $x_{ij}^m$ denote the differences in the $x$-coordinate of particles $i$ and $j$ at time-steps $m + 1$ and $m$, respectively. Also note that the kernel derivative terms must be evaluated at an appropriate time-step, and the particle speed of sound is constant for all particles, and is selected according to the recommendations of Equation (2.60). As well, since we are using quasi-incompressible SPH, the density must be evolved according to the mass conservation Equation (3.4) which in SPH form becomes

$$\rho_i^{m+1} = \rho_i^m + \Delta t \sum_{j=1}^{N} m_j \left[ v_x^{m+1}(x_i^j) - v_x^{m+1}(x_j^j) \right] \frac{x_{ij}^{m+1}}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}}$$

$$+ \frac{\Delta t}{2} \sum_{j=1}^{N} m_j \left[ v_x^m(x_i^j) - v_x^m(x_j^j) \right] \frac{x_{ij}^m}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}}$$

(3.19)
again using implicit Crank-Nicolson time integration. In the above equations, 
m_j is the constant mass of each SPH particle, found from dividing the total
mass by the number of particles according to \( m_j = \rho H L / N \) where \( H \) and \( L \)
are the height and length defining the SPH domain area.

**Figure 3.14** illustrates the SPH and analytical transient solutions to the
Poiseuille flow problem at times \( t = 0.01, t = 0.05, t = 0.1, \) and \( t = 0.3 \)
seconds, using the quintic spline kernel defined by Equation (2.13). A \( 40 \times 40 \)
array of particles were evenly distributed in the domain, and the parameters
were taken as \( H = 10^{-3} \) m, \( F_B = 10^{-4} \) N, \( \rho = 1000 \) kg/m\(^3\), and \( \mu = 10^{-3} \)
kg/m s. The particle speed of sound was estimated as \( c_s = \sqrt{FH/\Delta \rho/\rho} = 1.8 \times 10^{-3} \) m/s in accordance with Equation (2.60). The smoothing length
was set equal to 1.1 times the initial particle spacing, and the time step was
chosen as \( \Delta t = 0.0001 \) s, reaching the steady-state condition at \( t \approx 0.3 \) s.

![Graph showing SPH Poiseuille Flow Velocity Profile for N = 1600](image)

**Figure 3.14:** SPH Poiseuille Flow Velocity Profile for \( N = 1600 \)

In order to evolve the smoothing length, Equation (2.21) was used by specifying
the optimal number of neighbors as \( N_{opt} = 25 \). Inlet and outlet particles
were defined such that as a particle crossed a computational boundary defined by a convenient $x$-coordinate, that particle was mapped from the outlet to the inlet position, at the same $y$-coordinate, having the same velocity. This approach effectively simulates infinitely long plates, and is called the periodic boundary condition. As well, the CSPH equations were employed for near-boundary particles, as discussed in §2.2.6. Solid lines represent the analytical solution of Equation (3.17), and circles represent the SPH solution of Equations (3.18) and (3.19).

3.6 1D Couette Flow with No-Slip and Periodic Boundary Conditions

The second fluid flow test problem involved a simulation of one-dimensional incompressible Couette flow. Couette flow consists of two infinitely long plates, parallel to the $x$-axis, with a fluid in between them. These plates are separated by distance $H$, where $y = 0$ is the position of the bottom plate, and $y = H$ the position of the top plate. We assume that there is no spatial variation in the velocity in the $x$-direction, so that $v_x = v_x(y, t)$. Initially both plates are stationary, but at time $t = 0$ the top plate is set into motion with velocity

$$v_0 = 2.5 \times 10^{-5} \text{ m/s}$$

in the $x$-direction. The no-slip boundary conditions are employed at the plate/fluid interfaces $y = 0$ and $y = H$, so that

$$v_x(0, t) = 0 \quad \text{and} \quad v_x(H, t) = 2.5 \times 10^{-5} \text{ m/s}.$$ 

The initial condition for all interior particles is taken as $v_x(y, 0) = 0$. Figure 3.15 illustrates the problem domain.
3. SPH SIMULATION TEST PROBLEMS

\[ v_x(H, t) = 2.5 \times 10^5 \]

\[ v_x(y, 0) = 0 \]

\[ v_x(0, t) = 0 \]

Figure 3.15: 1D Couette Flow Problem Domain

The governing flow equation is the balance of linear momentum, Equation (3.5). As before, we obtain the one-dimensional \( x \)-direction momentum equation as

\[
\frac{dv_x}{dt} = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \frac{\mu}{\rho} \frac{\partial^2 v_x}{\partial y^2}.
\]

(3.20)

In the following simulation, the initial values \( \mu = 10^{-3} \) kg/m s and \( \rho = 1000 \) kg/m\(^3\) are used, producing a Reynolds number of \( \text{Re} = 2.5 \times 10^{-2} \). From Equations (2.62) and (2.60), the time-step was selected as \( \Delta t \leq 0.0001 \) s for physically accurate results. The analytical transient solution for Couette flow can be shown to be [24]

\[
v_x(y, t) = \frac{v_0}{H} y + \frac{2v_0}{\pi} \sum_{k=1}^{\infty} \frac{(-1)^k}{k} \sin \left( \frac{k\pi y}{H} \right) \exp \left( -\frac{\mu k^2 \pi^2 t}{\rho H^2} \right)
\]

(3.21)

where \( v_x(y, t) \) takes on a linear profile as \( t \to \infty \). The SPH version of Equation (3.20) using a Crank-Nicolson time integration is

\[
v_x^{m+1}(x_i) = v_x^m(x_i) + \frac{c_i^2 \Delta t}{2(\rho_i^{m+1})^2} \sum_{j=1}^{N} m_j x_j^m \rho_j^m \frac{1}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}}
\]

\[
+ \frac{c_i^2 \Delta t}{2(\rho_i^{m+1})^2} \sum_{j=1}^{N} m_j x_j^m \rho_j^m \frac{1}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}}
\]

\[
+ \frac{\mu_i \Delta t}{\rho_i^{m+1}} \sum_{j=1}^{N} \frac{m_j}{\rho_j^{m+1}} \left[ v_x^{m+1}(x_i) - v_x^{m+1}(x_j) \right] \frac{1}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}}
\]

\[
+ \frac{\mu_i \Delta t}{\rho_i^{m+1}} \sum_{j=1}^{N} \frac{m_j}{\rho_j^{m+1}} \left[ v_x^m(x_i) - v_x^m(x_j) \right] \frac{1}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}}.
\]

(3.22)
As in the previous test problem, the density is evolved according to Equation (3.19), and near-boundary particles use the CSPH equations of §2.2.6.

**Figure 3.16** illustrates the SPH and analytical transient solutions to the Couette flow problem at times $t = 0.01$, $t = 0.08$, and $t = 0.4$ seconds, using the quintic spline kernel defined by Equation (2.13). An evenly distributed arrangement of $20 \times 20$ particles was used, and the smoothing length chosen to be equal to 1.1 times the initial particle spacing.

![Figure 3.16: SPH Couette Flow Velocity Profile for $N = 400$](image)

It was found that the SPH simulation for Couette flow given in **Figure 3.16** compared well with the analytical solution, even though a relatively low particle resolution was used.
3.7 2D Flow Field Over a Cavity with No-Slip Boundary Conditions

The final test problem involved a two-dimensional simulation of a fluid flowing over a cavity. The no-slip boundary conditions are employed within the cavity at \( x = 0, \ y = 0, \) and \( x = L, \) producing

\[
v_x(0, y, t) = v_x(x, 0, t) = v_x(L, y, t) = 0
\]

and

\[
v_y(0, y, t) = v_y(x, 0, t) = v_y(L, y, t) = 0.
\]

The top of the cavity, defined by \( y = H \) has the boundary condition

\[
v_x(x, H, t) \equiv v_0 = 10^{-3} \text{ m/s} \quad \text{and} \quad v_y(x, H, t) = 0.
\]

The initial condition for all interior particles is taken as \( v_x(x, y, 0) = v_y(x, y, 0) = 0. \) Over time, the flow will reach a steady-state, and form a recirculation pattern within the domain. Figure 3.17 illustrates the problem domain.

![Figure 3.17: 2D Flow Cavity Problem Domain](image)

For ease of coding, it was decided to use explicit time integration for the 2D cavity problem. The governing flow equations are again the continuity
3. SPH SIMULATION TEST PROBLEMS

Equation (3.4), as well as the balance of linear momentum, Equations (3.5) and (3.6). The SPH versions of these equations using explicit time integration are

$$
\rho_{i}^{m+1} = \rho_{i}^{m} + \Delta t \sum_{j=1}^{N} m_{j} \left[ v_{x}^{m}(x_{i}^{\beta}) - v_{x}^{m}(x_{j}^{\beta}) \right] \frac{x_{ij}^{m}}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} \\
+ \Delta t \sum_{j=1}^{N} m_{j} \left[ v_{y}^{m}(x_{i}^{\beta}) - v_{y}^{m}(x_{j}^{\beta}) \right] \frac{y_{ij}^{m}}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} \quad (3.23)
$$

$$
v_{x}^{m+1}(x_{i}^{\beta}) = v_{x}^{m}(x_{i}^{\beta}) + \frac{c_{i}^{2} \Delta t}{(\rho_{i}^{m})^{2}} \sum_{j=1}^{N} m_{j} x_{ij}^{m} \rho_{ij}^{m} \frac{1}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} \\
+ \frac{2 \mu_{i} \Delta t}{\rho_{i}^{m}} \sum_{j=1}^{N} m_{j} \left[ v_{x}^{m}(x_{i}^{\beta}) - v_{x}^{m}(x_{j}^{\beta}) \right] \frac{1}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} \quad (3.24)
$$

$$
v_{y}^{m+1}(x_{i}^{\beta}) = v_{y}^{m}(x_{i}^{\beta}) + \frac{c_{i}^{2} \Delta t}{(\rho_{i}^{m})^{2}} \sum_{j=1}^{N} m_{j} y_{ij}^{m} \rho_{ij}^{m} \frac{1}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} \\
+ \frac{2 \mu_{i} \Delta t}{\rho_{i}^{m}} \sum_{j=1}^{N} m_{j} \left[ v_{y}^{m}(x_{i}^{\beta}) - v_{y}^{m}(x_{j}^{\beta}) \right] \frac{1}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} \quad (3.25)
$$

The parameters $H = L = 10^{-3}$ m, $\rho = 1000$ kg/m$^3$, and $\mu = 10^{-3}$ kg/m s were selected for this simulation, producing a Reynolds number Re = 1. Recall that the previous flow simulations both had Re = $2.5 \times 10^{-2}$. The time-step was chosen as $\Delta t = 5 \times 10^{-5}$ s in accordance with the CFL condition for explicit time integration, and steady-state was reached at approximately $t = 0.15$ s. The sound speed was initially selected according to the recommendations of Equation (2.60), yielding $c_{i} = 5.77 \times 10^{-3}$ m/s. However, it was found that this choice of sound speed led to a problem with the computed density field using Equation (3.23) due to the Lagrangian nature of SPH. These simulations did not adequately produce the known solution for the shear cavity problem. As particles were transported, numerical anomalies were produced at the top corners of the domain, where particles tended to accumulate on the right hand side, and disperse on the left hand side. Through
trial and error, by selecting a speed of sound of at least \( c_i = 0.18 \text{ m/s} \), the density anomalies at the top corners were eliminated. This selection was justified with the use of Equation (2.61) to select the sound speed. As well, particle field quantities were periodically remeshed in order to re-order the particles in the domain, which then produced the correct results. This particle remeshing occurred once every 0.05 seconds of simulation time. Therefore, the remeshing procedure was carried out a total of three times during a simulation to steady-state, where \( t = 0.15 \text{ s} \). **Figure 3.18** illustrates the velocity vectors at steady-state within the domain, using the quintic spline kernel defined by Equation (2.13) and an array of \( 21 \times 21 \) particles (\( N = 441 \)). The recirculation pattern within the cavity domain is clearly visible.

![Figure 3.18: SPH Flow Cavity Velocity Plot for \( N = 441 \) Particles](image)

**Figure 3.18** shows a comparison of the variation of the vertical velocity component \( v_y \) at the horizontal centerline of the domain for the SPH and finite difference (FD) solutions. The finite difference solution was obtained using a forward-time, centered-space differencing, along with successive over-relaxation. The finite difference grid was selected as a \( 51 \times 51 \) rectangular mesh to ensure the accuracy of the finite difference solution. The velocity has been normalized with the velocity at the top of the cavity, \( v_0 \).
Figure 3.19: SPH Flow Cavity $v_y$ at $y = H/2$ for $N = 441$ Particles

Figure 3.20 shows a comparison of the variation of the horizontal velocity component $v_x$ at the vertical centerline of the domain for the SPH and finite difference solutions. Again, the velocity has been normalized with the velocity at the top of the cavity, $v_0$.

Figure 3.20: SPH Flow Cavity $v_x$ at $x = L/2$ for $N = 441$ Particles

It can be seen that the SPH results do not compare well with the finite
difference solutions. However, in the Couette flow simulations, the relatively low particle resolution of $20 \times 20$ produced the correct results. The main difference between the Couette flow problem and the 2D cavity problem is the Reynolds number. Even the low Reynolds number of $\text{Re} = 1$ for the cavity problem requires an increase in the number of SPH particles to produce an accurate simulation. Therefore, the cavity simulation was again carried out, this time using a particle resolution of $61 \times 61$ ($N = 3721$) particles.

**Figure 3.21** illustrates the velocity vectors at steady-state for $61 \times 61$ particles. Again, the recirculation pattern within the cavity domain is clearly visible.

![SPH Flow Cavity Velocity Plot](image)

**Figure 3.21**: SPH Flow Cavity Velocity Plot for $N = 3721$ Particles

**Figure 3.22** compares the variation of the normalized vertical velocity component $v_y$ at the horizontal centerline of the domain for the SPH and finite difference solutions.
Figure 3.22: SPH Flow Cavity $v_y$ at $y = H/2$ for $N = 3721$ Particles

Figure 3.23 illustrates the comparison of the variation of the normalized horizontal velocity component $v_x$ at the vertical centerline of the domain for the SPH and finite difference solutions.

Figure 3.23: SPH Flow Cavity $v_x$ at $x = L/2$ for $N = 3721$ Particles

It can be seen that the SPH solutions agree fairly well with the finite difference results, providing that the number of particles used is sufficiently high,
and an appropriate speed of sound is selected. These facts are a major shortcoming of the SPH approach, since there is no rigorous formulation of how to select the speed of sound parameter, only vague rules of thumb. As well, when solving problems in general, the speed of sound must be selected before simulating can begin, making this method somewhat less robust than other finite volume approaches. The major advantage of the quasi-incompressible SPH approach is the elimination of determining the pressure in incompressible flow problems. This advantage greatly simplifies the SPH solution algorithm, as compared with other approaches, such as using PSPH, as discussed in §2.2.10.

For comparison with quasi-incompressible SPH, the flow cavity problem was also solved using PSPH, for particle resolutions of $21 \times 21$ and $61 \times 61$ particles ($N = 441$ and $N = 3721$). All parameters were kept constant, again producing a Reynolds number $Re = 1$. The boundary force of Equation (2.49) was again applied to interior particles close to the boundary, to prevent boundary penetration, and steady-state was reached at approximately $t = 0.15$ s.

Using an explicit time integration, the PSPH versions of the linear momentum equations for time-step $m$ are

$$
\bar{v}_x(x_i^\beta) = v_x^m(x_i^\beta) + \frac{2\mu_i \Delta t}{\rho_i \sum_{j=1}^{N} m_j} \left[v_x^m(x_i^\beta) - v_x^m(x_j^\beta)\right] \frac{1}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} \tag{3.26}
$$

$$
\bar{v}_y(x_i^\beta) = v_y^m(x_i^\beta) + \frac{2\mu_i \Delta t}{\rho_i \sum_{j=1}^{N} m_j} \left[v_y^m(x_i^\beta) - v_y^m(x_j^\beta)\right] \frac{1}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} \tag{3.27}
$$

where the barred velocities are intermediate velocities, since the pressure gradient is initially neglected. Once the intermediate velocities are computed, the pressure field can be calculated by solving the pressure Poisson equation

$$
\sum_{j=1}^{N} \frac{4m_j}{\rho_j (\rho_i + \rho_j)} \frac{p_{ij}}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} = \frac{\rho_i}{\Delta t} \sum_{j=1}^{N} m_j \left[\frac{\bar{v}_x(x_i^\beta)}{\rho_i^2} + \frac{\bar{v}_x(x_j^\beta)}{\rho_j^2}\right] \frac{x_{ij}^m}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} + \rho_i \sum_{j=1}^{N} m_j \left[\frac{\bar{v}_y(x_i^\beta)}{\rho_i^2} + \frac{\bar{v}_y(x_j^\beta)}{\rho_j^2}\right] \frac{y_{ij}^m}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}}. \tag{3.28}
$$
After the pressure field is determined, the actual velocities are found using the principle of Hodge decomposition, so that

\[
v_{x}^{m+1}(x_i^\beta) = \bar{v}_{x}(x_i^\beta) - \Delta t \sum_{j=1}^{N} m_j \left( \frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \frac{x_{ij}^m \partial W(x_{ij}, h)}{x_{ij}} \tag{3.29}
\]

\[
v_{y}^{m+1}(x_i^\beta) = \bar{v}_{y}(x_i^\beta) - \Delta t \sum_{j=1}^{N} m_j \left( \frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \frac{y_{ij}^m \partial W(x_{ij}, h)}{x_{ij}} \tag{3.30}
\]

The PSPH time-step was chosen as \( \Delta t = 1 \times 10^{-4} \) s, and the particle accumulation problem was avoided by a periodic redistribution of particles at \( t = 0.05 \), \( t = 0.1 \), and \( t = 0.15 \) s. Although the boundary force was employed in the PSPH simulations, it was observed that no special boundary treatment was required to obtain the correct results. The vector velocity plot for the flow is given below in **Figure 3.24**, for \( N = 3721 \) particles.

![Figure 3.24: PSPH Flow Cavity Velocity Plot for \( N = 3721 \) Particles](image)
Figure 3.25 gives the normalized vertical velocity component as a function of position along a horizontal centerline of the domain, computed with PSPH and the finite difference method, for both $N = 441$ and $N = 3721$ particles.

![Diagram](image1)

**Figure 3.25: PSPH Flow Cavity $v_y$ at $y = H/2$**

Figure 3.26 gives the normalized horizontal velocity components as a function of position along a vertical centerline of the domain for $N = 441$ and $N = 3721$ particles, respectively. The PSPH results are again compared with the finite difference solution, indicated by the solid lines.

![Diagram](image2)

**Figure 3.26: PSPH Flow Cavity $v_x$ at $x = L/2$**

As with the quasi-incompressible SPH results, it is observed from Figures 3.25 and 3.26 that both normalized velocity components are slightly underpredicted using the PSPH method. However, the minor discrepancy between
PSPH and finite difference solutions is acceptable, since the PSPH results approach the finite difference results as the number of SPH particles increases. Unfortunately, as the particle resolution increases, the computational time increases drastically due to a combination of the required reduction in time-step, and the increased number of particle neighbors to find. A study was carried out in order to determine the relationship between simulation times versus the number of particles \( N \), and the results are illustrated below in Figure 3.27. Note that these times include the determination of the finite difference results.

![Figure 3.27: Cavity Simulation Times Versus Number of Particles \( N \)](image-url)

The purpose of the above time study was to get a feel for the correlation between CPU time and \( N \), and to point out the dramatic increase in simulation times with only relatively small increases in the number of SPH particles. Due to the larger allowable time-steps and the elimination of the guess-work associated with the choice of the speed of sound, it was decided to use PSPH to simulate LPD crystal growth, instead of the quasi-incompressible approach.
4 Governing Equations for Crystal Growth by LPD

For the present application, the LPD crystal growth system consists of three sections: a polycrystalline Si source material, a binary Si-Ge liquid solution mixture, and a solid, single-crystal Ge substrate. Throughout the following simulations, we take the silicon concentration to be $x = 0.05$. Figure 4.1 illustrates the LPD computational domain [1]. Unit surface normal and tangential vectors $n_\alpha$ and $t_\alpha$ have been included in Figure 4.1 to facilitate the following discussion.

![Figure 4.1: Computational Domain of the LPD Crystal Growth System](image)

89
4. GOVERNING EQUATIONS FOR CRYSTAL GROWTH BY LPD

The materials are contained in a quartz crucible, and are exposed to an axial thermal gradient, $T(z)$. Initially, all materials are solid. As the system is heated, the germanium melts, whereas the silicon remains solid due to its higher melting temperature. Some silicon source material dissolves into the germanium solution, according to the binary thermal phase equilibrium, producing the Si-Ge liquid phase. Silicon species is then transported via various mechanisms to the germanium substrate. When the mixture in the vicinity of the substrate becomes supersaturated, solidification occurs, producing crystal growth. The silicon in the solution is then replenished by the source material, via diffusion.

The LPD growth system is most conveniently modelled in a cylindrical $(r, \theta, z)$ coordinate system having

$$g_{\alpha\beta} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & r^2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and $g = r^2$. The radius of the growth cell (the inner radius of the quartz crucible) is denoted by $R$, and due to symmetry considerations, the system can be modelled as a two-dimensional problem. Therefore, all dependence on the angular coordinate $\theta$ drops out of the following equations ($\partial/\partial \theta = 0$ and $v_\theta = 0$). The liquid phase equations represent the solution mixture of silicon and germanium. This solution mixture is assumed to be a viscous, heat-conducting fluid. The solid phase represents the substrate, polycrystalline source, and the growing crystal, as well as the quartz growth crucible. We now present the governing equations, found from Equations (3.1), (3.2), and (3.3) for the liquid and solid phases. When a distinction is required, liquid phase variables are denoted having subscript $L$, whereas solid phase variables have subscript $S$. Rigorous derivation of these equations, including development of the constitutive relations, can be found in [1].
4. Governing Equations for the Liquid Phase

The liquid phase is considered as an incompressible mixture. In cylindrical coordinates, the governing equations for the liquid solution are written in a Lagrangian frame with physical components as

**Conservation of Mass:**

\[
\frac{d\rho_L}{dt} = -\rho_L \left( \frac{\partial v_r}{\partial r} + \frac{v_r}{r} + \frac{\partial v_z}{\partial z} \right)
\]  

(4.2)

**Balance of Linear Momentum:**

\[
\begin{align*}
\frac{dv_r}{dt} &= -\frac{1}{\rho_L} \frac{\partial \bar{p}}{\partial r} + \nu \left( \frac{\partial^2 v_r}{\partial r^2} + \frac{1}{r} \frac{\partial v_r}{\partial r} + \frac{\partial^2 v_r}{\partial z^2} - \frac{v_r}{r^2} \right) \\
\frac{dv_z}{dt} &= -\frac{1}{\rho_L} \frac{\partial \bar{p}}{\partial z} + \nu \left( \frac{\partial^2 v_z}{\partial r^2} + \frac{1}{r} \frac{\partial v_z}{\partial r} + \frac{\partial^2 v_z}{\partial z^2} \right) \\
&\quad + a_g \beta_T (T - T_0) + a_g \beta_C (C - C_0)
\end{align*}
\]  

(4.3)

(4.4)

where \( \beta_T \) and \( \beta_C \) are thermal and solutal expansion coefficients, \( \bar{p} = p + \rho_L H_L a_g \) is the total pressure, \( a_g \) is the gravitational acceleration, \( H_L \) is the height of the solution, \( \nu = \mu/\rho_L \) is the kinematic viscosity, and \( T_0 \) and \( C_0 \) are the reference temperature and mass fraction of silicon, respectively.

**Balance of Energy:**

\[
\frac{dT}{dt} = \alpha_{DL} \left( \frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \frac{\partial T}{\partial r} + \frac{\partial^2 T}{\partial z^2} \right)
\]  

(4.5)

where \( T \) is the absolute temperature, and \( \alpha_{DL} \equiv \kappa_L/\rho_L c_p L \) is the thermal diffusivity of the liquid solution. In addition, since two species are present in the liquid solution, mass diffusion will occur, and is accounted for using the mass transport equation.

**Mass Transport:**

\[
\frac{dC}{dt} = D_L \left( \frac{\partial^2 C}{\partial r^2} + \frac{1}{r} \frac{\partial C}{\partial r} + \frac{\partial^2 C}{\partial z^2} \right)
\]  

(4.6)

where \( C \) is the mass fraction of silicon, and \( D_L \) is the diffusion coefficient of silicon in the liquid solution.
The liquid phase Equations (4.2)-(4.6) constitute a set of five equations for five unknowns \((v_r, v_z, \bar{p}, T, \text{ and } C)\). Therefore, this set of equations is mathematically closed.

4.2 Governing Equations for the Solid Phase

For the solid phase, the only non-trivial governing equation is the balance of energy,

**Balance of Energy:**

\[
\frac{dT}{dt} = \alpha_{DS} \left( \frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \frac{\partial T}{\partial r} + \frac{\partial^2 T}{\partial z^2} \right)
\]

(4.7)

where \(\alpha_{DS} \equiv \kappa_S/\rho_S c_p S\) is the thermal diffusivity of the solid phase.

4.3 PSPH Governing Equations for the Liquid and Solid Phases

In order to model LPD crystal growth, the governing Equations (4.2)-(4.7) must be converted to PSPH form. To this end, we use the results of §2.2.4, Equations (4.2)-(4.8) as follows.

**PSPH Conservation of Mass:**

The mass conservation relation should satisfy \(d\rho_L/dt = 0\) for an incompressible fluid. Because we are using PSPH, the mass conservation equation is not solved explicitly; conservation of mass is implicitly required, and is the foundation of projection methods.

**PSPH Balance of Linear Momentum:**

The PSPH linear momentum balance equations are found from Equation (4.48) to give the Laplacian of the velocity field. For the intermediate velocity field \(\bar{v}_r(x_i^3)\), Equation (4.3) then reads, upon writing in terms of physical compo-
nants and omitting the pressure gradient term for PSPH,
\[
\frac{d\bar{v}_r(x_i^\beta)}{dt} = 2\nu \sum_{j=1}^{N} \frac{m_j}{\rho_L} \left[ v_r(x_i^\beta) - v_r(x_j^\beta) \right] \frac{1}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}}.
\]
Now by using an explicit approach to time discretize the above equation, we obtain
\[
\bar{v}_r(x_i^\beta) - \bar{v}_r(x_i^{\beta'}) = 2\nu \sum_{j=1}^{N} \frac{m_j}{\rho_L} \left[ v_r(x_i^\beta) - v_r(x_j^\beta) \right] \frac{1}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}}
\]
where \(\Delta t\) is the time-step increment, \(m\) is the previous time index, and from Equation (2.11) the distance between particles in the two-dimensional cylindrical coordinate system is
\[
x_{ij} = \sqrt{r_{ij}^2 + z_{ij}^2}.
\] (4.8)
Upon rearranging the equation for \(\bar{v}_r(x_i^\beta)\), we obtain
\[
\bar{v}_r(x_i^\beta) = \bar{v}_r(x_i^{\beta'}) + 2\nu \Delta t \sum_{j=1}^{N} \frac{m_j}{\rho_L} \left[ v_r(x_i^\beta) - v_r(x_j^\beta) \right] \frac{1}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}}
\] (4.9)
A similar procedure applied to Equation (4.4) can be carried out to produce the intermediate velocity \(\bar{v}_z(x_i^\beta)\) as
\[
\bar{v}_z(x_i^\beta) = v_z^{m}(x_i^\beta) + 2\nu \Delta t \sum_{j=1}^{N} \frac{m_j}{\rho_L} \left[ v_z^{m}(x_i^\beta) - v_z^{m}(x_j^\beta) \right] \frac{1}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}}
\]
\[+ a_g \Delta t \left( \beta_T T_i^{m} + \beta_C C_i^{m} \right) - a_g \Delta t \left( \beta_T T_0 + \beta_C C_0 \right).
\] (4.10)
where we take values of \(T_i\) and \(C_i\) directly, as opposed to their SPH approximations, as these quantities will be computed from the SPH energy and mass transport equations. Note that by definition of the kinematic viscosity, \(\nu = \mu/\rho_L\). Recall that in the PSPH method, the velocities determined by Equations (4.9) and (4.10) are intermediate velocities which will not, in general, satisfy mass conservation. Once the intermediate velocity components are known, the pressure field at each time-step is then computed using
\[
\sum_{j=1}^{N} \frac{2m_j p_{ij}}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} = \frac{\rho_L}{\Delta t} \sum_{j=1}^{N} m_j \left[ \bar{v}_r(x_i^\beta) + \bar{v}_r(x_j^\beta) \right] \frac{r_{ij}}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}}
\]
\[+ \frac{\rho_L}{\Delta t} \sum_{j=1}^{N} m_j \left[ \bar{v}_z(x_i^\beta) + \bar{v}_z(x_j^\beta) \right] \frac{z_{ij}}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}}
\] (4.11)
as required by PSPH. Following determination of the pressure field consistent with mass conservation, Hodge decomposition is used to find the actual velocity components

\[
v^{m+1}_i (x^\beta_i) = \overline{v}_i (x^\beta_i) - \Delta t \sum_{j=1}^{N} \frac{m_j}{\rho_l} (p_i + p_j) \frac{r_{ij}}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} \tag{4.12}
\]

\[
v^{m+1}_z (x^\beta_i) = \overline{v}_z (x^\beta_i) - \Delta t \sum_{j=1}^{N} \frac{m_j}{\rho_L} (p_i + p_j) \frac{z_{ij}}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} \tag{4.13}
\]

**PSPH Balance of Energy:**

In PSPH, the liquid and solid phase energy balances can be computed as a single equation with a simple modification to the particle conductivities that effectively accounts for a continuous heat flux through material interfaces. The balance of energy is written using a Crank-Nicolson time discretization, as well as Equation (2.47) for the SPH Laplacian

\[
T^{m+1}_i = T^m_i + \alpha_{Di} \Delta t \sum_{j=1}^{N} \frac{m_j}{\rho_l} \frac{(T^{m+1}_i - T^{m+1}_j)}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} + \alpha_{Di} \Delta t \sum_{j=1}^{N} \frac{m_j}{\rho_l} \frac{(T^{m}_i - T^{m}_j)}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}}. \tag{4.14}
\]

where \(\alpha_{Di} = \kappa_i / \rho_i c_p_i\) is the thermal diffusivity for particle \(i\). To account for changes in thermal conductivity between different materials within the domain, the SPH conductivity

\[
\kappa_i \rightarrow \frac{2 \kappa_i \kappa_j}{\kappa_i + \kappa_j}
\]

was employed. Use of the above SPH conductivity eliminates the need for special thermal interface treatment, since this smoothed conductivity is derived [34] from the principle that the heat flux is continuous. Note that the above averaging of the conductivity is consistent for the special case where \(\kappa_i = \kappa_j\).
PSPH Mass Transport:
The mass transport equation for the liquid phase is also written using an explicit time discretization and Equation (2.47) for the PSPH Laplacian as

\[ C_{i}^{m+1} = C_{i}^{m} + D_{L} \Delta t \sum_{j=1}^{N} \frac{m_{j}}{\rho_{L}} \left( C_{i}^{m} - C_{j}^{m} \right) \frac{\partial W(x_{ij}, h)}{\partial x_{ij}}. \] (4.15)

With the governing equations for the liquid and solid phases presented, along with their PSPH approximations, the boundary and interface conditions for the LPD crystal growth system are now discussed.

### 4.4 Boundary, Initial, and Interface Conditions

**At the Vertical Crucible Wall, \( r = R \):**
At the vertical wall of the crucible, a no-slip boundary condition is enforced, where

\[ v_{r}(R, z, t) = v_{z}(R, z, t) = 0 \quad \text{and} \quad \frac{\partial C(R, z, t)}{\partial r} = 0 \] (4.16)

and the vertical walls are assumed impermeable to the species in the solution.

**At the Axis of Symmetry, \( r = 0 \):**
At the center of the crucible, the following conditions must be met in order to obtain a physical, finite solution

\[ \frac{\partial v_{z}(0, z, t)}{\partial r} = \frac{\partial T(0, z, t)}{\partial r} = \frac{\partial C(0, z, t)}{\partial r} = 0 \quad \text{and} \quad v_{r}(0, z, t) = 0. \] (4.17)

**At the Growth Interface:**
At the crystal growth interface, the species concentration is determined from the binary Si-Ge phase diagram \( C_{L, \text{eq}} = f(T) \) for thermal equilibrium. The Si-Ge phase diagram relation is determined experimentally for the liquidus curve at standard atmospheric conditions near the Ge-rich side as [1]

\[ C_{L, \text{eq}}(T) = 2.5984 \times 10^{-9} T^{3} - 8.7189 \times 10^{-6} T^{2} + 9.9041 \times 10^{-3} T - 3.822 \] (4.18)
and the solidus curve
\[ C_{\text{S,eq}}(T) = \frac{T - 1211.45}{639.63} \]  \hspace{1cm} (4.19)
where the temperature must be in Kelvin, and the concentration refers to the silicon mass fraction. As well, since the growth rates in LPD are several orders of magnitude smaller than the velocities of fluid particles, we have the growth interface boundary conditions
\[ v_r = v_z = 0 \quad \kappa_S \frac{\partial T_S}{\partial n} = \kappa_L \frac{\partial T_L}{\partial n} \]
\[ C = C_{\text{L,eq}} \quad \text{and} \quad \rho_S u_g (C_{\text{S,eq}} - C_{\text{L,eq}}) = \rho_L D_L \frac{\partial C}{\partial n} \]  \hspace{1cm} (4.20)
where \( u_g \) is the rate of displacement of the growth interface along a unit normal direction (i.e. the growth velocity). The initial concentration at the growth interface is taken as \( C(r, z, 0) = 0 \) since the crystal substrate is pure germanium. The unit normal, from **Figure 4.1** is denoted by \( n_\alpha \). Notice that the latent heat of fusion and the surface tension effects have been neglected in the above energy balance at the interface, as discussed in [1].

**At the Dissolution Interface:**
At the dissolution interface, the following boundary conditions are required
\[ C = C_{\text{L,eq}} \quad \text{and} \quad v_r = v_z = 0 \quad \text{and} \quad \kappa_S \frac{\partial T_S}{\partial n} = \kappa_L \frac{\partial T_L}{\partial n}. \]  \hspace{1cm} (4.21)
As well, initial conditions of \( C(r, z, 0) = C_0 \) as determined from Equation (4.18) and \( v_r(r, z, 0) = v_z(r, z, 0) = 0 \) are used.

**At the Boundaries of the Crucible:**
The solid phase energy balance equation also requires boundary conditions.
For the vertical wall of the crucible, and top and bottom solid surfaces, we require
\[ -\kappa_S \frac{\partial T}{\partial n} = \hat{H} [T - T_f(z)] \quad \text{and} \quad \frac{\partial T(0, z, t)}{\partial n} = 0 \]  \hspace{1cm} (4.22)
where \( \kappa_S \) is the thermal conductivity of the solid crystal, \( T_f(z) \) is the ambient temperature inside the furnace along the quartz crucible wall (or on the top
or bottom solid surface), and \( \tilde{H} \) is the modified heat transfer coefficient determined experimentally, to include the effects of convective and radiative heat transfer. At the axis of symmetry, it is required that \( \partial T(0, z, t)/\partial n = 0 \).

### 4.5 SPH Boundary, Initial, and Interface Conditions

All boundary conditions in the LPD system are either specified values, or specified gradients. If a field quantity is to be kept at a specified value, an SPH approximation is not required, since implementation of such specification is direct. For example, \( v_r(R, z, t) = 0 \) at the vertical wall of the crucible does not require an SPH approximation, since all that is required in the coding is that all particles at radial location \( r = R \) are forced to have radial velocity component \( v_r = 0 \).

For boundary conditions involving gradient specifications, Equation (2.44) can be used, where a factor of 2 is inserted to account for the one-sided gradient at the boundary. In other words, since there are only particles on one side of a boundary, the factor of 2 serves to balance the averaged gradient calculation. For example, the boundary condition \( \partial C(R, z, t)/\partial r = 0 \) can be written using Equation (2.44) as

\[
\left\langle \frac{\partial C_i(R, z, t)}{\partial r} \right\rangle = 2 \sum_{j=1}^{N} \frac{m_{ij}}{\rho_i} (C_j - C_i) \frac{r_{ij}}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} = 0
\]

where \( i \) is a particle on the boundary \( r = R \) and particles \( j \) will consist of boundary and interior particles. We now explicitly present the SPH boundary and initial conditions, where it is understood that all particles \( i \) represent particles located on the boundary in question.

**At the Vertical Crucible Wall, \( r_i = R \):**

\[
v_r = v_z = 0
\]
\[
\sum_{j=1}^{N} \frac{m_j}{\rho_i} (C_j - C_i) \frac{r_{ij}}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} = 0
\]

(4.23)

At the Axis of Symmetry, \( r_i = 0 \):

\[
\sum_{j=1}^{N} \frac{m_j}{\rho_i} \left[ v_z(x_j^\beta) - v_z(x_i^\beta) \right] \frac{r_{ij}}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} = 0
\]

\[
\sum_{j=1}^{N} \frac{m_j}{\rho_i} (T_j - T_i) \frac{r_{ij}}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} = 0
\]

\[
\sum_{j=1}^{N} \frac{m_j}{\rho_i} (C_j - C_i) \frac{r_{ij}}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} = 0
\]

\( v_r = 0 \).

(4.24)

At the Growth Interface:

\[ C_i = C_{L,eq}(T_i) = 2.5984 \times 10^{-9} T_i^3 - 8.7189 \times 10^{-6} T_i^2 + 9.9041 \times 10^{-3} T_i - 3.822 \]

\[ C_{S,eq}(T_i) = \frac{T_i - 1211.45}{639.63} \]

\[ v_r = v_z = 0 \]

\[ \rho_s u_g (C_{S,eq} - C_{L,eq}) = \rho_L D_L \frac{\partial C}{\partial n} \]

(4.25)

At the Dissolution Interface:

\[ C_i = C_{L,eq} \]

\[ v_r = v_z = 0. \]

(4.26)

As well, initial conditions of \( C_i(r, z, 0) = C_0 \) and \( v_r(r, z, 0) = v_z(r, z, 0) = 0 \) are used.

At the Boundaries of the Crucible:

\[
\sum_{j=1}^{N} \frac{\kappa_s m_j}{\rho_i} \left[ T_i(R, z, t) - T_j \right] \frac{r_{ij}}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} = \tilde{H} \left[ T_i(R, z, t) - T_f(z) \right]
\]
\[
\sum_{j=1}^{N} \frac{\kappa_{Si} m_j}{\rho_i} [T_i(r, 0, t) - T_j] \frac{z_{ij}}{x_{ij}} \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} = \frac{\partial W(x_{ij}, h)}{\partial x_{ij}} = 0
\]

4.6 Physical Parameters of Si\textsubscript{x}Ge\textsubscript{1-x}

In order to run the LPD SPH simulations, appropriate physical parameters must be known to obtain realistic results. These parameters include material densities, viscosities, conductivities, specific heat capacities, diffusion coefficients, and thermal and solutal expansion coefficients. Table 4.1 lists the numerical values for the physical parameters used in the current model [1]. All quantities with a dash in Table 4.1 are not required in the numerical simulations, and are therefore not applicable. Also note that in the linear momentum balance of Equation (4.4) we take the reference temperature and initial Si concentration as \( T_0 = 1211 \) K and \( C_0 = 0 \) kg/kg, respectively.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Si Source at ( T = 1300 ) K</th>
<th>Ge Substrate at ( T = 1210 ) K</th>
<th>( \text{Si}<em>x\text{Ge}</em>{1-x} ) Solid</th>
<th>( \text{Si-Ge} ) Liquid</th>
</tr>
</thead>
<tbody>
<tr>
<td>( c_p ) [J/kg K]</td>
<td>967</td>
<td>396.1</td>
<td>396.1 - 487</td>
<td>380 - 406</td>
</tr>
<tr>
<td>( \kappa ) [W/m K]</td>
<td>23.7</td>
<td>10.60</td>
<td>10.60</td>
<td>42.8</td>
</tr>
<tr>
<td>( D ) [m\textsuperscript{2}/s]</td>
<td>-</td>
<td>-</td>
<td>1.0 \times 10\textsuperscript{-20}</td>
<td>2.5 \times 10\textsuperscript{-8}</td>
</tr>
<tr>
<td>( \rho ) [kg/m\textsuperscript{3}]</td>
<td>2301.6</td>
<td>5323</td>
<td>4839 - 5323</td>
<td>5633</td>
</tr>
<tr>
<td>( \alpha_D ) [m\textsuperscript{2}/s]</td>
<td>1.1 \times 10\textsuperscript{-5}</td>
<td>5.0 \times 10\textsuperscript{-6}</td>
<td>5.0 \times 10\textsuperscript{-6}</td>
<td>1.9 \times 10\textsuperscript{-5}</td>
</tr>
<tr>
<td>( \beta_T ) [1/K]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1.2 \times 10\textsuperscript{-4}</td>
</tr>
<tr>
<td>( \beta_C ) [1/Wt% Si]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1.06 \times 10\textsuperscript{-2}</td>
</tr>
<tr>
<td>( \mu ) [kg/m s]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>7.35 \times 10\textsuperscript{-4}</td>
</tr>
<tr>
<td>( \nu ) [m\textsuperscript{2}/s]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1.305 \times 10\textsuperscript{-7}</td>
</tr>
</tbody>
</table>
Finally, Table 4.2 lists the numerical values for the applicable physical parameters for the quartz crucible [1].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Numerical Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_p$ [J/kg K]</td>
<td>1200</td>
</tr>
<tr>
<td>$\kappa$ [W/m K]</td>
<td>2.0</td>
</tr>
<tr>
<td>$\rho$ [kg/m$^3$]</td>
<td>2200</td>
</tr>
</tbody>
</table>
5 SPH Simulation of LPD Crystal Growth

In order to simulate LPD crystal growth using the SPH method, two discretized domains were selected, a coarse domain and a fine, high-resolution domain. The coarse domain consisted of an $18 \times 41$ array of $N = 738$ particles, where the initial smoothing length was taken as $h = 1.0$ mm. This domain was used primarily for preliminary results and code debugging. For more accurate results, a higher-resolution domain consisting of a $34 \times 81$ array of $N = 2754$ particles was also selected, where the initial smoothing length was chosen as $h = 0.5$ mm. These smoothing lengths are approximately equal to the initial particle spacings. The purpose of selecting two different domains for the SPH simulations of LPD was to compare the particle resolution effects on simulation accuracy.

Figure 5.1 illustrates both SPH domains used to discretize the LPD crystal growth system, where all dimensions are given in meters. The quartz crucible is represented by $\bullet$ particles, $\circ$ particles represent the Si-Ge solution, and $\Box$ and $\triangledown$ particles represent the germanium substrate and silicon source particles, respectively. As before, the $\ast$ symbol denotes particles that lie on the domain boundary, which may be quartz, solution, substrate, or silicon particles depending on boundary location.
The SPH simulation results were compared with the known solution as determined by Yildiz [1] using the finite volume approach. The finite volume LPD simulation domain consisted of separate computational regions for the Ge crystal substrate, Si-Ge liquid region, Si source material, and quartz ampoule. This domain is illustrated below in Figure 5.2. For all finite volume simulations, the respective mesh sizes for each region are given in Table 5.1.

<table>
<thead>
<tr>
<th>Domain Region</th>
<th>Mesh Size</th>
<th>Quartz Ampoule Mesh Size Adjacent to Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Si Source</td>
<td>10 × 50</td>
<td>10 × 10</td>
</tr>
<tr>
<td>Si-Ge Solution</td>
<td>80 × 50</td>
<td>80 × 10</td>
</tr>
<tr>
<td>Ge Substrate</td>
<td>30 × 50</td>
<td>30 × 10</td>
</tr>
</tbody>
</table>
Figure 5.2: Finite Volume Domain for LPD Simulation

Initially, the simulations were to be performed without moving the crystal growth interface. Neglecting the movement of the growth surface was acceptable for these first simulations since LPD was modelled to a time of $t = 0.5$ hour, over which time the furnace is heating up to a steady-state, and the system is stabilizing. After this time, it should be a straightforward matter to evolve the system and move the growth interface using Equation (4.25). The only modification in the code is to determine the shape of the crystal growth surface in order to compute $\partial C/\partial n$, at which point the growth velocity $u_g$ can be found. Here, $n$ represents the surface normal of the growth interface. Numerically, this interface can be determined from a polynomial curve fit subroutine that returns the polynomial coefficients of the interface, from which the surface normal $n$ can be deduced.

An AMD 64-bit 3500+ processor with 512 MB of RAM running MATLAB V6 was used to carry out all simulations for LPD crystal growth using PSPH.
5.1 SPH Coding for LPD Crystal Growth Simulation

All source code and corresponding subroutines were written as MATLAB m-files, and no built-in MATLAB subroutines were employed, with the exception of the plotting commands. The complete LPD simulation MATLAB code consisted of the main source code, called LPDMainPSPHReMesh.m. This source code employed use of the following ten m-file subroutines, as given in Table 5.2.

<table>
<thead>
<tr>
<th>Subroutine Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPDInitializeParticlesEven.m</td>
<td>Sets up the computational domain and initializes all global variables</td>
</tr>
<tr>
<td>LPDBoxNeighborFinder.m</td>
<td>Determines and stores all particle neighbors</td>
</tr>
<tr>
<td>SecondKernelDerivative.m</td>
<td>Computes the kernel value, all distance variables, and all derivatives</td>
</tr>
<tr>
<td>LPDTempSteadyState.m</td>
<td>Computes the steady-state temperature distribution</td>
</tr>
<tr>
<td>LPDReMeshPSPH.m</td>
<td>Redistributes all disordered particles to an ordered configuration, and interpolates all field quantities</td>
</tr>
<tr>
<td>KernelGradReMesh.m</td>
<td>Determines the kernel gradients during the particle redistribution portion; called in LPDReMeshPSPH.m</td>
</tr>
<tr>
<td>LPDConcentration.m</td>
<td>Determines the concentration field</td>
</tr>
<tr>
<td>KernelGradLPD.m</td>
<td>Computes the kernel gradient used in LPDConcentration.m</td>
</tr>
<tr>
<td>LPDVelocity.m</td>
<td>Determines the velocity field, as well as the pressure field using Hodge decomposition. This function calls LPDPPESolverPreConjGM.m to solve the pressure Poisson equation</td>
</tr>
<tr>
<td>LPDPPESolverPreConjGM.m</td>
<td>Solves the pressure Poisson equation with Neumann boundary conditions using the preconjugated gradient algorithm</td>
</tr>
</tbody>
</table>
Figure 5.3 illustrates the PSPH algorithm used for the LPD simulations. A simulation starts with an ordering of particles evenly distributed throughout the domain. The particle neighbors are then found, and the steady-state temperature field is computed. At this point the concentration field can be calculated. The principle of Hodge decomposition is then used to find the correct, incompressible velocity field used to move the particles to a new location, called advection.

Figure 5.3: PSPH Algorithm for LPD Simulations
Of course, as crystal growth progresses, particle properties will require changing at the growth and dissolution interfaces in accordance with the binary Si-Ge phase diagram. To eliminate error accumulation due to disordered particles, the field quantities are interpolated to a uniform distribution, and the particles reordered, as outlined in the literature [45]. The time level is then incremented by $\Delta t$, and the process continued until a specified time $t_{\text{max}}$ has been reached. Note that with this algorithm, the neighbors need only be found once, since the particles are redistributed at each time-step to a uniform arrangement.

5.2 LPD PSPH Simulation Results

It has been previously demonstrated [1] that the thermal profile may be computed independently of the flow field. That is, even though particles will move during the simulation according to the solution of the Navier-Stokes equations, this convective flow will not have a significant effect on the temperature field. This assumption was deduced from a dimensional analysis, where the computed Prandtl number $Pr = \nu/\alpha_D \cong 0.0075$ was determined for the LPD system. The Prandtl number is a ratio between the convective and conductive effects, and as long as $Pr \ll 1$, the thermal transport across the solution is dominated by conduction as opposed to the convective transport of fluid.

Equation (4.14) was used to compute the thermal profile, and the time-step was chosen as $\Delta t = 0.015$ s based on the limitation imposed by Equation (2.64) of $\Delta t \leq 0.0158$ s for the coarse particle distribution. For the high-resolution simulation, the time-step was taken as $\Delta t = 0.0025$ s based on the limitation imposed by Equation (2.64) of $\Delta t \leq 0.00395$ s, where now $h = 0.5$ mm.
To determine the SPH concentration field, the SPH mass balance Equation (4.15) was used. The time-step limitation for the concentration field can be found from Equation (2.64), with $\alpha_D$ replaced by the diffusivity $D$, producing $\Delta t \leq 12$ s for the coarse grid. The fine grid limitation is $\Delta t \leq 3$ s. However, since these respective time-steps are greater than those required by the flow field, the flow field time-step limitations are used for both the coarse and fine particle distributions, as the concentration field and velocity field must be computed simultaneously.

Concerning the flow field simulation, the flow regime is assumed to be laminar. The laminar flow assumption is justified by computation of the Reynolds number defined by

$$Re = \frac{||v^a||H_L}{\nu}$$

from Equation (2.59) producing a Reynolds number of approximately 300 since the maximal order of magnitude of the fluid velocity is computed as $10^{-3}$ m/s from the finite volume results [1]. As long as $Re \leq 2000$, the flow is considered to be laminar flow. If the flow was turbulent, some appropriate method of turbulence modelling would need to be incorporated into the Navier-Stokes equations, Equations (4.12) and (4.13). The time-step for the flow simulations was chosen as $\Delta t = 0.20$ s based on the limitation imposed by Equation (2.72) of $\Delta t \leq 0.25$ s for the coarse grid. The time-step for the flow simulations for the high-resolution simulation was chosen as $\Delta t = 0.10$ s based on the limitation imposed by Equation (2.72) of $\Delta t \leq 0.125$ s.

Figure 5.4 compares the coarse SPH and finite volume (FV) [1] solutions to the thermal profile at $t = 0.5$ hours, where all temperatures are given in Kelvin.
Figure 5.4: Coarse SPH and FV Temperature Field at $t = 0.5$ Hour

It is seen that the agreement between the SPH and finite volume results is very good, including the predicted initial location of the crystal growth interface, indicated by a curved black line. The initial location of the crystal growth interface was determined by the isothermal line corresponding to the melting temperature of germanium, approximately 1211 K. The range of temperature values are also in very close agreement between the two solution approaches. The coarse SPH solution, on the other hand, used a much smaller resolution of only 738 particles, as compared with the finite volume resolutions given in Table 5.1.

Figure 5.5 compares the high-resolution SPH and finite volume (FV) [1] solutions to the thermal profile at $t = 0.5$ hours, where again all temperatures are given in Kelvin.
Figure 5.5: Fine SPH and FV Temperature Field at $t = 0.5$ Hour

It was found that both high and low-resolution SPH thermal simulations produced nearly identical results, showing that for this simulation even the coarse particle distribution of only $N = 738$ particles was acceptable.

Unfortunately, the PSPH method did not simulate the concentration profile or the flow field correctly. After two months of code debugging, a correct simulation for these field quantities was not obtained. This result was frustrating, especially given the successful test problem simulations performed and presented in Chapter 3.

In order to meet the main objective of the present work, it was required to find out whether the erroneous results were due to faulty coding, or some other deficiency of SPH itself with respect to simulation of crystal growth.
5.3 Code Troubleshooting

Coding mistakes could have been contained in any or all of the ten subroutines outlined in Table 5.2. However, the function \textit{LPDTempSteadyState.m} was a stand-alone application which produced the correct temperature field for the LPD system, as indicated in Figures 5.4 and 5.5. Each other subroutine had its own unique possible errors, as well as a unique way of testing the function. Therefore, systematic testing of each function was carried out, with the exception of the function \textit{LPDTempSteadyState.m}, and these details are discussed here.

5.3.1 Function \textit{LPDInitializeParticlesEven.m}

This particle initialization function was essentially the same as those used for all SPH test problems as presented in Chapter 3. The differences were in the boundary particle indicators (called "flags"), which for the LPD simulations needed to be on the interior of the domain, at the interface between the quartz ampoule and the Si-Ge solution. However, this subroutine was tested and manually checked to ensure that all flags were accurate and producing the correct values. This function was further tested and used successfully in the function \textit{LPDTempSteadyState.m}. Therefore, the erroneous simulation could not have been due to an incorrect particle initialization.

5.3.2 Function \textit{LPDBoxNeighborFinder.m}

The particle neighbor-finding subroutine was tested extensively during the SPH test problems outlined in Chapter 3. The box algorithm used to determine particle neighbors was also tested against the standard \(N \times N\) neighbor-finding algorithm, producing identical results. Therefore, incorrect particle neighbor determination was not the cause of incorrect SPH simlu-
5. SPH SIMULATION OF LPD CRYSTAL GROWTH

sections of LPD.

5.3.3 Function SecondKernelDerivative.m, KernelGradLPD.m, and KernelGradReMesh.m

All subroutines used to compute kernel derivatives and values were extensively tested during the simulation of the SPH test problems of Chapter 3, and found to be accurate. Once again, the problem with the SPH LPD simulations could not have been linked with the kernel derivative subroutines.

5.3.4 Function LPDReMeshPSPH.m

As each particle advected to a new location, the global particle distribution became disordered. The remeshing subroutine was responsible for reordering the particles. The level of disorder depends greatly on the strain of the flow field, and will increase as the Reynolds number is increased. The function LPDReMeshPSPH.m takes the advected particle positions and subsequently finds all ordered particle neighbors. Once the ordered neighbors have been found, the disordered field quantities (namely velocity, pressure, and concentration) are interpolated onto the ordered particle mesh using the averaging properties of the kernel. This interpolation code was compared with various MATLAB interpolation routines, and actually found to outperform the MATLAB routines in SPH applications. Hence, this function was sufficiently tested and found to be accurate with respect to the flow cavity problem discussed in Chapter 3. As well, the PSPH results with the remeshing code were found to be in agreement with the quasi-incompressible SPH flow cavity results without remeshing. This fact is an indication that the remeshing/interpolation subroutine is indeed accurate.
There are only two obvious differences between the LPD simulation and the flow cavity simulation, namely, the Reynolds number and the inclusion of the body force in the linear momentum balance, Equation (4.4). The SPH simulations were carried out with and without inclusion of the body force term, and the results were incorrect in either case. In LPD Re \( \cong 300 \), whereas in the flow cavity test problem the Reynolds number was a much lower Re = 1. Because of this difference in Reynolds number, the remeshing subroutine had to be called at every time-step. Unfortunately, such frequent remeshing can introduce numerical errors, due to the numerical averaging and interpolations required when remeshing. However, frequency of remeshing tests at smaller than required time-steps indicated that these numerical errors should not be leading to the erroneous simulation results obtained. It was then concluded that the incorrect simulation results were not due to the remeshing subroutine.

### 5.3.5 Function LPDConcentration.m

The testing of the concentration field subroutine was straightforward in that all that was required was to run the function without advecting the particles. In this sense, the concentration field could be found in the same manner as the temperature field, by ignoring the convective flow of the fluid. **Figure 5.6** compares the coarse SPH and finite volume [1] solutions to the silicon concentration field at \( t = 0.5 \) hours. Note that the finite volume solution incorporates the convective fluid flow, whereas in this test the PSPH solution does not. It is seen from **Figure 5.6** that the SPH and finite volume solutions compare reasonably well with one another, implying that the concentration subroutine is working correctly.
5. SPH SIMULATION OF LPD CRYSTAL GROWTH

Figure 5.6: Coarse SPH and FV Concentration Field at $t = 0.5$ Hour

5.3.6 Function LPDPPESolverPreConjGM.m

Determination of the correct pressure field can be quite challenging. In the PSPH technique, one algorithm often used for solving the pressure Poisson equation (PPE) is the preconjugated gradient technique. In order to troubleshoot this subroutine, two other algorithms were written to solve the PPE to compare with the preconjugated gradient method results. In total, three functions were then tested, using Gauss-Siedel iteration, direct matrix inversion, and the preconjugated gradient approach. It was found that all three subroutines produced identical results, implying that the PPE was being solved correctly. However, Gauss-Siedel iteration took approximately 400 to 500 iterations to converge, whereas the preconjugated gradient approach took only around 50 iterations to converge for the coarse SPH particle distribution domain. Direct matrix inversion performed better than Gauss-Siedel iteration, but was not as fast as the preconjugated gradient approach. Since
all three functions produced the same pressure field, the \texttt{LPDPPESolverPre-ConjGM.m} subroutine can be assumed correct.

### 5.3.7 Function \texttt{LPDVelocity.m}

Since all of the other subroutines have been tested, we have, by a process of elimination, that the \texttt{LPDVelocity.m} subroutine must be the source of the erroneous LPD simulation results. Figure 5.7 shows the comparison between the coarse SPH and finite volume [1] solutions to the flow field at a simulation time \( t = 0.5 \) hours. The plotted contours are for the magnitude of the flow \( \|v^a\| = \sqrt{v_r^2 + v_z^2} \).

![Figure 5.7: Coarse SPH and FV Flow Field at \( t = 0.5 \) Hour](image)

Not only is the flow field qualitatively incorrect, but the flow magnitudes are up to ten times higher in the PSPH simulation. The high-resolution SPH simulation produces similar erroneous results.
However, the function \textit{LPDFVelocity.m} is almost identical to the velocity subroutine used in the successful flow cavity PSPH simulation of \textbf{Chapter 3}. The primary difference is the Reynolds number, suspected to be the root of the problem. A survey of the SPH literature reveals that almost all reported simulations are for extremely low Reynolds numbers. In fact, the only successful high Reynolds number simulations employ the use of very large numbers of particles. Chaniotis et al. [45] successfully simulated the driven cavity problem using quasi-incompressible SPH with \(N = 63,000\) particles for a Reynolds number \(Re = 400\). Having such a high number of particles, and consequently a miniscule time-step required by Equation (2.62), requires extremely powerful computational resources, and most likely parallel processing, although these details were overlooked in the paper by Chaniotis et al. [45]

One key difference in the results of Chaniotis et al. and the current LPD simulations is the drastic difference in number of particles \(N\). Recall that \(N\) had a significant impact on the accuracy of the flow cavity results given in \textbf{Chapter 3}. As \(N\) is increased, the computational time increases drastically due to both the increased number of neighbors to find and store, as well as the necessary reduction in time-step required by Equation (2.72).

Therefore, the erroneous PSPH LPD simulations almost certainly stem from a particle resolution that is too low. Following the results of Chaniotis et al., as the number of PSPH particles was increased to \(N = 57,600\) (a particle array of \(160 \times 360\)) the required time-step from Equation (2.72) reduced to \(\Delta t \leq 0.024\) s for \(h \approx 1.04 \times 10^{-4}\) m. For a \(t_{\text{max}} = 0.5\) hour crystal growth simulation, this time-step requires a total of 75,000 iterations. With this particle resolution, the CPU time per time-step was found to be an unacceptable \(640\) s running in MATLAB. Based on a simulation time of 0.5 hours, the total CPU time becomes over 18 months for a single simulation.
It is obvious that MATLAB cannot be used to simulate LPD using SPH at the current level of CPU power. Therefore, it was proposed to convert the MATLAB code into C++ or Fortran code and carry out the simulations in a more computationally efficient manner.

A particle neighbor-finding time test comparison between MATLAB and Fortran indicated that Fortran equivalent code is up to 90 times faster than MATLAB. A conservative estimate for the SPH simulation then produces a total CPU time of around six days with equivalent Fortran code. Note that the current finite volume simulations written in Fortran can produce a simulation to \( t_{\text{max}} = 8.0 \) hours in only one day [1, 46]. Based on the above arguments, it became clear that PSPH simulation of LPD crystal growth does not have the significant advantages initially anticipated, with the current computational power available. The benefit of automatic adaptability of SPH with respect to the crystal growth interface does not at this time outweigh the significant disadvantages in terms of CPU times compared with the finite volume simulation approach.
6 Conclusions

From the results of Chapter 5, it has been found that SPH is not feasible for simulation of growth of binary semiconducting crystals using LPD at this point in time. Although the averaging procedure of SPH is acceptable, the number of particles required to simulate fluid systems with realistic Reynolds numbers is too large, and requires simulation times much greater than equivalent finite volume simulations. In addition, the required time-steps for such simulations becomes increasingly small as the number of particles is increased. In its current formulation, the SPH method is not feasible for crystal growth simulations, and in terms of computational time required is far inferior to the currently employed finite volume technique.

However, it was found that the SPH temperature field was nearly identical to the results obtained using the finite volume approach. Unfortunately, the SPH flow and concentration fields were not simulated correctly, and therefore compared poorly with the finite volume simulation results. When the SPH particles were not permitted to move, however, the concentration field was simulated correctly using the SPH averaging procedure.

Since SPH is a relatively new technique, it is not surprising to see frequent modifications to the SPH equations in the literature. It is possible that in the future implicit time integration schemes in addition to accuracy-improvement modifications will be incorporated into SPH so that accurate crystal growth simulations can be obtained with a fewer number of particles than currently
required, thus reducing computational times.

### 6.1 Contributions

The work presented herein is unique in that it has investigated the application of the SPH method to crystal growth by LPD. In addition, the following specific contributions were made:

1. The PSPH equations for LPD crystal growth, along with appropriate boundary conditions were derived in a tensorial framework independent of a chosen coordinate system.

2. Modular PSPH MATLAB code was developed to simulate LPD crystal growth.

3. PSPH simulation of LPD crystal was determined to be inappropriate at this time, and some shortcomings of PSPH applied to crystal growth have been reported.

4. From first principles, an SPH vector Laplacian operator was derived and given in Equation (2.39) that contains only first-order derivatives.

### 6.2 Future Work

There are several possibilities for extensions of this work in the future. If a modular PSPH code is developed in Fortran, and if the computational power of computers has increased enough to make SPH simulation times comparable with mesh-dependent techniques, the following two recommendations are made:
6. CONCLUSIONS

1. That PSPH code be written to simulate an as yet undeveloped model of crystal growth from the vapour phase. The particle framework of SPH is well suited to such a model, as SPH particles can be used to model discrete gas particles.

2. That the SPH method be further compared with the finite volume and finite element crystal growth simulations in terms of the computation time required when tracking the crystal interface. For example, in the finite volume approach, the concentrations on both the solid and liquid sides of the interface must be solved iteratively in order to preserve the balances of mass and energy. In three-dimensional simulations, these iterations are very computationally expensive. If the crystal interface tracking can be implemented correctly using the SPH method, SPH has the potential to offer a major advantage over finite volume and finite element methods as applied to crystal growth.
References


REFERENCES


Appendix A: Covariant Derivatives

In general, vector quantities can be written using covariant or contravariant components. Covariant components are written as subscripts, whereas contravariant components are denoted with superscripts. Covariant components of a vector \( \vec{A} \) reference basis vectors \( \hat{e}^1, \hat{e}^2, \ldots, \hat{e}^N \) that are perpendicular to constant coordinate surfaces. An \( N \)-dimensional vector can be written with respect to this basis as

\[
\vec{A} = A_1 \hat{e}^1 + A_2 \hat{e}^2 + \cdots + A_N \hat{e}^N
\]

where \( A_\alpha \) (\( \alpha = 1, 2, \ldots, N \)) are the covariant components of \( \vec{A} \). A dual set of basis vectors \( \hat{e}_1, \hat{e}_2, \ldots, \hat{e}_N \) are those that are tangent to constant coordinate curves, producing the alternative representation of \( \vec{A} \) as

\[
\vec{A} = A^1 \hat{e}_1 + A^2 \hat{e}_2 + \cdots + A^N \hat{e}_N
\]

in terms of contravariant components \( A^\alpha \) (\( \alpha = 1, 2, \ldots, N \)). Higher order vectors are called tensors. Tensors that have both covariant and contravariant components are called mixed tensors.

A tensor entity \( T_{\beta_1 \beta_2 \cdots \beta_m}^{\alpha_1 \alpha_2 \cdots \alpha_m} \) of order \((m + n)\) can be defined as an entity that obeys the tensor transformation law:

\[
T_{\beta_1 \beta_2 \cdots \beta_m}^{\alpha_1 \alpha_2 \cdots \alpha_m} = T_{\lambda_1 \lambda_2 \cdots \lambda_m}^{\gamma_1 \gamma_2 \cdots \gamma_m} \frac{\partial x_{\lambda_1}}{\partial x_{\gamma_1}} \frac{\partial x_{\lambda_2}}{\partial x_{\gamma_2}} \cdots \frac{\partial x_{\lambda_m}}{\partial x_{\gamma_m}} \frac{\partial x_{\lambda_{m+1}}}{\partial x_{\gamma_{m+1}}} \cdots \frac{\partial x_{\lambda_n}}{\partial x_{\gamma_n}}
\]  \hspace{1cm} (A.1)

where we have two arbitrary coordinate systems with transformations \( x^\alpha = x^\alpha(x^1, x^2, \ldots, x^N) \) and \( \bar{x}^\beta \bar{=} \bar{x}^\beta(x^1, x^2, \ldots, x^N) \).

A covariant derivative with respect to \( x^\nu \) coordinates (denoted by \( ( )_{\nu} \) or sometimes by \( \nabla_\nu \)) for a general ordered tensor \( A_{\sigma \lambda \cdots \mu}^{\alpha \beta \cdots \gamma} \) can be formulated as:

\[
A_{\sigma \lambda \cdots \mu; \nu}^{\alpha \beta \cdots \gamma} = \frac{\partial}{\partial x^\nu} A_{\sigma \lambda \cdots \mu}^{\alpha \beta \cdots \gamma} + A_{\sigma \lambda \cdots \mu}^{\alpha i \cdots \gamma} \left\{ \begin{array}{c} \alpha \\ i \\ \nu \end{array} \right\} + A_{\sigma \lambda \cdots \mu}^{\beta i \cdots \gamma} \left\{ \begin{array}{c} \beta \\ i \\ \nu \end{array} \right\} + \cdots + A_{\sigma \lambda \cdots \mu}^{\gamma i \cdots \gamma} \left\{ \begin{array}{c} \gamma \\ i \\ \nu \end{array} \right\}.
\]

\[
-A_{\sigma \lambda \cdots \mu}^{\alpha \beta \cdots \gamma} \left\{ \begin{array}{c} i \\ \nu \end{array} \right\} - A_{\sigma \lambda \cdots \mu}^{\alpha i \cdots \gamma} \left\{ \begin{array}{c} i \\ \sigma \\ \nu \end{array} \right\} - A_{\sigma \lambda \cdots \mu}^{\beta i \cdots \gamma} \left\{ \begin{array}{c} i \\ \lambda \\ \nu \end{array} \right\} - \cdots - A_{\sigma \lambda \cdots \mu}^{\gamma i \cdots \gamma} \left\{ \begin{array}{c} i \\ \mu \\ \nu \end{array} \right\}
\]  \hspace{1cm} (A.2)
where the bracketed expressions are called Christoffel symbols of the second kind. These Christoffel symbols are defined in terms of the metric tensor $g_{\alpha\beta}$ as

$$\left\{ \begin{array}{c} \alpha \\ \beta \\ \gamma \end{array} \right\} \equiv \frac{g^{\alpha\lambda}}{2} \left( \frac{\partial g_{\beta\lambda}}{\partial x^\gamma} + \frac{\partial g_{\gamma\lambda}}{\partial x^\beta} - \frac{\partial g_{\beta\gamma}}{\partial x^\lambda} \right). \quad (A.3)$$

For Cartesian $(x, y, z)$ coordinates, Christoffel symbols of the second kind are all zero, and the covariant derivative simplifies to the regular partial derivative. In cylindrical $(r, \theta, z)$ coordinates, the only non-zero symbols are

$$\left\{ \begin{array}{c} \theta \\ r \\ \theta \end{array} \right\} = \left\{ \begin{array}{c} \theta \\ \theta \\ r \end{array} \right\} = \frac{1}{r}, \quad \text{and} \quad \left\{ \begin{array}{c} r \\ \theta \\ \theta \end{array} \right\} = -r.$$

In spherical $(r, \theta, \phi)$ coordinates, the non-zero Christoffel symbols of the second kind are

$$\left\{ \begin{array}{c} r \\ \theta \\ \theta \end{array} \right\} = -r \quad \left\{ \begin{array}{c} \theta \\ \phi \\ \phi \end{array} \right\} = -\sin \theta \cos \theta \quad \left\{ \begin{array}{c} r \\ \phi \\ \phi \end{array} \right\} = -r \sin^2 \theta$$

$$\left\{ \begin{array}{c} \theta \\ r \\ \theta \end{array} \right\} = \left\{ \begin{array}{c} \theta \\ \theta \\ r \end{array} \right\} = \left\{ \begin{array}{c} \phi \\ r \\ \phi \end{array} \right\} = \left\{ \begin{array}{c} \phi \\ \phi \\ r \end{array} \right\} = \frac{1}{r}$$

$$\left\{ \begin{array}{c} \phi \\ \theta \\ \phi \end{array} \right\} = \left\{ \begin{array}{c} \phi \\ \phi \\ \theta \end{array} \right\} = \cot \theta.$$

The use of Christoffel symbols in covariant differentiation ensures that $A_{\alpha\beta\gamma}^{\gamma}$ transforms according to the general tensor transformation law. Physically, covariant derivatives are required to account for the changes in local covariant (tangent) basis vectors ($\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_N$) as we move along coordinate curves. This is the physical interpretation of Christoffel symbols of the second kind. A contravariant derivative need not be defined since covariant and contravariant components can be interchanged by appropriate multiplication with the metric tensor $g_{\alpha\beta}$.

In general, for any orthogonal coordinate system, where $g_{\alpha\beta} = 0$ for $\alpha \neq \beta$, the Christoffel symbols of the second kind can be shown [14] to equal (where the summation convention does not apply):

$$\left\{ \begin{array}{c} \alpha \\ \beta \\ \gamma \end{array} \right\} = 0 \quad \left\{ \begin{array}{c} \alpha \\ \beta \\ \beta \end{array} \right\} = -\frac{1}{2g_{(\alpha\alpha)}} \frac{\partial g_{(\beta\beta)}}{\partial x^{(\alpha)}}$$

$$\left\{ \begin{array}{c} \alpha \\ \alpha \\ \beta \end{array} \right\} = \frac{\partial \ln \sqrt{g_{(\alpha\alpha)}}}{\partial x^{(\beta)}} \quad \left\{ \begin{array}{c} \alpha \\ \alpha \\ \alpha \end{array} \right\} = \frac{\partial \ln \sqrt{g_{(\alpha\alpha)}}}{\partial x^{(\alpha)}} \quad (A.4)$$

where $\alpha$, $\beta$, and $\gamma$ are not equal, and the brackets around indices remind us that in this instance, the summation convention is not in effect.