

Florida State University Libraries

Electronic Theses, Treatises and Dissertations

The Graduate School

A Novel Numerical Method for Resolving Micro-Structure Based on Supermesh for Multi-Material Systems

Yang Liu

FLORIDA STATE UNIVERSITY
COLLEGE OF ARTS AND SCIENCES

A NOVEL NUMERICAL METHOD FOR RESOLVING MICRO-STRUCTURE BASED ON
SUPERMESH FOR MULTI-MATERIAL SYSTEMS

By

YANG LIU

A Dissertation submitted to the
Department of Mathematics
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

2020

Yang Liu defended this dissertation on July 16, 2020.
The members of the supervisory committee were:

Mark Sussman
Professor Co-Directing Dissertation

M. Yousuff Hussaini
Professor Co-Directing Dissertation

Farrukh Alvi
University Representative

Kyle Gallivan
Committee Member

Nick Cogan
Committee Member

The Graduate School has verified and approved the above-named committee members, and certifies that the dissertation has been approved in accordance with university requirements.

To my mother, for your unconditionally love and support.

ACKNOWLEDGMENTS

First, I would like to offer my sincerely gratitude to my thesis advisors Professor Mark Sussman and Professor M. Yousuff Hussaini. My thanks to Dr Sussman for using his resourceful knowledge to interest and lead me to complete this research work which also makes my primary dream came true. My thanks to Dr Hussaini for his wise and inspirational mentoring which always reminds me to see from a general picture and how to be a better researcher. I would have not finished this work without their continued advice, assistance and support.

I would like to thank all the rest of my committee members, Professor Farrukh Alvi, Professor Kyle Gallivan and Professor Nick Cogan for their kind help and brilliant comments on the thesis.

Also special thanks to Professor Yongsheng Lian, Professor Kourosh Shoele for the pleasure of cooperation.

Finally, I would like to thank my friends, Dr. Xuan Ge, Dr. Mehdi Vahab, Peng Diao, Meng Wei, Dr. Haozhou Pu, Xuan Sheng, Dr Wenju Zhao for their help during my learning journey.

TABLE OF CONTENTS

List of Tables	vii
List of Figures	x
Abstract	xvi
1 Introduction	1
1.1 Multi-material Finite Volume Algorithm on Non-deforming Supermesh	6
1.2 The Stefan Problem	11
1.2.1 Our Intentions and Expectations for Solving the Stefan Problem	12
2 Problem Formulation and Equations	14
2.1 Multi-material Diffusion Problem	14
2.2 Multi-material Stefan Problem	15
2.2.1 Derivation of the Stefan Condition	15
3 Numerical Method	19
3.1 Multi-material Diffusion Solver	19
3.1.1 Centroid Initialization and MOF Reconstruction	20
3.1.2 Finite Volume Discretization for Space Derivatives and Backward Euler for Time	22
3.1.3 Flux Calculation	23
3.1.4 Calculation of Paired Area Fractions along Cell Faces	24
3.1.5 Calculation of Paired Area Fractions along Material Interfaces	25
3.1.6 “Connected Centroid” Approximation for Interfacial Fluxes	26
3.1.7 Numerical Approximation of Mixed Material Diffusivities	29
3.1.8 Justification for the Validity of the “Connected Centroid” Flux Algorithm	30
3.1.9 Linear Exact Algorithm for Dirichlet or Neumann Boundary Conditions	31
3.1.10 Solving the Linear System	33
3.2 Numerical Method on Stefan Problem	33
3.2.1 Gap Error Analysis	33
3.2.2 Solve the Interface Front Velocity	37
3.2.3 Solve the Advection Equation by Using Un-split Cell-Integrated Semi-Lagrangian Method	39
4 Numerical Results	42
4.1 Multi-material Diffusion	42
4.1.1 Error Calculation	42
4.1.2 Annulus Diffusion Test with Three Materials	45
4.1.3 Convergence Study for the Heat Flux through a Material Boundary	50
4.1.4 Penta-foil Test Problem Modified for Testing the Performance of Our FV-MOF Method in the Presence of Filaments	55
4.1.5 Hypocycloid-A Test with Six Materials	58

4.1.6	Example Where a Filamentary “Micro-scale Region” Effects the Macro-scale Solution	62
4.2	The Stefan Problem: Supermesh Method for Complex Deforming Boundaries.	71
4.2.1	Moving Planar Interface-Sanity Check	71
4.2.2	Shrinking Cylinder	71
4.2.3	Gingerbread Man Melting Test	72
4.2.4	The Unstable Solidification-Dendrite Growth Test	78
5	Conclusion	84
Appendix		
A	MOF boiling algorithm using Ghost Fluid method (NOT SUPERMESH)	87
A.1	Governing Equations	87
A.2	Material Advection	88
A.3	Momentum Advection	91
A.4	Energy Advection	96
A.5	Temperature Diffusion	96
A.6	Viscosity Term	97
A.7	Surface Tension Term	97
A.8	Projection Method	100
B	2D Geometry Intersection Algorithm	103
B.1	Algorithm	103
B.2	Tests	103
C	Our Micro-layer Model	105
D	Algorithm 3	108
Bibliography		109
Biographical Sketch		118

LIST OF TABLES

4.1	Problem set-up for annulus test	45
4.2	Convergent study, thick annulus ($\delta = 0.2$), homogeneous Neumann Boundary, $T_{stop} = 0.0125$, space and time refinement ratios are 2 and 4 respectively; “Linear Exact” method is used, $F^{\text{meas}} = 1$	48
4.3	Convergent study, thin annulus ($\delta = 0.01$), homogeneous Neumann Boundary, $T_{stop} = 0.0125$, space and time refinement ratios are both 2; “Linear Exact” method is used, $F^{\text{meas}} = 0.01$	48
4.4	Convergence study, thick annulus ($\delta = 0.2$), inhomogeneous Dirichlet Boundary, $T_{stop} = 0.0125$, space and time refinement ratios are 2 and 4 respectively; “Linear Exact” method is used, $F^{\text{meas}} = 1$	49
4.5	Convergence study, thin annulus ($\delta = 0.01$), inhomogenous Dirichlet Boundary, $T_{stop} = 0.0125$, space and time refinement ratios are both 2; “Linear Exact” method is used, $F^{\text{meas}} = 0.01$	50
4.6	Convergence study, thin annulus ($\delta = 0.01$), inhomogeneous Dirichlet Boundary, $T_{stop} = 0.0125$, space and time refinement ratios are both 2; “connected centroid” method is used, $F^{\text{meas}} = 0.01$	50
4.7	Convergence study, comparison with non-axisymmetric body fitted algorithm, inhomogenous Dirichlet Boundary, $T_{stop} = 0.004$, space and time refinement ratios are 2 and 4 respectively; “Linear Exact” method is used, $F^{\text{meas}} = 1$. The resolution of the body fitted algorithm is 128×256	53
4.8	Convergence study, comparison with non-axisymmetric body fitted algorithm, inhomogeneous Dirichlet Boundary, $T_{stop} = 0.004$, space and time refinement ratios are 2 and 4 respectively; “Linear Exact” method is used, $F^{\text{meas}} = 1$. The resolution of the body fitted algorithm is 256×512	53
4.9	Convergence study, material 2 temperature for the thick Penta-foil ($\xi = 0.2$), homogeneous Neumann Boundary, $T_{stop} = 0.0125$, space and time refinement ratios are both 2 respectively; “linear exact” method is used, $F^{\text{meas}} = 1.0$	56
4.10	Convergence study, material 2 for the thick Penta-foil ($\xi = 0.2$), inhomogeneous Dirichlet Boundary, $T_{stop} = 0.0125$, space and time refinement ratios are both 2 and 4 respectively; “linear exact” method is used, $F^{\text{meas}} = 1.0$	57
4.11	Convergence study, material 2 of the thin Penta-foil ($\xi = 0.05$), inhomogeneous Dirichlet Boundary, $T_{stop} = 0.0125$, space and time refinement ratios are both 2; “linear exact” method is used, $F^{\text{meas}} = 0.01$	57

4.12	Convergence study, Hypocycloid, two materials, convergence study for material 1 (inner material), 2000 connected markers, $k_1 = 0.1$, $k_2 = 10.0$, space and time refinement ratios are both 2 respectively; “simple” method is used, $F^{\text{meas}} = 1.0$	60
4.13	Convergence study, Hypocycloid, two materials, convergence study for material 1 (inner material), 4000 connected markers, $k_1 = 0.1$, $k_2 = 10.0$, space and time refinement ratios are both 2 respectively; “simple” method is used; $F^{\text{meas}} = 1.0$	60
4.14	Convergence study, Hypocycloid, six materials, convergence study for material 2 (the upright material of the inner region), 2000 connected markers, $k_1 = 10.0$, $k_2 = k_3 = k_4 = k_5 = k_6 = 0.1$, space and time refinement ratios are both 2 respectively; “simple” method is used, $F^{\text{meas}} = 1.0$	62
4.15	Convergence study, Hypocycloid, six materials, convergence study for material 2 (the upright material of the inner region), 2000 connected markers, $k_1 = 10.0$, $k_2 = k_4 = 0.5$, $k_3 = k_5 = 1.0$, $k_6 = 0.1$ space and time refinement ratios are both 2 respectively; “simple” method is used, $F^{\text{meas}} = 1.0$	62
4.16	Convergence study corresponding to the geometry illustrated in Figure 4.22; a thin filament with $\delta = 0.02$ separates a (smooth) circular shaped drop from the substrate. There are three materials and the convergence study is for material 2 (the drop). $k_{ratio} = 1000 : 1$, space and time refinement ratios are both 2, the “connected centroid” method (3.13) is used, $F^{\text{meas}} = 1.0$, and $T_{stop} = 2.0$ average flux: (16) – 0.131 (32) – 0.155 (64) – 0.177 (128) – 0.192 (256) – 0.197	66
4.17	Convergence study corresponding to the geometry illustrated in Figures 4.20 and 4.21; a thin filament with $\delta = 0.02$ separates irregular shaped drop from substrate. There are three materials and the convergence study is for material 2 (the drop). $k_{ratio} = 1000 : 1$, space and time refinement ratios are both 2, the “connected centroid” method (3.13) is used, $F^{\text{meas}} = 1.0$, and $T_{stop} = 2.0$ average flux: (16) – 0.121 (32) – 0.156 (64) – 0.177 (128) – 0.187 (256) – 0.191	67
4.18	Convergence study corresponding to the geometry illustrated in Figure 4.22; a thin filament with $\delta = 0.02$ separates smooth shaped drop from substrate. There are three materials and the convergence study is for material 2 (the drop). $k_{ratio} = 1000 : 1$, space and time refinement ratios are both 2, the “Dai and Scannapieco” [16] method (3.17) is used, $F^{\text{meas}} = 1.0$, and $T_{stop} = 2.0$ average flux: (16) – 0.131 (32) – 0.148 (64) – 0.172 (128) – 0.186 (256) – 0.195	67
4.19	Convergence study corresponding to the geometry illustrated in Figure 4.22; a thin filament with $\delta = 0.02$ separates smooth shaped drop from substrate. There are three materials and the convergence study is for material 2 (the drop). $k_{ratio} = 1000 : 1$, space and time refinement ratios are both 2, the “orthogonal projection” method (3.18) is used, $F^{\text{meas}} = 1.0$, and $T_{stop} = 2.0$ average flux: (16) – 0.137 (32) – 0.173 (64) – 0.185 (128) – 0.194 (256) – 0.199	67
4.20	Convergence study corresponding to the geometry illustrated in Figure 4.22; a thin filament with $\delta = 0.02$ separates (smooth) circular shaped drop from substrate. There are three materials and the convergence study is for material 2 (the drop). $k_{ratio} = 1 : 1$, space and time	

refinement ratios are both 2, the “connected centroid” method (3.13) is used, $F^{\text{meas}} = 1.0$, and $T_{\text{stop}} = 2.0$ average flux: (16) – 9.06 (32) – 9.31 (64) – 9.46 (128) – 9.54 (256) – 9.58	68
4.21 Convergence study corresponding to the geometry illustrated in Figures 4.20 and 4.21; a thin filament with $\delta = 0.02$ separates the irregular shaped drop from substrate. There are three materials and the convergence study is for material 2 (the drop). $k_{ratio} = 1 : 1$, space and time refinement ratios are both 2, the “connected centroid” method (3.13) is used, $F^{\text{meas}} = 1.0$, and $T_{\text{stop}} = 2.0$ average flux: (16) – 8.95 (32) – 9.73 (64) – 9.90 (128) – 9.99 (256) – 10.03	68
4.22 Convergence study of moving planar interface with the exact solution $x = 0.2$ at the end time $t = 1.0$	71
4.23 Convergence study of shrinking circle test.	72

LIST OF FIGURES

1.1 Nucleate boiling experiment results (left) from [88] and numerical simulation schematic (right) form [119]	1
1.2 Temperature contours and interfaces for nucleate boiling on a flat substrate. Left: $t = 0$, Right: $t = 0.142$. 128x256 grid.	2
1.3 Hydrate experiment results from [109]. Challenges: 1. Complex surficial mechanisms governing methane bubble hydrate formation and dissolution 2. Synergic action between bubble hydrodynamics, hydrate morphology	3
1.4 Heat pipe photo from Kundan et al [49] “Effect of capillary and Marangoni forces on transport phenomena in microgravity.”	3
1.5 From Kundan et al [49]. Left: expanded view of the interfacial flow region and the forces governing this flow. Right: Sketch of the streamlines in or near the central and the “pinch point” where Marangoni flow first touches the capillary flow	4
1.6 The example of simulating heat transfer and friction generated heat for splashing of lubricant against interlocking gears. Left figure from [115]	5
1.7 Material processing illustration from [23], Schematics of a coaxial laser metal deposition system with powder injection.	5
1.8 Filament types: (A) straight filaments and (B) triple point with filament both are “isolated” case; (C) Represent flows like bubbly flow and mist flow is the “non-isolated” case.	6
1.9 Comparison of the GFM, our supermesh method and FVM discretization on a region with a filament using a 2D example. Since all the level-set functions stored at cell centers (red dots) do not change sign for material 2, the filamentary region will not be captured by the ghost fluid method without AMR; The filament can be recognized by the FVM since the volume fraction of material 2 will not be zero; albeit the cell centered placement of unknowns will lead to poor matrix conditioning in comparison to our approach in which unknowns are stored at the centroids (black dots in the figure) (Dai& Scannapieco[16])	7
1.10 Illustration of “subdivision process” in Garimella et. al.[27] with a 2D example. Blue dot is the cell center and purple dots are centroids for different material regions. For the center cell, (a) Extend the reconstructed interfaces (thick blue line) from neighboring cells into the center target cell as red dotted lines. (b) Triangulate (using blue dotted line to represent this procedure) any polygon (polyhedron in 3D) region that is not triangle (tetrahedron in 3D). (c) A conforming triangular (tetrahedral) mesh is formed. Garimella et. al.’s method need 25 degrees of freedom (25 small triangular regions) while our method only need 3 (one for each material).	8

1.11	Example of a multi-material diffusion problem with MOF piecewise linear interface reconstruction. We use the same terminology as in [44] in which we call the light blue background rectangular mesh of the left most figure a “base mesh,” the partitioning of a single rectangular cell into material polygons a “minimesh,” and the union of each “minimesh,” derived from all of the cells on a base mesh, is called the “supermesh.” In contrast to the “basemesh,” the “supermesh” recognizes subgrid-scale filamentary regions. Note that our supermesh recognizes interface gaps at cell faces. This is similar to [16], but in contrast to [18].	11
2.1	Example of problem formulation, note: x_m is the centroid of region Ω_m occupied by material m , $m = 1, 2, 3$ in the figure. Assume the normal direction \mathbf{n} of a cell face is always point outward.	14
2.2	Stefan condition derivation using weak formulation. Note in the figure, $\Gamma \cup \Gamma_1 = \partial\Omega_1$, $\Gamma \cup \Gamma_2 = \partial\Omega_2$, $\Gamma \cap \Gamma_1 = \phi$, $\Gamma \cap \Gamma_2 = \phi$	15
3.1	Flow chart corresponding to our multi-material diffusion simulation algorithm.	19
3.2	Example of MOF reconstruction in two materials cell. Remark: the normal of the reconstructed interface is always point into the target material in this paper. For example in the figure, the normal n^1 points into material 1.	21
3.3	Illustration of MOF reconstruction with materials more than 3 using nested dissection algorithm Material 3 is the first material reconstructed. Material 2 is the second material reconstructed.	22
3.4	Flux integral calculation for material 3 along its boundary in the center cell: We calculate the flux through each subsection of the material 3 boundary. Each subsection has a unique material pair on its two sides. Note: the red normals n_1, n_2, n_3, n_4, n_5 are the ones used for calculating the heat flux integral, the blue normals $n^{(3)}, n^{(2)}$ are the normals of the MOF linear reconstructed interface.	23
3.5	From previous section 3.1.4: interface projections are shown in (a) and thin material slabs formation are shown in (b)	24
3.6	Pairing algorithm: In Figure (c), we subtract the blue shaded region from the right. After subtracting left material 1 region from the right, intuitively, as example shown in (d), $pair_{1,2} =$ right volume 2 before cut – right volume 2 after cut $pair_{1,1} =$ right volume 1 before cut – right volume 1 after cut	25
3.7	Example for illustrating the steps for calculating the area fractions along an interface	26
3.8	Example: “connected centroid” method for approximating the flux through a segment of a face of a cell or an interface.	28
3.9	Connected Centroid method for approximating the flux. The discretized material m temperature θ_m is stored at the material m centroid \mathbf{x}_m	28

3.10	Example: Dai and Scannapieco's method for approximating the flux through a segmentation of a face of a cell or an interface.	28
3.11	Example: orthogonal projection method for approximating the flux through a segmentation of a face of a cell or an interface. (see also Figure 3 of Dawes (2017) [18])	29
3.12	2D Linear exact algorithm example	33
3.13	Flow chart corresponding to our Stefan problem algorithm.	34
3.14	Example for an “area gap error”. The real interface in the cell is an arc with radius R, the centroid of material 2 region is the purple point; the reconstructed interface is the horizontal blue line coincide with x axis centered at (0, 0) dividing the cell into half. The gap error is indicated in the figure.	35
3.15	Phase change velocity calculation: find corresponding interpolation support domain and calculate the temperature gradient jump across the phase change front. Left: support domain for the vapor phase; Right: support domain for the liquid phase	38
3.16	Phase change velocity near filament region Since the macro probe, $x_{probe,2}$ is in material 3, we use the micro probe (immediately next to the red x) that is in material 2.	38
3.17	Phase change velocity calculation: velocity extension in cells ($2\Delta x < \phi_{liq} < 4\Delta x$ or $2\Delta x < \phi_{vap} < 4\Delta x$) and solid cells. Left: $x_{i,j}$ in vapor occupied cell within $4\Delta x$. Right: $x_{i,j}$ in solid cell	39
3.18	Illustration of 2D un-split Cell-Integrated Semi-Lagrangian (CISL) method: Left: Backward project (characteristic tracing of the value at the four vertices according to the velocity) the center cell $\Omega_{i,j}^{target}$ at time T^{k+1} onto the domain at time T^k Right: Show the intersection of P_1^{depart} with the T^k domain	40
3.19	Volume fraction correction after advection in > 2 material cells	41
4.1	Interface gap error: E_{gap} . Find the centroid of each small gap and substitute into the exact signed distance function then calculate the error.	42
4.2	Example for gradient error calculation	44
4.3	Left: a figure indicating interfaces and material ID. Right: MOF reconstruction and centroids for different materials of thick annulus ($\delta = 0.2$ and then $r_1 = 0.15, r_2 = 0.35$) mesh size 64×64	46
4.4	MOF reconstruction and centroids for different materials of thin annulus ($\delta = 0.01$ and then $r_1 = 0.245, r_2 = 0.255$). mesh size 64×64	47
4.5	Temperature Diffusion in the thick and thin annulus at time $T = 0.0125$. The embedded boundary conditions for the annular region are homogeneous Neumann. The grid resolution	

is 256×256 . The contour plot is “blanked out” in the regions where the material two volume fractions are zero.	47
4.6 Loglog plot referring to Table 4.2	48
4.7 Log log plot referring to Table 4.4	49
4.8 Illustration of normal probe algorithm for deriving the interfacial heat flux from the FV-MOF temperature field approximation mesh size 64×64	52
4.9 $\theta_1 = 2.0, \theta_2 = 3.0$, temperature diffusion at time 0.004, grid resolution 256×256	52
4.10 Loglog plot referring to Table 4.7	53
4.11 Log log plot referring to Table 4.8	54
4.12 Example for illustrating Interpolation Method for calculating temperature at normal probe. x_I is the closest point on the interface to the point x . The green box (containing 3×3 cells) is the region under consideration for normal probe x_{probe} (purple diamond).	54
4.13 Left: MOF reconstruction and centroids. Mesh size: 64×64 Right: A zoom in look at part of the left figure	55
4.14 Diffusion in Penta-foil shape at $T = 0.0125$. The initial conditions for the middle material (material 2) is given by (4.19). The embedded boundary conditions are Dirichlet for the result displayed here and the grid resolution is 256×256 . The contour plot is “blanked out” in the regions where the material two volume fractions are zero. Left: $\xi = 0.05$. Right: $\xi = 0.2$.	56
4.15 Loglog plot referring to Table 4.10	57
4.16 Left: MOF reconstruction and centroids. Mesh size: 64×64 Right: A zoom in look at part of the left figure	58
4.17 Hypocycloid, 6 materials, 2000 markers used to derive the signed distance function which, in turn, is used to derive the volume fractions and centroids. Left: MOF reconstruction and centroids. Mesh size: 64×64 Right: A zoom in look at part of the left figure Note: without losing generality, we set the center of the graph to be at the center of the whole domain which means the interfaces separating material 2 and 3, 3 and 4, 4 and 5, 5 and 2 from each other are aligned with the grid line.	59
4.18 Evolution of Hypocycloid 2 materials diffusion problem, mesh size 512×512 , $k_1 = 0.1$, $k_2 = 10.0$	61
4.19 Evolution of Hypocycloid, 6 materials type (iii) diffusion problem, mesh size 512×512 , $k_1 = 10.0, k_2 = k_4 = 0.5, k_3 = k_5 = 1.0, k_6 = 0.1$	63
4.20 Left: Problem geometry set-up with three materials defined in terms of four distance functions $\phi_i, i = 1, \dots, 4$. A filamentary region (filled with material 3) exists with thickness $\delta = 0.02$.	

Right: test for the average flux integral across line segment $0.375 < x < 0.625$ $y = 0.40625$ over the whole running time	64
4.21 MOF reconstruction and centroids for sharp cornered material 2. Mesh size 16×16	65
4.22 Left: Problem geometry set-up with three materials. Material 2 is a (smooth) circle. A filamentary region (filled with material 3) exists with thickness $\delta = 0.02$. The average flux integral test looks at the line $0.375 < x < 0.625$ $y = 0.59375$ over the whole running time. Right: MOF reconstruction and centroids; Mesh size 16×16	65
4.23 Comparison of thermal conductivity ratio 1:1 (left column) versus 1000 : 1 (right column); material two is a (smooth) circle. $\Delta T = 0.03125$, 256×256 mesh	69
4.24 Comparison of thermal conductivity ratio 1:1 (left column) versus 1000 : 1 (right column); material two has corners. $\Delta T = 0.03125$, 256×256 mesh	70
4.25 Convergence study of shrinking circle at $Time = 6.25E - 4$, expected radius= 0.3868. Left: interfaces for all the resolutions overlaid on top of each other red 64×64 , green 128×128 , blue 256×256 , black 512×512 ; Right: zoom-in look of the left figure.	73
4.26 Example for constructing the local level set function for each linear section.	74
4.27 Example for determining whether a point is "inside" or "outside" the element e_k . Note, the normal projection of the point x' onto element e_k is outside the element. x' would be considered as "inside" when looking at e_k and e_{k+1} , but outside when looking at e_{k+2} . $\mathbf{n}_{k,k+1} = (\mathbf{n}_k + \mathbf{n}_{k+1})/2$	74
4.28 Example of "uninitialized" cell (centered at red dots) next to initialized cell (centered at blue dot).	75
4.29 Interface reconstruction for gingerbread man melting test.	76
4.30 Left: Temperature field in sometime during the computation; Right: Interface evolution from red $Time = 0.0$ to green $Time = 0.02$	76
4.31 Comparison for different resolution at $Time = 0.02$. Red: 64×64 , 45 steps. Green: 128×128 , 91 steps. Blue: 256×256 , 159 steps.	77
4.32 Convergence observation for different resolution at $Time = 0.02$. Red: 64×64 , 50 steps. Green: 128×128 , 100 steps. Blue: 256×256 , 200 steps.	78
4.33 Gingerbread man evolution with area gap error analysis. The yellow boxes show where the area gap error exceeds the threshold of a radius of curvature less than Δx	79
4.34 Initial seed of dendrite growth test	80
4.35 Dendrite growth evolution at different time steps for resolution 128×128 at time $Time = 0.130, 0.367, 0.611, 0.800$	81

4.36	Comparison of dendrite growth without surface tension restriction (with regular Dirichlet boundary) with different resolution at $Time = 0.8$. $\epsilon_1 = \epsilon_2 = 0$. Red: 64×64 , Green: 128×128 . Blue: 256×256	82
4.37	Comparison of dendrite growth with surface tension restriction with different resolution at $Time = 0.464$ with $\epsilon_1 = 0.002, \epsilon_2 = 0.002$. Red: 64×64 , 123 Steps; Green: 128×128 , 267 steps; Blue: 256×256 , 651 steps.	82
4.38	Comparison of dendrite growth with surface tension restriction with different resolution at $Time = 0.32$ after 640 steps with $\epsilon_1 = 0.002, \epsilon_2 = 0.002, \Delta t = 1/2000$. Red: 64×64 ; Green: 128×128 ; Blue: 256×256	83
A.1	Material advection for material 1(yellow material) in target region $\Omega_{i,j}$	90
A.2	Momentum advection of $u_{i-1/2,j}$ on MAC grid in x-direction: backward characteristic tracing	92
A.3	Momentum advection of $u_{i-1/2,j}$ on MAC grid in x-direction for material 1 (yellow material): intersection and mapping	92
A.4	Momentum advection of $u_{i-1/2,j}$ on MAC grid in y-direction: backward characteristic tracing	95
A.5	Momentum advection of $u_{i-1/2,j}$ on MAC grid in y-direction for material 1 (yellow material): intersection and mapping	95
A.6	Find the crossing time and modify the temporal discretization	97
A.7	Height function orientation, $ a < b $, vertical	98
A.8	Height function orientation, $ a > b $, horizontal	98
A.9	Curvature interpolation in two material cells	99
A.10	Curvature interpolation for material number ≥ 3	99
A.11	Case1: two fluids with one solid material	100
A.12	Case2: three fluids with "Y-junction"	100
C.1	The microcell treatment is used whenever the solid distance function satisfies $\phi_{\text{substrate}}(\vec{x}_{\text{closest}}) > -h_{mac}$ where h_{mac} is a macroscale length that is independent of the mesh size Δx . \vec{x}_{closest} is the closest point on the vapor liquid interface to \vec{x}	107

ABSTRACT

A new static and deforming boundary supermesh numerical method is developed for the solution of the diffusion problem and the Stefan problem in a system of several materials. In such a system, the diffusion coefficients are piecewise continuous and jumps in their values can occur across the complex-shaped interfaces between contiguous materials. The boundary conditions along the complex-shaped interfaces can either be a jump condition boundary condition, a Neumann boundary condition, or a Dirichlet boundary condition. The moment-of-fluid (MOF) procedure is employed to reconstruct the interfaces. This procedure enables accurate reconstruction of any number of material interfaces in a computational cell. Furthermore, MOF is a volume preserving reconstruction, as well as capable of capturing thin filamentary regions without the necessity of adaptive mesh refinement. The proposed method is tested on multi-material diffusion problems which demonstrates its potential to enable numerical simulation of complex flows of technological importance relevant to predicting the heat transfer rate in materials and manufacturing processes. Results using the new method are reported on problems in complex (filamentary) domains, and it is found that the method is very efficient at approximating the temperature, the temperature gradient, and the interfacial heat flux, as compared to traditional approaches.

When the supermesh deforming boundary method is applied to computing solutions to the Stefan problem, the interface propagation equation is discretized using the unsplit cell-integrated semi-Lagrangian method. The level set method is also coupled during this process in order to assist in the initialization of the (transient) provisional velocity field. My method is validated on both canonical and challenging benchmark tests. Algorithm convergence results based on grid refinement are reported. It is found that the new method approximates solutions to the Stefan problem efficiently, compared to traditional approaches, due to the localized finite volume approximation stencil derived from the underlying supermesh. The new kind of supermesh approach provides a general framework for solving many complex deforming boundary problems containing inherent deforming micro-structural components.

CHAPTER 1

INTRODUCTION

There are many multi-material or multi-phase problems involving micro-structure in the real world, for example, classical phase change problem like nucleate boiling [88, 108, 83, 119, 9], evaporation[96, 102, 43], melting [32], freezing [64, 107] and also research related to surfactants [59, 63, 69, 98], lofted fire brands [46], cavitation flow [5, 6, 82, 106], turbulence boundary layer modeling [26, 50, 51], etc.

The left figure of Figure 1.1 shows a series of experimental results of a single bubble growing on a heated surface. The right schematic in 1.1 shows the typical way to model this problem. The following sample

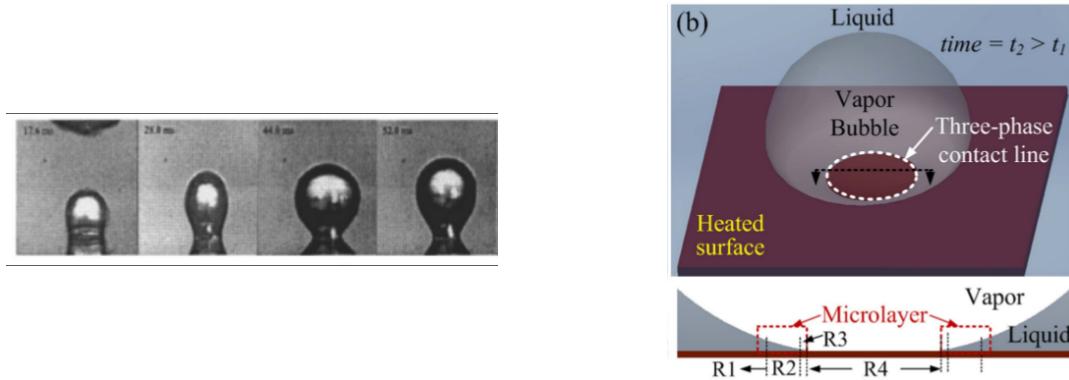


Figure 1.1: Nucleate boiling experiment results (left) from [88] and numerical simulation schematic (right) form [119]

nucleate boiling numerical results in Figure 1.2 are computed by our legacy code (algorithm referring to A.1) https://github.com/msussman42/amrex_implicit_interfaces. The legacy code is “pre” “deforming super mesh”. One of the computation difficulties is the region near the “triple point” where three phases co-exist. The key is to resolve the thermal layer (the thin liquid filament between vapor and substrate near the triple point) to obtain the accurate heat transfer rate (see [88, 108, 83, 119, 9]).

My research aim is to develop an algorithm to solve multi-scale phase change problems upon a system comprising several materials and containing complex deforming boundaries (including thin filaments) consisting of micro-structural regions on which being resolved by AMR is out of reach (See Figure 4.33 in the results section). My new deforming boundary supermesh approach will enable one to simulate general

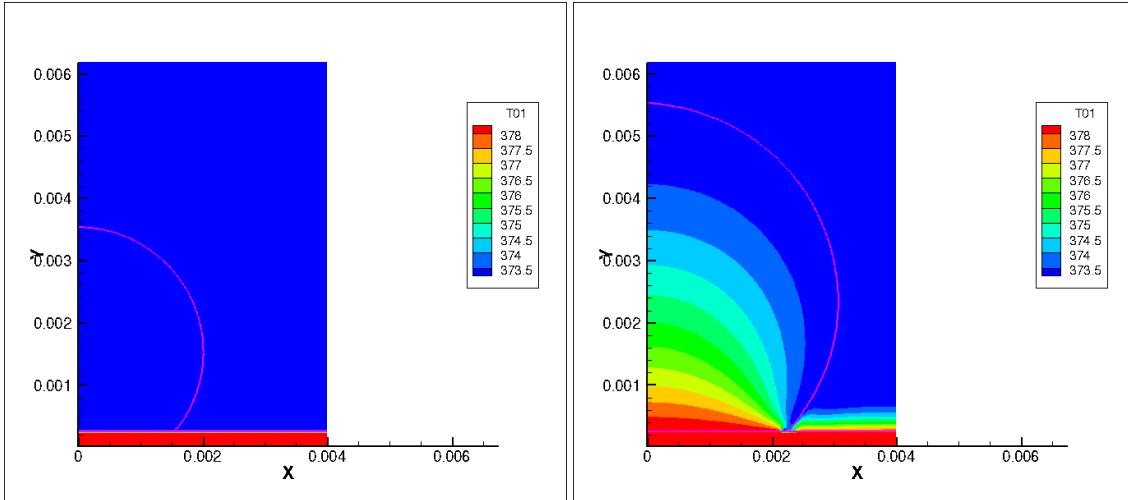


Figure 1.2: Temperature contours and interfaces for nucleate boiling on a flat substrate. Left: $t = 0$, Right: $t = 0.142$. 128x256 grid.

micro-structural influenced problems without the need for developing ad-hoc problem specific micro-layer models (as in [15, 73, 101, 110, 89, 88, 83, 9, 99, 39, 108, 119, 35]).

Another example for developing the deforming supermesh method is to simulate the hydrate problem [14, 85, 109]. Natural gas hydrates are solid crystalline which are formed under specific condition. They appear as solid coatings trapping the natural gas bubble which has escaped from the sea bed. Simulating the formation and dissolution of the hydrate is a challenge project due to the extremely complex surficial mechanisms of the thin hydrate layer. Figure [109] here depicts the experiment results that were obtained from observations of a single methane bubble in the high-pressure water tunnel facility at the National Energy Tech Lab.

Another potential application of great significance of the deforming supermesh method is the research in designing heat pipes (see references [89, 65, 36, 114, 62, 86, 17, 22, 74, 49, 48]). Heat pipes are passive heat transfer devices that can transfer heat over long distances with small temperature drops, without a requirement of external pumping power. They are light weight, reliable and require minimum maintenance when used in places where high heat fluxes are required. They are especially attractive in microgravity environments where the low Bond number allows for significant heat transfer [49, 48]. The Constrained Vapor Bubble (CVB) experiment [49, 48] concerns a transparent, simple, wickless heat pipe operated in the microgravity environment of the International Space Station (ISS) is illustrated in Figure 1.4.

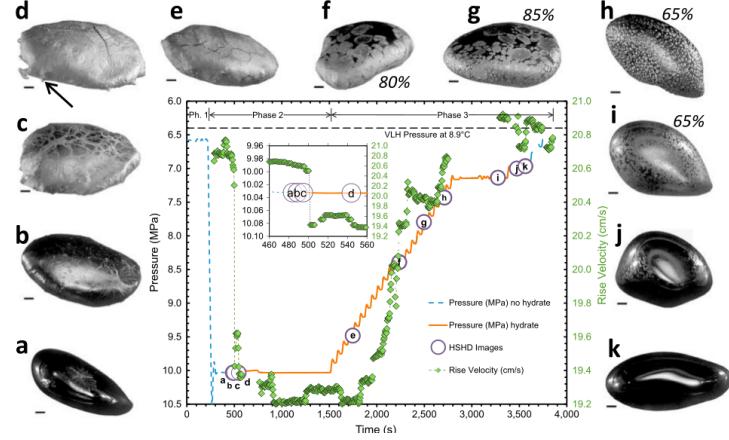


Figure 1.3: Hydrate experiment results from [109].

- Challenges:
1. Complex surficial mechanisms governing methane bubble hydrate formation and dissolution
 2. Synergic action between bubble hydrodynamics, hydrate morphology

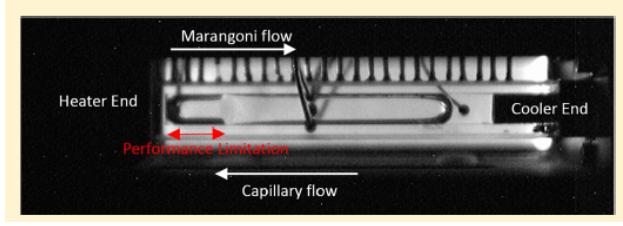


Figure 1.4: Heat pipe photo from Kundan et al [49] “Effect of capillary and Marangoni forces on transport phenomena in microgravity.”

The basic working principle of a heat pipe is dependent on the capillary pumping of the fluid from the cold end to the hot end using a wick or a wickless design. Liquid is evaporated at the heated end; the vapor then travels back to the cold end where it condenses and is then recirculated back to the hot end by the capillary pumping.

However, when taking into consideration of the Marangoni flow which is a surface tension driven flow caused either by a temperature or a concentration gradient, at high enough temperature gradients, the working fluid should be subjected to enough Marangoni force to force it away from the heater and ultimately, dry out the hot end. Yet, in Kundan et. al.’s CVB experiment [49, 48], a more complicated phenomenon in which opposing Marangoni and capillary forces lead to flooding of the device, as observed during the experiments [49, 48]. One can observe from Figures 1.4 and 1.5, that a thin liquid film exists between vapor bubble and the container. The challenge here in micro-gravity is being able to simulate the formation of the

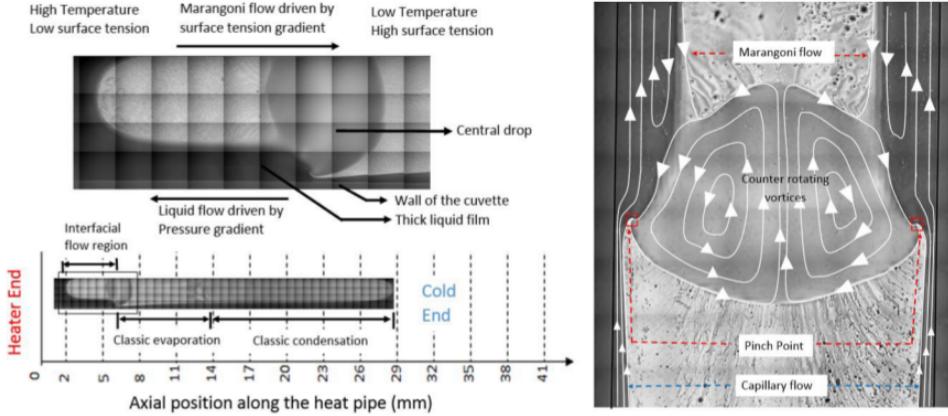


Figure 1.5: From Kundan et al [49].

Left: expanded view of the interfacial flow region and the forces governing this flow.
 Right: Sketch of the streamlines in or near the central and the “pinch point” where Marangoni flow first touches the capillary flow

mysterious “central drop” which is hypothesized to form due to the counter current of Marangoni flow and capillary flow; counter-intuitively the hot end floods.

The method we have developed is aimed at solving multi-scale problems. Consider, for example, the realms of research in the area of macro-scale fluid-mechanics and micro-scale contact modeling. The interaction of the two, from a DNS perspective, has not been investigated due, in large part, to the immense separation of length and time-scales that exist between the two as illustrated in Yazdani et.al.[115]. The simulation of heat transfer and friction generated heat for splashing of lubricant against interlocking gears (see Figure 1.6) is a good possible application of our method. Perhaps in the future, with supermesh grid, one can place custom models in deforming micro-structure regions.

Moreover, we believe our algorithm might be beneficial in simulating modern materials processing applications, for example, additive manufacturing. Additive manufacturing [23] can be achieved by powder-based spray (e.g., thermal spray [76, 1] or cold spray [91, 10]), sintering (e.g., selective laser sintering [47]), or fusion-based processes (direct metal deposition [45, 23]) which use an energy source (a laser beam, an electron beam, a plasma beam, or an electric arc) and either metallic powder or wire as feedstock. Numerical simulating of an additive manufacturing process is challenging due to the following two reasons: (1) Additive manufacturing processes involve a large number of process parameters, including total power and power intensity distribution of the energy source, travel speed, translation path, material feed rate and shielding gas pressure. These process parameters not only vary from part to part, but also frequently vary locally within

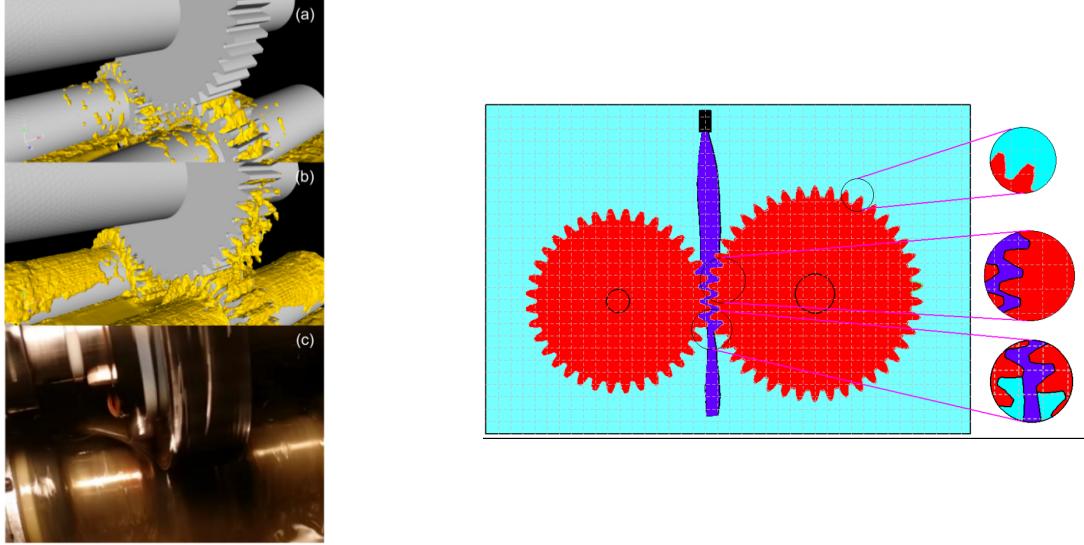


Figure 1.6: The example of simulating heat transfer and friction generated heat for splashing of lubricant against interlocking gears. Left figure from [115]

a single part to attain the desired deposit shape. (2) Additive manufacturing processes are an amalgamate of many complex physical phenomenon including melting/solidification and vaporization phase changes, surface tension-dominated free-surface flow, heat and mass transfer, and moving heat source. Delicate phase change process occurs on those thin material layers and it's very difficult to predict experimentally where defects appear in the printed structures [116]. Numerical modeling and simulation is a very important and useful tool for assessing the impact of process parameters and predicting optimized conditions in additive manufacturing processes.

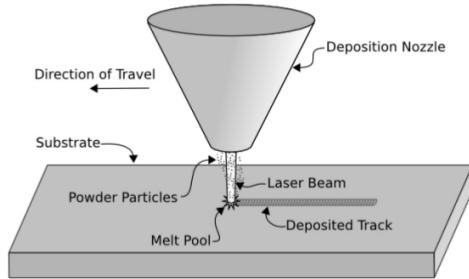


Figure 1.7: Material processing illustration from [23],
Schematics of a coaxial laser metal deposition system with powder injection.

For fluid dynamic problems in which there is a large separation between the “macro” scale and the

“micro” scale, and the scales interact (see Figures 1.1,1.3,1.5), the Adaptive Mesh Refinement technique is not a realistic option. We define “large” separation of scales as $\frac{l_{macro}}{l_{micro}} > 1000$. In this thesis we present a new supermesh algorithm which is a general approach for multi-scale problems when the multi-scale phenomena is “isolated” (see Figure 1.8). Existing approaches, “pre” supermesh, are very problem specific e.g.[88, 83, 9].

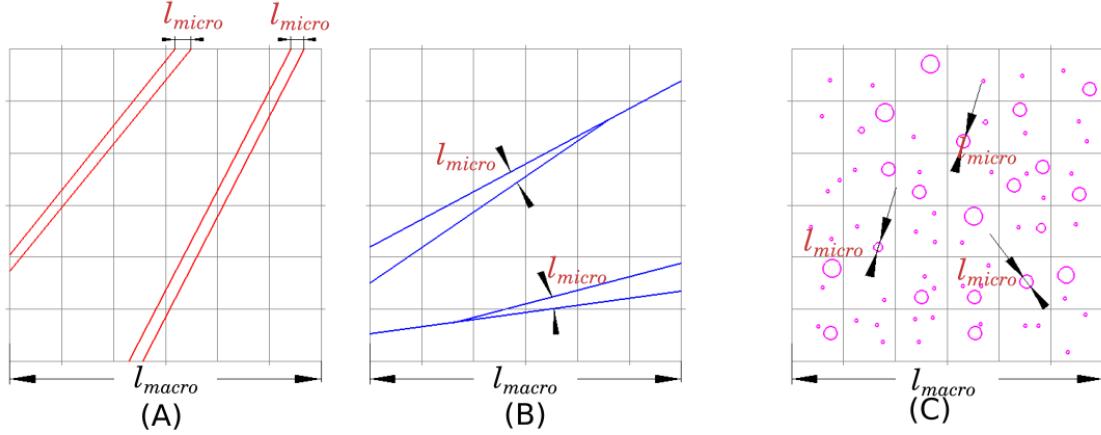


Figure 1.8: Filament types: (A) straight filaments and (B) triple point with filament both are “isolated” case;
(C) Represent flows like bubbly flow and mist flow is the “non-isolated” case.

For the “non-isolated” multi-scale problem, our “supermesh” approach will not be a viable option, in this scenario, researchers have developed particle methods [33] or mixture models [61].

In sum, our intention is to develop our method in a manner that it’s able to solve the multi-scale diffusion problem in multiple material systems with complex deforming boundaries without the aid of AMR.

1.1 Multi-material Finite Volume Algorithm on Non-deforming Supermesh

In a mathematical framework (that does not consider phase change), the heat transfer problem is represented by a parabolic (or elliptic for steady state) equation with a discontinuous coefficient yielding a solution involving discontinuous gradients across interfaces. There have been several numerical approaches [55, 38, 57, 31, 4, 27, 34, 16, 18, 44] proposed for this problem.

For solving elliptic or parabolic problems on embedded domains with discontinuities at the interfaces, the following classical methods have been widely used. Leveque and Li (1994) [55] [80] developed the second order immersed interface finite difference method for solving elliptic equations with discontinuous

diffusion coefficients. In Leveque and Li's work, interfaces, on which coefficients have jumps, are immersed within an uniform rectangular grid, (i.e. the grid is not aligned with the coefficient discontinuities). Leveque and Li's strategy in order to overcome the non-alignment issue is to introduce a locally aligned grid. Johansen and Colella (1998) [38] developed a second order finite volume method on a rectangular Cartesian grid for approximating solutions to the Poisson equation on an irregular embedded domain with Dirichlet boundary conditions. In their work, the multigrid method was implemented in order to solve the resulting large sparse matrix system. Liu and Fedkiw (2000) [57] and Gibou et al. (2002) [31] developed a finite difference Ghost Fluid Method for solving the variable density Poisson equation with a jump discontinuity in the coefficient at interfaces. In their work, the level set method (LSM) was used to represent interfaces. Chessa et al. [13] developed an “enriched finite element” method for simulating the multi-dimensional Stefan problem.

These classical methods need Adaptive Mesh Refinement (AMR) (e.g. block structured AMR [38], unstructured quad-tree/oct-tree AMR [75], unstructured anisotropic AMR[42], locally refined thermal grid[9]) in order to recognize the presence of a filamentary region. Without AMR, the activity in the filamentary region will be completely lost (for example in Figure 1.9). However, applying AMR means that more computational time and storage is required. A key feature of our method is being able to recognize thin filaments without the need of AMR.

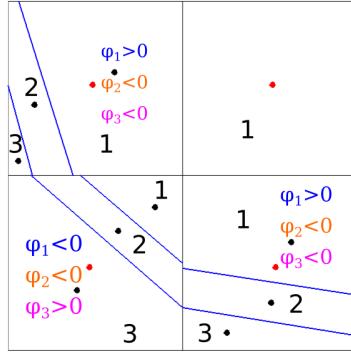


Figure 1.9: Comparison of the GFM, our supermesh method and FVM discretization on a region with a filament using a 2D example.

Since all the level-set functions stored at cell centers (red dots) do not change sign for material 2, the filamentary region will not be captured by the ghost fluid method without AMR;

The filament can be recognized by the FVM since the volume fraction of material 2 will not be zero; albeit the cell centered placement of unknowns will lead to poor matrix conditioning in comparison to our approach in which unknowns are stored at the centroids (black dots in the figure) (Dai& Scannapieco[16])

Focusing on solving multi-material diffusion problems (with possible thin or other fine scale features unrecognized by the standard Eulerian grid cut cell approaches) and preserving the energy, the following papers have made more sophisticated designs and modifications than classical numerical methods. Garimella and Lipnikov (2010) [27] developed a second-order method which subdivides the mesh along the reconstructed interface planes (“a supermesh approach”) into conforming tetrahedral (triangular in two dimensions) elements and then solves the diffusion equation on the new unstructured tetrahedral/triangular mesh using the mimetic finite difference (MFD) method. The generation of a conforming tetrahedral/triangular mesh by the subdivision process[27] results in a more accurate solution, however, the number of additional degrees of freedom increases rapidly for each subdivision (see Figure 1.10), especially in 3D. Our method (referring to

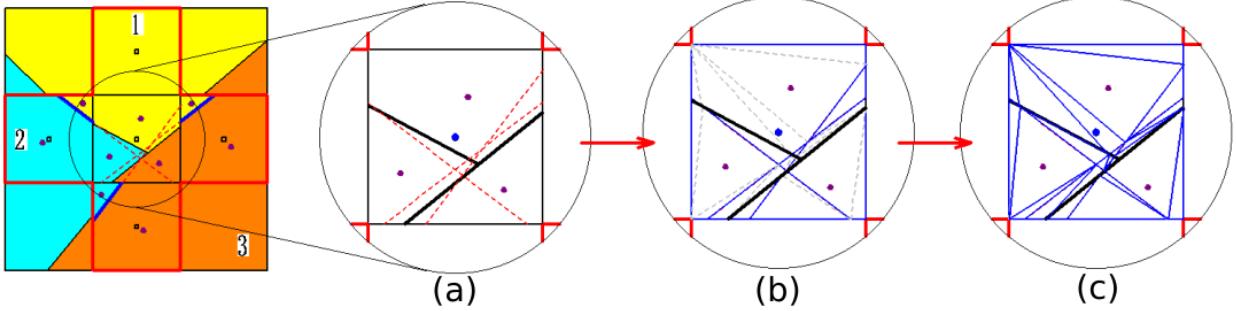


Figure 1.10: Illustration of “subdivision process” in Gariemlla et. al.[27] with a 2D example.

Blue dot is the cell center and purple dots are centroids for different material regions.

For the center cell, (a) Extend the reconstructed interfaces (thick blue line) from neighboring cells into the center target cell as red dotted lines. (b) Triangulate (using blue dotted line to represent this procedure) any polygon (polyhedron in 3D) region that is not triangle (tetrahedron in 3D). (c) A conforming triangular (tetrahedral) mesh is formed.

Garimella et. al.’s method need 25 degrees of freedom (25 small triangular regions) while our method only need 3 (one for each material).

Figure 3.4) requires only one degree of freedom for each material in order to achieve second order accuracy, does not require subdivision, is fairly easy to implement and extends to 3D, solely based on the information from the MOF reconstruction. Referring to Figure 1.10, Garimella et. al.’s method needs 25 degrees of freedom while our method only needs 3. The Volume-of-fluid (VOF) method was applied to construct the interfaces in Garimella & Lipnikov [27], which has limitations when compared with the Moment-of-fluid (MOF) method [20]; our algorithm in this thesis uses MOF. Multi-material interfaces are represented by the piecewise linear moment-of-fluid (MOF) reconstruction[20, 37]. We choose the MOF reconstruction algorithm because of the following important properties: (i) the MOF reconstruction algorithm relies on

information local to a given Eulerian cell. The MOF reconstruction does not need neighbor cell information (see Figure 1.11). Fluid parcels much smaller than a single grid cell, can be robustly represented using MOF method. (ii) the MOF algorithm is exact if the corresponding exact interface is linear. (iii) the MOF reconstruction preserves volume. (iv) the MOF reconstruction is second order accurate for smooth interfaces referring to Equation 4.1 ; we refer the reader to Section 4.1.1.1 and Table 4.20 where we define and report the level set “Gap error” respectively. (v) the MOF reconstruction algorithm recognizes micro-structural filamentary regions and enables one to robustly subdivide filamentary regions into thinner, also recognizable, filaments in order to selectively increase resolution in an interface normal direction[37]. Remark: the initial guess for the slope in our MOF procedure is determined using the “CLSVOF” slope [78, 95].

In order to overcome some of the complexity drawbacks of [27], Dai and Scannapieco (2014) [16] created a finite volume method using second-order differencing (second order in the bulk regions) to solve elliptic and parabolic equations over an unstructured polyhedral “supermesh”. The supermesh is generated by a gradient-based piecewise linear interface reconstruction; e.g. VOF or MOF. We illustrate the “supermesh” in Figure 1.11. Also, we refer the reader to [44] wherein the “supermesh” terminology was introduced. An additional half-time step is introduced by [16] in order to increase the accuracy in time. A related method is proposed by Dawes (2017) [18]. Both VOF and MOF produce disjoint interfaces at cell faces, illustrated in Figure 1.11 on the right. Dawes chose to remove the gaps, if possible, by locally adjusting the inter-cell connections to produce a surrogate supermesh. Dawes’ approach reduces the combinatorial overhead and complexity of the connections in a supermesh, particularly in 3D. (In our algorithm, we keep the gaps, see Figure 1.11 or Figure 4.1 for example). Kikinzon et al. (2017) [44] [117] developed a mimetic finite difference method to solve the multi-material diffusion-type problem on an unstructured grid. Within a given multi-material cell, materials are represented by material polygons (called the “minimesh” by [44]) embedded within a given cell, and each polygon contains only one material. In order to calculate the flux through each side of a material polygon, Kikinzon et al. assigned a different constant temperature corresponding to each face of a multi-material cell. The sides of different material polygons, that are located on the same face of a given multi-material cell, will share the same temperature. Once they defined a large sparse symmetric system of equations in terms of the unknown face temperatures (one temperature per face, not per material), they solved the resulting symmetric positive definite system for the face temperatures and then implemented a local backward substitution procedure to calculate the new temperature at each centroid of a multi-material cell.

Finally, we mention two promising approaches which have not yet been applied directly to the multi-material diffusion problem with complex embedded domains. The first approach is the Voronoi interface method[34]. Guittet et al. (2015) employ the Voronoi Interface Method to solve elliptic problems with discontinuities. They use the Level Set Method (LSM) in order to represent the interfaces at which jumps can occur. The feature that distinguishes the Voronoi Interface Method from [57] and [31] is the addition of extra degrees of freedom in the vicinity of the interface (represented by the level-set). They use the Voronoi diagram to construct a local mesh where the flux between two given degrees of freedom is guaranteed to be orthogonal to the face which is coincident with the interface connecting their respective control volumes. The second approach is the “Kinetically Deforming Skinned Mesh” (KDSM) method in which an auxiliary tetrahedral mesh overlays the Eulerian grid in the vicinity of under-resolved materials. The tetrahedral mesh, is glued to the Eulerian mesh[118, 52, 53, 112].

As far as we know, the above numerical methods for solving the heat equation on multi-material domains with complex embedded interfaces, have not been applied to the Dirichlet problem, and only a few have addressed the Neumann boundary condition at embedded interfaces. The Dirichlet and Neumann boundary condition cases are quite important in order to simulate phase change problems. In our method, we introduce the “connected centroid” flux discretization for all fluxes (see Figures 3.8 and 3.9). For embedded interfaces in which there is a Dirichlet or Neumann boundary condition, we introduce a “linear exact” correction term (see Section 3.1.9) to our “connected centroid” flux algorithm in order to guarantee second order accuracy in space for the temperature and its’ gradient. Our new “connected centroid” flux algorithm does not require us to subdivide material polygons into triangles/tetrahedra as in (Garimella&Lipnikov; Lee, Hyde et al. [27, 53]). The only degree of freedom for a given material polygon in a cell is at a materials’ centroid. We discretize the time derivative part of the heat equation (see Section 3.1.2) using the backward Euler method which is a discrete maximum principle preserving time discretization scheme. We discretize the spatial derivatives part of the heat equation (3.1.2) using the finite volume method since the finite volume method is a discrete energy preserving space discretization scheme[38]. A large sparse $n \times n$ non-symmetric and non-singular linear system is inverted at each time step using the Jacobi polynomial preconditioned BiCGStab method which is guaranteed to converge (assuming no round-off error) in at most n steps [100].

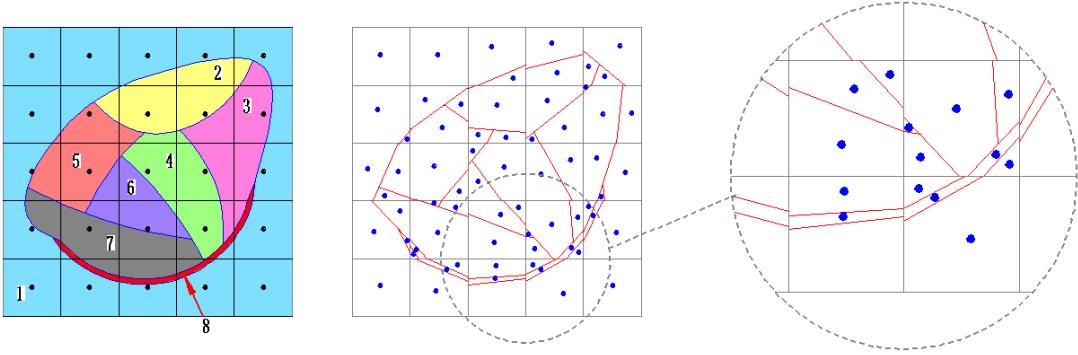


Figure 1.11: Example of a multi-material diffusion problem with MOF piecewise linear interface reconstruction.

We use the same terminology as in [44] in which we call the light blue background rectangular mesh of the left most figure a “base mesh,” the partitioning of a single rectangular cell into material polygons a “minimesh,” and the union of each “minimesh,” derived from all of the cells on a base mesh, is called the “supermesh.” In contrast to the “basemesh,” the “supermesh” recognizes subgrid-scale filamentary regions. Note that our supermesh recognizes interface gaps at cell faces. This is similar to [16], but in contrast to [18].

1.2 The Stefan Problem

For representing phase change phenomenon, the Stefan problem [105, 25, 71, 70, 66] is a special kind of boundary value problem for a heat equation with deforming interfaces. The classical two-phase Stefan problem describes the temperature distribution in a homogeneous medium undergoing phase change. The heat equation is solved by imposing the initial temperature distribution on the whole medium. The Stefan condition is enforced on the evolving interface (boundary) between its two phases.

The general class of Stefan problem was first introduced by the Slovenian physicist, Josef Stefan around 1890 in relation to problems of ice formation [105]. We refer the reader to Pruss et al.(2007) [79] for a proof of existence of solutions to the Stefan problem. The Stefan problem is a non-linear problem, but some analytically derived solutions do exist (for example, [25, 71, 70, 66]). Many different numerical algorithms have been applied or developed to solve the Stefan problem, as well.

Based on different approaches to treat deforming boundaries, we can divide the Stefan problem numerical algorithms into two major categories:

1. No tracking on the interface: For example, the classical enthalpy method. Voller et al. [104, 103, 87] developed an enthalpy method for convection/diffusion phase change problems. The essential feature

of their formulation is that latent heat effects are isolated in a source term. The interface between two phases is not explicitly tracked and is determined via post-processing.

2. The interface is represented implicitly [11, 29, 75] or explicitly [40] and an advection equation for tracking the deforming interface is coupled with the diffusion equation. We highlight several classical interface tracking methods and point out respective distinguishing algorithmic properties.

Juric and Tryggvason [40] presented an explicit front-tracking method to simulate time dependent two-dimensional dendrite solidification of pure substances. The method is based on a finite difference approximation of the heat equation. The solid-liquid interface is explicitly tracked by front-tracking technique. The method converges well under grid refinement. Note: no examples reported in Juric & Tryggvason in which the interface changed topology.

Chen et al. [11] developed an implicit tracking level set method for solving Stefan problems. An implicit finite difference scheme is employed for solving the heat equation. The front of the solid-liquid phases is captured by the zero contour of the level set method. Gibou and Fedkiw [29] developed a third-order accurate implicit finite difference method for solving Stefan problem on irregular domains. Level-set method is also applied here to capture the interface.

Papac et al.[75] presented a numerical method for simulating diffusion dominated phenomena on irregular domains and free moving boundaries with Robin boundary conditions on quadtree/octree adaptive meshes. They use a hybrid finite-difference and finite-volume framework that combines the level-set finite difference discretization and the treatment of Robin boundary conditions.

1.2.1 Our Intentions and Expectations for Solving the Stefan Problem

My ultimate goal is to apply our method on complex phase change problems like nucleate boiling which consists of complex deforming micro structural regions. For complex deforming boundary problems, the conventional approach is to approximate solutions to these problems using the level set method or VOF method using a “cut cell” approach. My supermesh-FVM-MOF approach, on the other hand, is the first approach to enable one to instantly and robustly form a dynamic locally body-fitted mesh on which the solution is solved. Also, as mentioned previously, due to the local property of MOF reconstruction, MOF is designed to represent. robustly, material parcels much smaller than a grid element. The Stefan problem is a good start for testing our supermesh-FVM-MOF diffusion solver performance from a theoretical numerical analysis perspective since the Stefan problem features complex deforming boundaries with micro-structure and good verification and validation test problems are available [11, 29, 40]. We predict that our supermesh-FVM-MOF method will be applicable to more complicated applications as well [113, 81, 31, 75, 29, 11, 88, 108, 83, 119, 9, 46, 5, 6, 82, 106, 89, 65, 36, 114, 62, 86, 17, 22, 74, 49, 48].

Starting with the FVM-MOF diffusion solver that we have developed, we next solve the advection equation which tracks the interface information based on the pure temperature diffusion induced interface front velocity.

We have our work focused on solving the Stefan problem on a dynamic supermesh, taking into account the solution in micro-structure in which no one has done before. We apply our Finite Volume Moment-of-Fluid method based on supermesh [58] to discretize the heat equation which in turn gives the front velocity (see Equation 2.5) The “linear exact” algorithm (see section 3.1.9) is used to enforce the Dirichlet boundary at the interface in cells in which the area fraction “Gap error” criteria (see section 3.2.1) is met. If the “gap error” is exceeded, we fall back to the low order, “maximum principle preserving” enforcement of Dirichlet boundary conditions (the “connected centroid” method without correction 3.1.6; see Equation 3.13). A few classic benchmark problems are tested to show the performance of our algorithm. Additionally, the proposed method is further tested on cases for multi-material Stefan problem and thin filament geometry, as well as unstable Stefan problem, which will provide evidence that the proposed method is suitable for complicated problems containing micro-structure, previously inaccessible via conventional AMR or cut-cell methods.

CHAPTER 2

PROBLEM FORMULATION AND EQUATIONS

2.1 Multi-material Diffusion Problem

The multi-material diffusion equation is considered in this paper. Referring to Figure 2.1, Ω_m represents the region occupied by material m ($m = 1, 2, \dots, M$) and the whole domain is $\Omega = \bigcup_m \Omega_m$. Γ_m represents the boundary of the material region occupied by m . We define the interface between materials m_1 and m_2 as $\Gamma_{m_1, m_2} = \Gamma_{m_1} \cap \Gamma_{m_2}$. For each material m in each cell, we solve:

$$\frac{\partial \theta_m}{\partial T} = \nabla \cdot (\kappa_m \nabla \theta_m) + \dot{Q} \quad [\theta]_\Gamma = 0, \quad [\kappa \nabla \theta \cdot \vec{n}]_\Gamma = 0 \quad (2.1)$$

where θ_m is the temperature profile of material m , $[\cdot]$ represents the jump condition across an interface, κ_m is the diffusion coefficient of material m and \dot{Q} is the heat source term. In our tests, without the loss of generality, we set both the temperature profile jump and the heat flux jump across the interfaces to be zero.

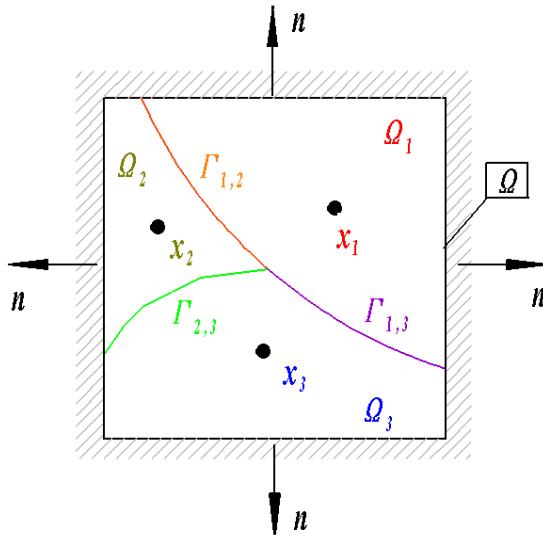


Figure 2.1: Example of problem formulation,
note: x_m is the centroid of region Ω_m occupied by material m , $m = 1, 2, 3$ in the figure. Assume the normal direction n of a cell face is always point outward.

2.2 Multi-material Stefan Problem

For classical two phase Stefan problem, assume $\rho = \text{constant}$, $C_v = \text{constant}$ and diffusion process dominating the heat transfer $\mathbf{u} = 0$.

$$\frac{\partial(\rho e)}{\partial t} = \nabla \cdot (\kappa \nabla \theta(\mathbf{x})) \quad \mathbf{x} \in \Omega \quad (2.2)$$

$$\theta(\mathbf{x}) = \theta_{\text{sat}}(\mathbf{x}, t) \quad \mathbf{x} \in \Gamma \quad (2.3)$$

$$F_t + V_n \nabla F = 0 \quad (2.4)$$

$$V_n = \frac{[\kappa \nabla \theta \cdot \mathbf{n}]_\Gamma}{L} \quad (2.5)$$

where $[\cdot]$ represents the jump condition across an interface, κ is the diffusion coefficient. The Dirichlet boundary condition at the interface is defined in Equation 2.3. In the dendrite growth test problem, the saturation temperature is a function of local interface curvature. Equation 2.4 is the volume fraction advection equation which tracks the moving interface location. Equation 2.5 is also called Stefan condition which defines the phase front velocity.

2.2.1 Derivation of the Stefan Condition

Starting from the energy equation, assume $\mathbf{u} = 0$.

$$\frac{\partial(\rho e)}{\partial T} = \nabla \cdot (\kappa \nabla \theta) \quad (2.6)$$

$$\text{where, } e = C_p \theta + LH(\phi) \quad (2.7)$$

$$H(\phi) = \begin{cases} 1 & \phi > 0 \quad (x \in \Omega_2) \\ 0 & \phi < 0 \quad (x \in \Omega_1) \end{cases} \quad (2.8)$$

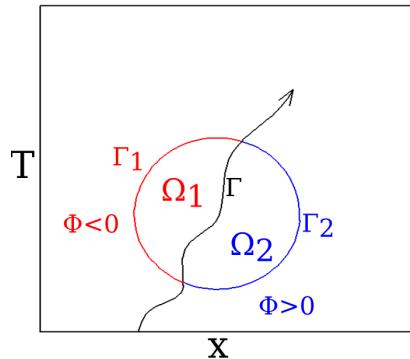


Figure 2.2: Stefan condition derivation using weak formulation.

Note in the figure, $\Gamma \cup \Gamma_1 = \partial\Omega_1$, $\Gamma \cup \Gamma_2 = \partial\Omega_2$, $\Gamma \cap \Gamma_1 = \phi$, $\Gamma \cap \Gamma_2 = \phi$

Without loss of generality, assume $C_p = 1$, $\rho = 1$

$$\frac{\partial e}{\partial T} = \nabla \cdot (\kappa \nabla \theta) \quad (2.9)$$

$$\text{where, } e = \theta + LH(\phi) \quad (2.10)$$

Rewrite Equation 2.9 into two ODEs, according to the Fourier law, introducing heat flux density $\mathbf{q} = -\kappa \nabla \theta$

$$\frac{\partial e}{\partial T} + \nabla \cdot \mathbf{q} = 0 \quad (2.11)$$

$$\mathbf{q} = -\kappa \nabla \theta \quad (2.12)$$

First, consider Equation 2.12, multiply a test function ψ which vanishes outside of Ω on both side of Equation 2.9, and integrate over Ω ($\Omega = \Omega_1 \cup \Omega_2$), then integrate by parts and apply divergence theorem,

$$\int_{\Omega} \frac{\mathbf{q} \cdot \psi}{\kappa} dV = \int_{\Omega} -\nabla \theta \cdot \psi dV \quad (2.13)$$

$$\int_{\Omega} \frac{\mathbf{q} \cdot \psi}{-\kappa} dV = \int_{\partial\Omega} \theta \psi \cdot \mathbf{n}_{\Omega} dS - \int_{\Omega} \theta (\nabla \cdot \psi) dV \quad (2.14)$$

Repeat this procedure again, integrate over Ω_1 and Ω_2 , similar to Equation 2.14, we get

$$\int_{\Omega_1} \frac{\mathbf{q} \cdot \psi}{-\kappa} dV = \int_{\partial\Omega_1} \theta_1 \psi \cdot \mathbf{n}_{\Omega_1} dS - \int_{\Omega_1} \theta_1 (\nabla \cdot \psi) dV \quad (2.15)$$

$$\int_{\Omega_2} \frac{\mathbf{q} \cdot \psi}{-\kappa} dV = \int_{\partial\Omega_2} \theta_2 \psi \cdot \mathbf{n}_{\Omega_2} dS - \int_{\Omega_2} \theta_2 (\nabla \cdot \psi) dV \quad (2.16)$$

Equation 2.15 + Equation 2.16 - Equation 2.14, we have

$$\int_{\Gamma} \theta_1 \psi \cdot \mathbf{n}_{\Gamma_1 \rightarrow 2} dS + \int_{\Gamma} \theta_2 \psi \cdot \mathbf{n}_{\Gamma_2 \rightarrow 1} dS = 0 \quad (2.17)$$

Since ψ is chosen randomly and $\mathbf{n}_{\Gamma_1 \rightarrow 2} = -\mathbf{n}_{\Gamma_2 \rightarrow 1}$,

$$\int_{\Gamma} (\theta_1 - \theta_2) \psi \cdot \mathbf{n}_{\Gamma_1 \rightarrow 2} dS = 0 \quad (2.18)$$

$$\theta_1 = \theta_2 \quad \text{i.e. } \theta \text{ is continuous across the interface.} \quad (2.19)$$

Equation 2.9 along with 2.10 will reduce to

$$\frac{\partial \theta}{\partial T} = \kappa_1 \Delta \theta \quad \text{if } x \in \Omega_1 \quad (2.20)$$

$$\frac{\partial \theta}{\partial T} = \kappa_2 \Delta \theta \quad \text{if } x \in \Omega_2 \quad (2.21)$$

Define $\bar{\nabla} \equiv \begin{pmatrix} \partial_T \\ \partial_x \end{pmatrix}$, multiply a test function ψ which vanishes outside of Ω on both side of Equation 2.9, and integrate over Ω ($\Omega = \Omega_1 \cup \Omega_2$), then integrate by parts and apply divergence theorem,

$$\int_{\Omega} [\bar{\nabla} \cdot \begin{pmatrix} e \\ -\kappa \nabla \theta \end{pmatrix}] \psi dV = \int_{\partial\Omega} \psi \begin{pmatrix} e \\ -\kappa \nabla \theta \end{pmatrix} \cdot \mathbf{n}_{\Omega} dS + \int_{\Omega} \bar{\nabla} \psi \cdot \begin{pmatrix} e \\ -\kappa \nabla \theta \end{pmatrix} dV = 0 \quad (2.22)$$

Split the boundary $\partial\Omega$ and domain Ω into two materials' part

$$\int_{F_1} \psi \begin{pmatrix} e \\ -\kappa \nabla \theta \end{pmatrix} \cdot \mathbf{n}_{\Gamma_1} + \int_{F_2} \psi \begin{pmatrix} e \\ -\kappa \nabla \theta \end{pmatrix} \cdot \mathbf{n}_{\Gamma_2} \xrightarrow{0} \int_{\Omega_1} \bar{\nabla} \psi \cdot \begin{pmatrix} e \\ -\kappa \nabla \theta \end{pmatrix} - \int_{\Omega_2} \bar{\nabla} \psi \cdot \begin{pmatrix} e \\ -\kappa \nabla \theta \end{pmatrix} = 0 \quad (2.23)$$

$$\int_{\Omega_1} \bar{\nabla} \psi \cdot \begin{pmatrix} e \\ -\kappa \nabla \theta \end{pmatrix} dV + \int_{\Omega_2} \bar{\nabla} \psi \cdot \begin{pmatrix} e \\ -\kappa \nabla \theta \end{pmatrix} dV = 0 \quad (2.24)$$

Since test function ψ vanishes at the boundary, the first two terms of 2.23 will be 0. Then in the same way, Equation 2.20 and 2.21 will be multiplied by test function ψ as well and then integrated over their corresponding domain Ω_1 and Ω_2 , respectively.

$$\int_{\Gamma} \psi \begin{pmatrix} e \\ -\kappa \nabla \theta \end{pmatrix} \cdot \mathbf{N}_1 dS + \int_{F_1} \psi \begin{pmatrix} e \\ -\kappa \nabla \theta \end{pmatrix} \cdot \mathbf{N}_1 dS \xrightarrow{0} \int_{\Omega_1} \bar{\nabla} \psi \cdot \begin{pmatrix} e \\ -\kappa \nabla \theta \end{pmatrix} dV = 0 \quad (2.25)$$

$$\int_{\Gamma} \psi \begin{pmatrix} e \\ -\kappa \nabla \theta \end{pmatrix} \cdot \mathbf{N}_2 dS + \int_{F_2} \psi \begin{pmatrix} e \\ -\kappa \nabla \theta \end{pmatrix} \cdot \mathbf{N}_2 dS \xrightarrow{0} \int_{\Omega_2} \bar{\nabla} \psi \cdot \begin{pmatrix} e \\ -\kappa \nabla \theta \end{pmatrix} dV = 0 \quad (2.26)$$

Add Equation 2.25 and 2.26 together and subtract Equation 2.24,

$$\int_{\Gamma} \left[\psi \begin{pmatrix} e_1 \\ -\kappa_1 \nabla \theta_1 \end{pmatrix} \cdot \mathbf{N}_1 + \psi \begin{pmatrix} e_2 \\ -\kappa_2 \nabla \theta_2 \end{pmatrix} \cdot \mathbf{N}_2 \right] dS = 0 \quad (2.27)$$

$$\int_{\Gamma} \psi \left[\begin{pmatrix} e_1 \\ -\kappa_1 \nabla \theta_1 \end{pmatrix} - \begin{pmatrix} e_2 \\ -\kappa_2 \nabla \theta_2 \end{pmatrix} \right] \cdot \mathbf{N}_1 dS = 0 \quad (2.28)$$

Simplify further due to the arbitrary choose of ψ ,

$$\left[\begin{pmatrix} e_1 \\ -\kappa_1 \nabla \theta_1 \end{pmatrix} - \begin{pmatrix} e_2 \\ -\kappa_2 \nabla \theta_2 \end{pmatrix} \right] \cdot \mathbf{N}_1 = 0 \quad (2.29)$$

where, $\mathbf{N}_1 = \begin{pmatrix} \phi_T \\ \nabla \phi \end{pmatrix}$, and at the interface Γ , the interface front velocity is $\frac{d\mathbf{x}}{dT} = s\mathbf{n}$

$$\phi(\mathbf{x}(T), T) = 0 \quad (2.30)$$

$$\frac{d\phi(\mathbf{x}(T), T)}{dT} = |\nabla \phi| \frac{d\mathbf{x}}{dT} + \frac{\partial \phi}{\partial T} = 0 \quad (2.31)$$

$$\phi_T = -s|\nabla \phi| \quad (2.32)$$

$$\text{Thus, } \mathbf{N}_1 = \begin{pmatrix} -s|\nabla \phi| \\ \nabla \phi \end{pmatrix} \quad (2.33)$$

Plug N_1 into Equation 2.29,

$$\left[\begin{pmatrix} e_1 \\ -\kappa_1 \nabla \theta_1 \end{pmatrix} - \begin{pmatrix} e_2 \\ -\kappa_2 \nabla \theta_2 \end{pmatrix} \right] \cdot \begin{pmatrix} -s |\nabla \phi| \\ \nabla \phi \end{pmatrix} = 0 \quad (2.34)$$

$$\left[\theta_1 - (\theta_2 + L) \right] (-s |\nabla \phi|) + (-\kappa_1 \nabla \theta_1 + \kappa_2 \nabla \theta_2) \cdot \nabla \phi = 0 \quad (2.35)$$

Because $[\theta] = 0$, rewrite the equation results in,

$$s = \frac{(\kappa_1 \nabla \theta_1 - \kappa_2 \nabla \theta_2) \frac{\nabla \phi}{|\nabla \phi|}}{L} \quad (2.36)$$

Finally, the Stefan condition is given by,

$$s = \frac{[\kappa \nabla \theta \cdot \mathbf{n}]_{\Gamma}}{L} \quad (2.37)$$

CHAPTER 3

NUMERICAL METHOD

3.1 Multi-material Diffusion Solver

An outline of our multi-material diffusion simulation tool is given in the flow chart shown in Figure 3.1, and the details are described in the following sections.

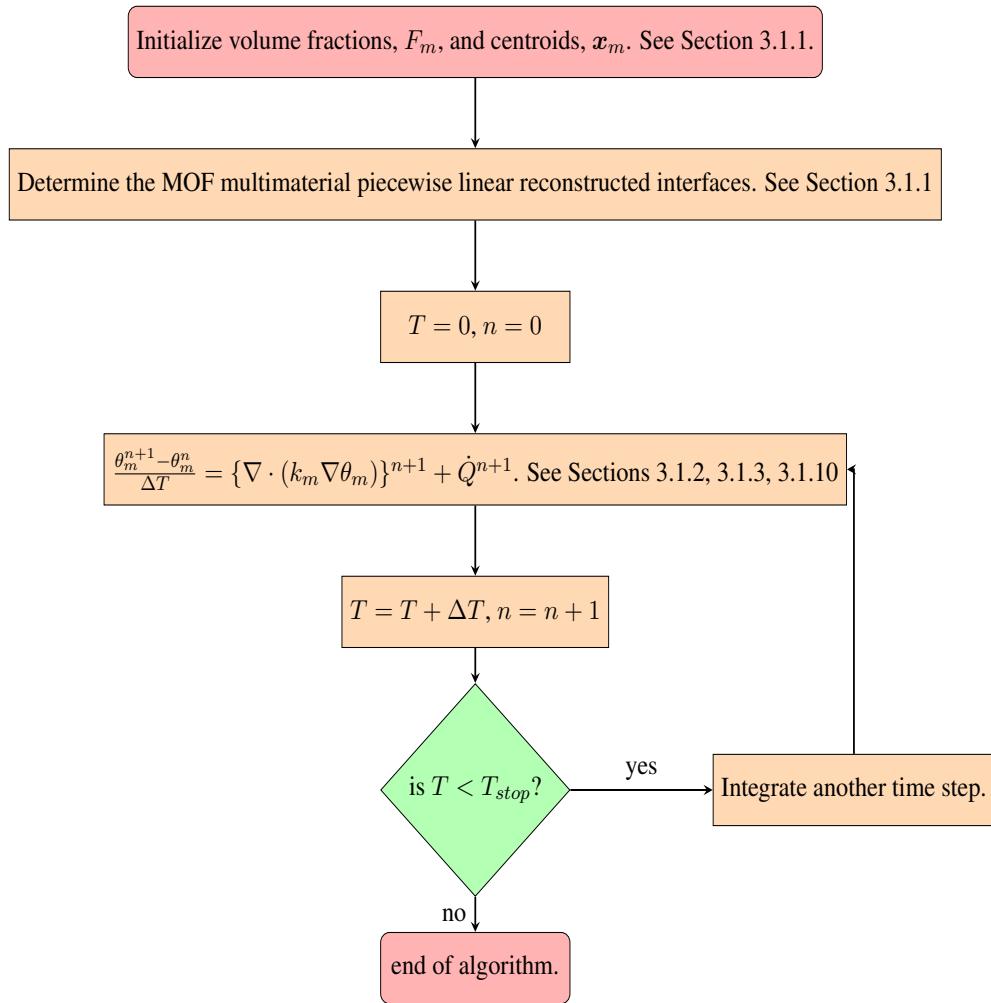


Figure 3.1: Flow chart corresponding to our multi-material diffusion simulation algorithm.

3.1.1 Centroid Initialization and MOF Reconstruction

Prior to the MOF reconstruction, one first must initialize the volume fractions F_m and centroids \mathbf{x}_m for each material in each cell.

We assume that each material region Ω_m is already implicitly defined by a level set function ϕ_m ,

$$\Omega_m = \{\mathbf{x} | \phi_m(\mathbf{x}) \geq 0\}. \quad (3.1)$$

The initial volume fraction F_m and centroid \mathbf{x} in a target cell are calculated as follows:

$$F_{m,cell} = \frac{\int_{\Omega_{cell}} H(\phi_m) dV}{|\Omega_{cell}|} \quad (3.2)$$

$$\mathbf{x}_{m,cell} = \frac{\int_{\Omega_{cell}} \mathbf{x} H(\phi_m) dV}{F_{m,cell} |\Omega_{cell}|}, \quad (3.3)$$

where $|\Omega_{cell}|$ is the area (volume in 3D) of cell Ω_{cell} and H is a Heaviside function defined as:

$$H(\phi_m) = \begin{cases} 1 & \phi_m > 0 \\ 0 & \phi_m \leq 0 \end{cases}$$

We must use adaptive quadrature in order to calculate the integrals in (3.2) and (3.3) so that volume fractions and centroids are recognized even for initializing filamentary materials. We truncate all volume fractions that are close to zero or one,

$$F_{m,cell} = 0 \quad \text{if } F_{m,cell} < \epsilon_{VFRAC}$$

$$F_{m,cell} = 1 \quad \text{if } |1 - F_{m,cell}| < \epsilon_{VFRAC}.$$

In our calculations, we have $\epsilon_{VFRAC} = 10^{-10}$.

The piecewise linear multi-material MOF [21] reconstruction method is applied for representing material interfaces (see Fig 3.2). In each computational grid cell, interfaces are approximated by straight lines (planes in 3D), see Equation 3.4. In a cell containing two materials, the normal \mathbf{n} and intercept point b of a line are derived from a reference volume fraction F_{ref} and reference centroid \mathbf{x}_{ref} .

$$\Gamma = \{\mathbf{x} | \mathbf{n} \cdot (\mathbf{x} - \mathbf{x}_0) + b = 0\} \quad (3.4)$$

We find an optimal slope \mathbf{n} and intercept b so that the centroid error E_{MOF} is minimized subject to the constraint that the volume error is zero (see Figure 3.2). In other words, we solve the following constrained

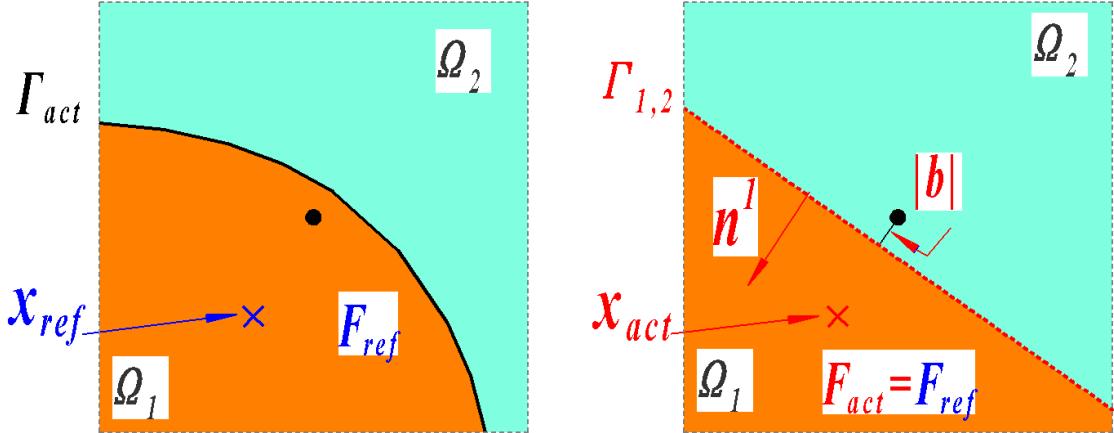


Figure 3.2: Example of MOF reconstruction in two materials cell.

Remark: the normal of the reconstructed interface is always point into the target material in this paper. For example in the figure, the normal n^1 points into material 1.

minimization problem:

$$\begin{aligned} (\mathbf{n}, \mathbf{b}) &= \operatorname{argmin}_{\mathbf{n}, \mathbf{b}} (E_{MOF}) = \operatorname{argmin}_{\mathbf{n}, \mathbf{b}} \|\mathbf{x}_{ref} - \mathbf{x}_{act}(\mathbf{n}, \mathbf{b})\| \\ \text{subject to } & |F_{act}(\mathbf{n}, \mathbf{b}) - F_{ref}(\mathbf{n}, \mathbf{b})| = 0 \end{aligned} \quad (3.5)$$

The Gauss-Newton iteration tolerance for finding the slope is $\epsilon_{slope} = 10^{-10}$ and the Brent's method iteration tolerance for finding the intercept is $\epsilon_{intercept} = 10^{-12}$. Note, modern reconstruction method due to analytical solution is presented in [19, 54, 67]

If the number of materials is more than two in one cell, we apply the nested dissection algorithm to perform the MOF reconstruction method [21], see Figure 3.3.

The nested dissection algorithm that we have implemented[37] is different from the classical algorithm[21]. the classical approach is to try all possible material reconstruction orderings, and then choose the ordering with the smallest overall centroid error (3.5). In our approach, we always choose the next reconstructed material to be the one whose centroid is the farthest to the centroid of the un-captured volume. It was found heuristically in [37] that the accuracy of the two approaches for selecting material reconstruction ordering is comparable.

Given that materials $m^{(1)}, m^{(2)}, \dots, m^{(p)}$ have been reconstructed, then $m^{(p+1)}$ is the material which satisfies:

$$\text{i) } F_{m^{(p+1)}} > 0$$

- ii) $m^{(p+1)} \neq m^{(1)}, m^{(2)}, \dots, m^{(p)}$
- iii) $\|\mathbf{x}^{uncaptured} - \mathbf{x}_{m^{(p+1)}}\| > \|\mathbf{x}^{uncaptured} - \mathbf{x}_r\|$ for all $r \neq m^q$, $q = 1, 2, \dots, p$ and $F_r > 0$.
 $\mathbf{x}^{uncaptured}$ is the centroid of the unfilled space in the cell.

A property of the MOF reconstruction algorithm is that if the actual interfaces are all straight lines (planes in 3D) and represented by a single cut, then the MOF reconstruction is exact.

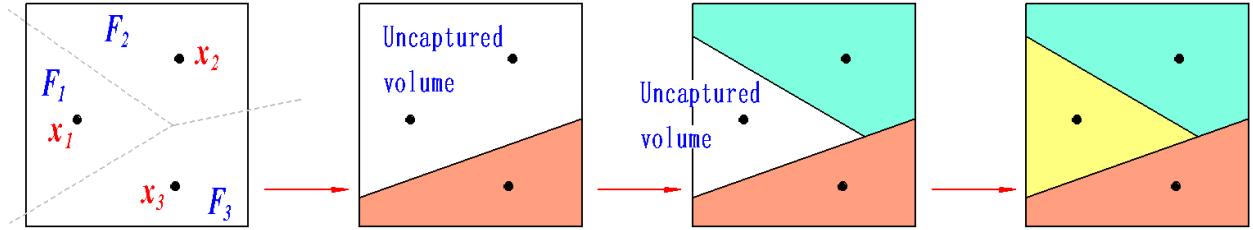


Figure 3.3: Illustration of MOF reconstruction with materials more than 3 using nested dissection algorithm
Material 3 is the first material reconstructed. Material 2 is the second material reconstructed.

3.1.2 Finite Volume Discretization for Space Derivatives and Backward Euler for Time

We discretize the time derivative part of the heat equation (2.1) using the backwards Euler method which is a discrete maximum principle preserving time discretization scheme. For the derivation of our spatial discretization algorithm, we integrate (2.1) and apply the divergence theorem in order to approximate $\nabla \cdot (k_m \nabla \theta_m)$ by a boundary integral (rather than finite difference or finite element method).

$$\int_{T^n}^{T^{n+1}} \left\{ \frac{1}{|\Omega_m|} \left[\int_{\Omega_m} (\theta_m)_T dV = \int_{\Omega_m} \nabla \cdot (\kappa_m \nabla \theta_m) + \dot{Q} dV \right] \right\} dT \quad (3.6)$$

$$\text{Define: } \bar{\theta}_m(T, \Omega_m) = \frac{1}{|\Omega_m|} \int_{\Omega_m} \theta_m dV = \theta_m(T, \mathbf{x}_m(\Omega_m)) + O(\Delta x^2)$$

$$\int_{T^n}^{T^{n+1}} \left\{ (\bar{\theta}_m)_T = \frac{1}{|\Omega_m|} \left[\int_{\Omega_m} \nabla \cdot (\kappa_m \nabla \theta_m) + \dot{Q} dV \right] \right\} dT \quad (3.7)$$

$$\int_{T^n}^{T^{n+1}} \left\{ (\bar{\theta}_m)_T = \frac{1}{|\Omega_m|} \left[\int_{\partial\Omega_m} \kappa_m \nabla \theta_m \cdot \mathbf{n} dS + \int_{\Omega_m} \dot{Q} dV \right] \right\} dT \quad (3.8)$$

$$|\Omega_m|(\theta(T^{n+1}, \mathbf{x}_m) - \theta(T^n, \mathbf{x}_m)) \doteq \left\{ \oint_{\partial\Omega_m} \kappa_m (\nabla \theta)(T^{n+1}, \mathbf{x}_m) \cdot \mathbf{n} dS + \int_{\Omega_m} \dot{Q}^{n+1} dV \right\} (T^{n+1} - T^n) \quad (3.9)$$

Referring to Figure 2.1, $x_m(\Omega_m)$ is the centroid of region Ω_m occupied by material m , $\partial\Omega_m$ represents the boundary of Ω_m , and $|\Omega_m|$ represents the area (volume for 3D) of Ω_m .

The details for approximating $\oint_{\partial\Omega_m} \kappa_m \nabla \theta_m \cdot \mathbf{n} dS$ is given in the next section (Section 3.1.3).

3.1.3 Flux Calculation

Given the finite volume formulation derived in Section 3.1.2, we approximate the flux integrals along the boundaries of each material element (material elements are denoted by Ω_1 , Ω_2 or Ω_3 in Figure 2.1) by the midpoint rule. The boundaries for a given material element Ω_m in a certain cell are comprised of cell faces and a material interface. Referring to Figure 3.4, the flux is piecewise continuous along the boundary of a material element. Each cell face or interface of a given material element is comprised of one or more material pairs corresponding to an inside adjoining material and an outside adjoining material. For each face (cell face or material interface) of a material element, it is important to know the area fractions of opposite side materials that share the same face. For example in Figure 3.4, the interface for material 3 ($L_1 \cup L_2$) has two pairs: (1) a material 2,3 pair (L_2) and (2) a material 1,3 pair (L_1). We treat each distinct material pair differently since each material pair corresponds to a different heat flux (see Figure (3.9)). The lengths (areas in 3D) of the interfaces and cell faces needed by the midpoint rule approximation, $(L_1, L_2, L_3, L_4, L_5)$, are computed with an algorithm that uses already existing MOF routines that were developed for the work reported in [37].

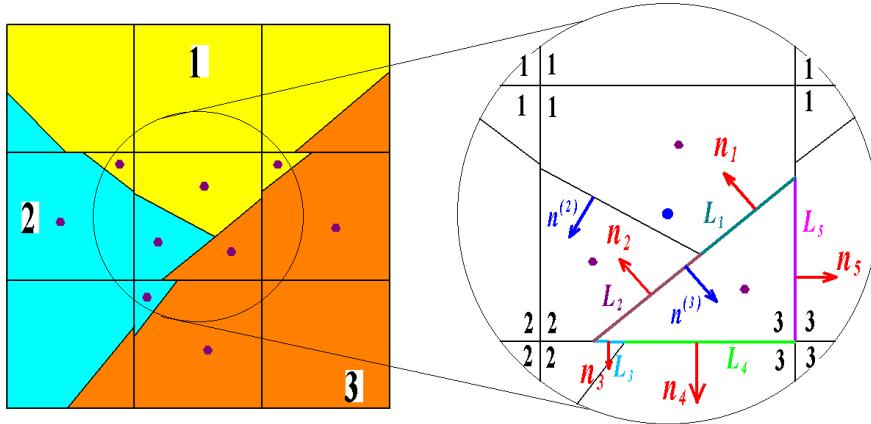


Figure 3.4: Flux integral calculation for material 3 along its boundary in the center cell:

We calculate the flux through each subsection of the material 3 boundary.

Each subsection has a unique material pair on its two sides.

Note: the red normals n_1, n_2, n_3, n_4, n_5 are the ones used for calculating the heat flux integral, the blue normals $n^{(3)}, n^{(2)}$ are the normals of the MOF linear reconstructed interface.

Thus, for a given multi-material cell, the flux integral algorithm is illustrated in Figure 3.4.

For each material element, we divide our flux integral algorithm into two parts: one for cell faces (e.g. (L_3, L_4, L_5) in Figure 3.4) and the other one is for interfaces (e.g. (L_1, L_2) in Figure 3.4).

For the finite volume method, in order to approximate integrals around a materials' boundary, the area fractions and area centroids for all possible material pairs need to be calculated. In the next section (Section 3.1.4), we describe our algorithm for accurately calculating the material pair areas and centroids, requiring only volume calculation routines.

3.1.4 Calculation of Paired Area Fractions along Cell Faces

For approximating the paired area fractions (length fractions in 2D) along the cell faces, we have developed a pairing algorithm that uses existing MOF-geometric library routines [56]. To elaborate the pairing algorithm, we use a 2D example-computing the material pairs between two neighboring cells (left blue cell and right orange cell) as shown in Figure 3.5(a). They both contain three materials marked as 1 2 3. The reconstructed interfaces are shown in the figure.

The steps for implementing the pairing algorithm are described in details as follows, referring to Figures 3.5 and 3.6,

1. Project interfaces to be orthogonal to the face which divides the adjoining cells (Figure 3.5(a)).

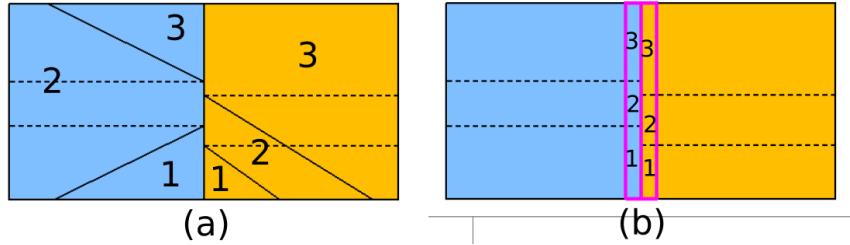


Figure 3.5: From previous section 3.1.4:
interface projections are shown in (a)
and thin material slabs formation are shown in (b)

2. Identify a thin strip Ω about the dividing face (Figure 3.5(b)).

$$\Omega_L = \Omega_{cell,L} \cap \Omega \quad (3.10)$$

$$\Omega_R = \Omega_{cell,R} \cap \Omega \quad (3.11)$$

3. Then we need to choose either to subtract material slabs of Ω_L from Ω_R or subtract those of Ω_R from Ω_L . In Figure 3.6, we illustrate the algorithm with subtracting Ω_L material slabs one by one from Ω_R .

4. Implement the pairing algorithm 1

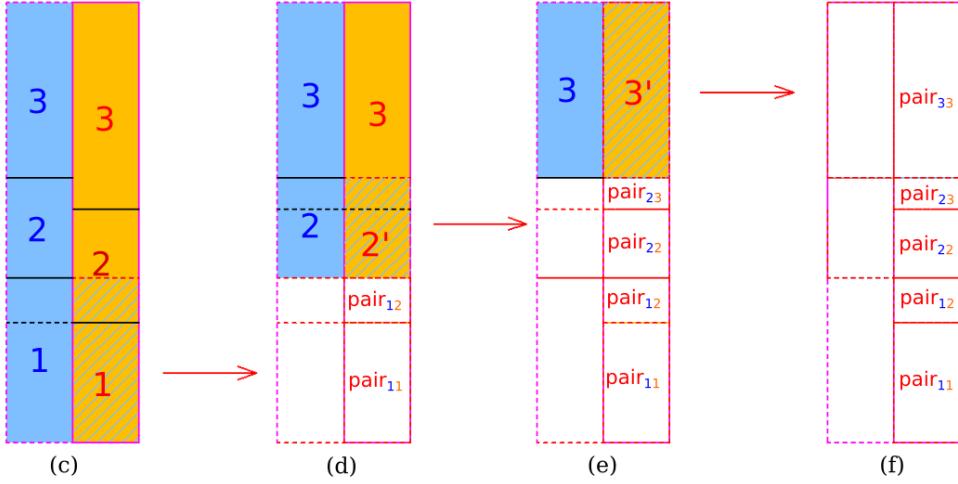


Figure 3.6: Pairing algorithm:

In Figure (c), we subtract the blue shaded region from the right. After subtracting left material 1 region from the right, intuitively, as example shown in (d),
 $\text{pair}_{1,2} = \text{right volume } 2 \text{ before cut} - \text{right volume } 2 \text{ after cut}$
 $\text{pair}_{1,1} = \text{right volume } 1 \text{ before cut} - \text{right volume } 1 \text{ after cut}$

3.1.5 Calculation of Paired Area Fractions along Material Interfaces

For multi-material reconstruction, we use the nested-dissection algorithm as mentioned (see also [21]) above. The reconstructed interface resulting from the MOF algorithm will come out in a specific order. Referring to Figure 3.7, the interface Γ^1 that separates material 3 from the other materials is reconstructed first, and Γ^2 (the interface that separates material 2 from 1) is reconstructed second. (We list the sorting rank on the upright corner of Γ in order to emphasize the distinction between the order at which a material is reconstructed versus the material ID). We follow this order to perturb each interface in its normal direction (The interface perturbation step is shown in the Appendix algorithm 3 step 5). Thus the general algorithm for calculating the paired area fractions (length fractions in 2D) of material pairs along a given material interface is described in Figure 3.7 and algorithm 3:

For example, as in Figure 3.7(c), $A_{3,2} = \frac{V_1}{V_{total}}$, $A_{3,1} = \frac{V_2}{V_{total}}$, where $V_{total} = \sum_m V_m$. In the expressions, A_{m_1, m_2} represents the area fraction (length fraction in 2D) of the subsection between materials m_1

Algorithm 1 Pairing algorithm for finding area fractions (length fractions in 2D) of cell faces

- 1: Initialize $\Omega_{uncapt} = \Omega_L$, $order = 1$
 - 2: **while** $\Omega_{uncapt} \neq \phi$ **do**
 - 3: Find the volumes of current slab material components of Ω_R within Ω_L , $|\Omega_{m,R}^{(L)} \cap \Omega_{uncapt}|$ $m = 1, \dots, M$
 - 4: Subtract $\Omega_{\tilde{m},L}$ from Ω_{uncapt} , $\Omega_{uncapt}^{new} = \Omega_{uncapt} \setminus \Omega_{\tilde{m},L}$, where \tilde{m} is the material reconstructed at the “order” rank.
 - 5: Find $|\Omega_{m,R}^{(L)} \cap \Omega_{uncapt}^{new}|$ $m = 1, \dots, M$
 - 6: $\Omega_{\tilde{m},m} = ||\Omega_{m,R} \cap \Omega_{uncapt}| - |\Omega_{m,R} \cap \Omega_{uncapt}^{new}||$ $m = 1, \dots, M$
 - 7: $\Omega_{uncapt} = \Omega_{uncapt}^{new}$, $order = order + 1$.
 - 8: **end while**
-

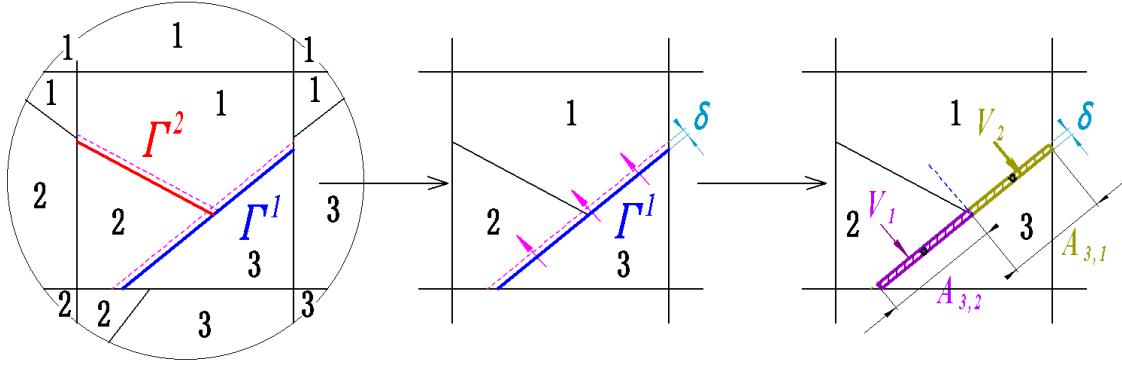


Figure 3.7: Example for illustrating the steps for calculating the area fractions along an interface

and m_2 , V_1 , V_2 and V_{total} represent the volume difference (area difference in 2D) due to perturbing a given material interface.

3.1.6 “Connected Centroid” Approximation for Interfacial Fluxes.

Given the area fractions (length fractions in 2D), areas (lengths in 2D), centroids and material temperatures, the flux integral around the boundary of a material element Ω_m is approximated as:

$$\int_{\partial\Omega_m} k_{m,m'} \nabla \theta_m \cdot \mathbf{n}_m dS \approx \sum_{iface=1}^F \sum_{m'=1}^M q_{iface,m,m'} A_{iface,m,m'} \quad (3.12)$$

$$q_{iface,m,m'} = k_{m,m'} \nabla \theta_m \cdot \mathbf{n}_m$$

where F is the number of faces of the material element Ω_m ; M is the maximum number of materials; $q_{iface,m,m'}$ is the heat flux across each face (including cell face and interface) of material element Ω_m ;

$A_{iface,m,m'}$ is the area (length in 2D) of each face; \mathbf{n}_m is the outward facing normal of the interface or cell face.

In this article we present the “connected centroid” method for approximating the flux. Referring to Figures 3.8 and 3.9, the “connected centroid” method is:

$$q_{iface,m,m'} = k_{m,m'} \frac{\theta_{m'} - \theta_m}{d_{m',m}} |\mathbf{n} \cdot \mathbf{n}_{iface}| = k_{m'} \frac{\theta_{m'} - \theta_{\text{intersect}}}{d_{m',\text{intersect}}} |\mathbf{n} \cdot \mathbf{n}_{iface}| = k_m \frac{\theta_{\text{intersect}} - \theta_m}{d_{\text{intersect},m}} |\mathbf{n} \cdot \mathbf{n}_{iface}| \quad (3.13)$$

where the index “intersect” indicates the intersection point of the line connecting two centroids and the cell face or the interface. $\mathbf{n} = \frac{\mathbf{x}_{m'} - \mathbf{x}_m}{\|\mathbf{x}_{m'} - \mathbf{x}_m\|}$, $d_{m',m}$ is the distance between $\mathbf{x}_{m'}$ and \mathbf{x}_m , \mathbf{n}_{iface} is the face/interface normal, and $k_{m,m'}$ is the weighted geometric average of the diffusion coefficients k_m and $k_{m'}$ (discussed in section 3.1.7; please see (3.19)).

In Figure 3.9, for updating θ_B (i.e. $m \equiv B$), the flux across L_1 (subsection of the face that separates middle cell from right cell) is approximated as

$$q_{BA} = k_{m_2,m_2} \frac{\theta_A - \theta_B}{d_1} |\mathbf{n}_1 \cdot \mathbf{n}_{f_1}| \quad (3.14)$$

the flux across L_2 (the interface that separates material 2 from material 1 in the center cell) is approximated as

$$q_{BC} = k_{m_1,m_2} \frac{\theta_C - \theta_B}{d_2} |\mathbf{n}_2 \cdot \mathbf{n}_{f_2}| \quad (3.15)$$

For the same face subsection L_1 , if θ_A is the one to update (i.e. $m \equiv A$), the flux is calculated as

$$q_{AB} = k_{m_2,m_2} \frac{\theta_B - \theta_A}{d_1} |\mathbf{n}_1 \cdot \mathbf{n}_{f_1}| \quad (3.16)$$

We note that the method proposed by Dai and Scannapieco [16], or a third valid method, the “Orthogonal projection” method (see for example Figure 3 of Dawes (2017) [18]), give comparable results to the “connected centroid” method for multi-material problems with jump condition embedded interface boundary conditions. We refer the reader to our comparison study in Section 4.1.6 and, for completeness, describe Dai and Scannapieco’s method and the “Orthogonal projection” method here:

1. Dai and Scannapieco’s method [16] (Figure 3.10):

$$q_{iface,m,m'} = k_{m'} \frac{\theta_{m'} - \theta_{\text{cen}}}{d_{m',\text{cen}}} |\mathbf{n}_{iface} \cdot \mathbf{n}_{m',\text{cen}}| = k_m \frac{\theta_{\text{cen}} - \theta_m}{d_{\text{cen},m}} |\mathbf{n}_{iface} \cdot \mathbf{n}_{\text{cen},m}| \quad (3.17)$$

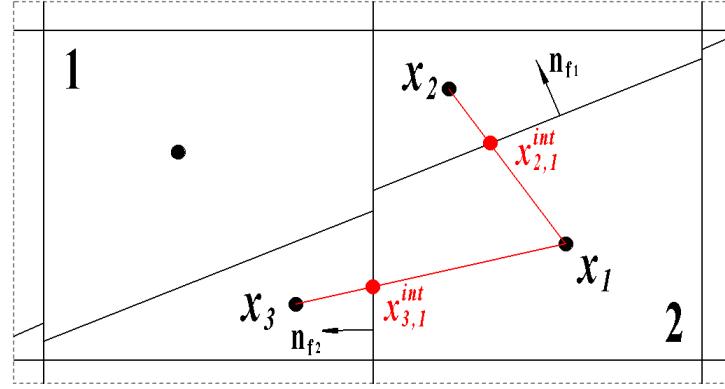


Figure 3.8: Example: “connected centroid” method for approximating the flux through a segment of a face of a cell or an interface.

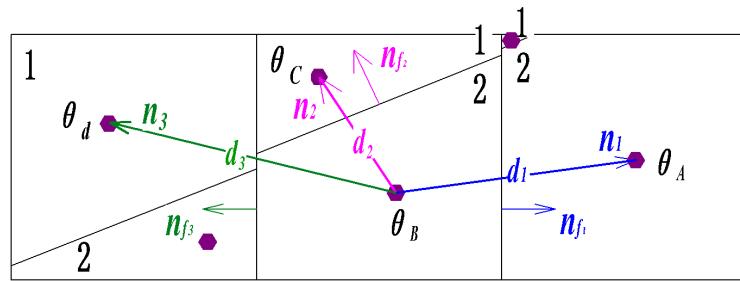


Figure 3.9: Connected Centroid method for approximating the flux.
The discretized material m temperature θ_m is stored at the material m centroid x_m .

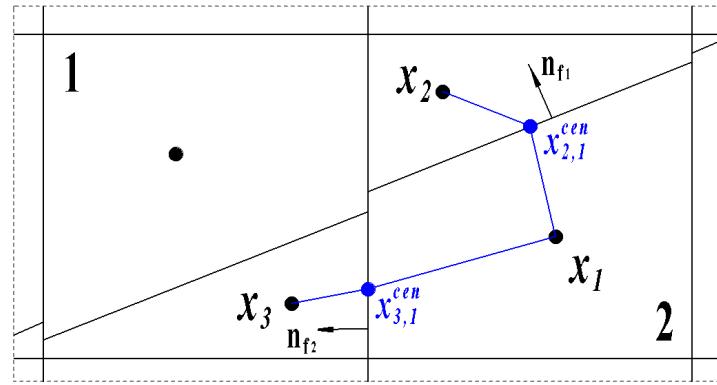


Figure 3.10: Example: Dai and Scannapieco’s method for approximating the flux through a segmentation of a face of a cell or an interface.

where index ‘‘cen’’ indicates the center of the cell face or interface; $\mathbf{n}_{m',\text{cen}}$ is the unit normal point from x_{cen} to $x_{m'}$.

2. Orthogonal projection method (Figure 3.11):

$$q_{\text{face},m,m'} = k_{m'} \frac{\theta_{m'} - \theta_{\text{pr}}}{d_{m',\text{pr}}} = k_m \frac{\theta_{\text{pr}} - \theta_m}{d_{\text{pr},m}} \quad (3.18)$$

where index ‘‘pr’’ indicates the projection of the centroid onto the cell face or interface.

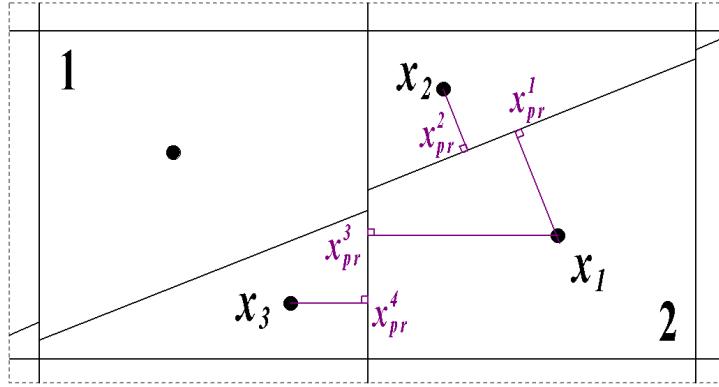


Figure 3.11: Example: orthogonal projection method for approximating the flux through a segmentation of a face of a cell or an interface. (see also Figure 3 of Dawes (2017) [18])

3.1.7 Numerical Approximation of Mixed Material Diffusivities

According to the different types of boundary conditions on the interfaces and cell faces, in the paper, we define the mixed material diffusion coefficient as follows:

Assume k_m is the diffusion coefficient of material m . k_{m_1,m_2} is calculated for the diffusion coefficient at the interface or cell face between materials m_1 and m_2 . We define l_m as an adjusted distance from the material m centroid to the interface or cell face,

$$l_m = \frac{d_m}{|\mathbf{n} \cdot \mathbf{n}_{\text{face}}|}.$$

Then the mixed material coefficient and gradient are defined as follows:

1. Multi-material jump boundary condition

$$k_{m_1,m_2} = \frac{(l_{m_1} + l_{m_2})k_{m_1}k_{m_2}}{l_{m_2}k_{m_1} + l_{m_1}k_{m_2}} \quad (3.19)$$

which is harmonic average derived from the following equations resulting from the flux continuity at the interface,

$$\frac{\theta_1 - \theta_I}{l_{m_1}} k_1 = \frac{\theta_2 - \theta_I}{l_{m_2}} k_2 = \frac{\theta_1 - \theta_2}{l_{m_1} + l_{m_2}} k_{m_1, m_2}$$

where θ_I represents the temperature at the face (cell face or interface) considered.

2. Homogeneous Neumann boundary

$$k_{m_1, m_2} = \begin{cases} 0, & \text{If } k_1 = 0 \text{ or } k_2 = 0, \\ 0, & \text{If } l_{m_1} = 0 \text{ or } l_{m_2} = 0. \end{cases} \quad (3.20)$$

3. Dirichlet boundary

On a Dirichlet boundary interface, there is no corresponding material on the opposite side. So the diffusion coefficient on the interface of material element Ω_m is always k_m ($k_m = k_{m,m}$).

3.1.8 Justification for the Validity of the “Connected Centroid” Flux Algorithm

For multi-material diffusion problems with the jump condition embedded interface boundary conditions, the flux approximation in (3.13) can be replaced with alternate approximations (for example, Dai and Scannapieco (3.17) or the “orthogonal projection” method (3.18)) without altering the order of convergence of the overall method. Implicit in our flux algorithm is that the following is a consistent approximation,

$$\nabla \theta \approx \frac{\theta_{\text{in}} - \theta_{\text{intersect}}}{L_{\text{in, intersect}}} \tilde{n}, \quad (3.21)$$

on either side of an internal or external face. A similar approximation is made in Dai and Scannapieco’s article (see Equation (3.17)) and for multi-material problems, we demonstrate convergence for the temperature and its’ gradient for all three flux algorithms tried (“Connected centroid”, Dai and Scannapieco (3.17), and Orthogonal projection (3.18), see Section 4.1.6). However (3.21) does not include the components tangent to \tilde{n} which results in an inconsistent method for predicting the temperature gradient (when $\tilde{n} \cdot n_{\text{face}} \neq 1$) when the embedded boundary condition is either Dirichlet or Neumann (instead of the jump condition case). In the next section, Section 3.1.9, we describe a new algorithm, which is a correction to our “connected centroid” algorithm, which has a “linear exact” property for problems with embedded Dirichlet or Neumann boundary conditions.

3.1.9 Linear Exact Algorithm for Dirichlet or Neumann Boundary Conditions

If an embedded interface has a Dirichlet boundary condition,

$$\theta_m(\mathbf{x}, t) = g(\mathbf{x}, t) \quad \mathbf{x} \in \Gamma_{m,m'},$$

or a Neumann boundary condition,

$$k_m \nabla \theta_m(\mathbf{x}, t) \cdot \mathbf{n} = g(\mathbf{x}, t) \quad \mathbf{x} \in \Gamma_{m,m'}, \quad \mathbf{n} \text{ is the outward facing normal to material } m$$

we have designed a correction to our connected centroid flux approximation (see Section 3.1.6) which is exact for the specific cases in which the exact solution is linear in space:

$$\theta(\mathbf{x}) = \theta_0 + \nabla \theta(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0).$$

Linear exact algorithm for interior flux approximation. Referring to Figure 3.12, the linear exact approximation for the material m interior flux on the left cell is:

$$q_{\text{int}} \approx \begin{cases} k_m \frac{g(\mathbf{x}_-^I) - \theta(\mathbf{x}_-)}{\|\mathbf{x}_- - \mathbf{x}_-^I\|} & \text{Dirichlet boundary condition} \\ g(\mathbf{x}_-^I) & \text{Neumann boundary condition} \end{cases}$$

Linear exact algorithm for exterior flux approximation.

Referring to Figure 3.12, the connected centroid flux approximation for q_{ext} is,

$$q_{\text{ext}} = k_m \frac{\theta(\mathbf{x}_+) - \theta(\mathbf{x}_-)}{\|\mathbf{x}_+ - \mathbf{x}_-\|} |\mathbf{n}' \cdot \mathbf{n}_{f,\text{ext}}|, \quad (3.22)$$

where

$$\mathbf{n}' \equiv \frac{\mathbf{x}_+ - \mathbf{x}_-}{\|\mathbf{x}_+ - \mathbf{x}_-\|}.$$

The approximation in (3.22) makes the assumption that

$$\theta(\mathbf{x}) \approx \theta(\mathbf{x}_-) + \frac{\theta(\mathbf{x}_+) - \theta(\mathbf{x}_-)}{\|\mathbf{x}_+ - \mathbf{x}_-\|} (\mathbf{x} - \mathbf{x}_-) \cdot \mathbf{n}' \quad (3.23)$$

The linear exact algorithm augments the approximation in (3.23) as follows,

$$\theta(\mathbf{x}) \approx \theta(\mathbf{x}_-) + \frac{\theta(\mathbf{x}_+) - \theta(\mathbf{x}_-)}{\|\mathbf{x}_+ - \mathbf{x}_-\|} (\mathbf{x} - \mathbf{x}_-) \cdot \mathbf{n}' + a_1 (\mathbf{x} - \mathbf{x}_-) \cdot \mathbf{t}_1 + a_2 (\mathbf{x} - \mathbf{x}_-) \cdot \mathbf{t}_2, \quad (3.24)$$

where \mathbf{t}_1 and \mathbf{t}_2 are orthogonal to each other and orthogonal to \mathbf{n}' . The coefficients a_1 and a_2 are determined from the following least squares algorithm:

- Pick the reconstructed plane with the largest surface area from the two adjoining cells; in Figure 3.12, the left plane has the largest surface area:

$$\Gamma_- = \left\{ \mathbf{x} | \mathbf{n}_{f,\text{int}} \cdot (\mathbf{x} - \mathbf{x}_-^I) = 0 \right\}$$

- Determine \mathbf{t}_1^I and \mathbf{t}_2^I which are orthogonal to each other and orthogonal to $\mathbf{n}_{f,\text{int}}$.

- Determine \mathbf{x}_2^I and \mathbf{x}_3^I on the plane Γ_- :

$$\begin{aligned}\mathbf{x}_2^I &= \mathbf{x}_-^I + \epsilon_T \|\mathbf{x}_- - \mathbf{x}_+\| \mathbf{t}_1^I \\ \mathbf{x}_3^I &= \mathbf{x}_-^I + \epsilon_T \|\mathbf{x}_- - \mathbf{x}_+\| \mathbf{t}_2^I\end{aligned}$$

Here, set $\epsilon_T = \frac{1}{100}$

- For the Dirichlet problem, define the following data points used for the least square procedure,

$$\begin{aligned}\mathbf{y}_1 &\equiv \mathbf{x}_2^I & \theta_1 &\equiv g(\mathbf{x}_2^I) \\ \mathbf{y}_2 &\equiv \mathbf{x}_3^I & \theta_2 &\equiv g(\mathbf{x}_3^I) \\ \mathbf{y}_3 &\equiv \mathbf{x}_-^I & \theta_3 &\equiv g(\mathbf{x}_-^I) \\ \mathbf{y}_4 &\equiv \mathbf{x}_+^I & \theta_4 &\equiv g(\mathbf{x}_+^I) \quad \text{if } \mathbf{x}_+^I \text{ exists.}\end{aligned}$$

For the Neumann problem, define the following data points used for the least square procedure,

$$\begin{aligned}\mathbf{y}_1 &\equiv \mathbf{x}_2^I & \theta_1 &\equiv \theta(\mathbf{x}_-) + \|\mathbf{x}_- - \mathbf{x}_-\| g(\mathbf{x}_2^I)/k_m \\ \mathbf{y}_2 &\equiv \mathbf{x}_3^I & \theta_2 &\equiv \theta(\mathbf{x}_-) + \|\mathbf{x}_- - \mathbf{x}_-\| g(\mathbf{x}_3^I)/k_m \\ \mathbf{y}_3 &\equiv \mathbf{x}_-^I & \theta_3 &\equiv \theta(\mathbf{x}_-) + \|\mathbf{x}_- - \mathbf{x}_-\| g(\mathbf{x}_-^I)/k_m \\ \mathbf{y}_4 &\equiv \mathbf{x}_+^I & \theta_4 &\equiv \theta(\mathbf{x}_+) + \|\mathbf{x}_+ - \mathbf{x}_+\| g(\mathbf{x}_+^I)/k_m \quad \text{if } \mathbf{x}_+^I \text{ exists.}\end{aligned}$$

- Solve the following optimization problem:

$$\min_{a_1, a_2} \sum_{j=1}^4 w_j \|\theta_j - \theta(\mathbf{y}_j)\|^2. \quad (3.25)$$

The weights are defined as:

$$w_j = \epsilon \quad j = 1, 2 \quad \epsilon = 1/100 \text{ for example} \quad (3.26)$$

$$w_3 = \frac{|\Gamma_-|}{|\Gamma_-| + |\Gamma_+|} \quad (3.27)$$

$$w_4 = \frac{|\Gamma_+|}{|\Gamma_-| + |\Gamma_+|} \quad (3.28)$$

- The flux will be calculated as $k_m \nabla \theta \cdot \mathbf{n}_{f,\text{ext}}$.

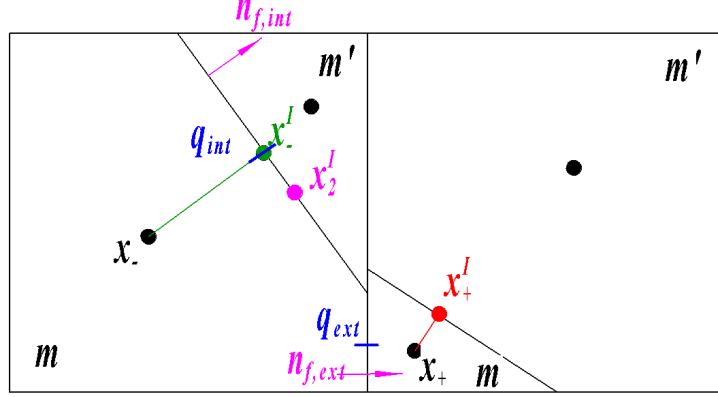


Figure 3.12: 2D Linear exact algorithm example

3.1.10 Solving the Linear System

Our new “linear exact” correction to the “connected centroid” flux algorithm, at Dirichlet and Neumann embedded interfaces, results in a non-symmetric linear system. It is proven in [100] that if there is no round-off error, and the matrix system is non-singular, the preconditioned BiCGStab method is guaranteed to converge in N steps where N is the number of unknowns. In our results section, we report the average number of (Jacobi Polynomial) preconditioned BiCGStab iterations needed in order that the L_2 norm of the residual is below $\epsilon_{BiCGStab} = 10^{-10}$.

3.2 Numerical Method on Stefan Problem

We have developed the following algorithm for numerically approximating solutions to the Stefan problem on a deforming supermesh (Figure 3.13):

3.2.1 Gap Error Analysis

Since the whole method is built on supermesh, and supermesh is achieved by piecewise linear MOF reconstruction. Interface gaps will appear from cell to cell which will influence the accuracy and the robustness of the overall method. We reported our way to measure the “gap error” in our previous paper [58] (see Section 4.1.1.1) and were able to show the gap error (previously defined as the length from the centroid of the gap to the interface represented in signed distance) is second order convergent to 0, as $\Delta x \rightarrow 0$, for smooth interfaces. In this section we give an alternate way to measure the gap error influence, “the area gap

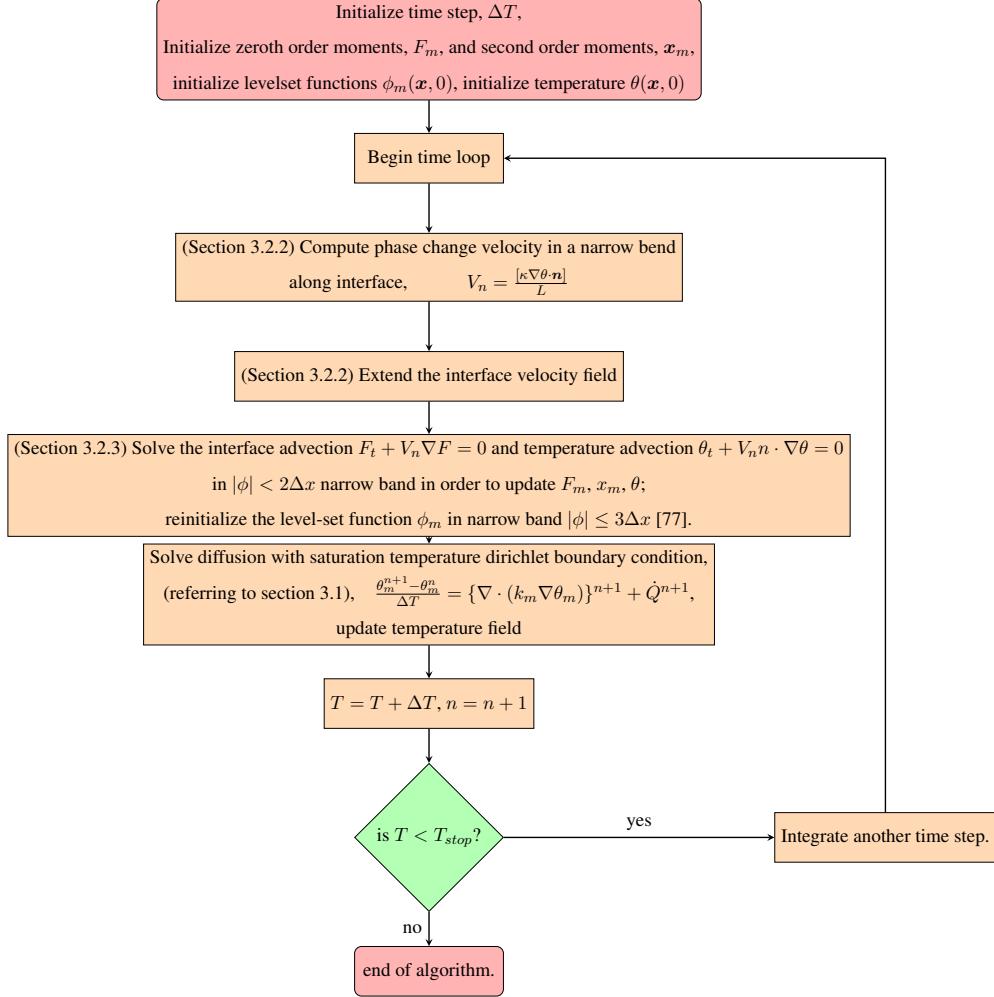


Figure 3.13: Flow chart corresponding to our Stefan problem algorithm.

error”,

$$(E_{gap})_\Omega = \max_{\substack{\text{all faces} \\ \text{of } \Omega}} \sum_{i=1}^{nmat} \sum_{\substack{j=1, \\ j \neq i}}^{nmat} A_{i,j}. \quad (3.29)$$

Figure 3.14 shows a piece of such “area gap error” on a face of the cell.

We use a simple case in Figure 3.14 to derive a relation between the reconstruction gap error E_{gap} and curvature of the reference arc (the exact interface before reconstruction) $\kappa = \frac{1}{R}$.

In the figure, the reference interface in the cell is an arc with radius R, the centroid of material 2 region is the purple point; the reconstructed interface is the horizontal blue line coincide with x axis centered at

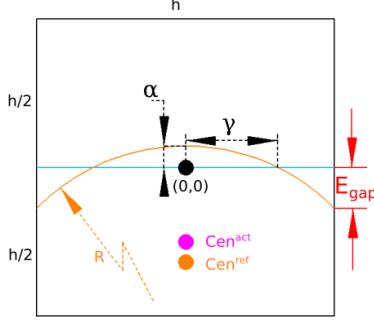


Figure 3.14: Example for an “area gap error”. The real interface in the cell is an arc with radius R , the centroid of material 2 region is the purple point; the reconstructed interface is the horizontal blue line coincide with x axis centered at $(0, 0)$ dividing the cell into half.

The gap error is indicated in the figure.

$(0, 0)$ dividing the cell into half. Starting from representing the reference interface with a function of y , we have the following derivation:

$$(y + (R - \alpha))^2 + x^2 = R^2 \quad (3.30)$$

$$y(x) = \sqrt{R^2 - x^2} - R + \alpha \quad (3.31)$$

Since the volume fraction stays the same after the MOF reconstruction,

$$\int_0^\gamma y dx = - \int_\gamma^{h/2} y dx \quad (3.32)$$

$$\sqrt{R^2 - \gamma^2} - R + \alpha = 0 \quad (3.33)$$

$$\alpha = R - \sqrt{R^2 - \gamma^2} \quad (3.34)$$

$$\int_0^\gamma \sqrt{R^2 - x^2} - \sqrt{R^2 - \gamma^2} dx = \int_\gamma^{h/2} \sqrt{R^2 - \gamma^2} - \sqrt{R^2 - x^2} dx \quad (3.35)$$

$$\int_0^{h/2} \sqrt{R^2 - x^2} - \sqrt{R^2 - \gamma^2} dx = 0 \quad (3.36)$$

$$\int_0^{h/2} \sqrt{R^2 - x^2} dx = h/2 \sqrt{R^2 - \gamma^2} \quad (3.37)$$

$$\text{Since, } \int \sqrt{a^2 - u^2} du = \frac{a^2}{2} \arcsin \frac{u}{a} + \frac{u}{2} \sqrt{a^2 - u^2} + C \quad (3.38)$$

$$\frac{R^2}{2} \arcsin \frac{h/2}{R} + \frac{h}{4} \sqrt{R^2 - \frac{h^2}{4}} = h/2 \sqrt{R^2 - \gamma^2} \quad (3.39)$$

Represent h and γ as multiples of R , $h = \beta R$, $\gamma = \sigma R$

$$\frac{R^2}{2} \arcsin \frac{\beta}{2} + \frac{\beta R}{4} \sqrt{R^2 - \frac{\beta^2 R^2}{4}} = (\beta R)/2 \sqrt{R^2 - \sigma^2 R^2} \quad (3.40)$$

$$2 \arcsin \frac{\beta}{2} + \beta \sqrt{1 - \frac{\beta^2}{4}} = 2\beta \sqrt{1 - \sigma^2} \quad (3.41)$$

$$\sqrt{1 - \sigma^2} = \frac{2 \arcsin \frac{\beta}{2} + \beta \sqrt{1 - \frac{\beta^2}{4}}}{2\beta} \quad (3.42)$$

Finally, evaluate value y at $x = h/2$

$$E_{Gap} = y(h/2) = \sqrt{R^2 - h^2/4} - \sqrt{R^2 - \gamma^2} \quad (3.43)$$

$$E_{Gap} = y(h/2) = R |\sqrt{1 - \beta^2/4} - \sqrt{1 - \sigma^2}| \quad (3.44)$$

Plug in Equation 3.42

$$E_{Gap} = R |\sqrt{1 - \beta^2/4} - \frac{2 \arcsin \frac{\beta}{2} + \beta \sqrt{1 - \frac{\beta^2}{4}}}{2\beta}| \quad (3.45)$$

$$E_{Gap} = R |\frac{1}{2} \sqrt{1 - \beta^2/4} - \frac{\arcsin \frac{\beta}{2}}{\beta}| \quad (3.46)$$

$$\text{Approximate } \arcsin(x) \approx x + \frac{x^3}{6}$$

$$\text{Approximate } \sqrt{1 - x^2} = \sqrt{1 - x} \sqrt{1 + x} = (1 + \frac{x}{2} - \frac{x^2}{8} + \dots)(1 - \frac{x}{2} - \frac{x^2}{8} + \dots) \approx 1 - \frac{x^2}{2}$$

$$E_{Gap} \approx R |\frac{1}{2} \left(1 - \frac{\beta^2}{8}\right) - \frac{\frac{\beta}{2} + \frac{\beta^3}{48}}{\beta}| \quad (3.47)$$

$$E_{Gap} \approx R |\frac{1}{2} \left(1 - \frac{\beta^2}{8}\right) - \frac{1}{2} - \frac{\beta^2}{48}| \quad (3.48)$$

$$E_{Gap} \approx \frac{\beta^2}{12} R \quad (3.49)$$

The linear exact hypothesis assumes that interface is linear relative to the mesh size. It is conjectured that the linear exact algorithm for approximating the flux is no better than the “simple method” when curvature radius is below some threshold $R = \eta h$, since E_{gap} is very large due to the large curvature of the real interface. Thus,

$$E_{Gap} \approx \frac{\beta^2}{12} R = \frac{\frac{h^2}{R^2}}{12} R = \frac{h^2}{12R} = \frac{h^2}{12\eta h} = \frac{h}{12\eta} \quad (3.50)$$

We believe that we can use this threshold to identify if we need to switch between “linear exact” and “connected centroid” flux calculation algorithm during the computing in order to increase the accuracy of our overall algorithm performance. In addition, it is easy for us to implement this approach since we have

already developed an algorithm in the code to detect and visualize where large area gap errors occur (see Figure 4.33). The choice of this threshold is still quite empirical right now, and it needs more future work.

3.2.2 Solve the Interface Front Velocity

During the phase change process, the deforming interface propagates with phase front velocity V_n . An advection equation is solved for updating the volume fractions and centroids.

The phase front velocity depends on the temperature gradient jump across the interface (see Equation 2.5). For computing the phase front velocity, we either use (A) the standard “macro-scale” normal probe method or (B) the “micro-scale” normal probe method.

3.2.2.1 “Macro-scale” normal probe method for determining the interface velocity.

The “macro-scale” normal probe method for approximating the interface velocity is as follows:

1. For each cell, we check the level set function ϕ_{vap} , if $|\phi_{vap}| \leq 2\Delta x$ or $|\phi_{liq}| \leq 2\Delta x$, then we mark this cell as active.
2. For an active cell $\Omega_{i,j}$, if either $\phi_{i,j}^{liq} > 0$ or $\phi_{i,j}^{vap} > 0$, we find the closest point $\vec{x}_{closest}$ on the interface from the cell center (See Figure 3.15).

$$\vec{x}_{closest} = \vec{x}_{i,j} - \phi_{i,j,p} \vec{n}_{i,j,p} \quad p = liq \text{ or } vap \quad (3.51)$$

3. From $\vec{x}_{closest}$ on the liquid-vapor interface, we draw two normal probes, each one of which points to a different phase (liquid or vapor) with magnitude of $\Delta x/2$. Then we have \vec{x}_{liq}^{probe} , and \vec{x}_{vap}^{probe} .
4. Locate the target cell that \vec{x}_i^{probe} (i represents vap or liq) is in. Check the 3×3 stencils centered at the target cell with $F_i > 0$, use least square method to interpolate the T^i 's at the centroids for each cell to \vec{x}_i^{probe} to get T_i^{probe} (Figure 3.15).
5. If $|\phi_{liq}(\vec{x}_{closest})| < \Delta x$, $|\phi_{vap}(\vec{x}_{closest})| < \Delta x$, $\phi_{liq}(\vec{x}_{liq}^{probe}) > 0$, $\phi_{vap}(\vec{x}_{vap}^{probe}) > 0$ and $\phi(\vec{x}_{closest}) < \min(\phi_{vap}(\vec{x}_{closest}), \phi_{liq}(\vec{x}_{closest}))$. The temperature gradient is calculated as (Figure 3.15)

$$\nabla T_i = \frac{T_i^{probe} - T_{sat}}{\Delta x/2} \quad (3.52)$$

and the corresponding velocity V_{I_p} stored on cell $\Omega_{i,j}$ center will be

$$V_n^{i,j} = \frac{\dot{m}}{\rho_v} = \frac{[\kappa \nabla T] \cdot (\vec{n}_{lv})}{\rho_v L}.$$

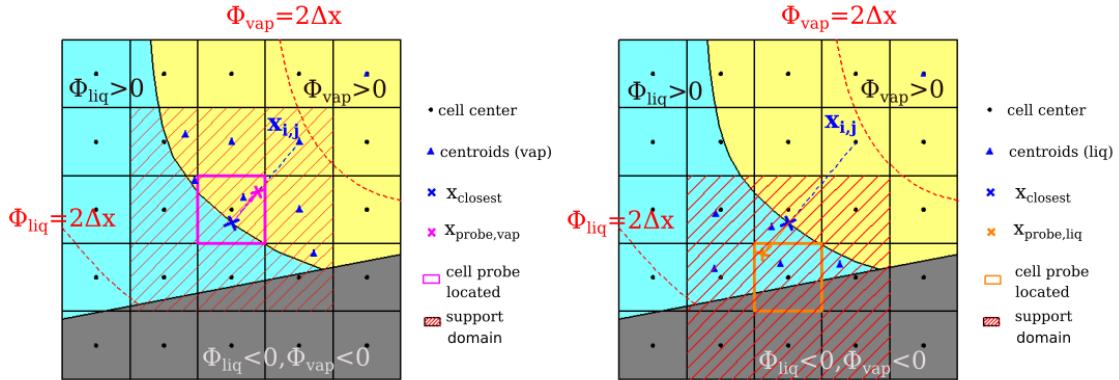


Figure 3.15: Phase change velocity calculation: find corresponding interpolation support domain and calculate the temperature gradient jump across the phase change front.
Left: support domain for the vapor phase; Right: support domain for the liquid phase

3.2.2.2 “Micro-scale” normal probe method for determining the interface velocity.

For calculating the interface velocity based on temperature gradient near the filamentary region, as shown in Figure 3.16, the normal probe $x_{probe,2}$ falls into material 3 region instead of material 2 region. In such case, we make the following algorithm changes.

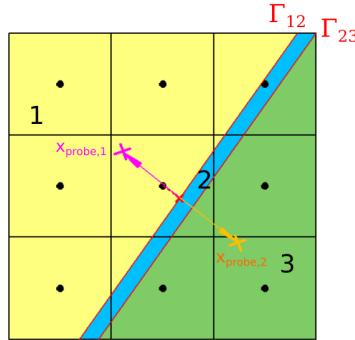


Figure 3.16: Phase change velocity near filament region
Since the macro probe, $x_{probe,2}$ is in material 3, we use the micro probe (immediately next to the red x) that is in material 2.

We introduce the “micro probes”

$$\mathbf{x}_{probe}^{\pm} = \mathbf{x}_{closest} \pm \epsilon \Delta x \mathbf{n}, \quad \epsilon = 10^{-2}. \quad (3.53)$$

If \mathbf{x}_{probe}^m is not in material “ m ”, then we find the cell containing $\mathbf{x}_{probe,micro}^m$ and the normal gradient is then

$$(\kappa \nabla \theta \cdot \mathbf{n})^\pm = \kappa^\pm \frac{\theta^\pm(\mathbf{x}_{centroid}^m) - \theta(\mathbf{x}_{closest})}{\|\mathbf{x}_{centroid}^m - \mathbf{x}_{closest}\|}. \quad (3.54)$$

Note: we also limit the phase change velocity as $|u^{phasechange}| < \frac{\Delta x}{\Delta T}$ in case there is stability issue near the filamentary region.

3.2.3 Solve the Advection Equation by Using Un-split Cell-Integrated Semi-Lagrangian Method

After the phase front velocity field is computed in the vicinity (narrow band) of the interface as in last section, the advection equation is ready to be solved.

First, we extend the phase change front velocity V_n (See Figure 3.17). Check the cells that have not been assigned a valid interface velocity yet. If $|\phi_{vap}| \leq 4\Delta x$ or $|\phi_{liq}| \leq 4\Delta x$, we find $\vec{x}_{closest}$ to either the vapor interface or the liquid interface. Take the one with the larger magnitude. After Locating the cell, we create a 3×3 stencil around it, then use the weighted average of all the velocities among the active cells as the interface velocity for the extended cell. Then, we interpolate V_n values (the velocity field) from the

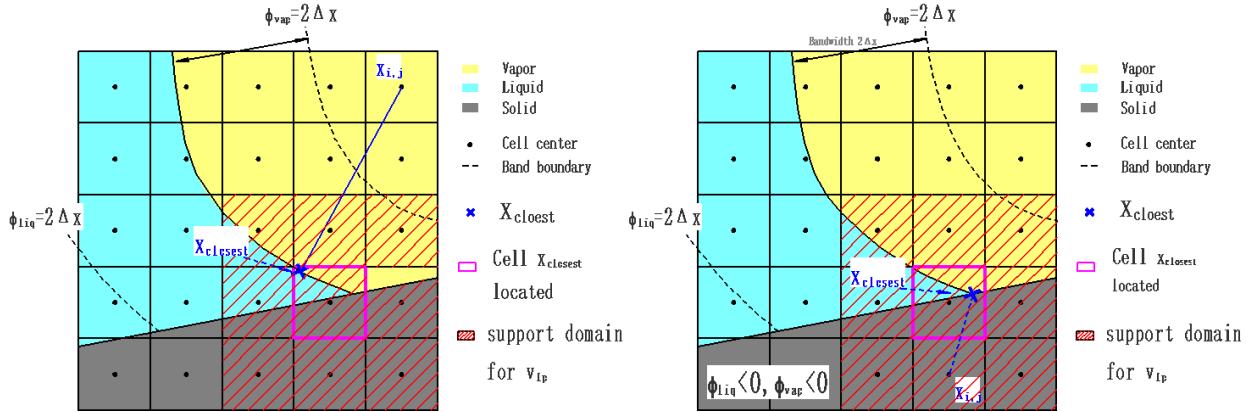


Figure 3.17: Phase change velocity calculation: velocity extension in cells ($2\Delta x < \phi_{liq} < 4\Delta x$ or $2\Delta x < \phi_{vap} < 4\Delta x$) and solid cells.

Left: $x_{i,j}$ in vapor occupied cell within $4\Delta x$. Right: $x_{i,j}$ in solid cell

cell centers to the cell nodes (four cell vertices in 2D). We apply the un-split Cell-integrated Semi-Lagrange (CISL) method to solve the advection equation 2.4 and update the volume fractions and centroids as phase change front moves.

The target region (a cell in current time step to be solved) will be traced back to the previous time step over a set of characteristic paths base on the velocity field stored at the vertices. State variables in a cell are updated using information from all sub-regions of the grid that map into that cell at the previous time step. Various methods exist for updating state variables using CISL advection [60, 72, 12]. CISL advection methods are natural for multiphase simulations because exact conservation of mass, momentum, and energy is trivial. Because the CISL advection method updates cell variables based on the dynamic intersection of the departure region or target region with the mesh, this advection strategy is not subject to the strict CFL restriction for stability. Figure 3.18 illustrates one 2D example of un-split CISL method.

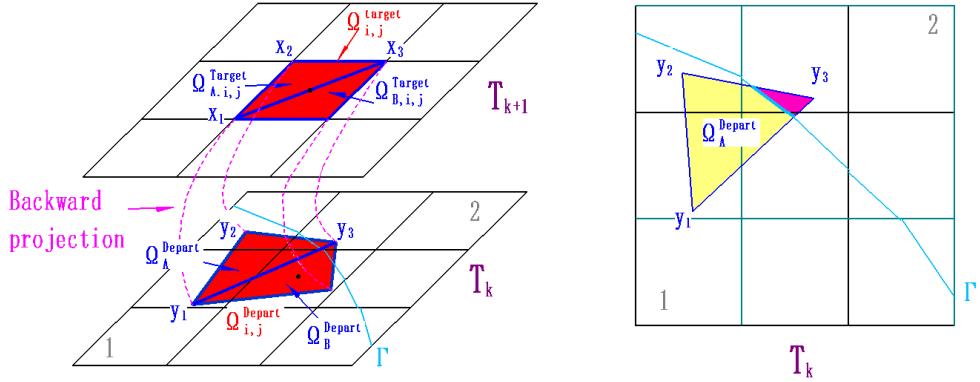


Figure 3.18: Illustration of 2D un-split Cell-Integrated Semi-Lagrangian (CISL) method:
Left: Backward project (characteristic tracing of the value at the four vertices according to the velocity) the center cell $\Omega_{i,j}^{target}$ at time T^{k+1} onto the domain at time T^k Right: Show the intersection of P_1^{depart} with the T^k domain

Remark: as shown in Figure 3.19, in the center cell which contains a triple point, interface Γ_3 is stationary that means no phase change occurs between material 3 and the rest two materials. Material 1 is the “source” material and material 2 is the “destination” material. In order to keep mass preserved , we need to check volume change of material 1 and 2 and make a correction if needed (referring to Algorithm 2).

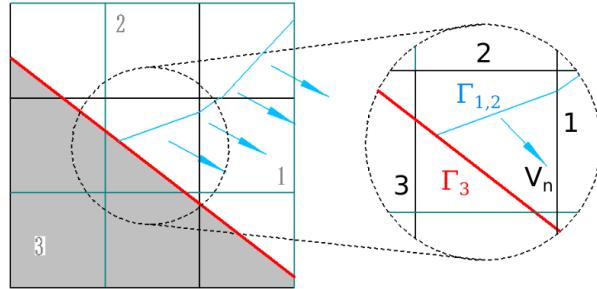


Figure 3.19: Volume fraction correction after advection in > 2 material cells

Algorithm 2 Volume fraction check algorithm (referring to Figure 3.19).

- 1: Input: $F_{dest}^{old}, F_{src}^{old}, F_{src}^{new,temp}, F_{dest}^{new,temp}$
 - 2: Output: $F_{dest}^{new,corrected}, F_{src}^{new,corrected}$
 - 3: $\Delta F_i = F_i^{new,temp} - F_i^{old}$
 - 4: $\Delta F = \min(|\Delta F_{dest}|, |\Delta F_{src}|)$
 - 5: If $F_{src}^{new,temp} < 0$ Then: $\Delta F = F_{src}^{old}$
 - 6: If $F_{dest}^{new,temp} > 1$ Then: $\Delta F = 1 - F_{dest}^{old}$
 - 7: $F_{dest}^{new,corrected} = F_{dest}^{old} + \Delta F$
 - 8: $F_{src}^{new,corrected} = F_{src}^{old} - \Delta F$
-

CHAPTER 4

NUMERICAL RESULTS

4.1 Multi-material Diffusion

In each numerical test, the computational domain is set to be $\Omega_D = [0, 1] \times [0, 1]$. We will indicate the interface set-up, boundary conditions, initial conditions and material properties in each test case. The L_1 , L_2 , L_∞ errors presented in each case are all calculated at the final time.

4.1.1 Error Calculation

4.1.1.1 Multi-material piecewise linear reconstruction validation; the “gap-error.” . We have implemented an algorithm for verifying, using an L_∞ measure, the accuracy of (1) the MOF reconstruction algorithm, (2) the initialization of volume fractions and centroids, (3) area fractions on either side of external or internal boundaries, and (4) the centroids of the piecewise linear MOF reconstruction gap regions, as in Figure (4.1). We have found experimentally that the Moment-of-Fluid reconstruction satisfies the following

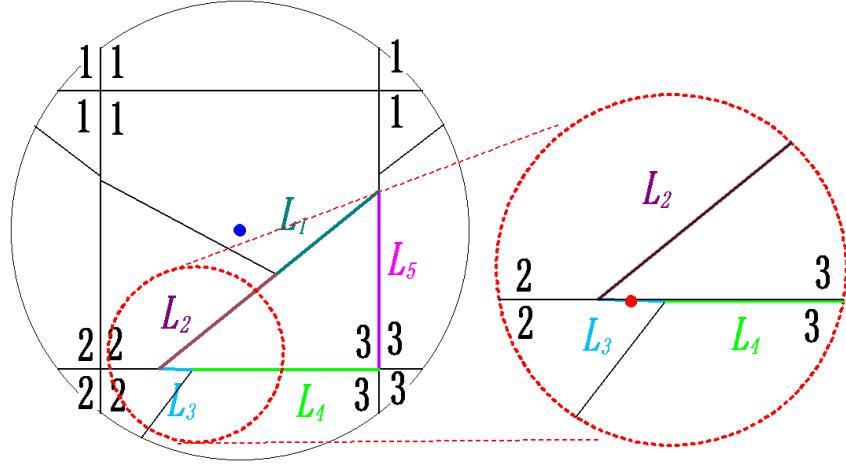


Figure 4.1: Interface gap error: E_{gap} .

Find the centroid of each small gap and substitute into the exact signed distance function
then calculate the error.

L_∞ error estimate,

$$E_{\text{gap}} \equiv \max_{\text{Gaps}} \min_{\boldsymbol{x} \in \Gamma_{m_1, m_2}} \|\boldsymbol{x} - \boldsymbol{x}_{\text{Gap}}^{\text{centroid}}\| < C \Delta x^2,$$

assuming appropriate regularity requirements of Γ_{m_1, m_2} . We report the gap error, E_{gap} , in Tables 4.20 and 4.21.

4.1.1.2 Temperature error and relative error calculation. Given the exact temperature $\theta_{i,j}^{\text{exact}}$, the error will be calculated as,

$$\theta_{\text{err},p} = \left(\frac{\sum_{i,j} |\theta_{i,j} - \theta_{i,j}^{\text{exact}}|^p F_{i,j} \Delta x \Delta y H_{i,j}}{\sum_{i,j} \Delta x \Delta y H_{i,j}} \right)^{\frac{1}{p}} \quad (4.1)$$

where, $H_{i,j} = \begin{cases} 1, & F_{i,j} \geq F^{\text{meas}}, \\ 0, & \text{otherwise.} \end{cases}$, F^{meas} is a prescribed volume fraction cutoff in which we only calculate the error for $F \geq F^{\text{meas}}$. F^{meas} equals one for non-filamentary test problems and $F^{\text{meas}} = 0.01$ for filamentary test problems.

When the exact temperature is unknown, we will report the relative error instead. For measuring the relative errors, only full material cells (cells that are fully occupied by a single material) are considered.

In order to compute the relative error $L_{2,h,2h}$, we perform the following steps:

1. Restrict the approximate solution on grid "h" (a.k.a "fine") to grid "2h" (a.k.a "coarse").
2. The restricted volume fraction RF^h is calculated as

$$(RF^h)_{ic,jc} = \frac{\sum_{i'=0}^1 \sum_{j'=0}^1 F_{2(ic)+i', 2(jc)+j'}^h}{4}. \quad (4.2)$$

3. The restricted temperature $R\theta^h$ is calculated as

$$(R\theta^h)_{ic,jc} = \frac{\sum_{i'=0}^1 \sum_{j'=0}^1 \theta_{2(ic)+i', 2(jc)+j'}^h}{4(RF^h)_{ic,jc}}. \quad (4.3)$$

4. Then the relative error will be calculated as

$$L_{p,h,2h} = \left(\frac{\sum_{ic} \sum_{jc} H_{ic,jc} |(R\theta^h)_{ic,jc} - (\theta^{2h})_{ic,jc}|^p}{\sum_{ic} \sum_{jc} H_{ic,jc}} \right)^{\frac{1}{p}}, \quad (4.4)$$

where,

$$H_{ic,jc} = \begin{cases} 1, & \text{if } (RF^h)_{ic,jc} = 1 \text{ and } F_{ic,jc} = 1, \\ 0, & \text{otherwise.} \end{cases} \quad (4.5)$$

4.1.1.3 Gradient error and gradient relative error calculation. We use Equation 4.2 for calculating the gradient between two cells for material m .

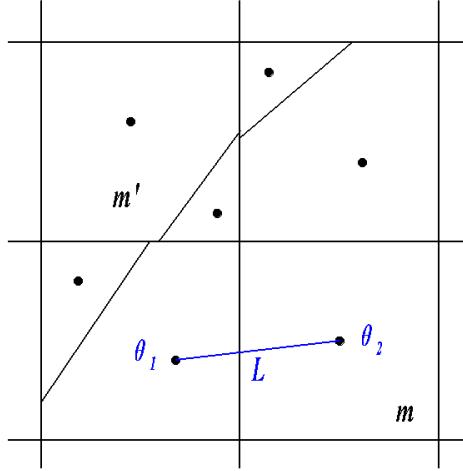


Figure 4.2: Example for gradient error calculation

$$\nabla \theta_{err,p} = \left(\frac{\sum_{i,j} |\nabla \theta_{i,j} - \nabla \theta_{i,j}^{exact}|^p F_{i,j} \Delta x \Delta y H_{i,j}}{\sum_{i,j} \Delta x \Delta y H_{i,j}} \right)^{\frac{1}{p}}, \quad (4.6)$$

where, $H_{i,j} = \begin{cases} 1, & F_{i,j} \geq F^{meas}, \\ 0, & \text{otherwise.} \end{cases}$

For example in Figure (4.2), $\nabla \theta = \frac{\theta_2 - \theta_1}{L}$, $\nabla \theta^{exact} = \frac{\theta_2^{exact} - \theta_1^{exact}}{L}$, L is distance between the two centroids. For gradient relative error, only material full cells are considered (i.e. $F^{meas} = 1.0$). The

restricted gradient will be calculated as

$$(R\nabla\theta^h)_{ic,jc} = \left(\frac{\sum_{j'=0}^1 (\theta_{2(ic)+1,2(jc)+j'}^h - \theta_{2(ic),2(jc)+j'}^h)}{2\Delta x}, \frac{\sum_{i'=0}^1 (\theta_{2(ic)+i',2(jc)+1}^h - \theta_{2(ic)+i',2(jc)}^h)}{2\Delta y} \right) \quad (4.7)$$

$$(\nabla\theta^{2h})_{ic,jc} = \left(\frac{(\theta_{ic+1,jc}^{2h} - \theta_{ic-1,jc}^{2h})}{2\Delta x}, \frac{(\theta_{ic,jc+1}^{2h} - \theta_{ic,jc-1}^{2h})}{2\Delta y} \right) \quad (4.8)$$

Similarly, we calculate the gradient relative error as follows,

$$L_{p,h,2h} = \left(\frac{\sum_{ic} \sum_{jc} H_{ic,jc}^{grad} |(R\nabla\theta^h)_{ic,jc} - (\nabla\theta^{2h})_{ic,jc}|^p}{\sum_{ic} \sum_{jc} H_{ic,jc}} \right)^{\frac{1}{p}} \quad (4.9)$$

where,

$$H_{ic,jc}^{grad} = \begin{cases} 1, & \text{if } (F^h)_{ic,jc} = (F^h)_{ic+1,jc} = (F^h)_{ic,jc+1} = (F^h)_{ic+1,jc+1} = 1 \text{ and } F_{ic,jc}^{2h} = 1, \\ 0, & \text{otherwise.} \end{cases} \quad (4.10)$$

4.1.2 Annulus Diffusion Test with Three Materials

Table 4.1: Problem set-up for annulus test

Test	Boundary conditions of computational domain	Material geometry	Boundary conditions on the interfaces	Diffusion coefficient
Test A	homogeneous Dirichlet	Refer to Figures 4.3 and 4.4	Homogeneous Neumann on Γ_1 and Γ_2	$\kappa_1 = \kappa_3 = 0$ $\kappa_2 = 1.0$
Test B	homogeneous Dirichlet	Refer to Figures 4.3 and 4.4	Inhomogeneous Dirichlet on Γ_1 and Γ_2	$\kappa_1 = \kappa_3 = 0$ $\kappa_2 = 1.0$

As shown in Figure 4.3, the whole domain is divided into 3 parts with two interfaces (two concentric circles) which are centered in the middle of the domain at $(0.5, 0.5)$ and defined as,

$$\Gamma_1 : (x - 0.5)^2 + (y - 0.5)^2 = r1$$

$$\Gamma_2 : (x - 0.5)^2 + (y - 0.5)^2 = r2$$

In order to investigate the convergence rate of the method, we use the manufactured solution method.

Assume we have a real solution for the diffusion problem (polar coordinates):

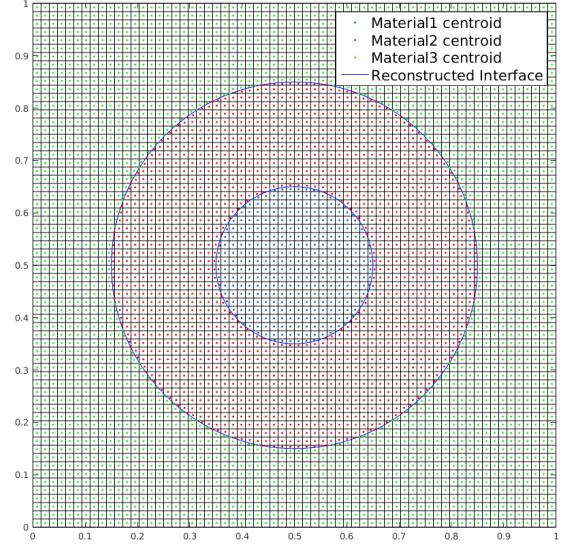
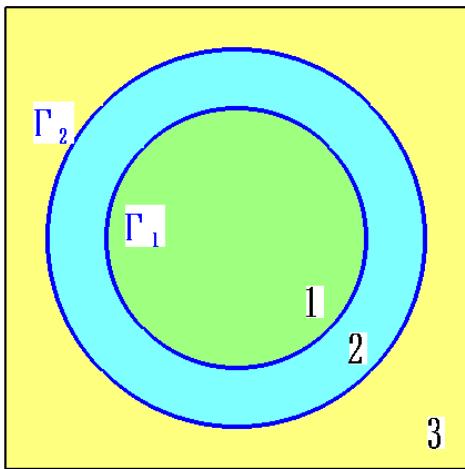


Figure 4.3: Left: a figure indicating interfaces and material ID.
 Right: MOF reconstruction and centroids for different materials of thick annulus ($\delta = 0.2$ and then
 $r_1 = 0.15, r_2 = 0.35$)
 mesh size 64×64

$$(u_{exact})_t = \kappa((u_{exact})_{rr} + \frac{1}{r}(u_{exact})_r + \frac{1}{r^2}((u_{exact})_{\theta\theta}) + F(r, \theta, t) \quad \kappa = 1.$$

We assume the exact solution is:

$$u_{exact}(r, \theta, t) = \begin{cases} 2 + \sin(\theta)e^{-t/(0.25)^2} & \text{Neumann condition} \\ 2 + \frac{r-0.25}{\delta} + \sin(\theta)e^{-t/(0.25)^2} & \text{Dirichlet condition} \end{cases}$$

After substituting the manufactured solution into the diffusion problem, we find the source term F is,

$$F(r, \theta, t) = (-1/(0.25)^2 + 1/r^2) \sin(\theta)e^{-t/(0.25)^2}$$

The convergence results for the annulus test are reported below in sections 4.1.2.1 and 4.1.2.2.

4.1.2.1 Test A: homogeneous Neumann boundary condition on Γ_1 and Γ_2 . We show the temperature profile at $T = 0.0125$ in Figure 4.5 and report the convergence study results in Tables 4.2 and 4.3. See also Figure 4.6. For the homogeneous Neumann boundary condition test, our method doesn't converge on

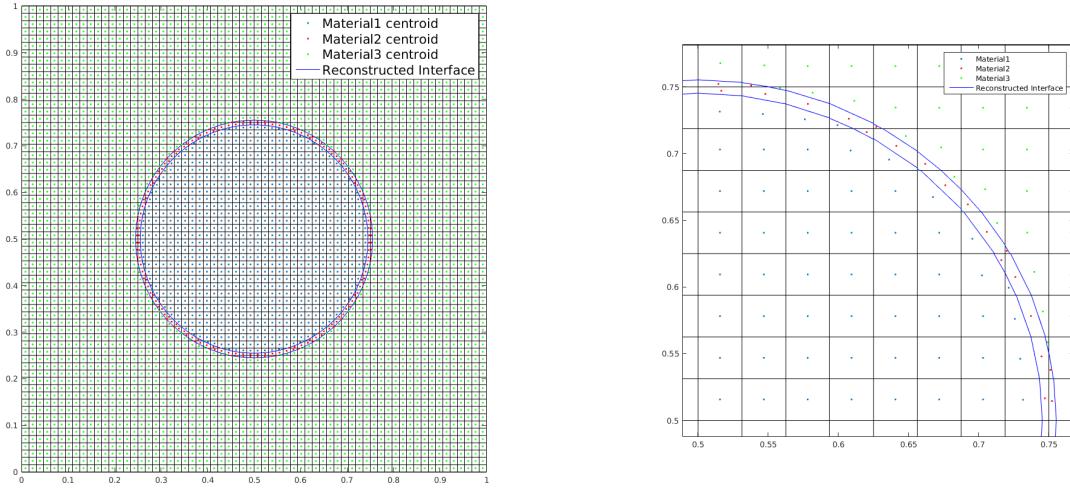


Figure 4.4: MOF reconstruction and centroids for different materials of thin annulus ($\delta = 0.01$ and then $r_1 = 0.245, r_2 = 0.255$).
mesh size 64×64

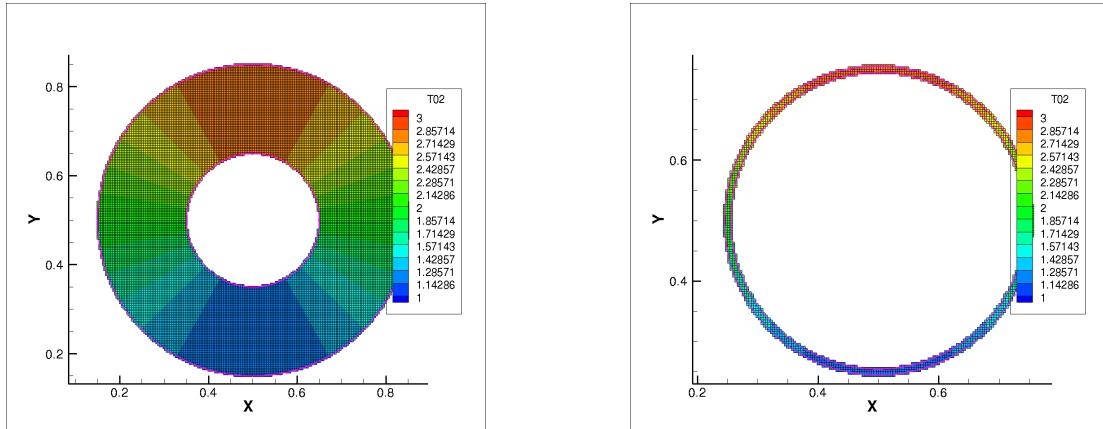


Figure 4.5: Temperature Diffusion in the thick and thin annulus at time $T = 0.0125$. The embedded boundary conditions for the annular region are homogeneous Neumann. The grid resolution is 256×256 .
The contour plot is “blanked out” in the regions where the material two volume fractions are zero.

the thin filament case at the beginning. But as the grid is refined and when the thin filament is resolved by the grid, the method starts to converge. We note that even in the under-resolved case, the L_1 error is around 1×10^{-2} which is quite small.

Table 4.2: Convergent study, thick annulus ($\delta = 0.2$), homogeneous Neumann Boundary, $T_{stop} = 0.0125$, space and time refinement ratios are 2 and 4 respectively; “Linear Exact” method is used, $F^{\text{meas}} = 1$.

N	M	L_1 err	L_2 err	L_∞	L_1^{grad}	L_2^{grad}	L_∞^{grad}	iter.
32	1	9.6E-3	1.1E-2	1.6E-2	2.5E-2	3.2E-2	8.4E-2	17
64	4	2.6E-3	2.9E-3	4.4E-3	6.8E-3	8.8E-3	2.7E-2	15
128	16	6.7E-4	7.5E-4	1.1E-3	1.8E-3	2.3E-3	1.1E-2	15
256	64	1.7E-4	1.9E-4	2.8E-4	4.5E-4	5.8E-4	3.1E-3	19

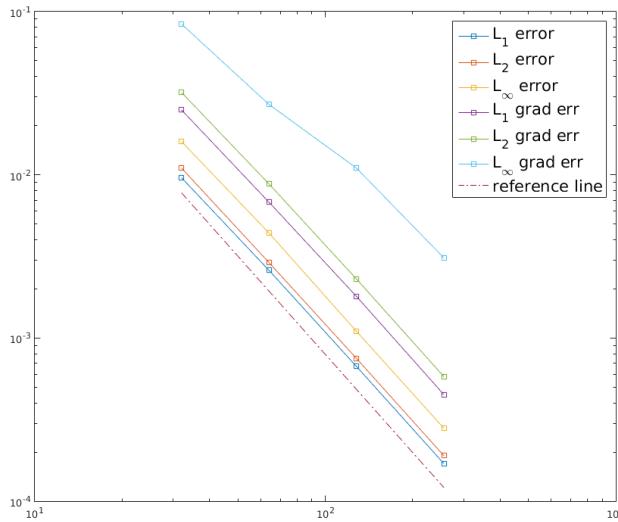


Figure 4.6: Loglog plot referring to Table 4.2

Table 4.3: Convergent study, thin annulus ($\delta = 0.01$), homogeneous Neumann Boundary, $T_{stop} = 0.0125$, space and time refinement ratios are both 2; “Linear Exact” method is used, $F^{\text{meas}} = 0.01$.

N	M	L_1 err	L_2 err	L_∞	L_1^{grad}	L_2^{grad}	L_∞^{grad}	iter
32	1	1.1E-2	1.3E-2	2.5E-2	2.5E-1	3.9E-1	1.2	22
64	2	1.0E-2	1.3E-2	2.7E-2	2.7E-1	3.4E-1	9.7E-1	31
128	4	8.4E-3	1.0E-2	2.4E-2	1.4E-1	1.9E-1	5.9E-1	37
256	8	1.3E-3	1.5E-3	2.1E-3	4.3E-3	5.8E-3	5.5E-2	55

4.1.2.2 Test B: inhomogeneous Dirichlet boundary condition on Γ_1 and Γ_2 . For the inhomogeneous Dirichlet boundary test, our “linear exact” method is second order accurate for the temperature and temperature gradient field for the thick annulus problem (see Table 4.4 and Figure 4.7). We note that the

interface flux converges with only first order accuracy, but this is because our method for extracting the interface flux from the temperature field is only a first order method (see Section 4.1.3.3).

For the thin annulus problem, neither the “connected centroid” method (Table 4.6) nor the “linear exact” approach (Table 4.5) converges for grids in which the mesh size is smaller than the filament thickness. Once the filament thickness is resolved by the grid, then the “linear exact” approach exhibits much faster convergence for both the temperature and temperature gradient than the “connected centroid” method.

Table 4.4: Convergence study, thick annulus ($\delta = 0.2$), inhomogeneous Dirichlet Boundary, $T_{stop} = 0.0125$, space and time refinement ratios are 2 and 4 respectively; “Linear Exact” method is used, $F^{\text{meas}} = 1$.

N	M	L_1 err	L_2 err	L_∞	L_1^{grad}	L_2^{grad}	L_∞^{grad}	$L_\infty^{\text{int. flux}}$	iter
32	1	2.6E-3	2.9E-3	5.5E-3	1.6E-2	2.1E-2	6.6E-2	1.2E-1	10
64	4	7.1E-4	8.2E-4	1.7E-3	5.7E-3	7.3E-3	2.8E-2	4.8E-2	12
128	16	1.7E-4	2.0E-4	4.2E-4	1.6E-3	2.1E-3	8.7E-3	2.4E-2	14
256	64	4.3E-5	5.1E-5	1.1E-4	4.2E-4	5.5E-4	2.7E-3	1.3E-2	16

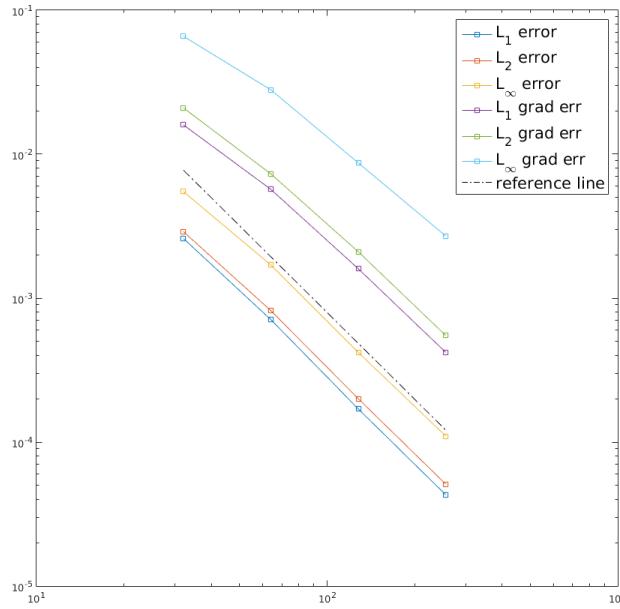


Figure 4.7: Log log plot referring to Table 4.4

Table 4.5: Convergence study, thin annulus ($\delta = 0.01$), inhomogenous Dirichlet Boundary, $T_{stop} = 0.0125$, space and time refinement ratios are both 2; “Linear Exact” method is used, $F^{\text{meas}} = 0.01$.

N	M	L_1^{err}	L_2^{err}	L_∞	L_1^{grad}	L_2^{grad}	L_∞^{grad}	iter
32	1	4.2E-4	6.9E-4	1.8E-3	3.8E-2	1.0E-1	5.4E-1	5
64	2	5.4E-4	7.1E-4	1.9E-3	6.5E-2	1.0E-1	4.5E-1	7
128	4	1.8E-4	2.5E-4	6.6E-4	3.6E-2	5.4E-2	1.9E-1	8
256	8	3.0E-6	3.6E-6	7.5E-6	4.7E-4	8.0E-4	3.6E-3	12

Table 4.6: Convergence study, thin annulus ($\delta = 0.01$), inhomogeneous Dirichlet Boundary, $T_{stop} = 0.0125$, space and time refinement ratios are both 2; “connected centroid” method is used, $F^{\text{meas}} = 0.01$.

N	M	L_1^{err}	L_2^{err}	L_∞	L_1^{grad}	L_2^{grad}	L_∞^{grad}	iter
32	1	1.1E-3	1.4E-3	3.3E-3	7.6E-2	1.3E-1	5.6E-1	2
64	2	4.4E-4	6.3E-4	1.7E-3	6.2E-2	9.5E-2	4.0E-1	2
128	4	2.0E-4	3.1E-4	1.2E-3	4.2E-2	6.0E-2	2.1E-1	2
256	8	8.6E-5	1.3E-4	6.2E-4	3.3E-2	5.4E-2	3.3E-1	3

4.1.3 Convergence Study for the Heat Flux through a Material Boundary

We further test our method by performing a grid refinement study for the computed heat flux through a material (embedded) boundary. We are motivated to perform this test since the rate of mass transfer for a material undergoing phase change is often governed by the jump in heat flux at a material interface [83, 97].

We still use the same thick annulus geometry set-up from the previous test. In order to check the error in approximating the flux through the interfaces Γ_1 and Γ_2 (the inner and outer circle), we developed a second order, body fitted, heat equation solver and used the gradient approximation of our body fitted solver as the “ground truth” exact solution. The body fitted, polar coordinates numerical method for measuring the “ground truth” heat flux through a material boundary was found to be sufficiently converged on a 256×512 polar coordinates grid (see Tables 4.7, 4.8).

4.1.3.1 Body fitted, polar coordinates, numerical method for measuring the “ground truth” heat flux through a material boundary. For determining the rate of convergence for our finite volume moment of fluid (FV-MOF) method, we find a “ground truth” approximation to the solutions of the heat equation in polar coordinates by using a body fitted method,

$$u_t = \kappa(u_{rr} + \frac{1}{r}u_r + \frac{1}{r^2}u_{\theta\theta}), \quad (4.11)$$

with initial condition,

$$u(r, \theta) = u_1 \frac{r_2 - r}{r_2 - r_1} + u_2 \frac{r - r_1}{r_2 - r_1} + 100 \sin \theta (r - r_1)(r_2 - r). \quad (4.12)$$

The boundary conditions are

$$u(x, y) = \begin{cases} u_1 & (x, y) \in \Gamma_1 \\ u_2 & (x, y) \in \Gamma_2 \end{cases} \quad (4.13)$$

The body fitted “ground truth” numerical method, which approximates solutions to (4.11) is discretized in time and space using the forward Euler and central differencing schemes respectively.

4.1.3.2 Deriving the heat flux through a material boundary from the results of our new FV-MOF

method. We use the following steps to approximate the interface gradient as derived from the temperature profile that is given from our FV-MOF method (see Figure 4.8 for the specific scenario where we derive the interfacial heat flux for material 2, the annulus region):

1. For each cell Ω with $F_2 > \epsilon$ (i.e. volume fraction of material 2 is bigger than a small tolerance), we find the signed distance from cell center to the interface $\phi_{i,j}$.
2. If the absolute value of the signed distance is within $2\Delta_x$ (Δ_x is the grid size), mark the cell as active and find the closet point x_I on the interface to the cell center.
3. Find the normal probe x_{probe} (diamonds in the figure) from the closet point x_I with probe length Δ_x .
4. Interpolate the temperature that is approximated using our FV-MOF method (temperature stored at material centroids) in order to find the temperature at the normal probe θ_{probe} by using the algorithm in Section 4.1.3.3.
5. Since the temperature at the interface satisfies the Dirichlet boundary condition $\theta = \theta_D$, we approximate the temperature gradient across the interface as $\nabla \theta \cdot \mathbf{n} \approx \frac{\theta_{probe} - \theta_D}{\Delta_x}$

We studied the sensitivity of the convergence rates admitted by our FV-MOF method with respect to the resolution used to find our “exact” solution using a body fitted method and have the following convergence analysis (see Tables 4.7, 4.8).

We can observe from the three tables, for each resolution used for the body fitted polar coordinates solver, the L_2 and infinity norm of the flux error are both convergent. As we refine the grid used by the body fitted polar coordinates solver, the convergence rate approaches 1.0.

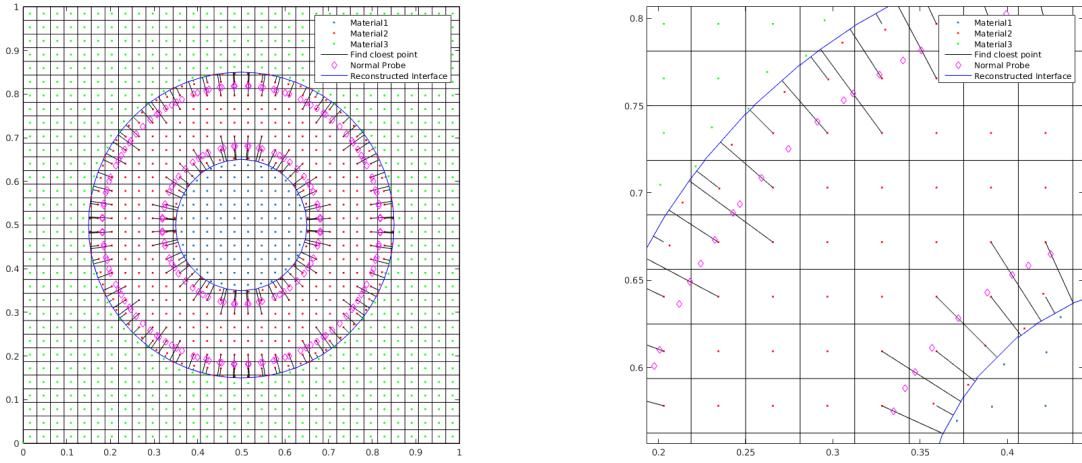


Figure 4.8: Illustration of normal probe algorithm for deriving the interfacial heat flux from the FV-MOF temperature field approximation
mesh size 64×64

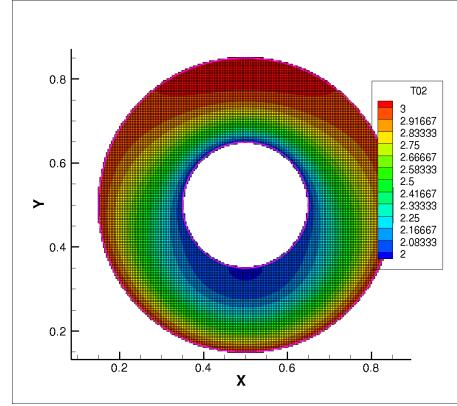


Figure 4.9: $\theta_1 = 2.0$, $\theta_2 = 3.0$, temperature diffusion at time 0.004, grid resolution 256×256

. The contour plot is “blanked out” in the regions where the material two volume fractions are zero.

4.1.3.3 Interpolation method for approximating the temperature admitted by the FV-MOF method at a normal probe . Referring to Figure 4.12, we approximate the normal probe temperature admitted by our FV-MOF method by implementing a least squares approximation method.

Given the 3×3 interpolation stencil about a target cell containing the normal probe, the cost function is,

$$C = \sum_{i,j=-1}^1 \omega_{i,j} (\theta_{i,j} - \theta(\mathbf{x}_{i,j}))^2, \quad (4.14)$$

Table 4.7: Convergence study, comparison with non-axisymmetric body fitted algorithm, inhomogenous Dirichlet Boundary, $T_{stop} = 0.004$, space and time refinement ratios are 2 and 4 respectively; “Linear Exact” method is used, $F^{\text{meas}} = 1$. The resolution of the body fitted algorithm is 128×256 .

N	M	L_1^{err}	L_2^{err}	L_∞	L_1^{grad}	L_2^{grad}	L_∞^{grad}	$L_\infty^{\text{int. flux}}$	iter.
32	1	6.9E-2	8.0E-2	1.5E-1	4.7E-1	6.2E-1	2.1	2.7	9
64	4	2.1E-2	2.4E-2	4.9E-2	1.8E-1	2.4E-1	7.5E-1	8.3E-1	8
128	16	5.3E-3	6.4E-3	1.4E-2	5.2E-2	6.8E-2	2.2E-1	2.5E-1	9
256	64	1.3E-3	1.6E-3	3.5E-3	1.4E-2	1.8E-2	6.1E-2	7.4E-2	10

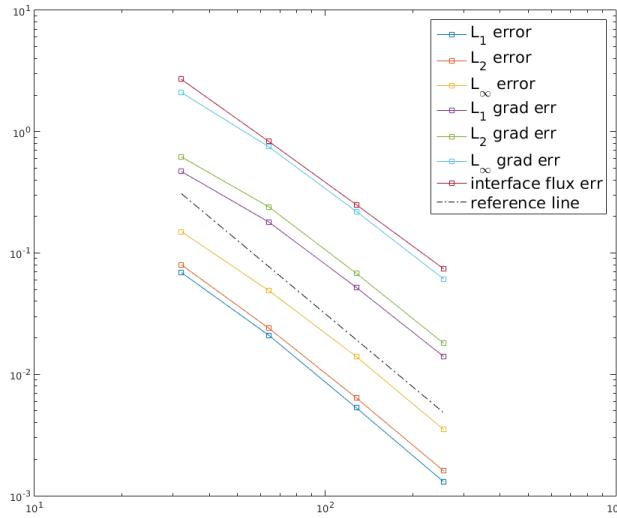


Figure 4.10: Loglog plot referring to Table 4.7

Table 4.8: Convergence study, comparison with non-axisymmetric body fitted algorithm, inhomogeneous Dirichlet Boundary, $T_{stop} = 0.004$, space and time refinement ratios are 2 and 4 respectively; “Linear Exact” method is used, $F^{\text{meas}} = 1$. The resolution of the body fitted algorithm is 256×512 .

N	M	L_1^{err}	L_2^{err}	L_∞	L_1^{grad}	L_2^{grad}	L_∞^{grad}	$L_\infty^{\text{int. flux}}$	iter.
32	1	6.9E-2	8.0E-2	1.5E-1	4.7E-1	6.2E-1	2.1	2.7	9
64	4	2.1E-2	2.4E-2	5.0E-2	1.8E-1	2.4E-1	7.5E-1	8.4E-1	8
128	16	5.3E-3	6.4E-3	1.3E-2	5.2E-2	6.8E-2	2.2E-1	2.6E-1	9
256	64	1.3E-3	1.6E-3	3.4E-3	1.4E-2	1.8E-2	6.0E-2	8.4E-2	10

where

$$\theta(\mathbf{x}) = \theta_{\text{sat}} + \nabla \theta \cdot (\mathbf{x} - \mathbf{x}_I), \quad (4.15)$$

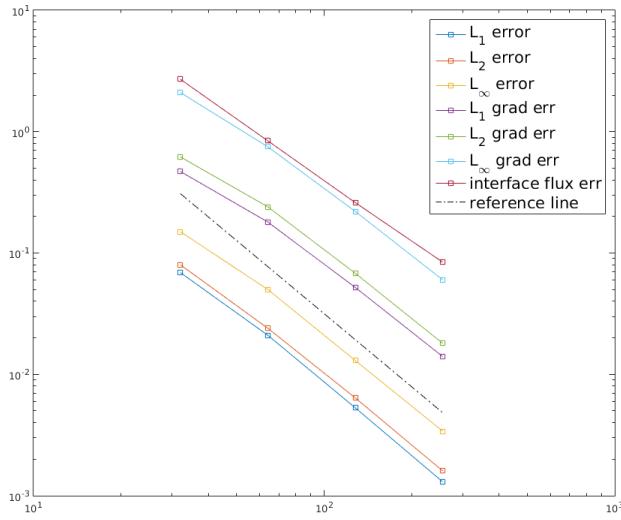


Figure 4.11: Log log plot referring to Table 4.8

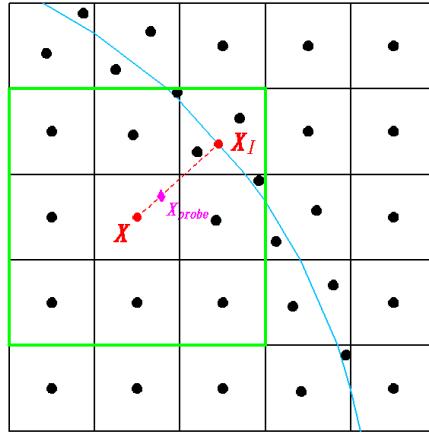


Figure 4.12: Example for illustrating Interpolation Method for calculating temperature at normal probe. x_I is the closest point on the interface to the point x . The green box (containing 3×3 cells) is the region under consideration for normal probe x_{probe} (purple diamond).

and the weights are,

$$\omega_{i,j} = \begin{cases} \frac{\epsilon}{\|x - x_{probe}\|^2} (F_{i,j})^2 & \text{if } F_{i,j} < 10^{-8} \\ 0 & \text{otherwise.} \end{cases} \quad (4.16)$$

4.1.4 Penta-foil Test Problem Modified for Testing the Performance of Our FV-MOF Method in the Presence of Filaments

We modified the penta-foil benchmark test (see [57, 55]) in order to test our FV-MOF method for problems with filamentary regions. The level-set function is defined as

$$\phi_1(x, y) = -\sqrt{(x - 0.02\sqrt{5})^2 + (y - 0.02\sqrt{5})^2} + r \quad (4.17)$$

$$\phi_2(x, y) = -\sqrt{(x - 0.02\sqrt{5})^2 + (y - 0.02\sqrt{5})^2} + (r + \xi) \quad (4.18)$$

where r is defined in polar coordinates as $r = 0.5 + 0.2 \sin(5\psi)$, $\psi \in [0, 2\pi]$; ξ is the parameter for us to adjust the thickness of the material in the middle. The MOF reconstruction is shown in Figure 4.13.

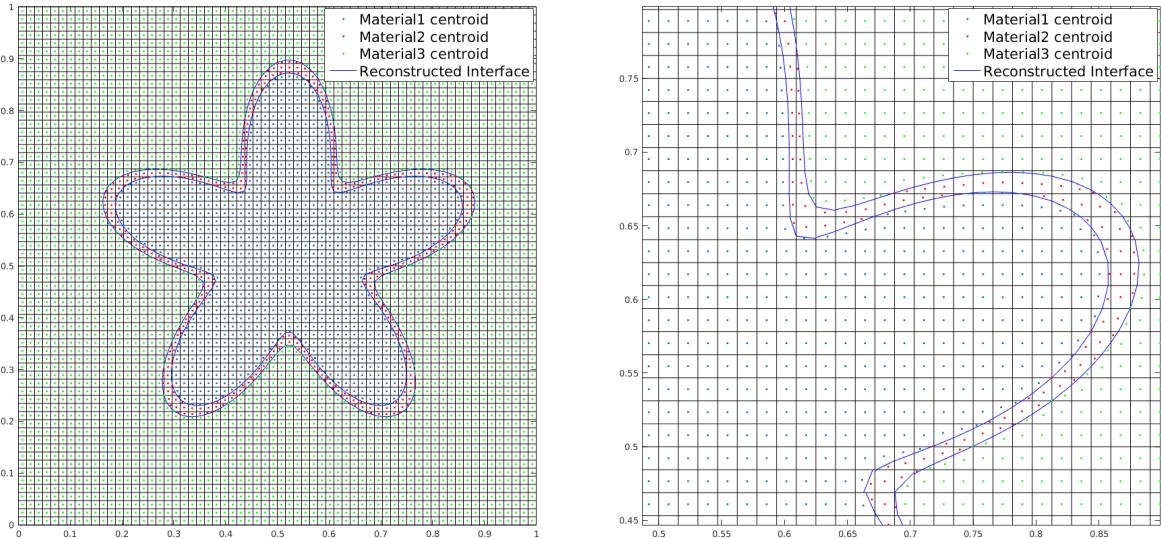


Figure 4.13: Left: MOF reconstruction and centroids. Mesh size: 64×64
Right: A zoom in look at part of the left figure

The problem statement is as follows:

1. Material ID:

1: $\phi_1 > 0$, 3: $\phi_2 < 0$, 2: $\phi_1 < 0$ and $\phi_2 > 0$. Note that the width of the material 2 is not uniform.

2. Initial conditions:

$$T(x, y) = ((x - refc)^2 + (y - refc)^2)e^{-t} \quad (4.19)$$

where, $refc = (1 + 0.02\sqrt{5})/2$

3. Boundary conditions on $\partial\Omega_D$: $\{T(x, y) = 0 : (x, y) \in \partial\Omega_D\}$
4. Boundary conditions at embedded interface: we consider both homogeneous Neumann and non-homogeneous Dirichlet boundary conditions.

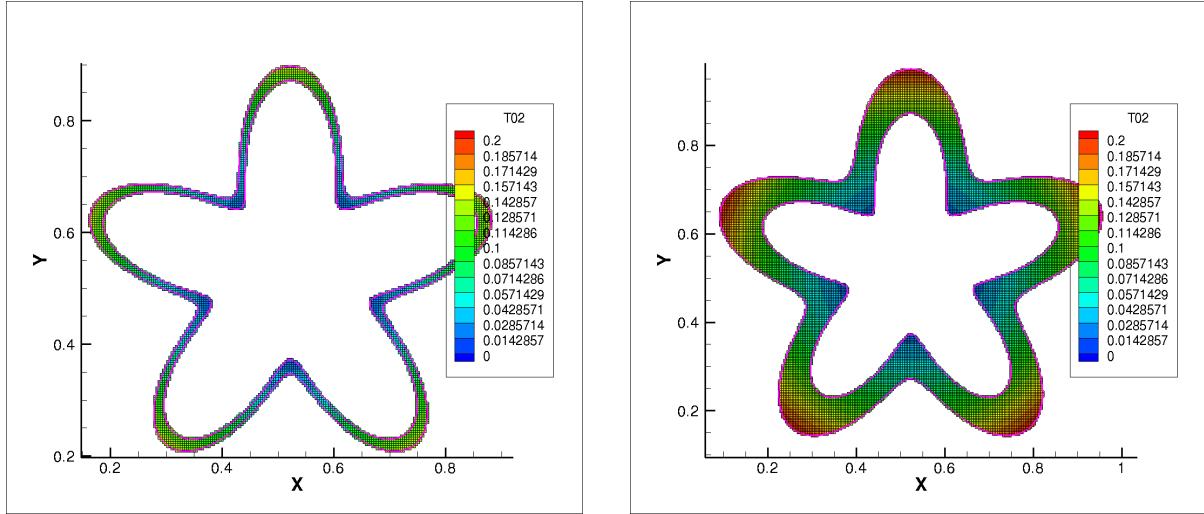


Figure 4.14: Diffusion in Penta-foil shape at $T = 0.0125$. The initial conditions for the middle material (material 2) is given by (4.19). The embedded boundary conditions are Dirichlet for the result displayed here and the grid resolution is 256×256 . The contour plot is “blanked out” in the regions where the material two volume fractions are zero.

Left: $\xi = 0.05$. Right: $\xi = 0.2$

Table 4.9: Convergence study, material 2 temperature for the thick Penta-foil ($\xi = 0.2$), homogeneous Neumann Boundary, $T_{stop} = 0.0125$, space and time refinement ratios are both 2 respectively; “linear exact” method is used, $F^{\text{meas}} = 1.0$.

N	M	L_1^{err}	L_2^{err}	L_∞	L_1^{grad}	L_2^{grad}	L_∞^{grad}	iter.
32-64	1-2	2.7E-3	3.0E-3	5.2E-3	3.7E-2	3.7E-2	3.9E-2	28
64-128	2-4	1.7E-3	1.9E-3	3.0E-3	1.8E-2	1.8E-2	2.6E-2	39
128-256	4-8	9.5E-4	1.1E-3	1.7E-3	9.3E-3	1.0E-2	1.6E-2	50
256-512	8-16	5.1E-4	5.6E-4	8.8E-4	4.9E-3	5.4E-3	9.2E-3	69

Table 4.10: Convergence study, material 2 for the thick Penta-foil ($\xi = 0.2$), inhomogeneous Dirichlet Boundary, $T_{stop} = 0.0125$, space and time refinement ratios are both 2 and 4 respectively; “linear exact” method is used, $F^{\text{meas}} = 1.0$.

N	M	$L_1\text{err}$	$L_2\text{err}$	L_∞	L_1^{grad}	L_2^{grad}	L_∞^{grad}	$L_1^{\text{int. flux}}$	$L_\infty^{\text{int. flux}}$	iter.
32	1	8.7E-5	9.7E-5	3.5E-4	6.5E-4	1.1E-3	5.6E-3	8.0E-2	4.8E-1	10
64	4	2.3E-5	2.4E-5	4.4E-5	1.5E-4	2.1E-4	8.4E-4	6.7E-2	5.7E-1	9
128	16	5.9E-6	6.0E-6	2.2E-5	3.0E-5	6.3E-5	1.1E-3	4.0E-2	5.4E-1	13
256	64	1.4E-6	1.5E-6	4.5E-6	1.0E-5	1.9E-5	4.3E-4	4.0E-2	6.2E-1	14

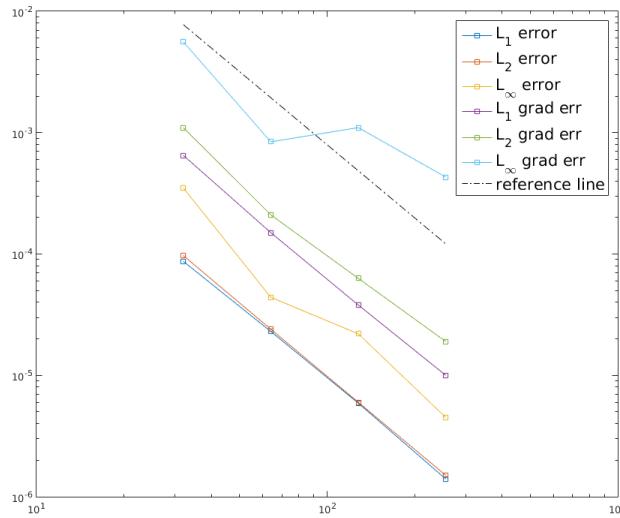


Figure 4.15: Loglog plot referring to Table 4.10

Table 4.11: Convergence study, material 2 of the thin Penta-foil ($\xi = 0.05$), inhomogeneous Dirichlet Boundary, $T_{stop} = 0.0125$, space and time refinement ratios are both 2; “linear exact” method is used, $F^{\text{meas}} = 0.01$.

N	M	$L_1\text{err}$	$L_2\text{err}$	L_∞	L_1^{grad}	L_2^{grad}	L_∞^{grad}	iter.
32	1	3.6E-4	5.9E-4	2.4E-3	2.4E-2	3.6E-2	1.5E-1	8
64	2	1.6E-4	2.2E-4	6.0E-4	1.5E-2	2.3E-2	8.7E-2	8
128	4	1.5E-5	2.2E-5	1.1E-4	2.3E-3	4.5E-3	3.1E-2	11
256	8	1.6E-6	1.7E-6	9.8E-6	1.7E-4	3.0E-4	3.5E-3	12

Temperature contour plots for our customized penta-foil test problem are shown in Figure 4.14. From Tables 4.9, 4.10 and 4.11, it is observed that our method is first order accurate for this multi-material Penta-

Foil benchmark problem. We hypothesize that we measured first order accuracy rather than second order accuracy because the penta-foil geometry has corners; i.e. the interface normal is not continuous.

4.1.5 Hypocycloid-A Test with Six Materials

In order to validate our method for problems with many materials, we designed the six material hypocycloid benchmark test in which we use, as a starting point, the hypocycloid shape with $k = 4$. The hypocycloid interface is defined by the curve:

$$\begin{aligned}\partial\Omega_{hypo} = \{(x(\psi), y(\psi)) | &x(\psi) = 0.2(k-1) \cos \psi + 0.2 \cos((k-1)\psi), \\ &y(\psi) = 0.2(k-1) \sin \psi - 0.2 \sin((k-1)\psi), \psi \in [0, 2\pi]\}\end{aligned}\quad (4.20)$$

where $k = 4$ is an asteroid as in Figure 4.16. Since the asteroid in Figure 4.16 does not have an explicit level set representation, we divide the hypocycloid into a large amount of connected markers in order to make the signed distance calculation accurate enough for deriving the volume fraction, F , and centroids, \mathbf{x} , from the signed distance function (see Equations (3.2) and (3.3)). We will report how many markers we use for each test. Figure 4.16 shows the MOF reconstruction and centroids that are derived from a distance function that, itself, was derived from 2000 markers.

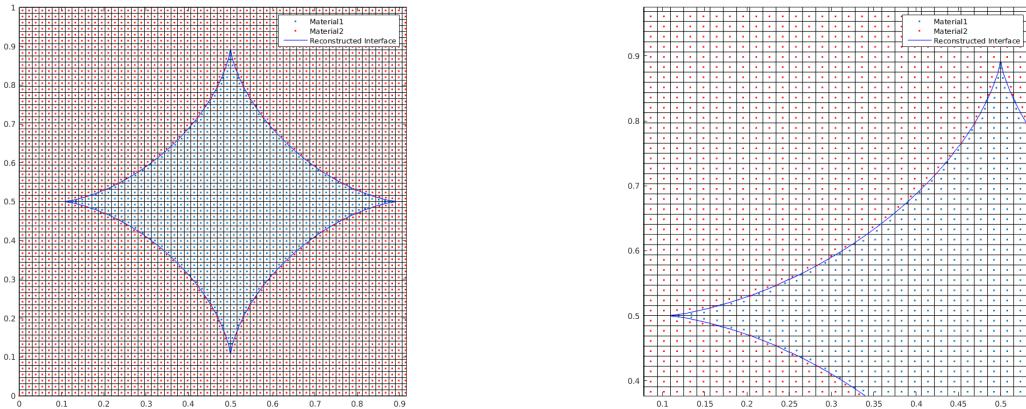


Figure 4.16: Left: MOF reconstruction and centroids. Mesh size: 64×64
Right: A zoom in look at part of the left figure

We set up an initial temperature field and then let it diffuse with time. We test FV-MOF on two cases: (i) two materials as in Figure 4.16, (ii) six materials in which we divide the inner region into four parts equally and add a diamond shape in the middle to be the sixth material, as Figure 4.17 shows.

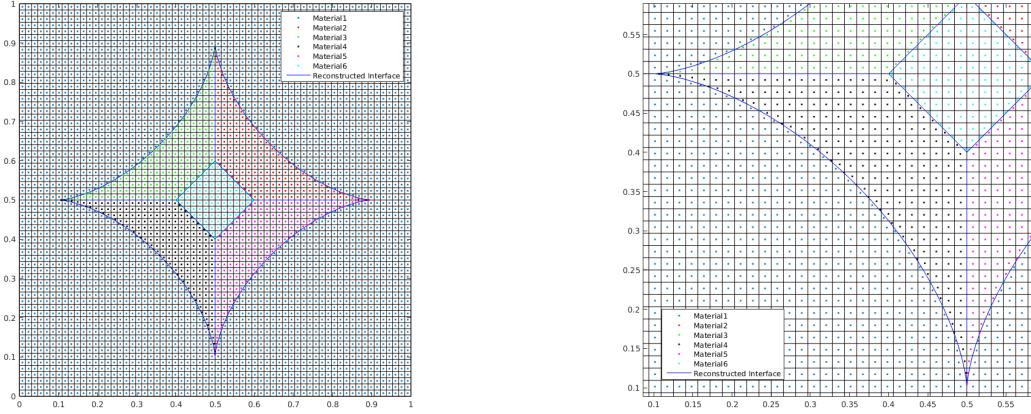


Figure 4.17: Hypocycloid, 6 materials,
2000 markers used to derive the signed distance function which, in turn, is used to derive the volume fractions and centroids.

Left: MOF reconstruction and centroids. Mesh size: 64×64

Right: A zoom in look at part of the left figure

Note: without losing generality, we set the center of the graph to be at the center of the whole domain which means the interfaces separating material 2 and 3, 3 and 4, 4 and 5, 5 and 2 from each other are aligned with the grid line.

1. Material ID: please refer to Figure 4.16 (case(i)) and 4.17(case(ii)).

2. Initial conditions (both cases (i) and (ii)): all materials are initialized as follows:

$$\theta(x, y) = 1 + \frac{(x - 0.5)^2 + (y - 0.5)^2}{D(\alpha)} (10 - 1) \quad (4.21)$$

where $D(\alpha)$ is the distance from the center to the boundary of Ω_D , $\alpha \in [0, 2\pi]$

3. Hypocycloid boundary condition at the computational domain boundaries (both cases (i) and (ii)):

$$\{\theta(x, y) = 10 \mid (x, y) \in \partial\Omega_D\}.$$

4. Multi-material interface boundary conditions (both cases (i) and (ii)): the jump condition, $[\kappa \nabla \theta \cdot \mathbf{n}] = 0$, is prescribed at material interfaces.

4.1.5.1 Hypocycloid numerical results. We perform one convergence study for the 2 material hypocycloid problem (type(i)) and two convergence studies for the 6 material hypocycloid problem (type(ii) and type (iii)). Only cells fully occupied by a single material are considered when measuring the relative error of temperature between two successively refined grids. From the relative errors, we then report the rate of convergence of our method for the hypocycloid tests. The details for our 3 hypocycloid tests are:

Type(i): two materials’ test case; we investigate the sensitivity of our convergence study for varying number of connected markers used to generate the signed distance function (which in turn, is used to generate the volume fractions and centroids, see (3.2) and (3.3)). The results will not be sensitive to the number of markers when this number is big enough. In the test below, we set the number of markers to be 2000. Results are presented in Figures 4.12.

Table 4.12: Convergence study, Hypocycloid, two materials, convergence study for material 1 (inner material), 2000 connected markers, $k_1 = 0.1$, $k_2 = 10.0$, space and time refinement ratios are both 2 respectively; “simple” method is used, $F^{\text{meas}} = 1.0$.

N	M	$L_1\text{err}$	$L_2\text{err}$	L_∞	L_1^{grad}	L_2^{grad}	L_∞^{grad}	iter.
32-64	1-2	1.4E-1	2.1E-1	5.5E-1	2.7	3.7	8.8	32
64-128	2-4	1.1E-1	1.6E-1	4.1E-1	3.0	3.9	7.7	62
128-256	4-8	7.2E-2	9.7E-2	2.1E-1	1.8	2.2	4.3	106
256-512	8-16	4.0E-2	5.3E-2	1.1E-1	1.0	1.2	2.5	177

Table 4.13: Convergence study, Hypocycloid, two materials, convergence study for material 1 (inner material), 4000 connected markers, $k_1 = 0.1$, $k_2 = 10.0$, space and time refinement ratios are both 2 respectively; “simple” method is used; $F^{\text{meas}} = 1.0$.

N	M	$L_1\text{err}$	$L_2\text{err}$	L_∞	L_1^{grad}	L_2^{grad}	L_∞^{grad}	iter.
32-64	1-2	1.4E-1	2.1E-1	5.5E-1	2.7	3.7	8.8	32
64-128	2-4	1.1E-1	1.6E-1	4.1E-1	3.0	3.9	7.8	62
128-256	4-8	7.2E-2	9.7E-2	2.1E-1	1.8	2.2	4.3	106
256-512	8-16	4.0E-2	5.3E-2	1.1E-1	1.0	1.2	2.6	177

We are able to see that when the number of connected markers are large enough, the measured rate of convergence is insensitive to the number of markers. The results are almost identical for these two cases. It is also observed, as we refine the grid, the convergence rate using either L_2 or L_∞ norms are approaching 1.0. Figure 4.18 shows the diffusion process at different time steps.

Type(ii): Six materials case with the same set-up as with the two materials case (i.e. all five materials inside the hypocycloid have the same diffusion coefficients as with the inner material in the two materials case). $k_1 = 10.0$, $k_2 = k_3 = k_4 = k_5 = k_6 = 0.1$. The results are shown in Table 4.14.

It is observed that the convergence rate of the six material case is very close to that of the previous two material case with the same set-up, as expected.

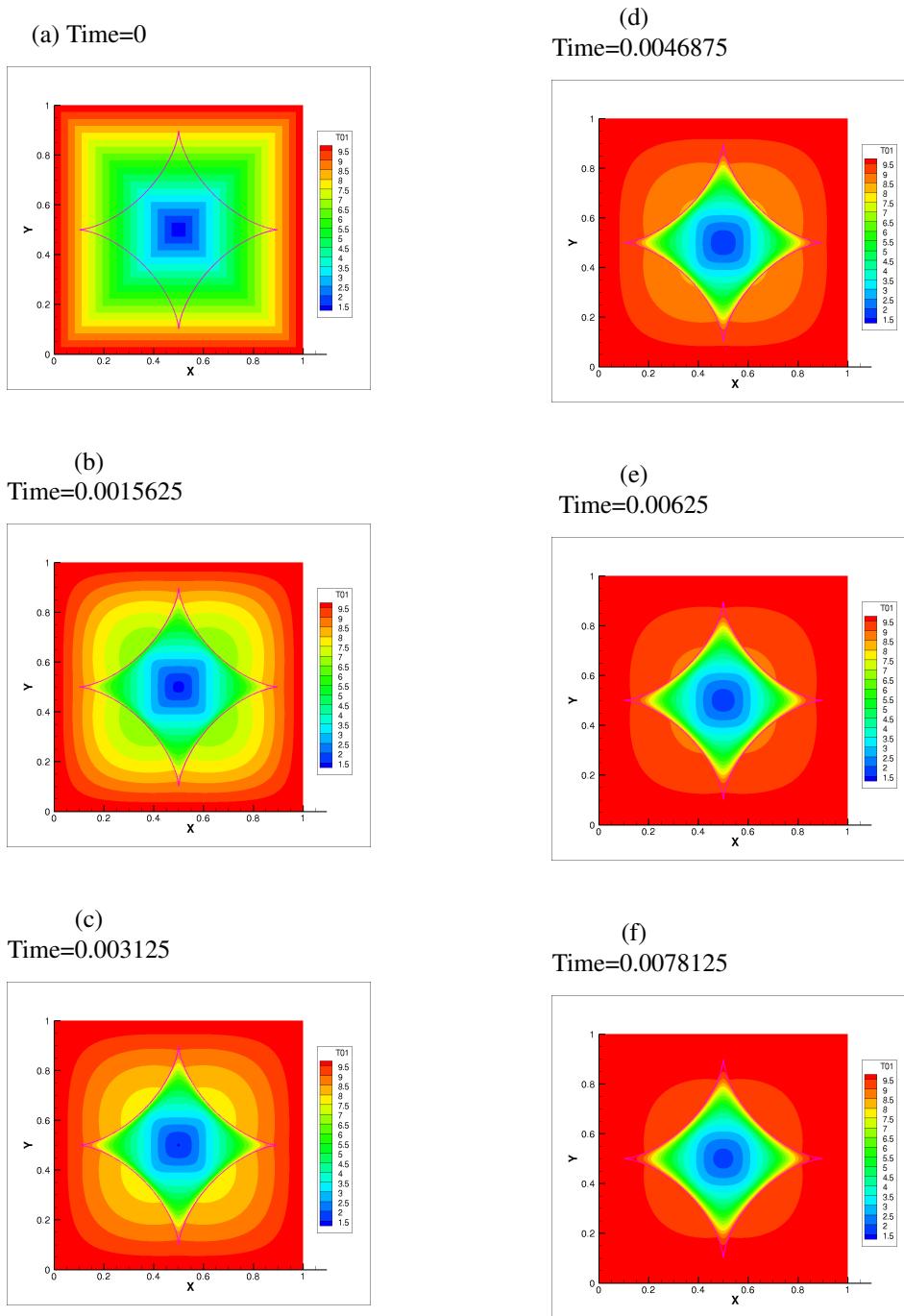


Figure 4.18: Evolution of Hypocycloid 2 materials diffusion problem, mesh size 512×512 , $k_1 = 0.1$, $k_2 = 10.0$.

Table 4.14: Convergence study, Hypocycloid, six materials, convergence study for material 2 (the upright material of the inner region), 2000 connected markers,

$k_1 = 10.0, k_2 = k_3 = k_4 = k_5 = k_6 = 0.1$, space and time refinement ratios are both 2 respectively; “simple” method is used, $F^{\text{meas}} = 1.0$.

N	M	L_1^{err}	L_2^{err}	L_{∞}	L_1^{grad}	L_2^{grad}	L_{∞}^{grad}	iter.
32-64	1-2	1.9E-1	2.5E-1	5.5E-1	5.7	6.0	8.8	34
64-128	2-4	1.3E-1	1.8E-1	4.1E-1	4.2	4.9	7.8	64
128-256	4-8	8.1E-2	1.0E-1	2.1E-1	2.2	2.5	4.3	105
256-512	8-16	4.5E-2	5.6E-2	1.1E-1	1.1	1.3	2.5	180

Type (iii): Six materials with $k_1 = 10.0, k_2 = k_4 = 0.5, k_3 = k_5 = 1.0, k_6 = 0.1$; the convergence rate for this test is reported in Table 14.

Table 4.15: Convergence study, Hypocycloid, six materials, convergence study for material 2 (the upright material of the inner region), 2000 connected markers,

$k_1 = 10.0, k_2 = k_4 = 0.5, k_3 = k_4 = 1.0, k_6 = 0.1$ space and time refinement ratios are both 2 respectively; “simple” method is used, $F^{\text{meas}} = 1.0$.

N	M	L_1^{err}	L_2^{err}	L_{∞}	L_1^{grad}	L_2^{grad}	L_{∞}^{grad}	iter.
32-64	1-2	4.3E-1	4.3E-1	5.5E-1	3.1	3.2	4.2	49
64-128	2-4	2.7E-1	2.7E-1	3.1E-1	1.2	1.4	2.3	81
128-256	4-8	1.5E-1	1.5E-1	1.8E-1	7.9E-1	8.9E-1	1.8	133
256-512	8-16	7.9E-2	8.1E-1	1.0E-1	5.2E-1	5.9E-1	1.4	186

It is observed from Table 4.15 that the convergence rate is approaching 1.0. Figure 4.19 shows this diffusion process at different time steps. As expected, the results that we reported in Figure 4.19 are distinct from those reported in Figure 4.18 in a manner that is consistent with the differences between the material configurations and heat conductivities of the two respective test cases. (And we note that the results from Figure 4.18 are almost identical to the results obtained from Type(ii).)

4.1.6 Example Where a Filamentary “Micro-scale Region” Effects the Macro-scale Solution

In many problems of practical interest, for instance, those problems described in [84] and [109], the heat transfer in micro-scale regions can effect the overall (macro-scale) heat-transfer properties. We have designed a benchmark problem which investigates the performance of how our new method captures the micro-scale region effect on macro-scale heat transfer.

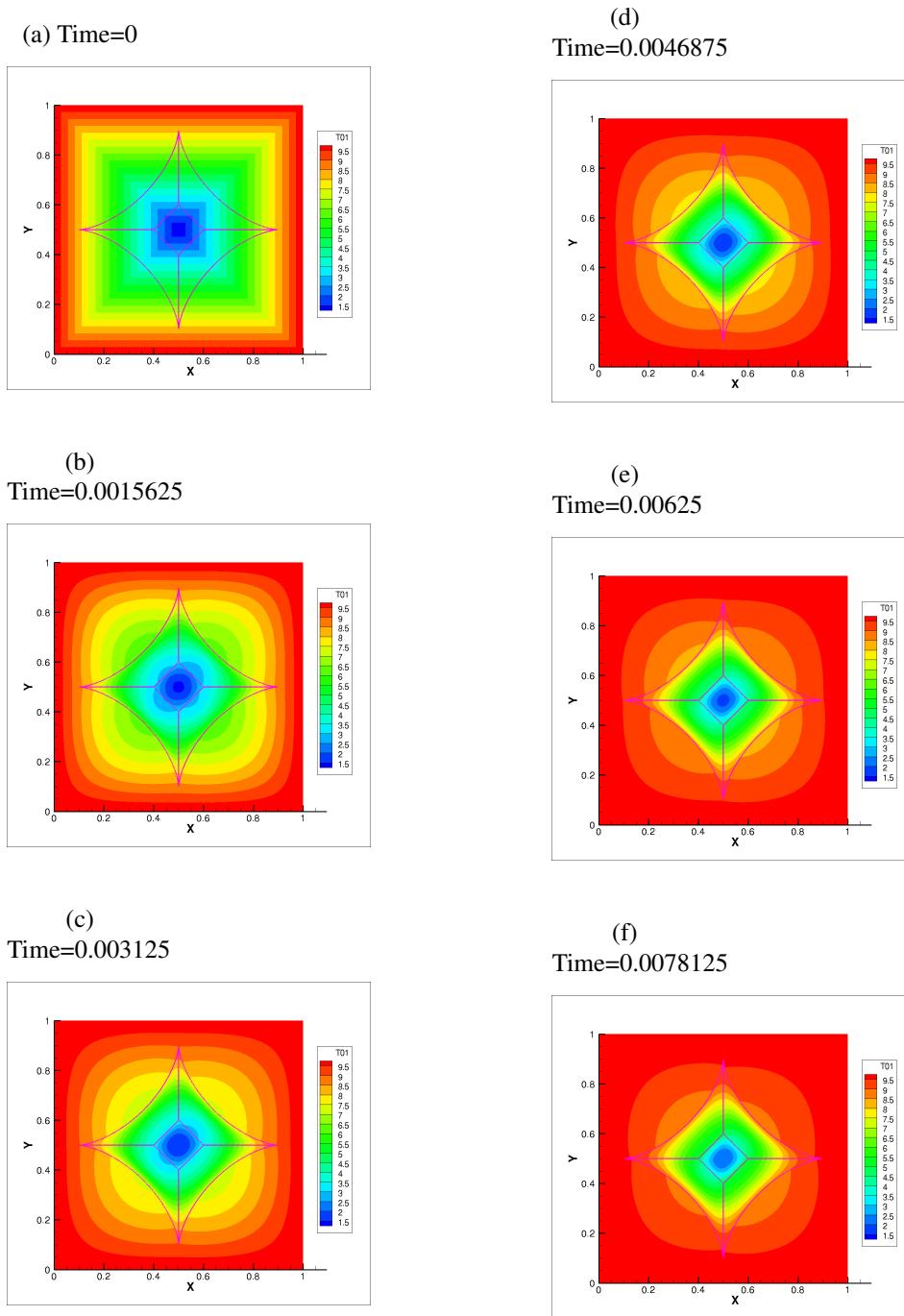


Figure 4.19: Evolution of Hypocycloid, 6 materials type (iii) diffusion problem, mesh size 512×512 ,
 $k_1 = 10.0, k_2 = k_4 = 0.5, k_3 = k_5 = 1.0, k_6 = 0.1$

Referring to Figure 4.20, there are three materials defined in terms of four distance functions:

$$\phi_1(x, y) = y - 0.55 \quad (4.22)$$

$$\phi_2(x, y) = \sqrt{(x - 0.5)^2 + (y - 0.55)^2} - 0.29 \quad (4.23)$$

$$\phi_3(x, y) = \sqrt{(x - 0.5)^2 + y^2} - 0.5 \quad (4.24)$$

$$\phi_4(x, y) = \sqrt{(x - 0.5)^2 + (y - 0.55)^2} - (0.29 - \delta) \quad (4.25)$$

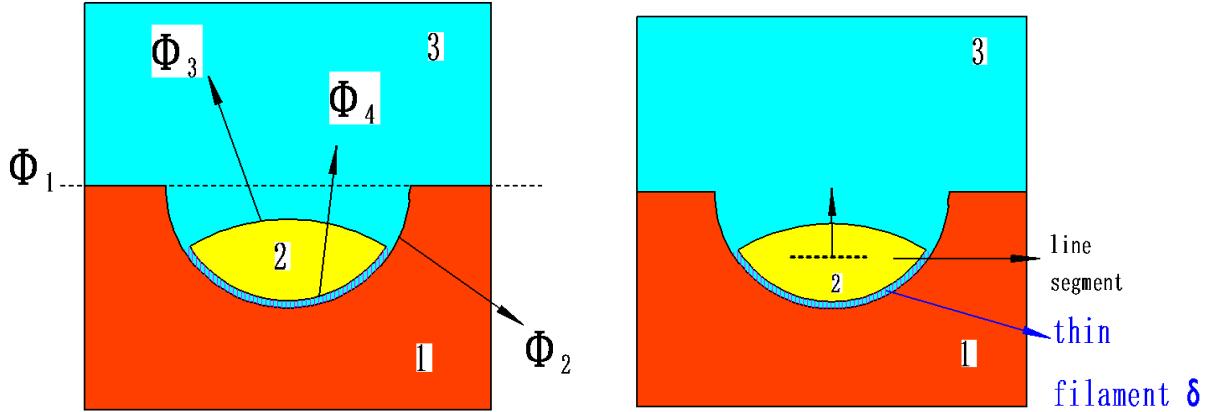


Figure 4.20: Left: Problem geometry set-up with three materials defined in terms of four distance functions $\phi_i, i = 1, \dots, 4$.
A filamentary region (filled with material 3) exists with thickness $\delta = 0.02$.

Right: test for the average flux integral across line segment $0.375 < x < 0.625$ $y = 0.40625$ over the whole running time

We prescribe the micro-scale filamentary thickness δ in Equation (4.25) to be $\delta = 0.02$. Figure 4.21 shows the piecewise linear MOF reconstruction of the case in which material 2 has two sharp corners. Figure 4.22 shows the piecewise linear MOF reconstruction for the case in which material 2 is a smooth circle.

The boundary conditions of the computational domain Ω_D are defined as

$$\begin{cases} \frac{\partial \theta}{\partial n} = 0, & \text{if } x = 0 \text{ or } x = 1, \\ \theta(x, y) = 0, & \text{if } y = 1, \\ \theta(x, y) = 10, & \text{if } y = 0. \end{cases} \quad (4.26)$$

We set the diffusion coefficient to be $k_1 = k_2 = 1.0$ and $k_3 = 0.001$.

The initial temperature field in all material regions is defined as

$$\theta(x, y) = H(y - 0.2)(5\cos(\pi y/0.2) + 5) \quad \text{where, } H(0.2 - y) = \begin{cases} 1 & \text{if } y \leq 0.2 \\ 0 & \text{otherwise} \end{cases} \quad (4.27)$$

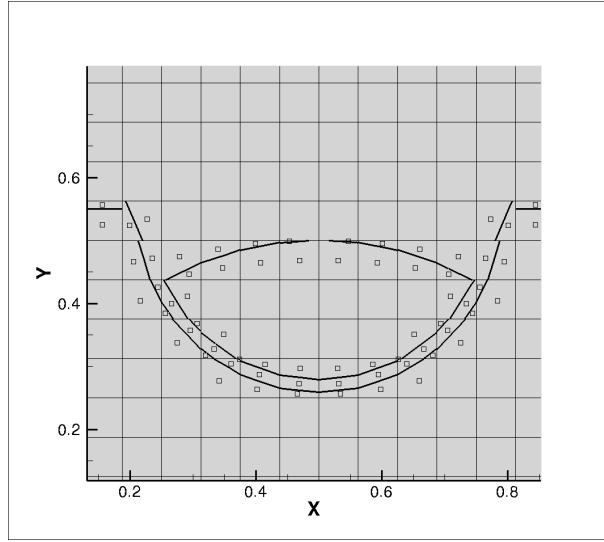


Figure 4.21: MOF reconstruction and centroids for sharp cornered material 2. Mesh size 16×16

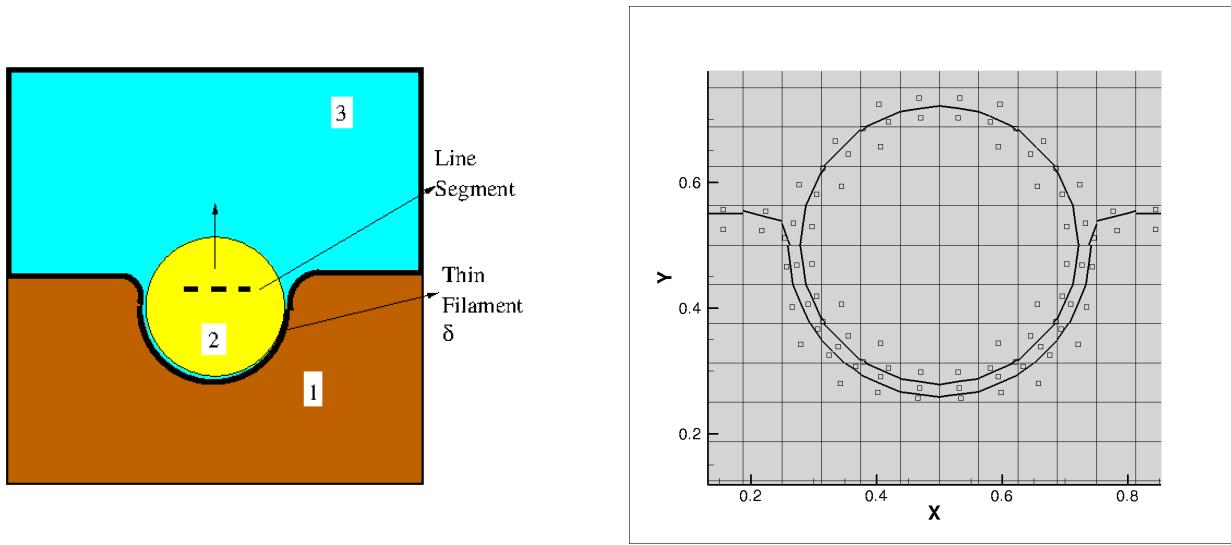


Figure 4.22: Left: Problem geometry set-up with three materials. Material 2 is a (smooth) circle. A filamentary region (filled with material 3) exists with thickness $\delta = 0.02$. The average flux integral test looks at the line $0.375 < x < 0.625$ $y = 0.59375$ over the whole running time.
Right: MOF reconstruction and centroids; Mesh size 16×16

In order to measure the macro-scale effect from the micro-scale region, we measure an average heat flux across a line segment placed in material 2. The line segment is defined as,

$$\Gamma_{flux} = \{(x, y) \mid 0.375 \leq x \leq 0.625, y = 0.40625\} \quad (4.28)$$

for the problem illustrated in Figures 4.20 and 4.21, and

$$\Gamma_{flux} = \{(x, y) \mid 0.375 \leq x \leq 0.625, y = 0.59375\} \quad (4.29)$$

for the problem illustrated in Figure 4.22.

The average heat flux is calculated as follows,

$$Avg_flux = \frac{\int_0^{T_N} \int_{\Gamma_{flux}} k \nabla \theta \cdot \mathbf{n} \, dx dT}{T_N \cdot L_{tot}} \doteq \frac{\sum_{n=1}^N \int_{\Gamma_{flux}} k \nabla \theta(T^n, \mathbf{x}) \cdot \mathbf{n} dx}{N \cdot L_{tot}} \approx \frac{\sum_{n=1}^N \sum_{i=ilo}^{ihi} k \left(\frac{\theta_{i,j+1}^n - \theta_{i,j}^n}{\Delta y} \right)}{N(ihi - ilo + 1)} \quad (4.30)$$

$$x_{ilo} = 0.375, \quad x_{ihi} = 0.625, \quad y_j = 0.40625 - \frac{\Delta y}{2} \text{ or } y_j = 0.59375 - \frac{\Delta y}{2}$$

where, $T_N = 2.0$ is the total computing time, $L_{tot} = x_{ihi} - x_{ilo}$ is the total length of the line segment, $ihi - ilo + 1$ is the total number of cells covered by the line segment, N is the total number of time steps, $\Delta y = \Delta x$ is the grid size.

Table 4.16: Convergence study corresponding to the geometry illustrated in Figure 4.22; a thin filament with $\delta = 0.02$ separates a (smooth) circular shaped drop from the substrate. There are three materials and the convergence study is for material 2 (the drop). $k_{ratio} = 1000 : 1$, space and time refinement ratios are both 2, the “connected centroid” method (3.13) is used, $F^{\text{meas}} = 1.0$, and $T_{stop} = 2.0$
average flux: (16) – 0.131 (32) – 0.155 (64) – 0.177 (128) – 0.192 (256) – 0.197

N	M	dx	L_1 err	L_2 err	L_∞	L_1^{grad}	L_2^{grad}	L_∞^{grad}	iter.
16-32	4-8	0.03125	1.2E-1	1.2E-1	1.2E-1	1.3E-2	1.5E-2	2.5E-2	25
32-64	8-16	0.015625	3.1E-1	3.1E-1	3.1E-1	2.2E-2	1.0E-1	5.6E-1	45
64-128	16-32	0.0078125	1.9E-1	1.9E-1	1.9E-1	1.3E-2	1.3E-2	1.8E-2	84
128-256	32-64	3.90625E-3	8.0E-2	8.0E-2	8.1E-2	5.2E-3	5.3E-3	9.3E-3	138

The data from Tables 4.16, 4.17, 4.18, 4.19, 4.20, and 4.21, can be condensed into the following observations:

- For multi-material problems with jump condition interfacial boundary conditions and complex geometry, all three approaches, (i) the connected centroid method (3.13), (ii) Dai and Scannapieco’s method (3.17), and (iii) the orthogonal projection method (3.18) converge with grid refinement (temperature and its gradient) and the errors for all three approaches are comparable.
- For non-constant k across the filamentary region, convergence is not observed until after the filament thickness δ is greater than the mesh size Δx . We expect this result since just one point along the thickness of the filament is not enough to represent a solution profile with different gradients on either side of the filament if there

Table 4.17: Convergence study corresponding to the geometry illustrated in Figures 4.20 and 4.21; a thin filament with $\delta = 0.02$ separates irregular shaped drop from substrate. There are three materials and the convergence study is for material 2 (the drop). $k_{ratio} = 1000 : 1$, space and time refinement ratios are both 2, the “connected centroid” method (3.13) is used, $F^{meas} = 1.0$, and $T_{stop} = 2.0$
 average flux: (16) – 0.121 (32) – 0.156 (64) – 0.177 (128) – 0.187 (256) – 0.191

N	M	dx	L_1 err	L_2 err	L_∞	L_1^{grad}	L_2^{grad}	L_∞^{grad}	iter.
16-32	4-8	0.03125	STENCIL NOT AVAILABLE FOR GRADIENT ERROR						24
32-64	8-16	0.015625	3.7E-1	3.7E-1	3.8E-1	1.3E-2	1.3E-2	1.3E-2	42
64-128	16-32	0.0078125	2.4E-1	2.4E-1	2.4E-1	6.4E-3	6.8E-3	1.3E-2	79
128-256	32-64	3.90625E-3	1.3E-1	1.3E-1	1.3E-1	3.7E-3	4.0E-3	9.2E-3	133

Table 4.18: Convergence study corresponding to the geometry illustrated in Figure 4.22; a thin filament with $\delta = 0.02$ separates smooth shaped drop from substrate. There are three materials and the convergence study is for material 2 (the drop). $k_{ratio} = 1000 : 1$, space and time refinement ratios are both 2, the “Dai and Scannapieco” [16] method (3.17) is used, $F^{meas} = 1.0$, and $T_{stop} = 2.0$
 average flux: (16) – 0.131 (32) – 0.148 (64) – 0.172 (128) – 0.186 (256) – 0.195

N	M	dx	L_1 err	L_2 err	L_∞	L_1^{grad}	L_2^{grad}	L_∞^{grad}	iter.
16-32	4-8	0.03125	2.1E-1	2.1E-1	2.2E-1	2.1E-2	2.3E-2	3.4E-2	26
32-64	8-16	0.015625	3.3E-1	3.3E-1	3.3E-1	2.5E-2	2.5E-2	3.0E-2	46
64-128	16-32	0.0078125	2.0E-1	2.0E-1	2.0E-1	1.4E-2	1.4E-2	2.0E-2	84
128-256	32-64	3.90625E-3	1.2E-1	1.2E-1	1.2E-1	7.7E-3	7.8E-3	2.1E-2	138

Table 4.19: Convergence study corresponding to the geometry illustrated in Figure 4.22; a thin filament with $\delta = 0.02$ separates smooth shaped drop from substrate. There are three materials and the convergence study is for material 2 (the drop). $k_{ratio} = 1000 : 1$, space and time refinement ratios are both 2, the “orthogonal projection” method (3.18) is used, $F^{meas} = 1.0$, and $T_{stop} = 2.0$
 average flux: (16) – 0.137 (32) – 0.173 (64) – 0.185 (128) – 0.194 (256) – 0.199

N	M	dx	L_1 err	L_2 err	L_∞	L_1^{grad}	L_2^{grad}	L_∞^{grad}	iter.
16-32	4-8	0.03125	4.6E-3	4.7E-3	5.5E-3	4.3E-3	4.7E-3	7.8E-3	25
32-64	8-16	0.015625	1.3E-1	1.3E-1	1.3E-1	9.6E-3	9.6E-3	1.3E-2	46
64-128	16-32	0.0078125	1.2E-1	1.2E-1	1.2E-1	7.9E-3	8.1E-3	1.3E-2	85
128-256	32-64	3.90625E-3	6.2E-2	6.2E-2	6.2E-2	3.9E-3	4.0E-3	7.1E-3	138

- For constant k across the filamentary region, convergence is observed, albeit less than first order accurate, even for grid sizes larger than the filament thickness. We expect this result too since one point along the thickness of the filament is enough to capture the solution whose gradients on either side of the filament approach each other.

Table 4.20: Convergence study corresponding to the geometry illustrated in Figure 4.22; a thin filament with $\delta = 0.02$ separates (smooth) circular shaped drop from substrate. There are three materials and the convergence study is for material 2 (the drop). $k_{ratio} = 1 : 1$, space and time refinement ratios are both 2, the “connected centroid” method (3.13) is used, $F^{meas} = 1.0$, and $T_{stop} = 2.0$
 average flux: (16) – 9.06 (32) – 9.31 (64) – 9.46 (128) – 9.54 (256) – 9.58

N	M	dx	L_1 err	L_2 err	L_∞	L_1^{grad}	L_2^{grad}	L_∞^{grad}	iter.	E_{gap}
16	4	0.0625	4.7E-2	5.6E-2	1.0E-1	2.5E-1	3.3E-1	5.5E-1	13	3.9E-3
32	8	0.03125	3.2E-2	4.0E-2	8.6E-2	1.7E-1	2.3E-1	4.2E-1	24	1.4E-3
64	16	0.015625	1.7E-2	2.1E-2	4.7E-2	8.7E-2	1.1E-1	2.7E-1	45	4.5E-4
128	32	0.0078125	7.9E-3	9.8E-3	2.3E-2	4.1E-2	5.4E-2	2.5E-1	77	1.1E-4
256	64	3.90625E-3	4.0E-3	4.9E-3	1.1E-2	2.1E-2	2.0E-2	2.2E-1	132	3.0E-5

Table 4.21: Convergence study corresponding to the geometry illustrated in Figures 4.20 and 4.21; a thin filament with $\delta = 0.02$ separates the irregular shaped drop from substrate. There are three materials and the convergence study is for material 2 (the drop). $k_{ratio} = 1 : 1$, space and time refinement ratios are both 2, the “connected centroid” method (3.13) is used, $F^{meas} = 1.0$, and $T_{stop} = 2.0$
 average flux: (16) – 8.95 (32) – 9.73 (64) – 9.90 (128) – 9.99 (256) – 10.03

N	M	dx	L_1 err	L_2 err	L_∞	L_1^{grad}	L_2^{grad}	L_∞^{grad}	iter.	E_{gap}
16	4	0.0625	6.9E-2	7.5E-2	1.0E-1	4.8E-1	7.7E-1	3.3	12	1.2E-2
32	8	0.03125	5.2E-2	5.5E-2	9.4E-2	3.5E-1	6.1E-1	3.4	23	8.2E-3
64	16	0.015625	2.3E-2	2.5E-2	5.0E-2	1.8E-1	3.8E-1	3.5	45	1.6E-3
128	32	0.0078125	1.2E-2	1.2E-2	2.5E-2	1.0E-1	2.9E-1	3.4	76	9.3E-4
256	64	3.90625E-3	5.8E-3	6.2E-3	1.3E-2	5.6E-2	2.0E-1	3.7	130	9.2E-4

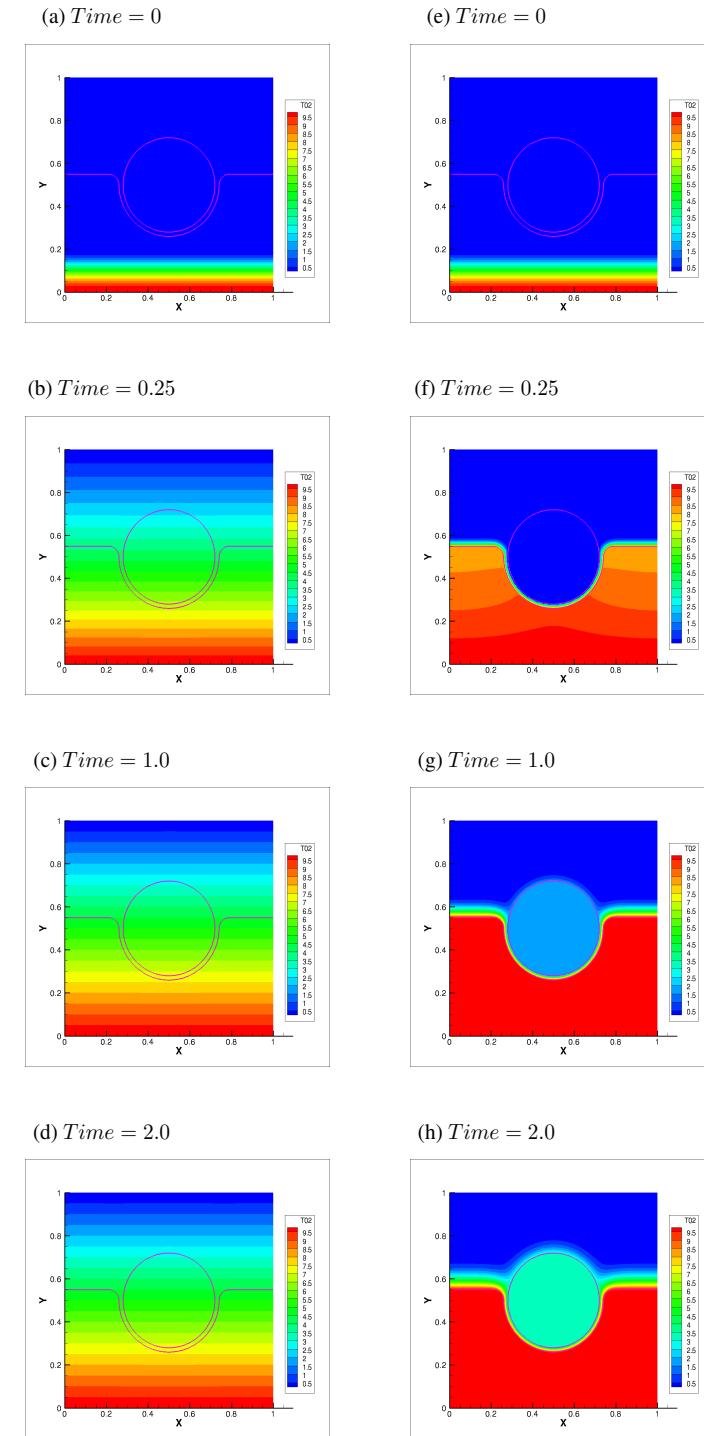


Figure 4.23: Comparison of thermal conductivity ratio 1:1 (left column) versus 1000 : 1 (right column); material two is a (smooth) circle. $\Delta T = 0.03125$, 256×256 mesh

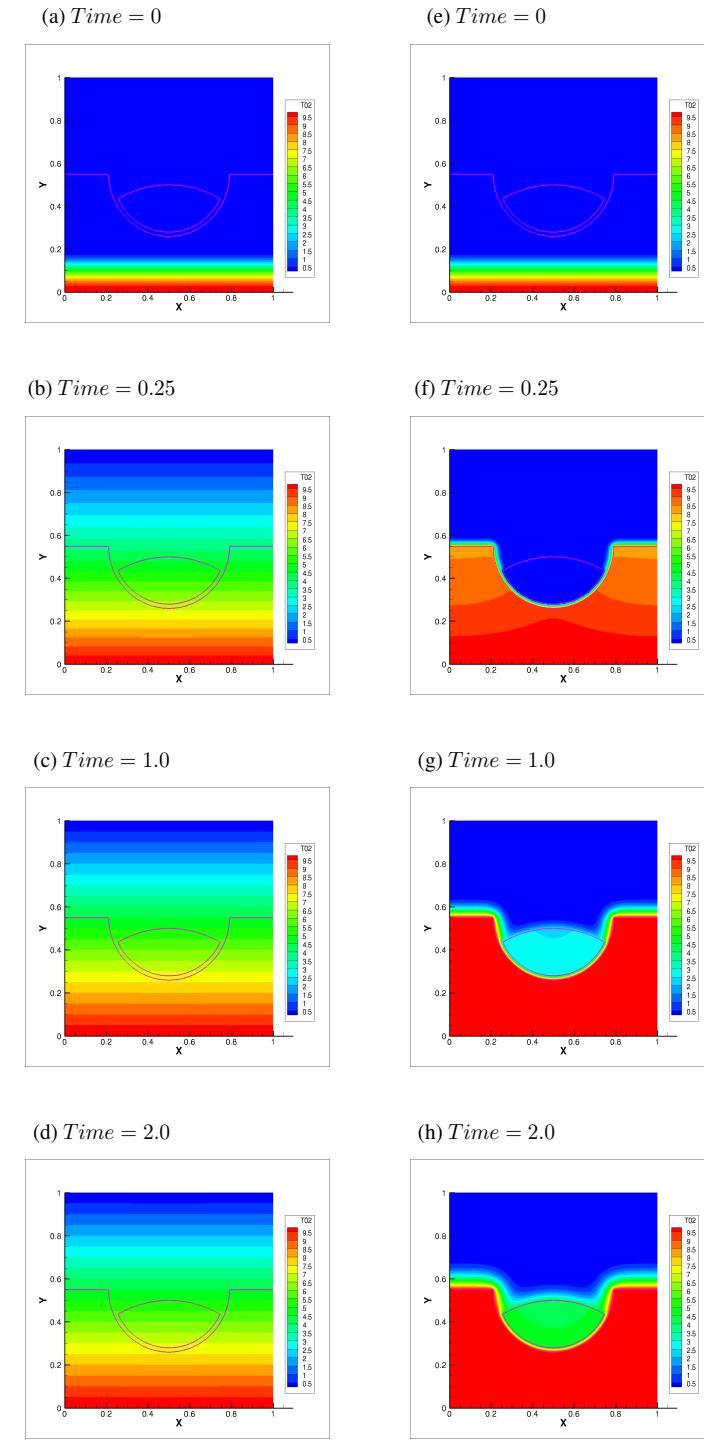


Figure 4.24: Comparison of thermal conductivity ratio 1:1 (left column) versus 1000 : 1 (right column); material two has corners. $\Delta T = 0.03125$, 256×256 mesh

4.2 The Stefan Problem: Supermesh Method for Complex Deforming Boundaries.

We tested our algorithm for solving the Stefan problem, described in Chapter 3, for the following tests: (A) the moving planar interface with exact solution, (B) the shrinking circle comparing with 1D body-fitted algorithm, (C) gingerbread man melting and (D) unstable solidification.

4.2.1 Moving Planar Interface-Sanity Check

In this test, the interface is set to be a steadily advancing planar surface, $\Gamma(t) = \{x = 0.1 + Vt, y \in \mathbb{R}\}$, where the front velocity V is constant and set to be 1.

The exact solution to this Stefan problem is given by

$$\theta(\mathbf{x}, t) = \begin{cases} 272 + e^{-V(x-0.1-Vt)}, & x > 0.1 + Vt, \\ 273, & x \leq 0.1 + Vt \end{cases} \quad (4.31)$$

The boundary conditions at the computational domain boundaries are Dirichlet boundary at the left side of the domain and non-homogeneous Neumann boundary condition on all the other walls. Note that the last few columns of cells with $x \in [1 - 1/16, 1]$ on the right side of the domain are continuously overwritten with exact solution.

We applied our deforming boundary supermesh method to this planar front and measured the error between the exact and computed solutions (see Table 4.22). One can observe a first order convergence of the error under grid refinement.

Table 4.22: Convergence study of moving planar interface with the exact solution $x = 0.2$ at the end time $t = 1.0$

Space resolution	computing result	error
16×16	0.1913	0.0087
32×32	0.1953	0.0047
64×64	0.1974	0.0026
128×128	0.1987	0.0013

4.2.2 Shrinking Cylinder

In this test, the interface is a circle centered at the middle of the computational domain $[0, 1] \times [0, 1]$ which separates the inner material 1 and outer material 2. Initial conditions for the problem are,

1. Initial radius: $r_0 = 0.4$;
2. Initial temperature field

$$\theta(r, 0) = \begin{cases} -4 \cos(0.5\pi r/r_0) + T_{sat} & r < r_0 \\ T_{sat} & r > r_0 \end{cases} \quad (4.32)$$

where $T_{sat} = 273$.

For this test, the “melt” is inside the circle and the ”ice” (a.k.a. “solid”) is outside the circle. Since $\nabla\theta \neq 0$ in material 1 and $\nabla\theta = 0$ in material 2, the circle will be driven to shrink due to the temperature gradient inside the circle. Since there is no analytical solution for this test, we compare the computed radius with the results computed by a 1D body-fitted code.

The computational results are reported in Table 4.23. We could see that the radius errors at the end time of different resolutions are first order convergent. We also notice that there is a super-linear convergence at the finest resolution. We hypothesize that the super-linear convergence observed for the finest resolution case is due to the fact that the resolution for the 1D body-fitted Stefan problem code was chosen to be 1024 points which is near the resolution for the 512×512 case. In Figure 4.25, we provide the visualized results.

Table 4.23: Convergence study of shrinking circle test.

Space resolution	Time steps	error
64×64	5	0.0024
128×128	10	0.0015
256×256	20	0.0008
512×512	40	0.0002

From the zoom-in look of the circles in different resolutions, we can observe the convergence, as well.

4.2.3 Gingerbread Man Melting Test

In this test, we create a gingerbread man geometry, and set the top and bottom wall temperature boundary conditions to be heated (i.e. Dirichlet $\theta > \theta_{sat}$) sources on the boundaries of the computational domain. We expect the temperature diffusion induced phase front velocity to propel the interface and finally melt down the gingerbread man outline to nothing.

The gingerbread man initial geometry is created by the following steps:

1. Sketch the outline using CAD graphics software.

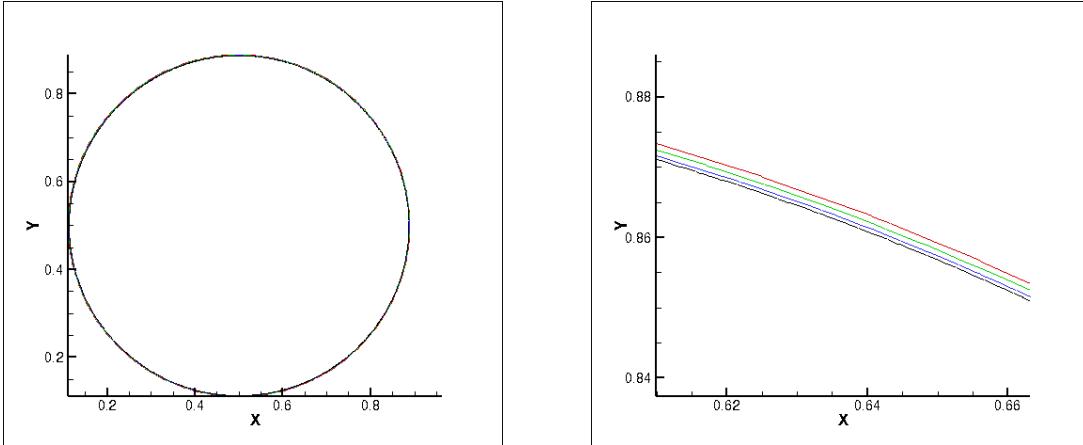


Figure 4.25: Convergence study of shrinking circle at $Time = 6.25E - 4$, expected radius= 0.3868.

Left: interfaces for all the resolutions overlaid on top of each other red 64×64 , green 128×128 , blue 256×256 , black 512×512 ; Right: zoom-in look of the left figure.

2. Output the geometry produced by the CAD software and input the CAD geometry into a mesh generation program in order to generate an unstructured triangular mesh consisting of nodes and elements.
3. Since the mesh generation software creates a tessellation of the boundary and inside of the Gingerbread Man shape, we wrote a program to extract just the line segments (in 2D) from the boundary of the Gingerbread Man.

After the above steps, the outline data of the gingerbread man is stored as piecewise linear sections. We still need another algorithm to generate the signed distance based on the outline file, converting Lagrangian geometry to Eulerian signed distance function.

The algorithm we have developed is as follows:

1. On the Eulerian grid, initialize all cells as “not processed”.
2. Loop through all the boundary elements,

For $k = 1, 2, \dots, N_{elements}$,

- (a) Construct the local level set function $\phi_k = \mathbf{n} \cdot (\mathbf{x} - \mathbf{x}_k)$ (referring to Figure 4.26).

In order to compute the normal for line section $\overrightarrow{\mathbf{x}_1, \mathbf{x}_2}$, we extend 2D domain into 3D and introduce $\mathbf{x}'_1, \mathbf{x}'_2$.

Then the normal will be computed as:

$$\mathbf{n} = \frac{(\mathbf{x}'_2 - \mathbf{x}_2) \times (\mathbf{x}_2 - \mathbf{x}_1)}{\|(\mathbf{x}'_2 - \mathbf{x}_2) \times (\mathbf{x}_2 - \mathbf{x}_1)\|} \quad (4.33)$$

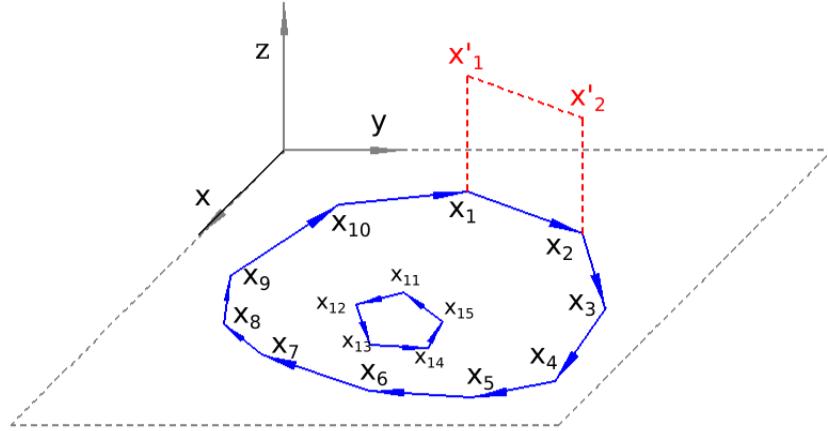


Figure 4.26: Example for constructing the local level set function for each linear section.

Remark: in order to determine whether a point is "inside" or "outside" the element e_k , we do the following

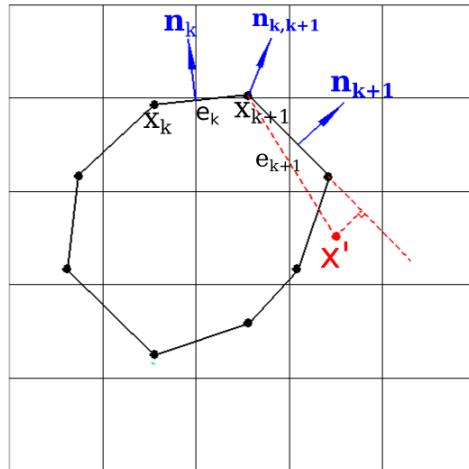


Figure 4.27: Example for determining whether a point is "inside" or "outside" the element e_k . Note, the normal projection of the point x' onto element e_k is outside the element. x' would be considered as "inside" when looking at e_k and e_{k+1} , but outside when looking at e_{k+2} . $\mathbf{n}_{k,k+1} = (\mathbf{n}_k + \mathbf{n}_{k+1})/2$

- (b) Find Eulerian bounding box about the current element " e_k ".
- (c) For each cell in the bounding box, we find a closest distance $d^{candidate}$ and a sign $s^{candidate}$. For each element,
 - i. check the normal,
 - ii. check the endpoints of the element,

iii. determine sign, e.g. in Figure 4.27, $\omega = \mathbf{x}' - \mathbf{x}_{k+1}$. If $\omega \cdot \mathbf{n}_{k,k+1} < 0$, then \mathbf{x}' is inside the object, otherwise outside the object.

Note, if $\omega \cdot \mathbf{n}$ is “close” to zero, then there can be difficulties.

- (d) If the cell is “not initialized”, then tag the cell as “initialized” and assign the closest distance and sign to the cell. If the cell is already initialized, then replace the current closest point information with the new information if $|d^{candidate}| < |d^{current}|$.

End looping through all the elements.

3. Assign all the “uninitialized” cells to have $|d^{large}|$.

4. (a) Repeat
 - (b) for each uninitialized cell, with an initialized neighbor, assign the “sign” of an initialized neighbor and tag cell as “initialized”, referring to Figure 4.28
 - (c) go back to (a) if some cells still “uninitialized”.

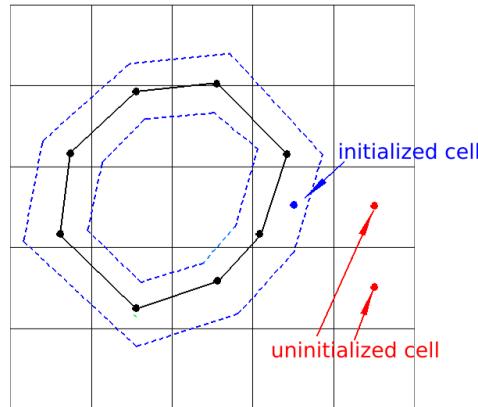


Figure 4.28: Example of ”uninitialized” cell (centered at red dots) next to initialized cell (centered at blue dot).

The specific conditions for the gingerbread man melting test list as follows:

- $\kappa_1 = \kappa_2 = 1, L = 1$
- Initial temperature field

$$\theta(x, y, 0) = \begin{cases} 274, & \text{if } y = 0, \text{ or } y = 1, \\ 273, & \text{everywhere else in the domain.} \end{cases} \quad (4.34)$$

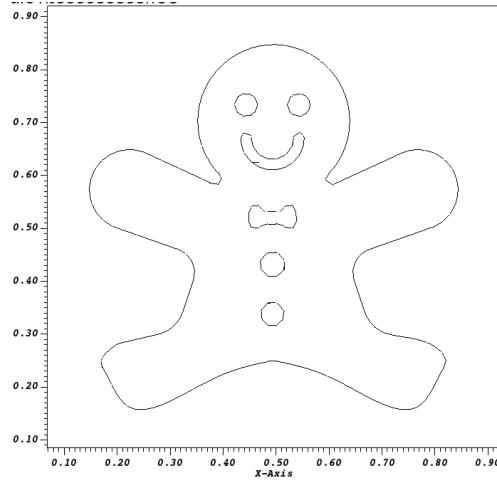


Figure 4.29: Interface reconstruction for gingerbread man melting test.

- Dirichlet boundary condition is set at the top and bottom of the computational domain $\theta_{top} = \theta_{bot} = 274$. Homogeneous Neumann boundary condition is set on the left and right of the computational domain.
- Dirichlet boundary condition $\theta_{sat} = 273$ is set at the deforming interface.

Figure 4.30 shows the temperature field and the interface evolution at two different times. For this initial temperature field, the interface is expected to be compressed in y direction.

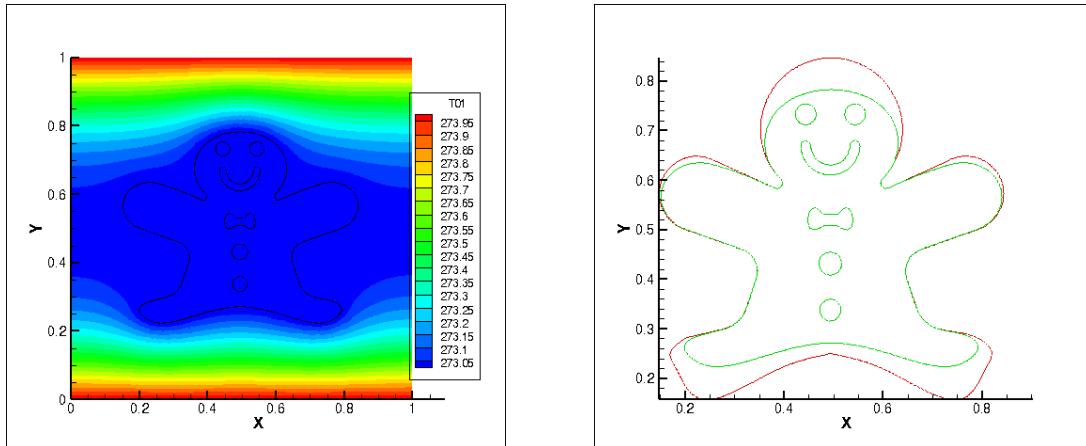


Figure 4.30: Left: Temperature field in sometime during the computation; Right: Interface evolution from red $Time = 0.0$ to green $Time = 0.02$.

Figure 4.31 shows a convergence study for three different resolutions in which Δt is variable for each resolution tried. One can clearly observe that the results do not converge.

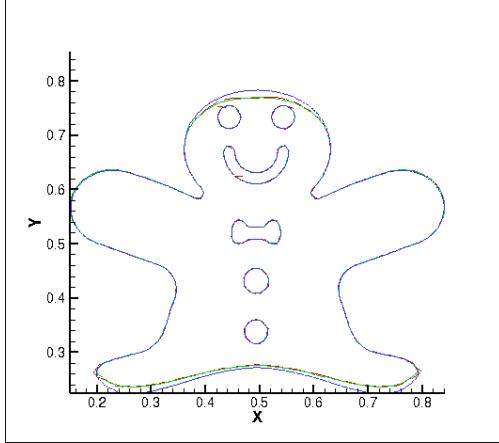


Figure 4.31: Comparison for different resolution at $Time = 0.02$.

Red: 64×64 , 45 steps.

Green: 128×128 , 91 steps.

Blue: 256×256 , 159 steps.

My thinking for why the results with 45, 91, and 159 steps do not converge is that for the 256×256 (159 steps) case, the time step is not as uniform, $\Delta t = \frac{\Delta x}{2 \max[\frac{k|\nabla \theta|}{L}]}$, as the lower resolution cases. The lack of uniformity can have an unpredictable effect on the local truncation error analysis (If we had simultaneously calculated the front speed, new front location, and temperature for each time step, we predict that a variable Δt would not adversely effect our convergence study. Our method though, is an "asynchronous method" in which the velocity $V = [\kappa \nabla \theta \cdot \mathbf{n}]/L$ is computed using the lagged temperature, prior to temperature diffusion and advection, i.e.

$$V^{(n+1)} = [\kappa \nabla \theta^{(n)} \cdot \mathbf{n}^{(n)}]/L$$

$$\theta_t = \nabla(\kappa \nabla \theta), \quad \theta = \theta_{SAT} \text{ on } \Gamma^{(n+1)}.$$

So we repeated the convergence test, except with uniform Δt for each grid resolution. As shown in Figure 4.32, from the zoom-in look, convergent results are achieved. We emphasize that for this gingerbread man test, the deforming interface has corners, and we switch from the linear exact method (see Section 3.1.9) to the "pragmatic" simple method in cells in which the area gap error (see Section 3.2.1) exceeds the threshold of the predicted radius of curvature being less than Δx . Figure 4.33 shows the cells (marked as yellow) with large gap error in different time steps.

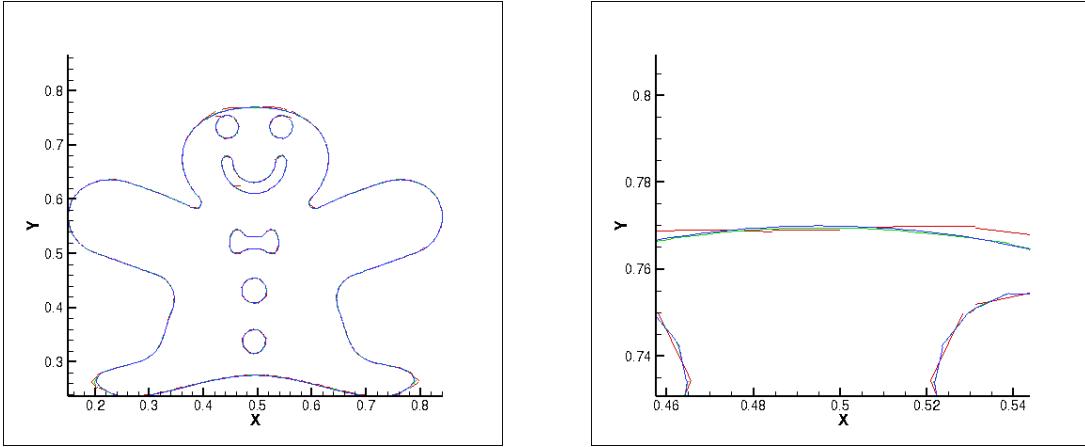


Figure 4.32: Convergence observation for different resolution at $Time = 0.02$.

Red: 64×64 , 50 steps.

Green: 128×128 , 100 steps.

Blue: 256×256 , 200 steps.

4.2.4 The Unstable Solidification-Dendrite Growth Test

For the last test of the Stefan problem, we would like to test our algorithm on an unstable Stefan problem, an unstable solidification (dendrite growth) test.

In fact, it is believed that the unstable evolution of the interface is much more common in the Stefan problem. This test has also been done by the papers, Juric & Tryggvason [40], Chen et al[11], Gibou et al.[30] and Kaiser et al. [41], etc.

To simulate the conditions of supercooling, a small frozen seed of material is placed in a surrounding region of undercooled liquid. Initial conditions for this test are as follows:

1. The initial interface is defined parametrically as:

$$\begin{aligned} x(s) &= (R + P \cos(8\pi s)) \cos(2\pi s) \\ y(s) &= (R + P \cos(8\pi s)) \sin(2\pi s), \\ \text{where } R &= 0.1, \quad P = 0.02. \end{aligned}$$

Figure 4.34 shows the shape of the initial interface. We assume the region inside the seed is D

2. The initial temperature field is

$$\theta = \begin{cases} 273, & \text{if } x \in D, \\ 272, & \text{otherwise.} \end{cases} \quad (4.35)$$

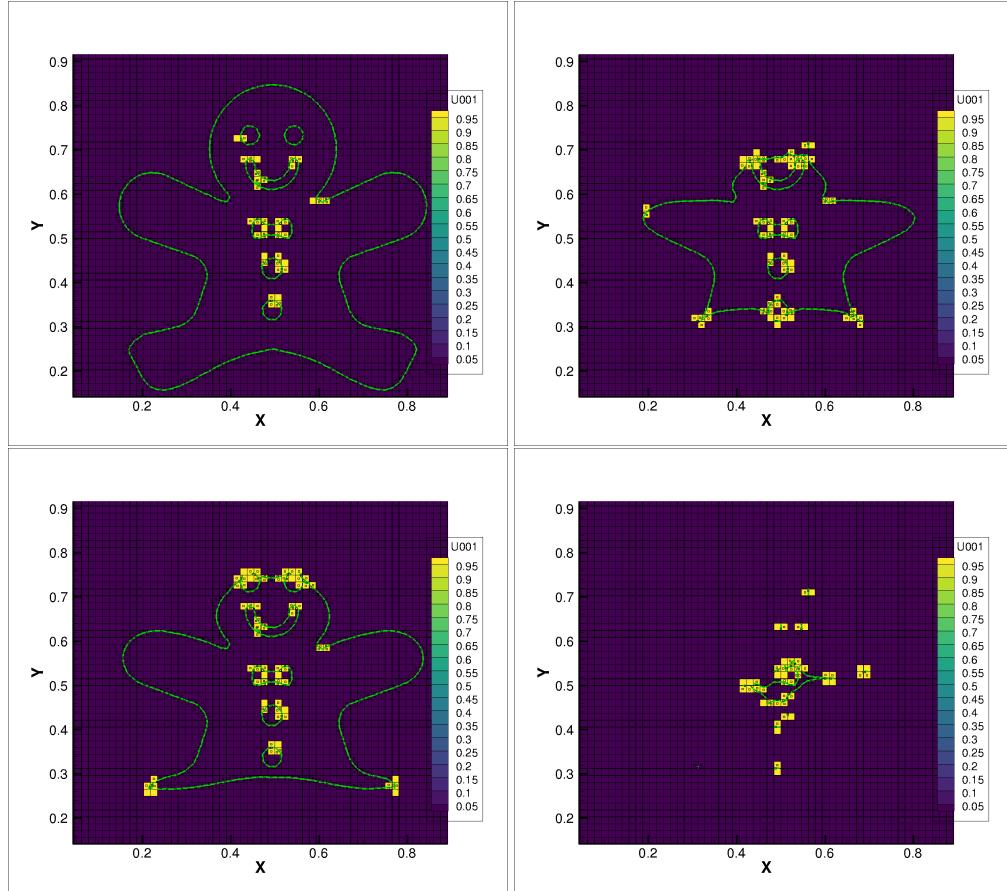


Figure 4.33: Gingerbread man evolution with area gap error analysis.

The yellow boxes show where the area gap error exceeds the threshold of a radius of curvature less than Δx .

The boundary conditions at all the computational domain boundaries are homogeneous Neumann boundary.

We test this problem with two different set-up for the Dirichlet boundary at the deforming interface.

1. Regular Dirichlet deforming boundary: the temperature is always a constant and the saturation temperature at the interface; $\theta_{SAT} = 273$.
2. Enforce the Gibbs-Thompson relation: the temperature is variational depending on the local phase-front velocity and local curvature.

Note: the Gibbs-Thompson relation (see Equation 4.36) we use in the test is a simplified version for only the isotropic case.

$$\theta(\mathbf{x}, t) = -\epsilon_1 \kappa - \epsilon_2 V \quad (4.36)$$

where κ is the local interface curvature, ϵ_1 is the surface tension coefficient and ϵ_2 is the molecular kinetic coefficient.

From the Stefan condition (Equation 2.5), we know that locally at the interface, the phase front velocity is depending on local temperature. And in the Gibbs-Thompson relation, the local saturation temperature is also depending on the phase front velocity. Here, we give the algorithm we use for solving this non-linear system, in order to determine the spatially varying normal freezing front velocity.

1. Initialize $\theta_I^{(0)} = \theta_{SAT} - \epsilon_1 \kappa - \epsilon_2 V^{(0)}$,

$$V^{(0)} = 0.$$

2. $l = 0$.

Repeat:

$$V^{(l+1)} = \frac{[\kappa \nabla \theta^{(l)} \cdot \mathbf{n}]}{L},$$

$$\theta_I^{(l+1)} = \theta_I^{(l)} + \epsilon_2 V^{(l)} - \epsilon_2 V^{(l+1)},$$

until convergence ($|V^{(l)} - V^{(l+1)}| < \epsilon$).

Figure 4.35 shows the computational result for dendrite growth at different time for 128×128 resolution when $\epsilon_1 = \epsilon_2 = 0$.

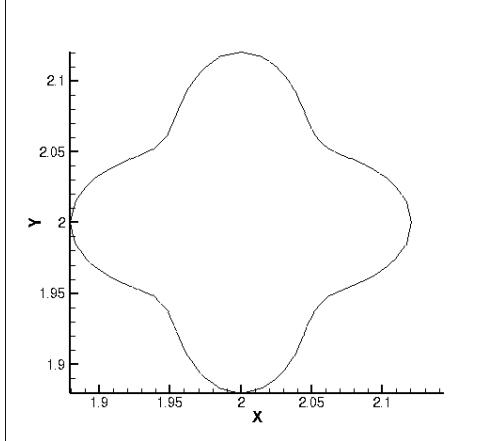


Figure 4.34: Initial seed of dendrite growth test

Figure 4.36 compares the results at the same end time $Time = 0.8$ for three different resolutions with regular Dirichlet boundary at the interface ($\epsilon_1 = \epsilon_2 = 0$).

Figure 4.37 and 4.38 compares the results at the same end time for three different resolutions with Gibbs-Thompson relation enforced at the interface. The number of time steps corresponding to each resolution

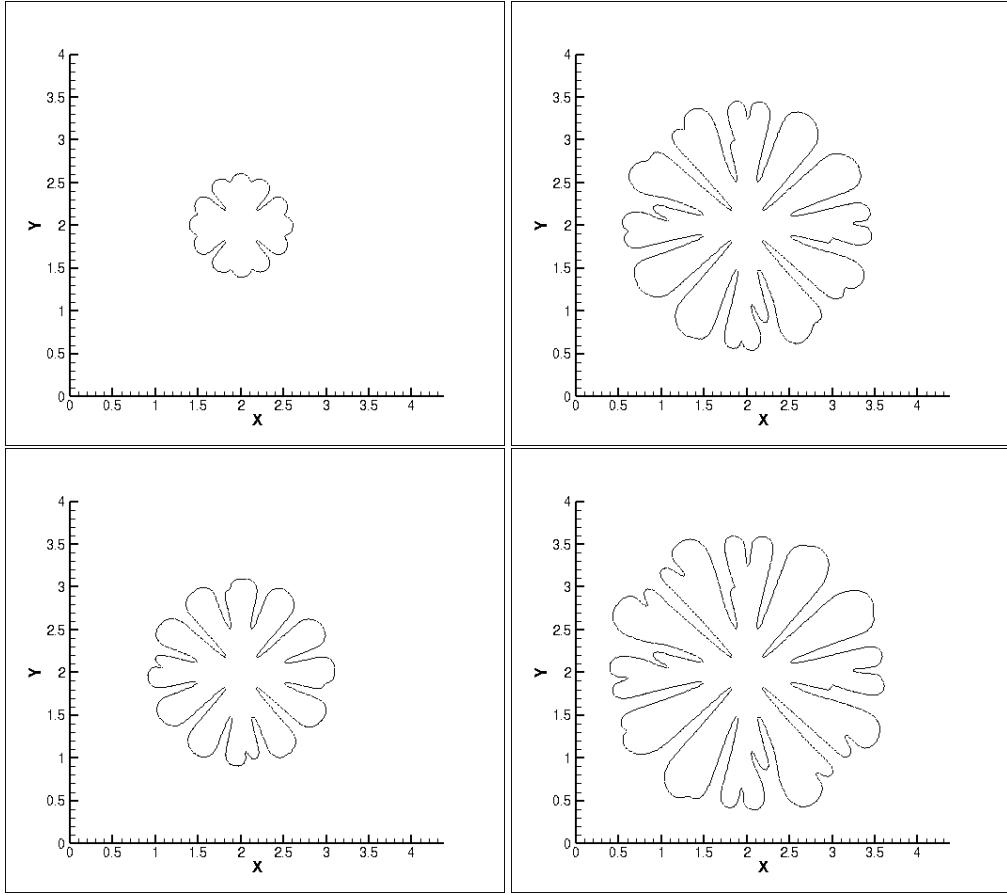


Figure 4.35: Dendrite growth evolution at different time steps for resolution 128×128 at time
 $Time = 0.130, 0.367, 0.611, 0.800$.

shown in Figure 4.37 is increasing with increasing resolution: 123 for the 64×64 case, 267 for the 128×128 case, and 651 for the 256×256 case. The number of time steps corresponding to each resolution shown in Figure 4.38 is fixed with increasing resolution: $\Delta t = 1/2000$ for all resolutions, the same as that used in the results reported by Chen et al.[11].

It is observed that neither results in Figure 4.37 or 4.38 indicate convergence for this stabilized Dendrite growth test problem. We note that convergence was not reported in Juric & Tryggvason [40], Chen et al.[11], Gibou et al.[30] and Kaiser et al.[41], either. Also the results are all different when comparing these previously reported results to each other and to our results. We really do expect convergence for this stabilized dendrite test case, and it will be a matter of future research in order to understand (a) why we do not observe convergence, and (b) why all the previous reported results differ.

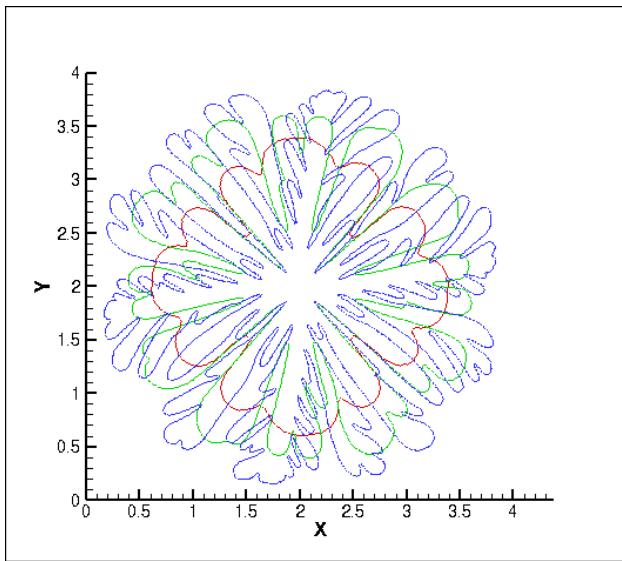


Figure 4.36: Comparison of dendrite growth without surface tension restriction (with regular Dirichlet boundary) with different resolution at $Time = 0.8$. $\epsilon_1 = \epsilon_2 = 0$.
 Red: 64×64 , Green: 128×128 . Blue: 256×256

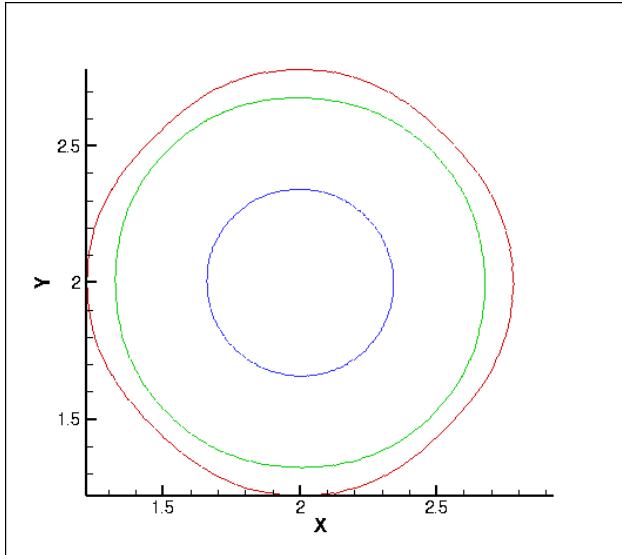


Figure 4.37: Comparison of dendrite growth with surface tension restriction with different resolution at $Time = 0.464$ with $\epsilon_1 = 0.002, \epsilon_2 = 0.002$.
 Red: 64×64 , 123 Steps; Green: 128×128 , 267 steps; Blue: 256×256 , 651 steps.

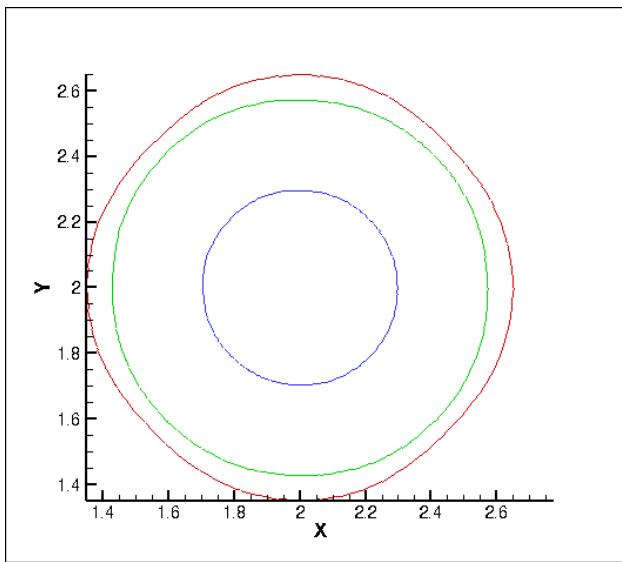


Figure 4.38: Comparison of dendrite growth with surface tension restriction with different resolution at
 $Time = 0.32$ after 640 steps with $\epsilon_1 = 0.002, \epsilon_2 = 0.002, \Delta t = 1/2000$.
Red: 64×64 ; Green: 128×128 ; Blue: 256×256 .

CHAPTER 5

CONCLUSION

Prior to our undertaking this research, it was completely unknown whether one can robustly and accurately solve complex deforming boundary problems on a supermesh. In the need of capturing micro-structure over multi-material multiphase systems, we have developed a simple, robust and conservative finite-volume method based on the completely novel deforming boundary supermesh framework. The deforming boundary Supermesh approach is found to have the property that all parcels of microscopic material, regardless of size, can be recognized in the solution of PDEs on such a supermesh. Prior work in the field of supermesh [27, 16, 44] only considered select stationary domains. In this thesis, we have shown something revolutionary: we have shown that one can viably (efficiently, accurately, robustly) solve deforming boundary problems on a deforming supermesh with as complex deformations and as small micro-structure, as one pleases. Please see Figures 4.36 where we do grid refinement studies of the unstable dendrite problem. As expected, as the grid is refined, more structure appears. For each grid resolution, the simulation marches on efficiently regardless of the structure of the supermesh grid, or the distribution of microscopic parcels, on a given time step. Please also see Figures 4.30, in which we start with a "complex geometry", the Gingerbread man, and then solve the Stefan problem starting with this initial complex geometry. As time evolves for this Gingerbread man problem, the melting object assumes progressively more and more complex shapes as the gingerbread melts to nothing. As with the unstable dendrite problem, our method solves on the evolving melting "Gingerbread" character efficiently, robustly, and accurately, all the way to the point where the "Gingerbread" man has completely vanished. We have shown that in smooth regions, that is where the area fraction gap error is below a quantifiable gap error criterion (see Section 3.14), we can use our new linear exact method which results in convergence of solutions under grid refinement (see sections 4.2.1, 4.2.3, 4.2.2). When the area gap error exceeds a threshold we resort to the "pragmatic" approach, which gives meaningful results for any deforming boundary problem that we have tried.

The proposed method has been tested on stationary interface problems as well as complex deforming boundary problem, the Stefan problem. The embedded interface boundary conditions can be either (i) jump condition boundary condition, (ii) Dirichlet boundary condition, or (iii) Neumann boundary condition.

For the stationary interface case, the proposed method is validated in the case of canonical configurations, such as a thick or thin annulus (3 materials), pentafoil with thin filament (3 materials), hypocycloid (2 or 6 materials) and a multi-scale filament test problem (3 materials). For the Stefan problem, the proposed method is validated in the case of the moving planar interface with exact solution, the shrinking circle comparing with 1D body-fitted algorithm, gingerbread man melting and unstable solidification.

For the stationary interface case, convergence results for the temperature field and its gradient indicate first order in time and second order in space convergence for Dirichlet and Neumann embedded interfacial boundary conditions, and first order in time and first order in space for jump condition embedded interfacial boundary conditions. For the Stefan problem, the tests show a first order convergence of the proposed method due to the first order “normal probe” technique for computing the temperature normal gradient at the interfaces.

The supermesh framework is based on MOF reconstruction which gives accurate multi-material interface reconstruction and is able to support conservative and robust numerical methods which recognize any sized micro-structure without applying AMR technique or approximating by constructing additional models. For fluid dynamic problems in which there is a large separation between the “macro” scale and the “micro” scale $\frac{l_{macro}}{l_{micro}} > 1000$, and the scales interact, the Adaptive Mesh Refinement technique is not a realistic option. Specializing in problems with large scale ratio, the proposed method could be considered as a complement of problems which are inaccessible via conventional AMR or cut-cell method. The new deforming boundary supermesh algorithm is a general approach for multi-scale problems when the multi-scale phenomena is “isolated” (see Figure 1.8). Existing approaches, “pre” supermesh, are very problem specific e.g.[88, 83, 9].

For problems with laminar boundary layer (thermal or viscous) or problems with “micro-layers”, we believe the present method is ideal (see for example results in Figures 4.33, 4.23). On the other hand, the present method will not be of much use if there is large variation in fluid states tangential to deforming interfaces (such as turbulent boundary viscous or thermal layers) or where microscopic material is torn away from interfaces. In such a case a hybrid method such as AMReX+GPU+MPI+OMP+SUPERMESH might be ideal as far as DNS goes or perhaps some kind of hybrid supermesh method +turbulent boundary layer model trained with assimilated PIV data is more realistic.

In the future, we plan to extend our deforming boundary supermesh approach to more general phase change problems in which there is micro-structure. Examples include evaporation of droplets in internal

combustion engines[7], evaporation of virus laden droplets expelled from singing or coughing[68], heat transfer and phase change in geared turbofan lubrication systems[115], nuclear boiling in microgravity[88, 108, 83, 119, 9], hydrate formation in the ocean depths[14, 85, 109], heat pipe in microgravity[49, 48], thermal spray material processing systems[76, 1], and additive manufacturing systems[23].

Looking even further down the line, we are interesting in coupling our new supermesh technology with data assimilation, parameter estimation, and optimization approaches in order to design multiphase turbulence models[50, 51] or capture turbulent thermal or viscous boundary layers at deforming boundaries changing phase[8].

APPENDIX A

MOF BOILING ALGORITHM USING GHOST FLUID METHOD (NOT SUPERMESH)

A.1 Governing Equations

The governing equations to be solved are for incompressible, immiscible, multiphase flows, which are given as follows:

For each material m ,

$$\nabla \cdot \vec{u} = 0 \quad (\text{A.1})$$

$$(\Phi_m)_t + \vec{V}_I \cdot \nabla \Phi_m = 0, \quad m = 1, 2, \dots, M \quad (\text{A.2})$$

$$\frac{\partial}{\partial t}(\rho_m \vec{u}) + \nabla \cdot (\rho_m \vec{u} \otimes \vec{u}) = -\nabla \cdot (p_m \mathbb{I}) + \nabla \cdot \vec{r} + \rho_m \vec{g} \quad \text{if } \phi_m(\vec{x}, t) > 0 \quad (\text{A.3})$$

$$\frac{\partial}{\partial t}(\rho_m C_{p,m} T) + \nabla \cdot (\vec{u}_m \rho_m C_{p,m} T) = \nabla \cdot (\kappa_m \nabla T) + \vec{r} : \nabla \vec{u} \quad \text{if } \phi_m(\vec{x}, t) > 0 \quad (\text{A.4})$$

where

- For material m , ρ_m is the density,

$$\rho_m = \rho_{m0}(1 + \alpha_m(T_m - T_{m0})), \quad (\text{A.5})$$

p_m is the pressure, μ_m is the dynamic viscosity, \vec{g} is the acceleration due to gravity, $C_{p,m}$ is the heat capacity per unit of mass at constant pressure, k_m is the thermal conductivity.

- V_I is the interface velocity for which we use the same model as used by [28][97]:

$$\vec{V}_I = \vec{V}_{vap} - \frac{\dot{m}}{\rho_{vap}} \vec{n}_{lv} = \vec{V}_{liq} - \frac{\dot{m}}{\rho_{liq}} \vec{n}_{lv} \quad (\text{A.6})$$

$$\dot{m} = \frac{[\kappa \nabla T \cdot \vec{n}_{lv}]_\Gamma}{L_{vap}} \quad (\text{A.7})$$

where \vec{n}_{lv} is the normal vector at the interface Γ which points from liquid to vapor phase, and L_{vap} is the latent heat of vaporization.

- ϕ_m is the level set function for material m

$$\phi_m(\vec{x}, t) = \begin{cases} > 0, & \vec{x} \in \text{material m,} \\ \leq 0, & \text{otherwise.} \end{cases}$$

- $\vec{u} = (u, v, w)$ is the velocity field.
- We assume the temperature at the phase change front(vapor- liquid interface) is always the saturation temperature T_{sat}
- $\vec{\tau}$ is the deviatoric stress tensor for a Newtonian fluid,

$$\vec{\tau} = \mu_m (\nabla \vec{u} + \nabla \vec{u}^T)$$

- $\vec{\tau} : \nabla \vec{u}$ is the viscous diffusion term which we include here in the formulation, but neglect in the numerical implementation.

The advection part of the time derivative in (A.2) through (A.4) is further split by way of directional splitting. For example, if advection in the x direction is approximated first, then we solve:

$$(\phi_m)_\tau + (u\phi_m)_x = u_x \phi_m \quad (\text{A.8})$$

$$(H_m)_\tau + (uH_m)_x = u_x H_m \quad (\text{A.9})$$

$$(u\rho_m H_m)_\tau + (\rho_m uu H_m)_x = 0 \quad (\text{A.10})$$

$$(u\rho_m H_m)_\tau + (\rho_m uv H_m)_x = 0 \quad (\text{A.11})$$

$$(\rho_m C_{p,m} T)_\tau + (u\rho_m C_{p,m} T)_x = 0 \quad (\text{A.12})$$

$$\frac{\partial \vec{x}_m}{\partial \tau} = \frac{\left(\oint_{\partial(\Omega_m \cap \Omega)} \vec{x} \vec{v}_b \cdot \vec{n} dS \right) \left(\int_{\Omega_m \cap \Omega} dV \right) - \left(\oint_{\partial(\Omega_m \cap \Omega)} \vec{v}_b \cdot \vec{n} dS \right) \left(\int_{\Omega_m \cap \Omega} \vec{x} dV \right)}{\left(\int_{\Omega_m \cap \Omega} dV \right)^2} \quad (\text{A.13})$$

where, H_m is the Heaviside function which defined as

$$H_m = H(\phi_m) = \begin{cases} 1, & \text{if } \phi_m \geq 0, \\ 0, & \text{otherwise.} \end{cases}$$

\vec{x}_m is the centroid of material m , v_b is the Eulerian velocity of the boundary $\partial(\Omega_m \cap \Omega)$

We perform the advection in $X - Y - Z$ orderings and reverse it in next time step as $Z - Y - X$. And we keep on this alternating ordering for advection.

A.2 Material Advection

To advect equation A.8 and A.9, we apply the method introduced by Weymouth and Yue[111] with directional split cell integrated semi-Lagrange method(CISL) due to the perfect guarantee of volume conservation both in 2D and 3D.

Weymouth and Yue's algorithm for material advection solves the following directionally splitting formulation:

$$(F_m)_t + (u_d F_m)_{x_d} = 0, \quad \text{if } F_m < 1/2 \quad (\text{A.14})$$

$$(F_m)_t + (u_d F_m)_{x_d} = u_{x_d}, \quad \text{if } F_m \geq 1/2 \quad (\text{A.15})$$

where $d = 1, 2, 3$ represents the different directions.

Note here, there is a tiny drawback for Weymouth and Yue's method is the time step. comparing with EI-LE approach which satisfy $CFL < 1/2$, the Weymouth and Yue's method better has $CFL < 1/4$ in 2D and $CFL < 1/6$ in 3D to avoid overshoot or undershoot problem. In addition,

to simplify the problem, for equation A.15, we solve the compliment problem when $G = 1 - F_m < 1/2$:

$$(F_m)_t + (u_d F_m)_{x_d} = u_{x_d}, \quad \text{if } F_m \geq 1/2$$

$$\text{Let } G = 1 - F_m$$

$$F_m = 1 - G$$

$$-G_t + (u_d(1 - G))_x = u_{x_d}$$

$$-G_t - (u_d G)_x = 0$$

$$G_t + (u_d G)_x = 0$$

Then we need to know how to solve equation A.14, we apply directionally splitting CISL method which do the following steps,

1. Backward tracing of characteristics
2. Intersection of departure region and target region

Here we illustrate these procedures in the x direction (The CISL algorithm in the other directions is carried out similarly).

For a cell $\Omega_{i,j}$, we have velocity $u_{i-1/2}, u_{i+1/2}$ stored on the face center of the left and right boundary $x_{i-1/2}, x_{i+1/2}$.

$$\Omega_{target} = \Omega_{i,j} \quad (= \{(x, y) | x_{i-1/2} < x < x_{i+1/2}, y_{i-1/2} < y < y_{i+1/2}\}) \quad (\text{A.16})$$

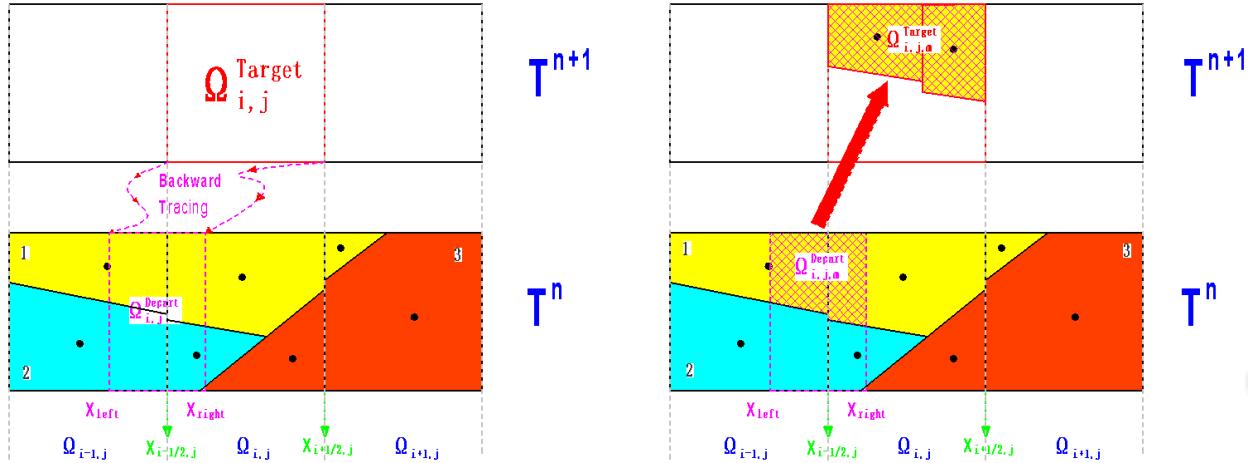


Figure A.1: Material advection for material 1(yellow material) in target region $\Omega_{i,j}$

- Backward tracing of Characteristic

$$x_{left} = x_{i-1/2} - \Delta t u_{i-1/2} \quad (\text{A.17})$$

$$x_{right} = x_{i+1/2} - \Delta t u_{i+1/2} \quad (\text{A.18})$$

$$\Omega_{departure} = \{(x, y) | x_{left} < x < x_{right}, y_{i-1/2} < y < y_{i+1/2}\} \quad (\text{A.19})$$

And we can define this linear backward mapping function as

$$T_{i,j}^{CISL} = f(\alpha x + \beta, y) \quad (\text{A.20})$$

where, $\alpha = \frac{\Delta x}{x_{right} - x_{left}}$, $\beta = x_{i-1/2} - \alpha x_{left}$.

- Intersect the departure region with target region

Since, the advection scheme is constrained by CFL number, the departure region is intersected with the target and its two neighboring cells, to get the volume fraction and centroid for the target cell in the next time step. This procedure is formulated as following:

$$F_{m,i,j}^{n+1} = \frac{\sum_{i'=-1}^1 \int_{T_{i,j}^{CISL}(\Omega_{m,i+i',j}^n) \cap \Omega_{i,j}} dV}{\int_{\Omega_{i,j}} dV} \quad (\text{A.21})$$

$$\vec{x}_{m,i,j}^{n+1} = \frac{\sum_{i'=-1}^1 \int_{T_{i,j}^{CISL}(\Omega_{m,i+i',j}^n) \cap \Omega_{i,j}} \vec{x} dV}{F_{m,i,j}^{n+1} \int_{\Omega_{i,j}} dV} \quad (\text{A.22})$$

Note: The interface is reconstructed using the updated volume fractions and centroids after each directional sweep, and the updated volume fractions and centroids are used for the next directional sweep. We truncate

the volume fraction value in the regions away from the interface,

$$F_{m,i,j} = \begin{cases} 0, & F_{m,i,j} < \epsilon \text{ and } \phi_{m,i,j} < -2 \max(\Delta x, \Delta y) \\ 1, & F_{m,i,j} > 1 - \epsilon \text{ and } \phi_{m,i,j} > 2 \max(\Delta x, \Delta y) \\ F_{m,i,j}, & \text{otherwise.} \end{cases} \quad (\text{A.23})$$

and normalize the volume fractions for each cell

$$F_{m,i,j}^{normalized} = \frac{F_{m,i,j}}{\sum_{m'=1}^M F_{m',i,j}} \quad m = 1, 2, \dots, M. \quad (\text{A.24})$$

A.3 Momentum Advection

To advect momentum equation A.10, we still use directionally split backward tracing for CISL method.

We solve

$$(u_j \rho_m)_t + (u_j \rho_m u_i)_x = 0 \quad (\text{A.25})$$

where, $i, j = 1, 2, 3$ for 3D problem

Here we only discuss the x momentum in 2D

$$(u \rho_m)_t + (u \rho_m u)_x = 0 \quad (\text{A.26})$$

$$(u \rho_m)_t + (u \rho_m v)_y = 0 \quad (\text{A.27})$$

The velocities are still stored on the MAC grid(face center), but the resulting momentum will either be stored on the MAC grid or cell center. We made a comparison in our numerical experiment.

- Momentum on the material centroids

In this case, we did the same with material advection since the information are still stored on the cell centers.

- Momentum on the MAC grid(face centers)

1. First, we divide each cell into half evenly.

For advection in x direction, we divide the cell vertically.

So for cell $\Omega_{i,j} = \{(x, y) | x_{i-\frac{1}{2}} < x < x_{i+\frac{1}{2}}, y_{i-\frac{1}{2}} < y < y_{i+\frac{1}{2}}\}$

– Left half: $\Omega_{i-\frac{1}{2},R} = \{(x, y) | x_{i-\frac{1}{2}} < x < x_i, y_{i-\frac{1}{2}} < y < y_{i+\frac{1}{2}}\}$

Note: R here represents this left half of cell $\Omega_{i,j}$ is to the right of face $x_{i-\frac{1}{2}}$

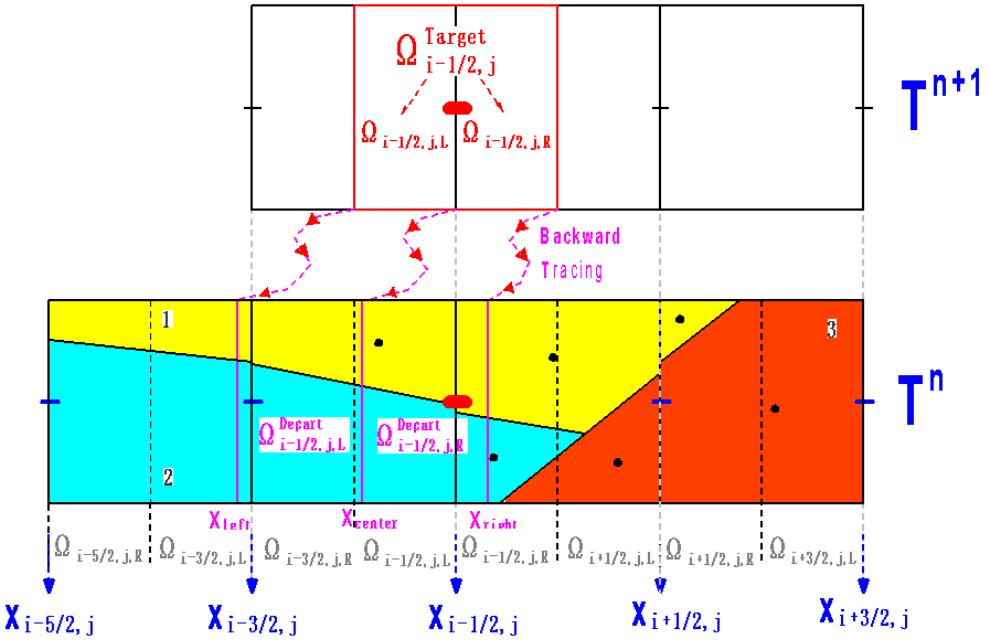


Figure A.2: Momentum advection of $u_{i-1/2,j}$ on MAC grid in x-direction: backward characteristic tracing

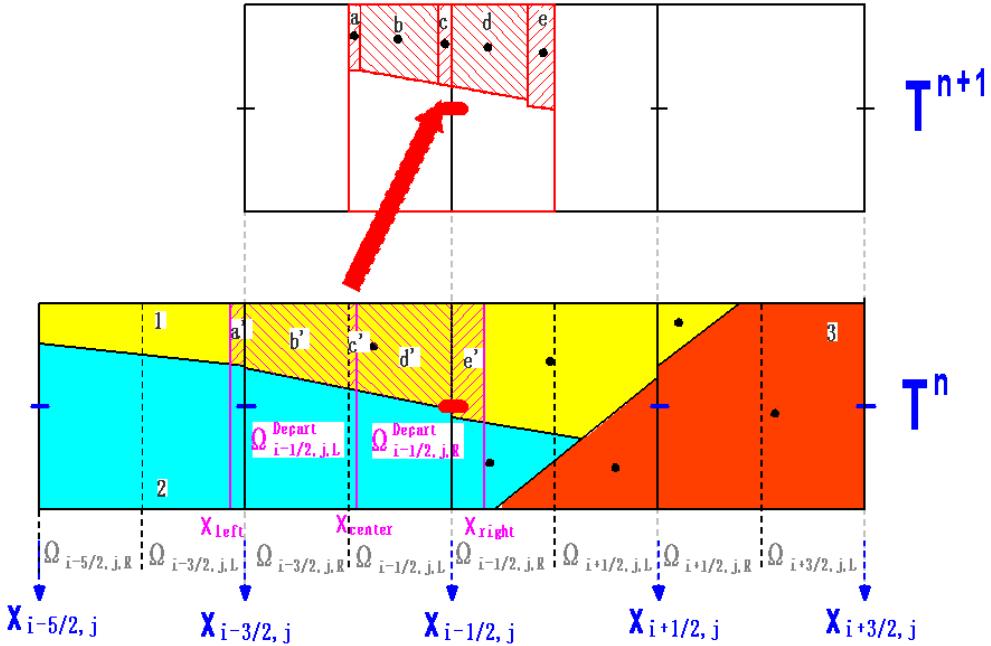


Figure A.3: Momentum advection of $u_{i-1/2,j}$ on MAC grid in x-direction for material 1 (yellow material): intersection and mapping

– Right half: $\Omega_{i+\frac{1}{2},L} = \{(x,y) | x_i < x < x_{i+\frac{1}{2}}, y_{i-\frac{1}{2}} < y < y_{i+\frac{1}{2}}\}$

Note: Similarly, L here represents this right half of cell $\Omega_{i,j}$ is to the left of face $x_{i+\frac{1}{2}}$

2. After the division in first step, we can tie each half-cell to its corresponding cell faces.

So the target cell $\Omega_{i-\frac{1}{2},j}^{target}$ which centered at cell face $x_{i-\frac{1}{2}}$ is comprised of

$$\Omega_{i-\frac{1}{2},j}^{target} = \Omega_{i-\frac{1}{2},j,L} \cap \Omega_{i-\frac{1}{2},j,R} \quad (\text{A.28})$$

3. Then we backward trace the boundaries of target region to find the corresponding departure region.

$$x_{left} = x_{i-1,j} - \Delta t \frac{(u_{i-\frac{3}{2},j} + u_{i-\frac{1}{2},j})}{2} \quad (\text{A.29})$$

$$x_{center} = x_{i-\frac{1}{2},j} - \Delta t u_{i-\frac{1}{2},j} \quad (\text{A.30})$$

$$x_{right} = x_{i,j} - \Delta t \frac{(u_{i-\frac{1}{2},j} + u_{i+\frac{1}{2},j})}{2} \quad (\text{A.31})$$

Note: here, since the velocities are stored at center of faces. The velocity at the node u_{i-1} is defined as the average of $u_{i-\frac{3}{2}}$ and $u_{i-\frac{1}{2}}$. Then the departure region is:

$$\Omega_{i-\frac{1}{2},j}^{departure} = \Omega_{i-\frac{1}{2},j,L}^{departure} \cap \Omega_{i-\frac{1}{2},j,R}^{departure} \quad (\text{A.32})$$

where,

$$\Omega_{i-\frac{1}{2},j,L}^{departure} = \{(x,y) | x_{left} < x < x_{center}, y_{i-\frac{1}{2}} < y < y_{i+\frac{1}{2}}\} \quad (\text{A.33})$$

$$\Omega_{i-\frac{1}{2},j,R}^{departure} = \{(x,y) | x_{center} < x < x_{right}, y_{i-\frac{1}{2}} < y < y_{i+\frac{1}{2}}\} \quad (\text{A.34})$$

4. The last step is to intersect the departure region which is backward tracing region from target region at time step t^{n+1} with the half-cells in time step t^n .

(Note: Under the constrains of CFL condition, $\Omega_{i-\frac{1}{2},j,L}^{departure}$ can only fall onto cells $\Omega_{i+i',j}$ where $i' = -2, -1, 0$, yet it is only possible for region $\Omega_{i-\frac{1}{2},j,R}^{departure}$ to fall onto cells $\Omega_{i+i',j}$ where $i' = -1, 0, 1$.

So if we divide all cells into half-cells, all half-cells that may have intersection with $\Omega_{i-\frac{1}{2},j,L}^{departure}$, which in our example from left to right including $\Omega_{i-3/2,j,L}$, $\Omega_{i-3/2,j,R}$, $\Omega_{i-1/2,j,L}$, $\Omega_{i-1/2,j,R}$ and $\Omega_{i+1/2,j,L}$. Similarly, the half-cells which could possibly have intersection with $\Omega_{i-\frac{1}{2},j,R}^{departure}$ is also 5 including $\Omega_{i-3/2,j,R}$, $\Omega_{i-1/2,j,L}$, $\Omega_{i-1/2,j,R}$, $\Omega_{i+1/2,j,L}$ and $\Omega_{i+1/2,j,R}$.) In order to simplify my formulation representation, I include one more half-cell for each case.

The intersection region for material m is calculated as following:

$$V_{i',i-\frac{1}{2},j}^{L,x} = \sum_{k=L \text{ or } R} \Omega_{m,i+(1/2+i'),j,k} \cap \Omega_{i-\frac{1}{2},j,L}^{departure} \quad \text{where, } i' = -2, -1, 0 \quad (\text{A.35})$$

$$V_{i',i-\frac{1}{2},j}^{R,x} = \sum_{k=L \text{ or } R} \Omega_{m,i+(1/2+i'),j,k} \cap \Omega_{i-\frac{1}{2},j,R}^{departure} \quad \text{where, } i' = -1, 0, 1 \quad (\text{A.36})$$

5. Finally, the new velocity profile at $\tilde{u}_{i-1/2,j}$ is updated calculated by finding the momentum in the intersection regions, divided by the density of materials in them. In our numerical method, we assume all the material in the same cell have the same velocity profile.

$$\tilde{u}_{i-1,j}^x = \frac{\sum_{m=1}^M \left(\sum_{i'=-2}^0 \int_{V_{i',i-\frac{1}{2},j}^{L,x}} \rho_m u_{m,i+(1/2+i'),j} dx + \sum_{i'=-1}^1 \int_{V_{i',i-\frac{1}{2},j}^{R,x}} \rho_m u_{m,i+(\frac{1}{2}+i'),j} dx \right)}{\sum_{m=1}^M \left(\sum_{i'=-2}^0 \int_{V_{i',i-\frac{1}{2},j}^L} \rho_m dx + \sum_{i'=-1}^1 \int_{V_{i',i-\frac{1}{2},j}^R} \rho_m dx \right)}$$

After we updated every u (horizontal velocity) on the MAC grid as \tilde{u}^x , which is an intermediate values during directionally splitting, we then start to advect it in transverse velocity v on the MAC grid. We go through the steps as we discussed above again.

1. The target region doesn't change, and it is comprised of two vertical regions.

$$\Omega_{i-\frac{1}{2},j}^{target} = \Omega_{i-\frac{1}{2},j,L} \cap \Omega_{i-\frac{1}{2},j,R}$$

2. we advect the target region in y-direction, using the top $v_{i-1,j+1/2}$ and bottom $v_{i-1,j-1/2}$ for left part $\Omega_{i-1/2,j,L}$. Also, $v_{i,j+1/2}$ and $v_{i,j-1/2}$ for the right part $\Omega_{i-1/2,j,R}$ to find the departure region.

$$y_{L,top} = y_{j+\frac{1}{2}} - \Delta t v_{i-1,j+1/2} \quad (\text{A.37})$$

$$y_{L,bot} = y_{j-\frac{1}{2}} - \Delta t v_{i-1,j-1/2} \quad (\text{A.38})$$

$$y_{R,top} = y_{j+\frac{1}{2}} - \Delta t v_{i,j+1/2} \quad (\text{A.39})$$

$$y_{R,bot} = y_{j-\frac{1}{2}} - \Delta t v_{i,j-1/2} \quad (\text{A.40})$$

$$\Omega_{i-1/2,j,L}^{Depart,y} = \{(x, y) | x_{i-1} < x < x_{i-1/2}, y_{L,bot} < y < y_{L,top}\} \quad (\text{A.41})$$

$$\Omega_{i-1/2,j,R}^{Depart,y} = \{(x, y) | x_{i-1} < x < x_{i-1/2}, y_{R,bot} < y < y_{R,top}\} \quad (\text{A.42})$$

$$(\text{A.43})$$

3. Intersect the departure region with related cells.

$$V_{j',i-\frac{1}{2},j}^{L,y} = \Omega_{m,i-1,j+j'} \cap \Omega_{i-\frac{1}{2},j,L}^{departure} \quad \text{where, } j' = -1, 0, 1 \quad (\text{A.44})$$

$$V_{j',i-\frac{1}{2},j}^{R,y} = \Omega_{m,i,j+j',k} \cap \Omega_{i-\frac{1}{2},j,R}^{departure} \quad \text{where, } j' = -1, 0, 1 \quad (\text{A.45})$$

4. Finally the velocity \tilde{u} which we got from advecting in x-direction is updated and we complete our advection for velocity u and get new u profile u^{new} .

$$u_{i-1,j}^{new} = \frac{\sum_{m=1}^M \left(\sum_{j'=-1}^1 \int_{V_{j',i-\frac{1}{2},j}^{L,y}} \rho_m \tilde{u}_{m,i-\frac{1}{2},j+j'}^x dx + \sum_{j'=-1}^1 \int_{V_{j',i-\frac{1}{2},j}^{R,y}} \rho_m \tilde{u}_{m,i-\frac{1}{2},j+j'}^x dx \right)}{\sum_{m=1}^M \left(\sum_{j'=-1}^1 \int_{V_{j',i-\frac{1}{2},j}^L} \rho_m dx + \sum_{j'=-1}^1 \int_{V_{j',i-\frac{1}{2},j}^R} \rho_m dx \right)} \quad (\text{A.46})$$

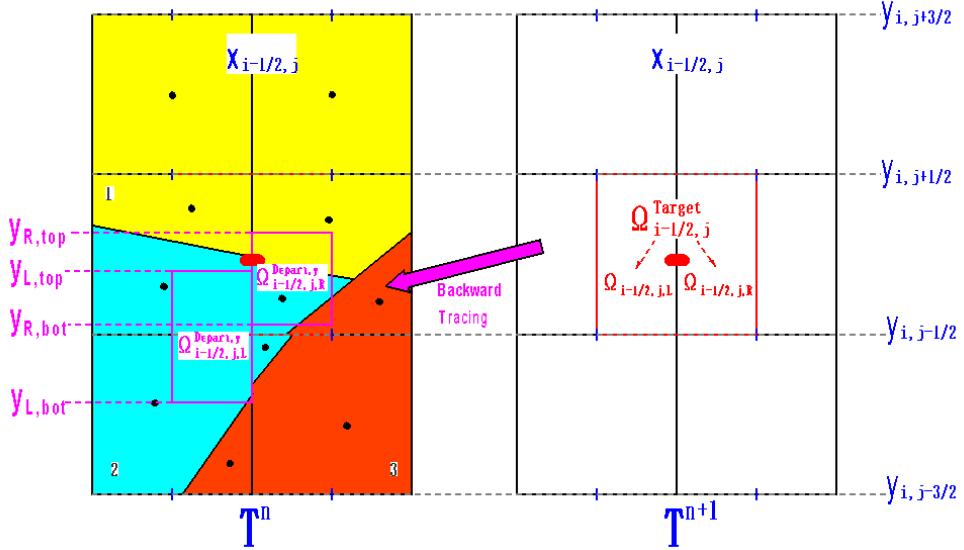


Figure A.4: Momentum advection of $u_{i-1/2,j}$ on MAC grid in y-direction: backward characteristic tracing

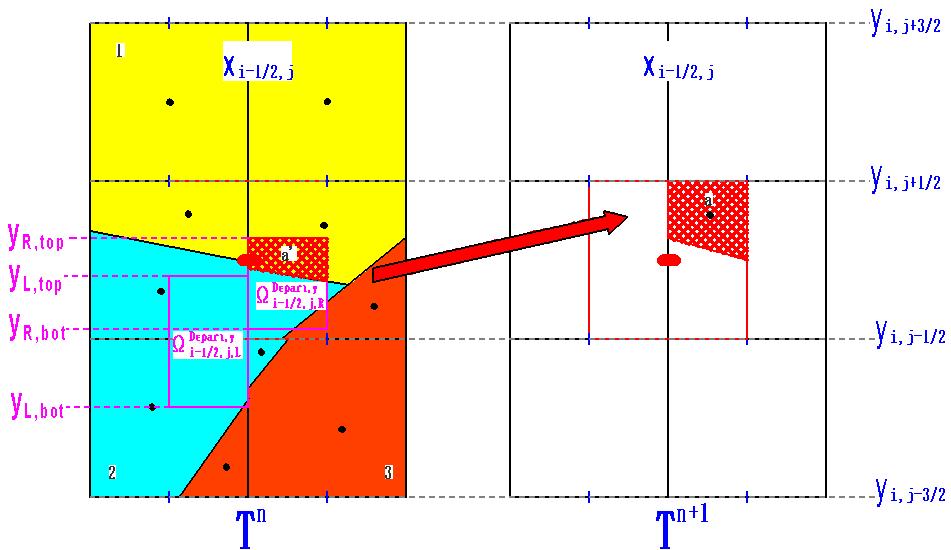


Figure A.5: Momentum advection of $u_{i-1/2,j}$ on MAC grid in y-direction for material 1 (yellow material): intersection and mapping

A.4 Energy Advection

Since the temperature profile is stored on the material centroids, the procedure of energy advection is the same with material advection which we discussed earlier in the paper. Directionally splitting cell integrated semi-Lagrange method is applied here. For the method in x direction, the temperature is updated as following:

$$T_{m,i,j}^x = \frac{\sum_{i'=-1}^1 \int_{V_{m,i+i',j}^n} \rho_m C_{p,m} T_m dx}{\sum_{i'=-1}^1 \int_{V_{m,i+i',j}^n} \rho_m C_{p,m} dx} \quad (\text{A.47})$$

A.5 Temperature Diffusion

We solve the temperature diffusion using ghost fluid method. The diffusion operator $\nabla T \cdot (\kappa_m \nabla T_m)$ is discretized with standard central difference second order method, while considering that Dirichlet boundary condition, a ghost value $T = T_{sat}$ is enforced on the liquid-gas interface(boiling front).

A cell is tagged as newly boiling cell when it does not belong to the vapor phase at time step t^n and by the boiling process become part of the vapor phase at time step t^{n+1} . We use linear interpolation to find the crossing time (A.48) where the temperature equals saturation temperature T_{sat} and then modify the temporal discretization with this time (A.49).

$$t_{v,i,j}^{crossing} = \frac{|\phi_{v,i,j}^{new}| t^n + |\phi_{v,i,j}^{old}| t^{n+1}}{|\phi_{v,i,j}^{new}| + |\phi_{v,i,j}^{old}|} \quad (\text{A.48})$$

$$\rho_v C_{p,v} \frac{T_{v,i,j}^{n+1} - T_{sat}}{t^{n+1} - t_{v,i,j}^{crossing}} = \nabla \cdot \kappa_{vap} \nabla T_{v,i,j}^{n+1} \quad (\text{A.49})$$

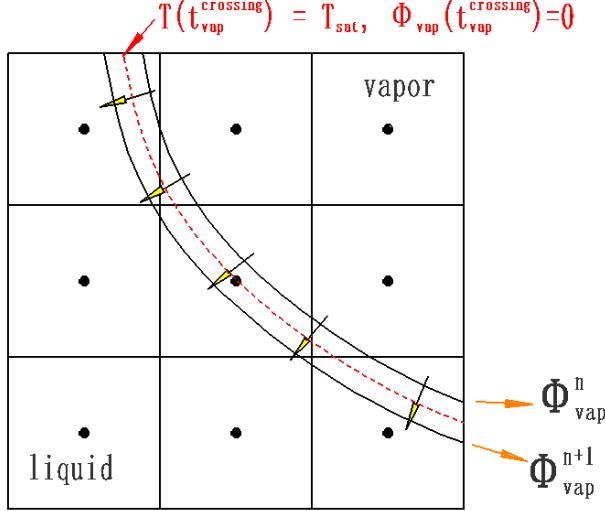


Figure A.6: Find the crossing time and modify the temporal discretization

A.6 Viscosity Term

$$u^{n+1} - \Delta t \frac{\nabla \cdot (2\mu \mathbb{D}^{n+1})}{\rho} = u^* \quad (\text{A.50})$$

BiCGSTAB with multigrid preconditioner, red-black Gauss-Seidel decoupled smoother is applied.[90]

A.7 Surface Tension Term

Ghost fluid method is applied to calculate the surface tension force if only two materials are involved.

$-\Delta t \frac{\sum_{m=1}^M \gamma_m \kappa_m \nabla H(\phi_m)}{\rho}$ is discretized at cell faces as

$$F_{i+1/2,j,m}^{MAC,tension} = -\Delta t \frac{\gamma_m \kappa_{i+1/2,j,m} (H(\phi_{i+1,j,m}) - H(\phi_{i,j,m}))}{\rho_{i+1/2,j,m}^{n+1} \Delta x} \quad (\text{A.51})$$

The surface tension force is discretized at cell centers [24] as

$$F_{i,j,m}^{tension} = \frac{\rho_{i+1/2,R} F_{i+1/2,j,m}^{MAC,tension} + \rho_{i-1/2,L} F_{i-1/2,j,m}^{MAC,tension}}{2\rho_{i+1/2,j,m}^{n+1}} \quad (\text{A.52})$$

Level set height function technique [94][93][92] is applied to approximate the curvature $\kappa_{i+1/2,j}$.[2] In 2D, we construct either a 7×3 or 3×7 stencil around each interface cell (depending on $\nabla \phi_{i,j}$).Figure A.7 and A.8

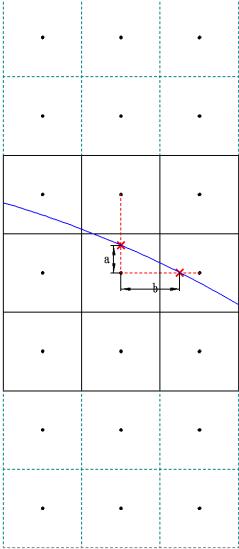


Figure A.7: Height function orientation, $|a| < |b|$, vertical

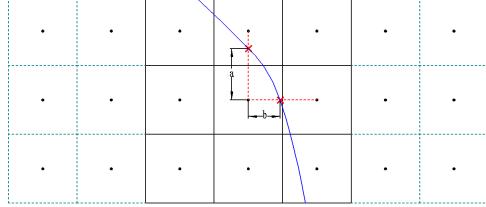


Figure A.8: Height function orientation, $|a| > |b|$, horizontal

For example, the center cell (i, j) in figure A.7, we determine the heights h_{i-1}, h_i, h_{i+1} by looking for the zero crossing of level set function ϕ for example:

$$h_{i'} = (1 - \alpha_{i'})y_{i',j} + \alpha_{i'}y_{i',j}$$

$$\text{where, } \alpha_{i'} = \frac{|\phi_{i',j}|}{|\phi_{i',j}| + |\phi_{i',j+1}|} i' = i-1, i, i+1$$

Then the heights $h_{i'}$ are then used to compute the interface normal \vec{n} and curvature κ at cell centers:

$$h' \approx \frac{h_{i+1} - h_{i-1}}{2\Delta x} \quad h'' \approx \frac{h_{i+1} + h_{i-1} - 2h_i}{(\Delta x)^2} \quad (\text{A.53})$$

$$\kappa_{i,j} = \frac{-h''}{(1 + (h')^2)^{3/2}} \quad (\text{A.54})$$

Then in order to apply ghost fluid method later, we need to interpolate the curvature from cell centers to interface (if needed). For two adjacent cells which the level set function ϕ changes signs. For example, cell (i, j) and $(i, j + 1)$. If the height function is oriented vertically, then

$$\kappa_I = \begin{cases} \kappa_{i,j}, & \text{if } |\phi_{i,j}|/\Delta_y < 1/2 \\ \kappa_{i,j+1}, & \text{otherwise} \end{cases} \quad (\text{A.55})$$

Then if the height function is oriented horizontally, then we interpolate the $\kappa_{i,j}$ and $\kappa_{i,j+1}$ to find κ_I , $(\kappa_I = (1 - \theta)\kappa_{i,j} + \theta\kappa_{i,j+1})$.

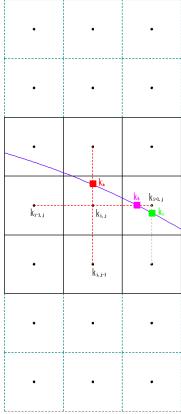


Figure A.9: Curvature interpolation in two material cells

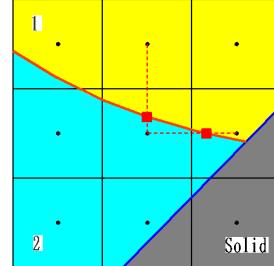


Figure A.10: Curvature interpolation for material number ≥ 3

For example, in figureA.9, the red square $\kappa_a = \kappa_{i,j}$; then we need the value of purple square, which we have to interpolate it with the green $\kappa_c = \kappa_{i+1,j}$ and red square.

For material number ≥ 3 , height function technique is not viable anymore, we use ghost fluid method with finite difference method instead. we treat it in two different cases: two fluids with one solid material, figureA.11; three fluids with "Y-junction",figureA.12.

The main idea here is to use level set functions which are stored at the cell centers to calculate the curvature

$$\kappa_{i,j} = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \quad (\text{A.56})$$

$$\int_{\Omega_{i,j}} \kappa_{i,j} d\Omega = \oint_{\partial \Omega_{i,j}} \frac{\nabla \phi}{|\nabla \phi|} \cdot \vec{n} dS \quad (\text{A.57})$$

$$\kappa_{i,j} = \frac{1}{(\Delta x)(\Delta y)} \sum_{l=1}^4 \frac{\nabla \phi_l}{|\nabla \phi_l|} \vec{n}_l \quad (\text{A.58})$$

For example, for case 1 which is showed in figure A.11,we do the following

1. Use central differencing method to calculate the normal level set function $\frac{\phi}{|\phi|}$ at the four corner of the center cell, which denoted as n^1, n^2, n^3, n^4 .

For example, $n^1 = \left(\frac{(\phi_{i,j+1} - \phi_{i-1,j+1})/\Delta x, (\phi_{i,j+1} - \phi_{i,j})/\Delta y}{\|(\phi_{i,j+1} - \phi_{i-1,j+1})/\Delta x, (\phi_{i,j+1} - \phi_{i,j})/\Delta y\|} \right)$

Note: $n^4 = n^{ghost}$, we use a ghost normal to replace the unit normal at the right bottom corner, because it involves the solid material at the right corner. We have a preset contact angle and also

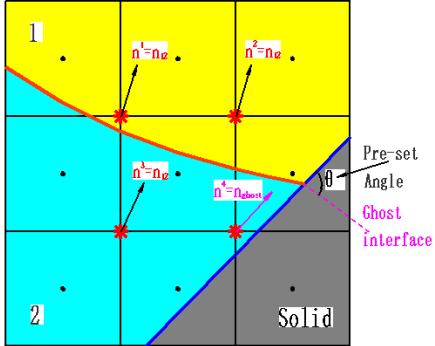


Figure A.11: Case1: two fluids with one solid material

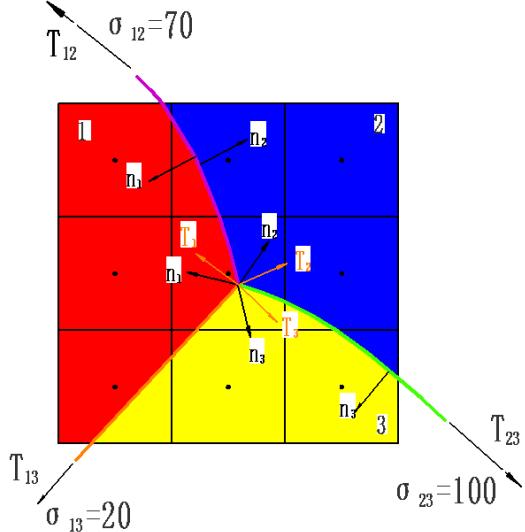


Figure A.12: Case2: three fluids with "Y-junction"

the corresponding ghost interface, then we get the ghost normal which is perpendicular to the ghost interface pointing from material 2 to material 1.

2. Then we interpolate the 4 normals on the corner to the cell faces ϕ_l , and the curvature will be calculated as the divergence of the level set function normals around the cell boundaries see equation A.58.
3. After calculating all the curvature at the cell center, we need to interpolate it back to the interface in order to calculate the surface tension force using ghost fluid method which is shown in figure A.10

A.8 Projection Method

The final step is the temperature projection.

$$\nabla \cdot \frac{\nabla p}{\rho} = \frac{\nabla \cdot u^*}{\Delta t} + \dot{m} \quad (\text{A.59})$$

$$(\text{A.60})$$

BiCGSTAB with multigrid preconditioner, red-black Gauss-Seidel decoupled smoother is applied.

Also, we deal with the expansion of liquid during phase change in this step.

$$u_{n+1} = u^* - \Delta t \frac{\nabla p}{\rho} \quad (\text{A.61})$$

$$\nabla \cdot u^{n+1} = \begin{cases} J\left(\frac{\Delta t}{vol_{cell}}\right), & \text{in expansion cells,} \\ 0, & \text{otherwise.} \end{cases} \quad (\text{A.62})$$

$$J\left(\frac{\Delta t}{vol_{cell}}\right) = \nabla \cdot u^{n+1} = \nabla \cdot u^* - \Delta t \nabla \cdot \frac{\nabla p}{\rho} \quad (\text{A.63})$$

So in an expansion cell: $\nabla \cdot u^* - \Delta t \nabla \cdot \frac{\nabla p}{\rho}$ (A.64)

$$-\nabla \cdot \frac{\nabla p}{\rho} = -\frac{\nabla \cdot u^*}{\Delta t} + J\left(\frac{1}{vol_{cell}}\right) \quad (\text{A.65})$$

where term J could be derived as:

$$J\left(\frac{\Delta t}{vol_{cell}}\right) = \frac{\Delta y(u_{liq} - u_{vap})}{vol_{cell}} = \frac{\Delta y V_{I_p} \left(\frac{\rho_{liq} - \rho_{vap}}{\rho_{vap}}\right)}{vol_{cell}} = \frac{\Delta y}{vol_{cell}} V_{I_p} \left[\frac{1}{\rho}\right] \rho_{liq} \frac{\Delta t}{\Delta t} \quad (\text{A.66})$$

$$= \frac{1}{\Delta t \cdot vol_{cell}} (vol_{vap}^{new} - vol_{vap}^{old}) \left[\frac{1}{\rho}\right] \rho_{liq} \quad (\text{A.67})$$

So, $J = \frac{1}{(\Delta t)^2} \left[\frac{1}{\rho}\right] \rho_{liq} (vol_{vap}^{new} - vol_{vap}^{old})$

Expansion term calculation Steps:

1. Calculate the rate of boiling(phase change velocity $V_{I_p} = \frac{[\kappa \nabla \theta \cdot \vec{n}]}{\rho_{liq} L}$) then extend the velocity.(See discussion in 3.2.2)

2. Use V_{I_p} to advect the interface through unsplit CISL method.

3. Loop over all the cells to narrow down the expansion cells need to be taken care. Mark the cells.

If $F_{i,j,vap} < \frac{1}{2}$ (volume fraction of the vapor in cell(i, j)) and $J > 0$ then cell(i, j) is a "donor cell".

If $F_{i,j,vap} \geq \frac{1}{2}$ and $\phi(i, j, vap) < 0$ (level set function) then cell(i, j) is a "receiving cell".

4. For each receiving cell, loop over its neighbouring 5×5 box of cells, if there is a donor cell, then check the 3×3 box of cells centered at the donor cell for any donor cell

- For each receiving cell evaluate the magnitude of the level set gradient in the direction to the donor cell

$$|\nabla_{d-r} \cdot \phi_{receiving}| = \frac{|\phi_{donor} - \phi_{receiving}|}{|\vec{x}_{donor} - \vec{x}_{receiving}|} \quad (\text{A.68})$$

Find the maximum value $|\nabla_{d-r} \cdot \phi|_{max}$.

- Evaluate and limit the weights based on the level set gradient

$$\omega_{receiving} = \begin{cases} 0, & \text{if } \frac{|\nabla_{d-r} \cdot \phi_{receiving}|}{|\nabla_{d-r} \cdot \phi|_{max}} < \sqrt{0.8} \\ 1, & \text{if } |\nabla_{d-r} \cdot \phi|_{max} = 0 \\ \frac{|\nabla_{d-r} \cdot \phi_{receiving}|}{|\nabla_{d-r} \cdot \phi|_{max}}, & \text{otherwise.} \end{cases} \quad (\text{A.69})$$

- Distribute the expansion term in the donor cell to the neighbor receiving neighbor cells using the calculated weights

$$\varepsilon_{receiving} \leftarrow \varepsilon_{receiving} + \varepsilon_{donor} \frac{\omega_{receiving}}{\sum \omega_{receiving}} \quad (\text{A.70})$$

APPENDIX B

2D GEOMETRY INTERSECTION ALGORITHM

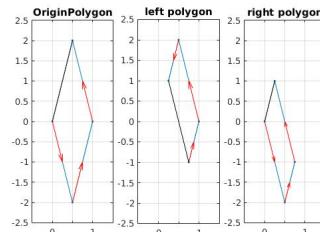
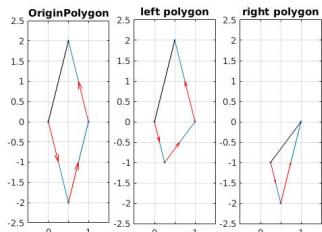
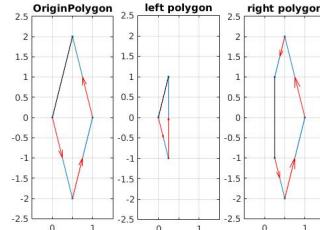
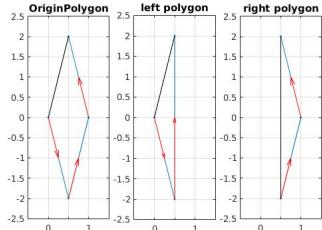
B.1 Algorithm

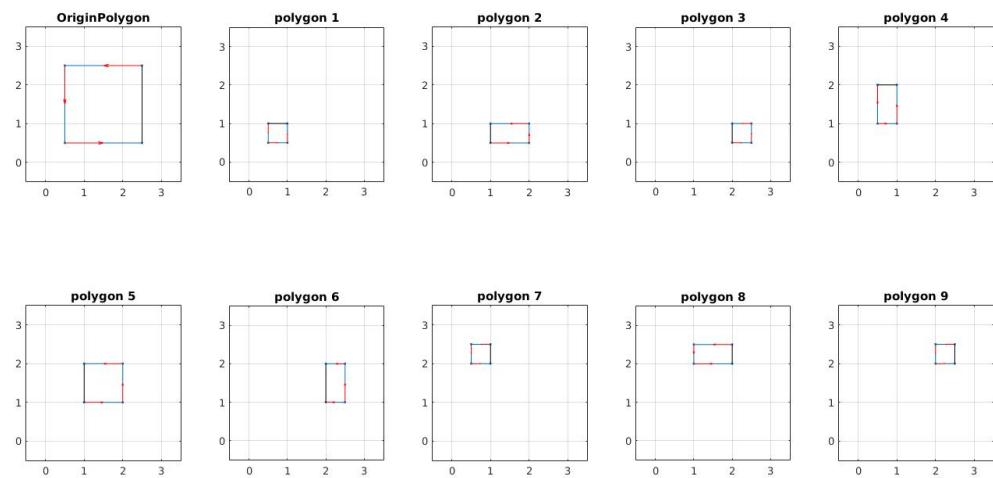
This section is to present an alternate algorithm the author has developed for implementing polygon intersections.

By Using a linked list data structure to search and add to cut the polygon dynamically, the algorithm is very efficient referring to the discussion in details in [3]:

B.2 Tests

Polygon intersection samples:





APPENDIX C

OUR MICRO-LAYER MODEL

The discretization formula in (3.52) is an inconsistent discretization formula for approximating the normal gradient of the temperature when the closest point on the vapor/liquid interface is near the substrate. This is because the temperature right at the junction where vapor, liquid, and solid meet is simultaneously prescribed to be the saturation temperature on the vapor/liquid interface, and the wall temperature on the fluid/substrate interface. The limiting value for the discretization formula (3.52) is infinite as the mesh size Δx approaches zero.

Using (C.1) as a guide, we hypothesize that the second derivatives of the temperature function satisfy the following asymptotic relation in the limit as the distance ξ to the triple point approaches zero:

$$T_{\xi\xi} = O(1/\xi^2) \quad \xi \rightarrow 0.$$

This means that (3.52) is a consistent approximation as long as the absolute distance to the substrate satisfies,

$$|\phi_{\text{substrate}}(\vec{x})| > O(\Delta x^{1/2}).$$

In our algorithm, if the substrate level set function at the closest point on the interface, \vec{x}_{closest} (see (3.51)), satisfies (see Figure C.1),

$$\phi_{\text{substrate}}(\vec{x}_{\text{closest}}) > -h_{mac},$$

then the following microcell model is used to replace (3.52).

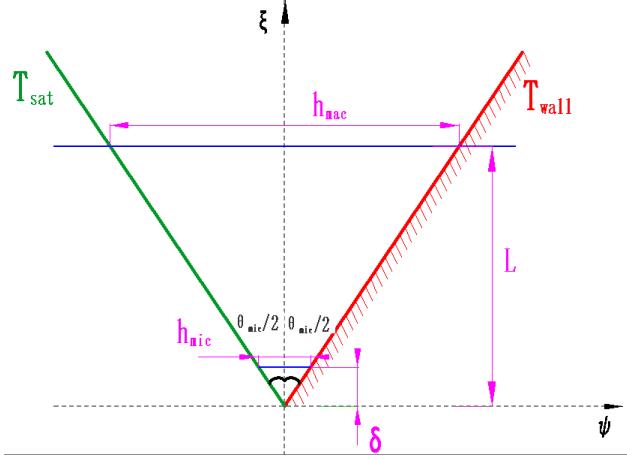
Temperature profile formulation:

$$\begin{aligned}\frac{(h_{mac}/2)}{L} &= \tan(\theta_{mic}/2) \\ m &= \tan(\theta_{mic}/2) \\ \psi_{wall} &= h(\xi) = m\xi \\ \psi_{Interface} &= -h(\xi) = -m\xi \\ T(\psi, \xi) &= \left(\frac{T_s + T_w}{2}\right) \frac{\psi + h(\xi)}{0 + h(\xi)} - \frac{T_s \psi}{h(\xi)}\end{aligned}$$

Set $\tilde{T} = \frac{T_s + T_w}{2}$, rewrite $T(\psi, \xi)$

$$T(\psi, \xi) = \tilde{T} + \frac{(\tilde{T} - T_s)\psi}{h(\xi)}$$

Consider $\psi = -h(\xi) = -m\xi$ on the gas-liquid interface:



$$\begin{aligned}T_\psi &= \frac{\tilde{T} - T_s}{h(\xi)} \\ T_\xi &= \frac{-h'}{h^2} (\tilde{T} - T_s) \psi \\ \text{LS: } \phi &= \psi + m\xi \\ \mathbf{n} &= \frac{\nabla \phi}{|\nabla \phi|} = \begin{bmatrix} 1 \\ m \end{bmatrix} \frac{1}{\sqrt{1+m^2}}\end{aligned}\tag{C.1}$$

So for gradient side from the liquid:

$$\begin{aligned}\nabla T \cdot \mathbf{n} &= \begin{bmatrix} T_\psi \\ T_\xi \end{bmatrix} \cdot \mathbf{n} = \begin{bmatrix} \frac{\tilde{T} - T_s}{h(\xi)} \\ \frac{-h'}{h^2} (\tilde{T} - T_s) \psi \end{bmatrix} \cdot \begin{bmatrix} 1 \\ m \end{bmatrix} \frac{1}{\sqrt{1+m^2}} \\ &= \frac{\tilde{T} - T_s}{h} \left[\sqrt{1+m^2} \right]\end{aligned}$$

Since:

$$\begin{aligned}\frac{1}{L-\delta} \int_\delta^L \frac{1}{h(\xi)} d\xi &= \frac{1}{L-\delta} \frac{1}{m} \ln \xi|_\delta^L = \frac{\ln \frac{L}{\delta}}{m(L-\delta)} \\ \text{where } 2\delta m &= h_{mic}\end{aligned}$$

The average flux can be calculated as:

$$\frac{\tilde{T} - T_s}{m(L-\delta)} \left(\ln \frac{L}{\delta} \right) \sqrt{1+m^2} = \frac{T_w - T_s}{h_{mac} - h_{mic}} \left(\ln \frac{h_{mac}}{h_{mic}} \right) \sqrt{1+m^2}$$

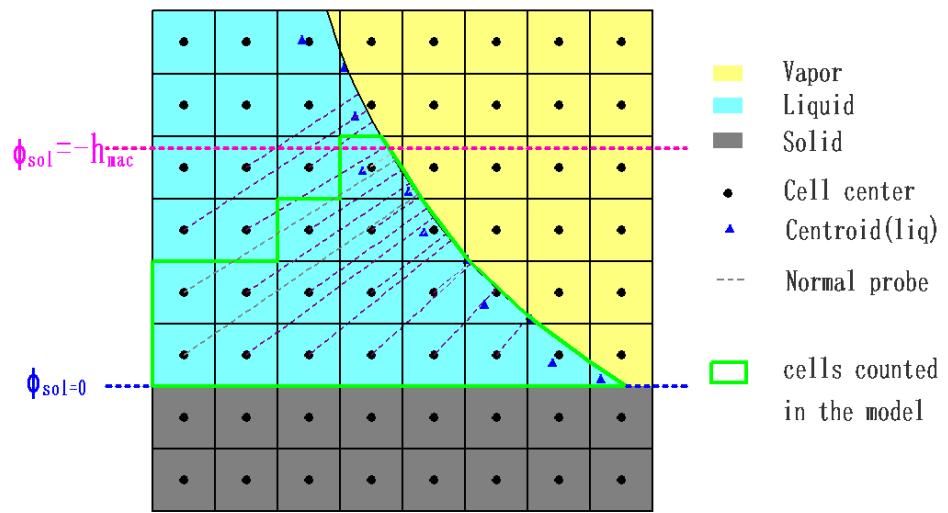


Figure C.1: The microcell treatment is used whenever the solid distance function satisfies $\phi_{\text{substrate}}(\vec{x}_{\text{closest}}) > -h_{\text{mac}}$ where h_{mac} is a macroscale length that is independent of the mesh size Δx . \vec{x}_{closest} is the closest point on the vapor liquid interface to \vec{x} .

APPENDIX D

ALGORITHM 3

Algorithm 3 For finding area fractions (length fractions in 2D) of interfaces

- 1: **Input:** MOFdata (information resulting from MOF reconstruction) including:
 - a. M , maximum number of materials.
 - b. m^i , $i = 1, 2, \dots, M$, material ID number in the order of reconstruction. (e.g. in Figure 3.7, $m^1 = 3$ represents material 3 which is the first reconstructed material.)
 - c. Γ_{m^i} , reconstructed interfaces, $i = 1, 2, \dots, M - 1$ (Note: interface Γ_{m^i} separates material m^i from m^j , $i < j < M$)
 - d. \mathbf{n}^{m_i} $i = 1, 2, \dots, M - 1$ is the normal that points towards Ω_{m_i}
 - e. α^{m_i} , intercept, $i = 1, 2, \dots, M - 1$
 - f. f_{m^i} , volume fraction, $i = 1, 2, \dots, M$
 - g. \mathbf{x}_{m^i} , material centroid, $i = 1, 2, \dots, M$
 - 2: **Output:** A_j^i , area fractions of material j along material i interface. ($i = 1, 2, \dots, M - 1$, $i < j < M$)
 - 3:
 - 4: **for** $i = 1$ to $M - 1$ **do**
 - 5: Perturb Γ_{m^i} in its normal direction $-\mathbf{n}^{m_i}$ (away from m^i 's centroid) by δ . ie. We shift the reconstructed linear interface $\mathbf{n}^{m_i} \cdot (\mathbf{x} - \mathbf{x}_0) + \alpha^{m_i}$ to $\mathbf{n}^{m_i} \cdot (\mathbf{x} - \mathbf{x}_0 + \delta \mathbf{n}^{m_i}) + \alpha^{m_i}$.
 - 6: ▷ Note: for this step, the perturbed interface could only have intersections with materials m_j which are captured later than it, ie. all j where $i < j < M$
 - 7:
 - 8: **for** $j = i + 1$ to M **do**
 - 9:
$$A_{m_j}^i = \frac{|(V_{m_j}(\mathbf{n}^{m_i}, \alpha^{m_i} + \delta) - V_{m_j}(\mathbf{n}^{m_i}, \alpha^{m_i}))|}{\sum_m V_m}$$
 - 10: ▷ Find the area (length in 2D) of each subsection by computing the difference of subsection volume (area in 2D) and divide by the total slab volume.
 - 11: **end for**
 - 12: **end for**
-

BIBLIOGRAPHY

- [1] Abba A Abubakar, Abul Fazal M Arif, Syed Sohail Akhtar, and Javad Mostaghimi. Splat formation, interaction and residual stress evolution in thermal spray coating using a hybrid computational model. *Journal of Thermal Spray Technology*, 28(3):359–377, 2019.
- [2] S Afkhami, Stephane Zaleski, and M Bussmann. A mesh-dependent model for applying dynamic contact angles to vof simulations. *Journal of Computational Physics*, 228(15):5370–5389, 2009.
- [3] Hyung Taek Ahn and Mikhail Shashkov. Multi-material interface reconstruction on generalized polyhedral meshes. *Journal of Computational Physics*, 226(2):2096–2132, 2007.
- [4] Frédéric Alauzet and Adrien Loseille. A decade of progress on anisotropic mesh adaptation for computational fluid dynamics. *Computer-Aided Design*, 72:13–39, 2016.
- [5] A Andriotis, M Gavaises, and C Arcoumanis. Vortex flow and cavitation in diesel injector nozzles. *Journal of Fluid Mechanics*, 610:195–215, 2008.
- [6] C Arcoumanis, M Gavaises, H Flora, and H Roth. Visualisation of cavitation in diesel engine injectors. *Mécanique & industries*, 2(5):375–381, 2001.
- [7] Marco Arienti and Mark Sussman. A numerical study of the thermal transient in high-pressure diesel injection. *International Journal of Multiphase Flow*, 88:205–221, 2017.
- [8] JW Bao, JM Wilczak, JK Choi, and LH Kantha. Numerical simulations of air-sea interaction under high wind conditions using a coupled model: A study of hurricane development. *Monthly Weather Review*, 128(7):2190–2210, 2000.
- [9] Edip Can and Andrea Prosperetti. A level set method for vapor bubble dynamics. *Journal of computational physics*, 231(4):1533–1552, 2012.
- [10] Victor K Champagne et al. *The cold spray materials deposition process*. Elsevier, 2007.
- [11] S Chen, B Merriman, S Osher, and P Smereka. A simple level set method for solving stefan problems. *Journal of Computational Physics*, 135(1):8–29, 1997.
- [12] Vincent Le Chenadec and Heinz Pitsch. A 3d unsplit forward/backward volume-of-fluid approach and coupling to the level set method. *Journal of Computational Physics*, 233(0):10 – 33, 2013.
- [13] Jack Chessa, Patrick Smolinski, and Ted Belytschko. The extended finite element method (xfem) for solidification problems. *International Journal for Numerical Methods in Engineering*, 53(8):1959–1977, 2002.

- [14] Richard L Christiansen and E Dendy Sloan Jr. Mechanisms and kinetics of hydrate formation. *Annals of the New York Academy of Sciences*, 715(1):283–305, 1994.
- [15] MG Cooper and AJP Lloyd. The microlayer in nucleate pool boiling. *International Journal of Heat and Mass Transfer*, 12(8):895–913, 1969.
- [16] William W Dai and Anthony J Scannapieco. Second-order accurate interface-and discontinuity-aware diffusion solvers in two and three dimensions. *Journal of Computational Physics*, 281:982–1002, 2015.
- [17] Shyama Prasad Das, Frédéric Lefèvre, Jocelyn Bonjour, and Sameer Khandekar. b, parametric study of a two-phase oscillating flow in a capillary tube,. In *Proceedings of 15th International Heat Pipe Conference*, 2010.
- [18] AS Dawes. Three-dimensional multi-material polyhedral method for diffusion. *Computers & Fluids*, 156:485–495, 2017.
- [19] Steven Diot and Marianne M François. An interface reconstruction method based on an analytical formula for 3d arbitrary convex cells. *Journal of Computational Physics*, 305:63–74, 2016.
- [20] Vadim Dyadechko and Mikhail Shashkov. Moment-of-fluid interface reconstruction. *Los Alamos report LA-UR-05-7571*, 2005.
- [21] Vadim Dyadechko and Mikhail Shashkov. Reconstruction of multi-material interfaces from moment data. *Journal of Computational Physics*, 227(11):5361–5384, 2008.
- [22] Amir Faghri. Review and advances in heat pipe science and technology. *Journal of heat transfer*, 134(12), 2012.
- [23] Zhiqiang Fan and Frank Liou. Numerical modeling of the additive manufacturing (am) processes of titanium alloy. *Titanium alloys—towards achieving enhanced properties for diversified applications*, pages 3–28, 2012.
- [24] Marianne M Francois, Sharen J Cummins, Edward D Dendy, Douglas B Kothe, James M Sicilian, and Matthew W Williams. A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. *Journal of Computational Physics*, 213(1):141–173, 2006.
- [25] Frederick Charles Frank. Radially symmetric phase growth controlled by diffusion. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 201(1067):586–599, 1950.
- [26] M Fulgosi, D Lakehal, S Banerjee, and V De Angelis. Direct numerical simulation of turbulence in a sheared air–water flow with a deformable interface. *Journal of fluid mechanics*, 482:319–345, 2003.

- [27] RV Garimella and Konstatin Lipnikov. Solution of the diffusion equation in multi-material domains by sub-division of elements along reconstructed interfaces. *International Journal for Numerical Methods in Fluids*, 65(11-12):1423–1437, 2011.
- [28] Frédéric Gibou, Liguo Chen, Duc Nguyen, and Sanjoy Banerjee. A level set based sharp interface method for the multiphase incompressible navier–stokes equations with phase change. *Journal of Computational Physics*, 222(2):536–555, 2007.
- [29] Frédéric Gibou and Ronald Fedkiw. A fourth order accurate discretization for the laplace and heat equations on arbitrary domains, with applications to the stefan problem. *Journal of Computational Physics*, 202(2):577–601, 2005.
- [30] Frédéric Gibou, Ronald Fedkiw, Russel Caflisch, and Stanley Osher. A level set approach for the numerical simulation of dendritic growth. *Journal of Scientific Computing*, 19(1-3):183–199, 2003.
- [31] Frederic Gibou, Ronald P Fedkiw, Li-Tien Cheng, and Myungjoo Kang. A second-order-accurate symmetric discretization of the poisson equation on irregular domains. *Journal of Computational Physics*, 176(1):205–227, 2002.
- [32] Chris Gordon, Claire Cooper, Catherine A Senior, Helene Banks, Jonathan M Gregory, Timothy C Johns, John FB Mitchell, and Richard A Wood. The simulation of sst, sea ice extents and ocean heat transports in a version of the hadley centre coupled model without flux adjustments. *Climate dynamics*, 16(2-3):147–168, 2000.
- [33] Mikhael Gorokhovski and Marcus Herrmann. Modeling primary atomization. *Annu. Rev. Fluid Mech.*, 40:343–366, 2008.
- [34] Arthur Guittet, Mathieu Lepilliez, Sebastien Tanguy, and Frédéric Gibou. Solving elliptic problems with discontinuities on irregular domains—the voronoi interface method. *Journal of Computational Physics*, 298:747–765, 2015.
- [35] Susann Hänsch and Simon Walker. Microlayer formation and depletion beneath growing steam bubbles. *International Journal of Multiphase Flow*, 111:241–263, 2019.
- [36] C Höhmann and P Stephan. Microscale temperature measurement at an evaporating liquid meniscus. *Experimental Thermal and Fluid Science*, 26(2-4):157–162, 2002.
- [37] Matthew Jemison, Mark Sussman, and Mikhail Shashkov. Filament capturing with the multimaterial moment-of-fluid method. *Journal of Computational Physics*, 285:149–172, 2015.
- [38] Hans Johansen and Phillip Colella. A cartesian grid embedded boundary method for poisson’s equation on irregular domains. *Journal of Computational Physics*, 147(1):60–85, 1998.

- [39] Satbyoul Jung and Hyungdae Kim. An experimental method to simultaneously measure the dynamics and heat transfer associated with a single bubble during nucleate boiling on a horizontal surface. *International Journal of Heat and Mass Transfer*, 73:365–375, 2014.
- [40] Damir Juric and Grétar Tryggvason. A front-tracking method for dendritic solidification. *Journal of computational physics*, 123(1):127–148, 1996.
- [41] JWJ Kaiser, S Adami, IS Akhatov, and NA Adams. A semi-implicit conservative sharp-interface method for liquid-solid phase transition. *International Journal of Heat and Mass Transfer*, 155:119800, 2020.
- [42] Mehdi Khalloufi, Youssef Mesri, Rudy Valette, Elisabeth Massoni, and Elie Hachem. High fidelity anisotropic adaptive variational multiscale method for multiphase flows with surface tension. *Computer Methods in Applied Mechanics and Engineering*, 307:44–67, 2016.
- [43] Mehdi Khalloufi, Rudy Valette, and Elie Hachem. Adaptive eulerian framework for boiling and evaporation. *Journal of Computational Physics*, 401:109030, 2020.
- [44] Evgeny Kikinzon, Yuri Kuznetsov, Konstantin Lipnikov, and Mikhail Shashkov. Approximate static condensation algorithm for solving multi-material diffusion problems on meshes non-aligned with material interfaces. *Journal of Computational Physics*, 347:416–436, 2017.
- [45] PA Kobryn, NR Ontko, LP Perkins, and JS Tiley. Additive manufacturing of aerospace alloys for aircraft structures. Technical report, Air Force Research Lab Wright-Patterson AFB OH Materials and Manufacturing , 2006.
- [46] Eunmo Koo, Patrick J Pagni, David R Weise, and John P Woycheese. Firebrands and spotting ignition in large-scale fires. *International Journal of Wildland Fire*, 19(7):818–843, 2010.
- [47] Jean-Pierre Kruth, Peter Mercelis, J Van Vaerenbergh, Ludo Froyen, and Marleen Rombouts. Binding mechanisms in selective laser sintering and selective laser melting. *Rapid prototyping journal*, 11(1):26–36, 2005.
- [48] Akshay Kundan, Thao TT Nguyen, Joel L Plawsky, Peter C Wayner Jr, David F Chao, and Ronald J Sicker. Arresting the phenomenon of heater flooding in a wickless heat pipe in microgravity. *International Journal of Multiphase Flow*, 82:65–73, 2016.
- [49] Akshay Kundan, Joel L Plawsky, and Peter C Wayner Jr. Effect of capillary and marangoni forces on transport phenomena in microgravity. *Langmuir*, 31(19):5377–5386, 2015.
- [50] Djamel Lakehal. Highly-resolved les of turbulent convective flow along a pwr rod bundle. *International Journal of Heat and Mass Transfer*, 122:785–794, 2018.
- [51] Djamel Lakehal. Status and future developments of large-eddy simulation of turbulent multi-fluid flows (leis and less). *International Journal of Multiphase Flow*, 104:322–337, 2018.

- [52] Minjae Lee, David Hyde, Michael Bao, and Ronald Fedkiw. A skinned tetrahedral mesh for hair animation and hair-water interaction. *IEEE transactions on visualization and computer graphics*, 25(3):1449–1459, 2018.
- [53] Minjae Lee, David Hyde, Kevin Li, and Ronald Fedkiw. A robust volume conserving method for character-water interaction. *arXiv preprint arXiv:1902.00801*, 2019.
- [54] Antoine Lemoine, Stéphane Glockner, and Jérôme Breil. Moment-of-fluid analytic reconstruction on 2d cartesian grids. *Journal of Computational Physics*, 328:131–139, 2017.
- [55] Randall J Leveque and Zhilin Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis*, 31(4):1019–1044, 1994.
- [56] Guibo Li, Yongsheng Lian, Yisen Guo, Matthew Jemison, Mark Sussman, Trevor Helms, and Marco Arienti. Incompressible multiphase flow and encapsulation simulations using the moment-of-fluid method. *International Journal for Numerical Methods in Fluids*, 79(9):456–490, 2015.
- [57] Xu-Dong Liu, Ronald P Fedkiw, and Myungjoo Kang. A boundary condition capturing method for poisson’s equation on irregular domains. *Journal of computational Physics*, 160(1):151–178, 2000.
- [58] Yang Liu, Mark Sussman, Yongsheng Lian, and M Yousuff Hussaini. A moment-of-fluid method for diffusion equations on irregular domains in multi-material systems. *Journal of Computational Physics*, 402:109017, 2020.
- [59] Zheng Yuan Luo, Xing Long Shang, and Bo Feng Bai. Influence of pressure-dependent surface viscosity on dynamics of surfactant-laden drops in shear flow. *Journal of Fluid Mechanics*, 858:91–121, 2019.
- [60] Bennert Machenhauer and Markus Olk. The implementation of the semi-implicit scheme in cell-integrated semi-lagrangian models. *Atmosphere-Ocean*, 35(sup1):103–126, 1997.
- [61] Mikko Manninen, Veikko Taivassalo, Sirpa Kallio, et al. On the mixture model for multiphase flow, 1996.
- [62] Mulugeta Markos, Vladimir S Ajaev, and GM Homsy. Steady flow and evaporation of a volatile liquid in a wedge. *Physics of Fluids*, 18(9):092102, 2006.
- [63] David W Martin, Tamonotubo George, and François Blanchette. Simulations of clean drops rising into a layer of dissolved surfactant. *Physical Review Fluids*, 4(1):014302, 2019.
- [64] Masakazu Matsumoto, Shinji Saito, and Iwao Ohmine. Molecular dynamics simulation of the ice nucleation and growth process leading to water freezing. *Nature*, 416(6879):409, 2002.
- [65] Yu F Maydanik. Loop heat pipes. *Applied thermal engineering*, 25(5-6):635–657, 2005.

- [66] Anvarbek Mukatovich Meirmanov. The Stefan Problem, volume 3. Walter de Gruyter, 2011.
- [67] Thomas Milcent and Antoine Lemoine. Moment-of-fluid analytic reconstruction on 3d rectangular hexahedrons. Journal of Computational Physics, 409:109346, 2020.
- [68] Rajat Mittal, Rui Ni, and Jung-Hee Seo. The flow physics of covid-19. Journal of fluid Mechanics, 894, 2020.
- [69] Florian Moreau, Pierre Colinet, and Stéphane Dorbolo. Explosive leidenfrost droplets. Physical Review Fluids, 4(1):013602, 2019.
- [70] William W Mullins and RF Sekerka. Stability of a planar interface during solidification of a dilute binary alloy. Journal of applied physics, 35(2):444–451, 1964.
- [71] William W Mullins and Robert F Sekerka. Morphological stability of a particle growing by diffusion or heat flow. Journal of applied physics, 34(2):323–329, 1963.
- [72] Ramachandran D Nair and Bennert Machenhauer. The mass-conservative cell-integrated semi-lagrangian advection scheme on the sphere. Monthly Weather Review, 130(3):649–667, 2002.
- [73] RR Olander and RG Watts. An analytical expression of microlayer thickness in nucleate boiling. 1969.
- [74] Sashidhar S Panchamgam, Arya Chatterjee, Joel L Plawsky, and Peter C Wayner Jr. Comprehensive experimental and theoretical study of fluid flow and heat transfer in a microscopic evaporating meniscus in a miniature heat exchanger. International Journal of Heat and Mass Transfer, 51(21–22):5368–5379, 2008.
- [75] Joseph Papac, Asdis Helgadottir, Christian Ratsch, and Frederic Gibou. A level set approach for diffusion and stefan-type problems with robin boundary conditions on quadtree/octree adaptive cartesian grids. Journal of Computational Physics, 233:241–261, 2013.
- [76] Lech Pawłowski. The science and engineering of thermal spray coatings. John Wiley & Sons, 2008.
- [77] Chaoxu Pei, Mark Sussman, and M Yousuff Hussaini. A hierarchical block structured space-time spectral element method for simulating complex multiphase flows.
- [78] James Edward Pilliod Jr and Elbridge Gerry Puckett. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. Journal of Computational Physics, 199(2):465–502, 2004.
- [79] Jan Prüss, Jurgen Saal, and Gieri Simonett. Existence of analytic solutions for the classical stefan problem. Mathematische Annalen, 338(3):703–756, 2007.

- [80] Fangfang Qin, Jinru Chen, Zhilin Li, and Mingchao Cai. A cartesian grid nonconforming immersed finite element method for planar elasticity interface problems. *Computers & Mathematics with Applications*, 73(3):404–418, 2017.
- [81] Theron Rodgers, Jonathan D Madison, John A Mitchell, and Veena Tikare. Numerical simulation of microstructural evolution during additive manufacturing of metals. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2017.
- [82] H Roth, M Gavaises, and C Arcoumanis. Cavitation initiation, its development and link with flow turbulence in diesel injector nozzles. *SAE Transactions*, pages 561–580, 2002.
- [83] Yohei Sato and Bojan Ničeno. A sharp-interface phase change model for a mass-conservative interface tracking method. *Journal of Computational Physics*, 249:127–161, 2013.
- [84] Yohei Sato and Bojan Niceno. A depletable micro-layer model for nucleate pool boiling. *Journal of Computational physics*, 300:20–52, 2015.
- [85] Eberhard J. Sauter, Sergey I. Muyakshin, Jean-Luc Charlou, Michael Schlter, Antje Boetius, Kerstin Jerosch, Ellen Damm, Jean-Paul Foucher, and Michael Klages. Methane discharge from a deep-sea submarine mud volcano into the upper water column by gas hydrate-coated methane bubbles. *Earth and Planetary Science Letters*, 243(3):354 – 365, 2006.
- [86] R Savino and D Paterna. Marangoni effect and heat pipe dry-out. *Physics of Fluids*, 18(11):118103, 2006.
- [87] S Srinivas Shastri and RM Allen. Method of lines and enthalpy method for solving moving boundary problems. *International communications in heat and mass transfer*, 25(4):531–540, 1998.
- [88] G Son, VK Dhir, and N Ramanujapu. Dynamics and heat transfer associated with a single bubble during nucleate boiling on a horizontal surface. *Journal of Heat Transfer*, 121(3):623–631, 1999.
- [89] PC Stephan and CA Busse. Analysis of the heat transfer coefficient of grooved heat pipe evaporator walls. *International Journal of heat and mass transfer*, 35(2):383–391, 1992.
- [90] P.A. Stewart, N. Lay, M. Sussman, and M. Ohta. An improved sharp interface method for viscoelastic and viscous two-phase flows. *Journal of Scientific Computing*, 35(1):43–61, 2008.
- [91] T Stoltenhoff, H Kreye, and HJ Richter. An analysis of the cold spray process and its coatings. *Journal of Thermal spray technology*, 11(4):542–550, 2002.
- [92] M Sussman and M Ohta. High-order techniques for calculating surface tension forces. In *Free Boundary Problems, from International Series of Numerical Mathematics*, volume 154, pages 425–434. Birkhauser Verlag Basel/Switzerland, 2006.

- [93] M. Sussman and M. Ohta. Improvements for calculating two-phase bubble and drop motion using an adaptive sharp interface method. *Fluid Dynamics and Materials Processing*, 3(1):21–36, 2007.
- [94] M. Sussman and M. Ohta. A stable and efficient method for treating surface tension in incompressible two-phase flow. *SIAM J. Sci. Comput.*, 31(4):2447–2471, 2009.
- [95] Mark Sussman and Elbridge Gerry Puckett. A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows. *Journal of Computational Physics*, 162(2):301–337, 2000.
- [96] Sébastien Tanguy, Thibaut Ménard, and Alain Berlemont. A level set method for vaporizing two-phase flows. *Journal of Computational Physics*, 221(2):837–853, 2007.
- [97] Sébastien Tanguy, Michaël Sagan, Benjamin Lalanne, Frédéric Couderc, and Catherine Colin. Benchmarks and numerical methods for the simulation of boiling flows. *Journal of Computational Physics*, 264:1–22, 2014.
- [98] Uwe Thiele, Jacco H Snoeijer, Sarah Trinschek, and Karin John. Equilibrium contact angle and adsorption layer properties with surfactants. *Langmuir*, 34(24):7210–7221, 2018.
- [99] Yoshio Utaka, Yuki Kashiwabara, and Michio Ozaki. Microlayer structure in nucleate boiling of water and ethanol at atmospheric pressure. *International Journal of Heat and Mass Transfer*, 57(1):222–230, 2013.
- [100] Henk A Van der Vorst. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631–644, 1992.
- [101] SJD Van Stralen, MS Sohal, R Cole, and WM Sluyter. Bubble growth rates in pure and binary systems: combined effect of relaxation and evaporation microlayers. *International Journal of Heat and Mass Transfer*, 18(3):453–467, 1975.
- [102] Lucia Rueda Villegas, Romain Alis, Mathieu Lepilliez, and Sébastien Tanguy. A ghost fluid/level set method for boiling flows and liquid evaporation: application to the leidenfrost effect. *Journal of Computational Physics*, 316:789–813, 2016.
- [103] Vaughan R Voller. An enthalpy method for modeling dendritic growth in a binary alloy. *International Journal of Heat and Mass Transfer*, 51(3-4):823–834, 2008.
- [104] Vaughan R Voller, M Cross, and NC Markatos. An enthalpy method for convection/diffusion phase change. *International journal for numerical methods in engineering*, 24(1):271–284, 1987.
- [105] Cornelis Vuik. Some historical notes about the stefan problem. 1993.

- [106] Guoyu Wang, Inanc Senocak, Wei Shyy, Toshiaki Ikohagi, and Shuliang Cao. Dynamics of attached turbulent cavitating flows. *Progress in Aerospace sciences*, 37(6):551–581, 2001.
- [107] Tianbao Wang, Liqiang Ai, Yusi Zhou, and Min Chen. Numerical simulation of supercooled water droplets impacting ice with rapid crystal growth taken into consideration. *Langmuir*, 2020.
- [108] Gopinath R. Warrier, Vijay K. Dhir, and David F. Chao. Nucleate pool boiling experiment (npbx) in microgravity: International space station. *International Journal of Heat and Mass Transfer*, 83(Supplement C):781 – 798, 2015.
- [109] Robert P Warzinski, Ronald Lynn, Igor Haljasmaa, Ira Leifer, Frank Shaffer, Brian J Anderson, and Jonathan S Levine. Dynamic morphology of gas hydrate on a methane bubble in water: Observations and new insights for hydrate film models. *Geophysical Research Letters*, 41(19):6841–6847, 2014.
- [110] PC Wayner Jr, YK Kao, and LV LaCroix. The interline heat-transfer coefficient of an evaporating wetting film. *International Journal of Heat and Mass Transfer*, 19(5):487–492, 1976.
- [111] GD Weymouth and Dick K-P Yue. Conservative volume-of-fluid method for free-surface simulations on cartesian-grids. *Journal of Computational Physics*, 229(8):2853–2865, 2010.
- [112] Jane Wu, Zhenglin Geng, Hui Zhou, and Ronald Fedkiw. Skinning a parameterization of three-dimensional space for neural network cloth. *arXiv preprint arXiv:2006.04874*, 2020.
- [113] Qingrong Xiong, Joseph D. Robson, Litao Chang, Jonathan W. Fellowes, and Mike C. Smith. Numerical simulation of grain boundary carbides evolution in 316h stainless steel. *Journal of Nuclear Materials*, 508:299 – 309, 2018.
- [114] Li Yang and GM Homsy. Steady three-dimensional thermocapillary flows and dryout inside a v-shaped wedge. *Physics of Fluids*, 18(4):042107, 2006.
- [115] Miad Yazdani, Marios C Soteriou, Fanping Sun, and Zaffir Chaudhry. Prediction of the thermo-fluids of gearbox systems. *International Journal of Heat and Mass Transfer*, 81:337–346, 2015.
- [116] Lang Yuan. Solidification defects in additive manufactured materials. *JOM*, 71(9):3221–3222, 2019.
- [117] Alexander Zhiliakov, Daniil Svyatskiy, Maxim Olshanskii, Evgeny Kikinzon, and Mikhail Shashkov. A higher order approximate static condensation method for multi-material diffusion problems. *Journal of Computational Physics*, 2019.
- [118] Bo Zhu, Ed Quigley, Matthew Cong, Justin Solomon, and Ronald Fedkiw. Codimensional surface tension flow on simplicial complexes. *ACM Transactions on Graphics (TOG)*, 33(4):111, 2014.
- [119] An Zou, Ashish Chanana, Amit Agrawal, Peter C Wayner Jr, and Shalabh C Maroo. Steady state vapor bubble in pool boiling. *Scientific reports*, 6:20240, 2016.

BIOGRAPHICAL SKETCH

The author was born in Shenyang, China. The author studied in Dalian University of Technology, Dalian, China from year 2004 to 2012, and successively obtained a bachelor of science degree of Applied Mathematics, a duo degree on English, and a master degree on Fluid Machinery. The author was admitted into the Florida state University in the year 2013.