Journal of Scientific Computing manuscript No.
(will be inserted by the editor)

A numerical method for computing solutions to the multi-material Stefan problem on a complex deforming supermesh

Yang Liu · Mark Sussman · Yongsheng Lian · M. Yousuff Hussaini · Mehdi Vahab · Kourosh Shoele

the date of receipt and acceptance should be inserted later

Abstract A novel supermesh method that was developed for computing solutions to the multimaterial heat equation in complex stationary geometries will be applied for computing solutions to the Stefan problem involving complex deforming geometries. The supermesh is established by combining the structured rectangular grid and the piecewise linear interfaces reconstructed by the moment-of-fluid method. The temperature diffusion equation with Dirichlet boundary at the interfaces is solved by our linear exact multi-material finite volume method upon the supermesh. The interface propagation equation is resolved by 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. Our 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 opens the door for solving many complex deforming boundary problems in which the method has the efficiency prop-

Y. Liu
Department of Mathematics, Florida State University, Tallahassee, FL, 32306, USA.
E-mail: whliu13@gmail.com

M. Sussman
Department of Mathematics, Florida State University, Tallahassee, FL, 32306, USA.
E-mail: sussman@math.fsu.edu

Yongsheng Lian
Department of Mechanical Engineering , University of Louisville, Louisville, KY 40292, USA.
E-mail: yongsheng.lian@louisville.edu

M. Yousuff Hussaini
Department of Mathematics, Florida State University, Tallahassee, FL, 32306, USA.
E-mail: myh@math.fsu.edu

Mehdi Vahab
Department of Mathematics, Florida State University, Tallahassee, FL, 32306, USA.
E-mail: mvahab@fsu.edu

Kourosh Shoele
Department of Mathematics, Florida State University, Tallahassee, FL, 32306, USA.
E-mail: kshoele@fsu.edu

erties of a body fitted mesh combined with the robustness of a "cut-cell" (a.k.a. "embedded boundary" or "immersed") method.

Keywords Moment-of-fluid · Multi-phase flow · Multi-material flow · Stefan problem

1 Introduction

For representing phase change phenomenon, the Stefan problem [41,9,26,25,24] 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 problems were introduced by the Slovenian physicist, Josef Stefan around 1890 for modeling problems of ice formation [41]. We refer the reader to Pruss etal.(2007) [31] 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, [9,26,25,24]). 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 of interface is done. For example, the classical enthalpy method. Voller et al. [40,39,36] 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 [4,10,29] or explicitly [15] 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 [15] 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 the front-tracking technique. The method converges well under grid refinement. Note: no examples were reported in Juric& Tryggvason in which the interface changed topology. Chen et al. [4] developed a level set method for solving the 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 function. Gibou and Fedkiw [10] developed a third-order accurate level-set method for solving the Stefan problem on irregular deforming domains. Papac et al.[29] 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.

Our 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. Our supermesh-FVM-MOF approach in the previous paper [20], 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 problem 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 [4,10,15]. We predict that our supermesh-FVM-MOF method will be applicable to more complicated applications (see [44,32,12,29,10,4,37,43,34,46,3,17,1,2,33,42,38,23,13,45,22,35,6,8,28,19, 18]) as well.

Starting with the FVM-MOF diffusion solver that we have developed [20], we next solve the multi-material 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, a feature unique to our supermesh FVM-MOF approach. We apply our Finite Volume Moment-of-Fluid method based on supermesh to discretize the heat equation which in turn gives the front velocity (see Equation 4). The "linear exact" algorithm (see section 3.4.3) is used to enforce the Dirichlet boundary at the interface in cells in which the area fraction "Gap error" criteria (see section 3.4.4) 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 3.4.2; see Equation 18). 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.

2 Problem Formulation

For the classical two phase Stefan problem, we assume $\rho = constant$, $C_v = constant$ and that diffusion processes dominate the heat transfer $\boldsymbol{u} = 0$.

$$\frac{\partial(\rho e)}{\partial t} = \nabla \cdot (\kappa \nabla \theta(\boldsymbol{x})) \quad \boldsymbol{x} \in \Omega \tag{1}$$

$$\theta(\boldsymbol{x}) = \theta_{sat}(\boldsymbol{x}, t) \quad \boldsymbol{x} \in \Gamma \tag{2}$$

$$F_t + V_n |\nabla F| = 0 \tag{3}$$

$$V_n = \frac{[\kappa \nabla \theta \cdot \boldsymbol{n}]_\Gamma}{L} \tag{4}$$

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. In the dendrite growth test problem, the saturation temperature is a function of local interface curvature. Equation 3 is the volume fraction advection equation which tracks the moving interface location. Equation 4 lists the Stefan condition which defines the phase front normal velocity V_n.

3 Numerical method

We have developed the following algorithm for numerically approximating solutions to the Stefan problem on a deforming supermesh (Figure 1):

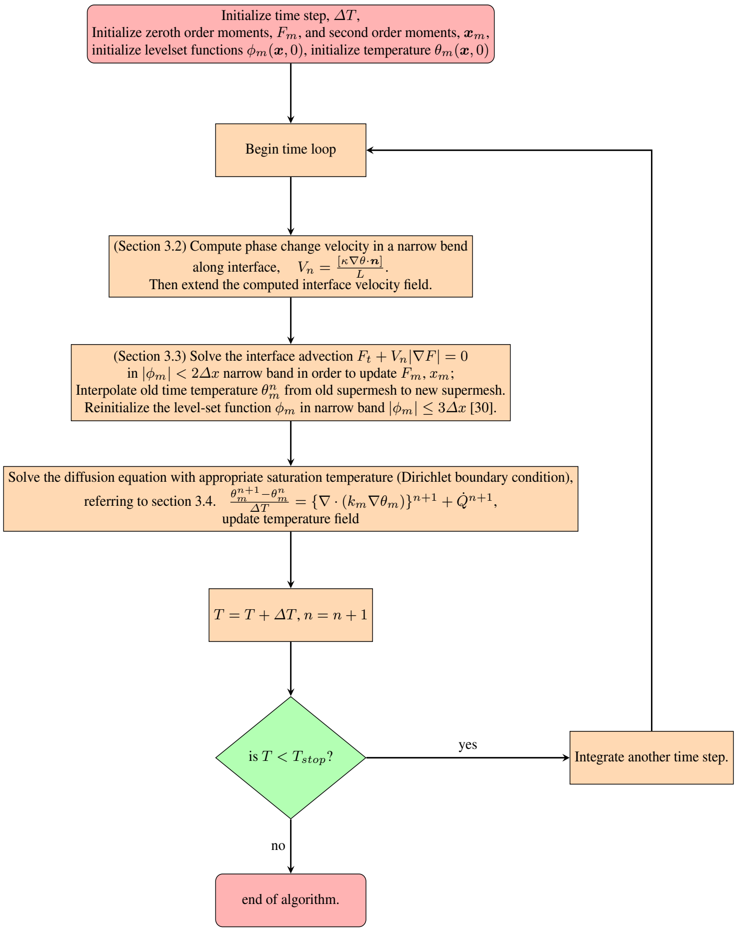
Initialize time step, $\varDelta T$,
Initialize zeroth order moments, F_m, and second order moments, $\boldsymbol{x}_m$,
initialize levelset functions $\phi_m(\boldsymbol{x},0)$, initialize temperature $\theta_m(\boldsymbol{x},0)$
Begin time loop
(Section 3.2) Compute phase change velocity in a narrow bend
along interface, $V_n = \frac{[\kappa\nabla\theta\cdot\boldsymbol{n}]}{L}$.
Then extend the computed interface velocity field.
(Section 3.3) Solve the interface advection $F_t + V_n|\nabla F| = 0$
in $|\phi_m| < 2\varDelta x$ narrow band in order to update F_m, x_m;
Interpolate old time temperature θ_m^n from old supermesh to new supermesh.
Reinitialize the level-set function ϕ_m in narrow band $|\phi_m| \leq 3\varDelta x$ [30].
Solve the diffusion equation with appropriate saturation temperature (Dirichlet boundary condition),
referring to section 3.4. $\frac{\theta_m^{n+1}-\theta_m^n}{\varDelta T} = \{\nabla\cdot(k_m\nabla\theta_m)\}^{n+1} + \dot{Q}^{n+1}$,
update temperature field
$T = T + \varDelta T, n = n + 1$
is $T < T_{stop}$?
yes
Integrate another time step.
no
end of algorithm.

Fig. 1: Flow chart corresponding to our Stefan problem algorithm.

3.1 Interface representation

The piecewise linear multi-material MOF [7] reconstruction method is applied for representing material interfaces (see Fig 2). In each computational grid cell, interfaces are approximated by straight lines (planes in 3D), see equation 5.

$$\varGamma = \{\boldsymbol{x}|\boldsymbol{n}\cdot(\boldsymbol{x}-\boldsymbol{x_0})+b=0\} \tag{5}$$

In a cell containing two materials, the normal $\boldsymbol{n}$ and intercept point b of a line are derived from a reference volume fraction F_{ref} and reference centroid $\boldsymbol{x}_{ref}$. We solve the following constrained minimization problem:

$$\begin{aligned} (\boldsymbol{n}, b) &= \operatorname{argmin}(E_{MOF}) = \underset{\boldsymbol{n},b}{\operatorname{argmin}} \|\boldsymbol{x}_{ref} - \boldsymbol{x}_{act}(\boldsymbol{n}, b)\| \\ \text{subject to } & |F_{act}(\boldsymbol{n}, b) - F_{ref}(\boldsymbol{n}, b)| = 0 \end{aligned} \tag{6}$$

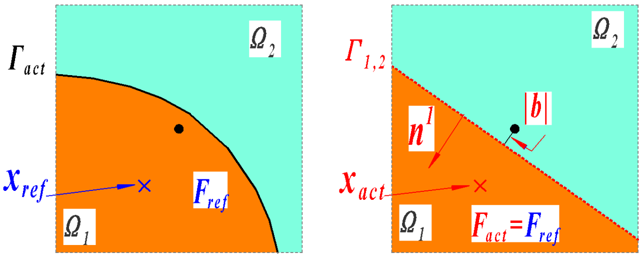

Fig. 2: Example of MOF reconstruction in two materials cell.
Remark: the normal of the reconstructed interface is always point into the target material in this paper. For example in the figure, the normal n^1 points into material 1.

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 $\epsilon_{intercept} = 10^{-12}$.

If the number of materials is more than two in one cell, we apply the nested dissection algorithm to perform the MOF reconstruction method [7], see Figure 3.

The nested dissection algorithm that we have implemented[14] is different from the classical algorithm[7]. 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 [14] that the accuracy of the two approaches for selecting material reconstruction ordering is comparable.

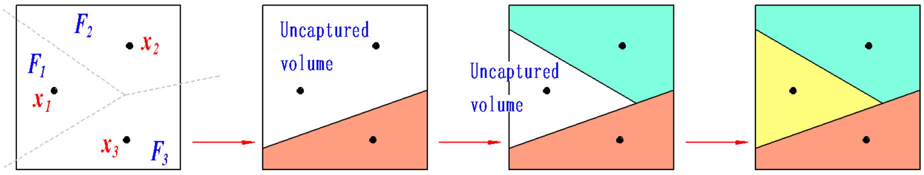

Fig. 3: Illustration of MOF reconstruction with materials more than 3 using nested dissection algorithm
Material 3 is the first material reconstructed. Material 2 is the second material reconstructed.

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.

3.2 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 4). For computing the phase front velocity, we either use (A) the standard “macro-scale” normal probe method (see Figure 4) or (B) the “micro-scale” normal probe method (see Figure 5).

3.2.1 “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 $\overrightarrow{x}_{closest}$ on the interface from the cell center (See Figure 4).

$$\overrightarrow{x}_{closest} = \overrightarrow{x}_{i,j} - \phi_{i,j,p}\overrightarrow{n}_{i,j,p} \qquad p = liq \text{ or } vap \tag{7}$$

3. From $\overrightarrow{x}_{closest}$ 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 $\overrightarrow{x}_{liq}^{probe}$, and $\overrightarrow{x}_{vap}^{probe}$.
4. Locate the target cell that $\overrightarrow{x}_{i}^{probe}$ (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^is at the centroids for each cell to $\overrightarrow{x}_{i}^{probe}$ to get T_i^{probe} (Figure 4).
5. If $|\phi_{liq}(\overrightarrow{x}_{closest})| < \Delta x$, $|\phi_{vap}(\overrightarrow{x}_{closest})| < \Delta x$, $\phi_{liq}(\overrightarrow{x}_{liq}^{probe}) > 0$, $\phi_{vap}(\overrightarrow{x}_{vap}^{probe}) > 0$ and $\phi(\overrightarrow{x}_{closest}) < \min(\phi_{vap}(\overrightarrow{x}_{closest}), \phi_{liq}(\overrightarrow{x}_{closest}))$. The temperature gradient is calculated as (Figure 4)

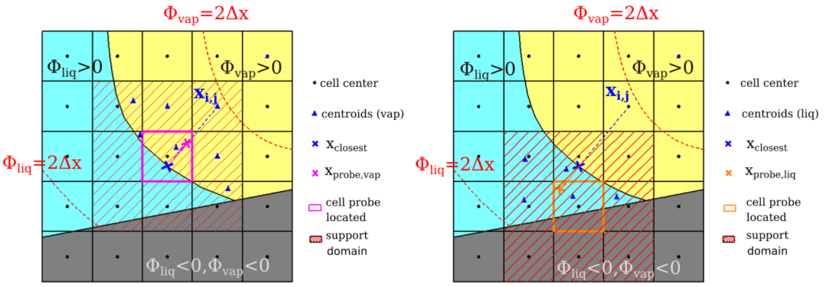

Fig. 4: Phase change velocity calculation: find corresponding interpolation support domain and calculate the temperature gradient jump across the phase change front. Left: support domain for the vapor phase; Right: support domain for the liquid phase

$$\nabla\theta_i = \frac{\theta_i^{probe} - \theta_{sat}}{\Delta x/2} \tag{8}$$

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

$$V_n^{i,j} = \frac{\dot{m}}{\rho_v} = \frac{[\kappa\nabla\theta]\cdot(\overrightarrow{n_{lv}})}{\rho_v L}.$$

3.2.2 “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 5, the normal probe $\boldsymbol{x}_{probe,2}$ falls into material 3 region instead of material 2 region. In such case, we make the following algorithm changes.

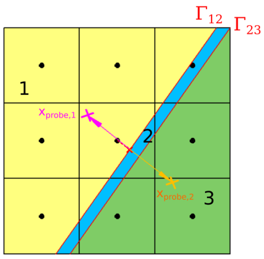

Fig. 5: Phase change velocity near filament region
Since the macro probe, $x_{probe,2}$ is in material 3, we use the micro probe (immediately next to the red x) that is in material 2.

We introduce the “micro probes”

$$\boldsymbol{x}^{\pm}_{probe} = \boldsymbol{x}_{closest} \pm \epsilon\Delta x\boldsymbol{n}, \quad \epsilon = 10^{-2}. \tag{9}$$

If $\boldsymbol{x}^m_{probe}$ is not in material “m”, then we find the cell containing $\boldsymbol{x}^m_{probe,micro}$.

- If the volume fraction (F_m) of the target material (m) in the containing cell is less than a tolerance (i.e. the target material does not exist in the containing cell), then we directly set $\theta^{probe}_m = \theta_{sat}$.
  The normal gradient is

$$(\kappa\nabla\theta\cdot\boldsymbol{n})^{\pm} = \kappa^{\pm}\frac{\theta_{sat} - \theta(\boldsymbol{x}_{closest})}{\|\boldsymbol{x}^m_{probe} - \boldsymbol{x}_{closest}\|}. \tag{10}$$

- Otherwise, the normal gradient is

$$(\kappa\nabla\theta\cdot\boldsymbol{n})^{\pm} = \kappa^{\pm}\frac{\theta^{\pm}(\boldsymbol{x}^m_{centroid}) - \theta(\boldsymbol{x}_{closest})}{\|\boldsymbol{x}^m_{centroid} - \boldsymbol{x}_{closest}\|}. \tag{11}$$

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

3.3 Solve the Advection Equation by 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 6). 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 $\overrightarrow{x}_{closest}$ 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 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 3 and update the volume fractions and centroids as phase change front moves.

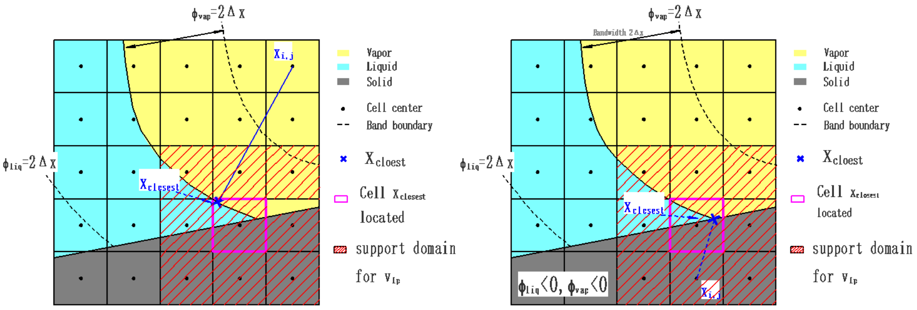

Fig. 6: Phase change velocity calculation: velocity extension in cells ($2\Delta x < \phi_{liq} < 4\Delta x$ or $2\Delta x < \phi_{vap} < 4\Delta x$) and solid cells.
Left: $x_{i,j}$ in vapor occupied cell within $4\Delta x$. Right: $x_{i,j}$ in solid cell

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 [21,27,5]. 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 7 illustrates one 2D example of un-split CISL method.

Remark: as shown in Figure 8, 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 1).

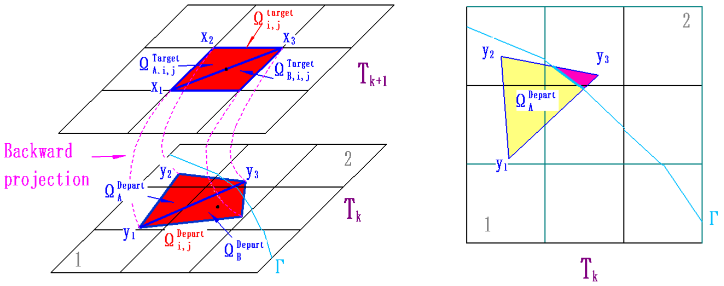

Fig. 7: Illustration of 2D un-split Cell-Integrated Semi-Lagrangian (CISL) method: Left: Backward project (characteristic tracing of the value at the four vertices according to the velocity) the center cell $\Omega^{target}_{i,j}$ at time T^{k+1} onto the domain at time T^k Right: Show the intersection of P^{depart}_1 with the T^k domain

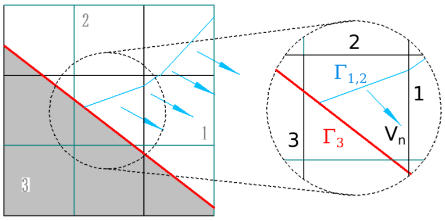

Fig. 8: Volume fraction correction after advection in > 2 material cells

Algorithm 1 Volume fraction check algorithm (referring to Figure 8).

```
Input: $F^{old}_{dest}$, $F^{old}_{src}$, $F^{new,temp}_{src}$, $F^{new,temp}_{dest}$
Output: $F^{new,corrected}_{dest}$, $F^{new,corrected}_{src}$
$\Delta F_i = F^{new,temp}_i - F^{old}_i$
$\Delta F = \min(|\Delta F_{dest}|, |\Delta F_{src}|)$
If $F^{new,temp}_{src} < 0$    Then: $\Delta F = F^{old}_{src}$
If $F^{new,temp}_{dest} > 1$    Then: $\Delta F = 1 - F^{old}_{dest}$
$F^{new,corrected}_{dest} = F^{old}_{dest} + \Delta F$
$F^{new,corrected}_{src} = F^{old}_{src} - \Delta F$
```

3.4 Solve the diffusion equation

We apply the multi-material supermesh FVM-MOF method we have developed in the previous paper to solve the heat diffusion part of the Stefan problem, which is linearly exact for Dirichlet boundary condition at the interfaces.

3.4.1 Discretization

We discretize the time derivative using the backward Euler method which is a discrete maximum principle preserving time discretization scheme. For the derivation of our spatial discretization algorithm, we integrate (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).

$$\int_{\tilde{T}^n}^{T^{n+1}} \Big\{ \frac{1}{|\Omega_m|} \Big[\int_{\Omega_m} (\theta_m)_T dV = \int_{\Omega_m} \nabla \cdot (\kappa_m \nabla \theta_m) + \dot{Q}\ dV \Big] \Big\} dT \tag{12}$$

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

$$\int_{\tilde{T}^n}^{T^{n+1}} \Big\{ (\bar{\theta}_m)_T = \frac{1}{|\Omega_m|} \Big[\int_{\Omega_m} \nabla \cdot (\kappa_m \nabla \theta_m) + \dot{Q}\ dV \Big] \Big\} dT \tag{13}$$

$$\int_{\tilde{T}^n}^{T^{n+1}} \Big\{ (\bar{\theta}_m)_T = \frac{1}{|\Omega_m|} \Big[\int_{\partial\Omega_m} \kappa_m \nabla \theta_m \cdot \boldsymbol{n} dS + \int_{\Omega_m} \dot{Q}\ dV \Big] \Big\} dT \tag{14}$$

$$|\Omega_m|(\theta(T^{n+1}, \boldsymbol{x_m}) - \theta_\Gamma(\tilde{T}^n, \boldsymbol{x_m})) \doteq \Big\{ \oint_{\partial\Omega_m} \kappa_m (\nabla\theta)(T^{n+1}, \boldsymbol{x_m}) \cdot \boldsymbol{n} dS + \int_{\Omega_m} \dot{Q}^{n+1}\ dV \Big\} (T^{n+1} - \tilde{T}^n) \tag{15}$$

$x_m(\Omega_m)$ 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.

Note, θ^n must be interpolated from old T^n supermesh to the new T^{n+1} supermesh with the aid of T_Γ.

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:

$$\begin{gathered} \int_{\partial\Omega_m} k_{m,m'} \nabla \theta_m \cdot \boldsymbol{n}_m dS \approx \sum_{iface=1}^{F} \sum_{m'=1}^{M} q_{iface,m,m'} A_{iface,m,m'} \\ q_{iface,m,m'} = k_{m,m'} \nabla \theta_m \cdot \boldsymbol{n}_m \end{gathered} \tag{16}$$

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; $\boldsymbol{n}_m$ is the outward facing normal of the interface or cell face.

3.4.2 "Connected Centroid" Approximation for Interfacial Fluxes.

In previous article we present the "connected centroid" method for approximating the flux. Referring to Figure 9, the "connected centroid" method is:

$$q_{iface,m,m'} = k_{m,m'} \frac{\theta_{m'} - \theta_m}{d_{m',m}} |\boldsymbol{n} \cdot \boldsymbol{n_{iface}}| \tag{17}$$

$$= k_{m'} \frac{\theta_{m'} - \theta_{\text{intersect}}}{d_{m',\text{intersect}}} |\boldsymbol{n} \cdot \boldsymbol{n_{iface}}| = k_m \frac{\theta_{\text{intersect}} - \theta_m}{d_{\text{intersect},m}} |\boldsymbol{n} \cdot \boldsymbol{n_{iface}}| \tag{18}$$

where the index "intersect" indicates the intersection point of the line connecting two centroids and the cell face or the interface. $\boldsymbol{n} = \dfrac{\boldsymbol{x_{m'}} - \boldsymbol{x_m}}{\|\boldsymbol{x_{m'}} - \boldsymbol{x_m}\|}$, $d_{m',m}$ is the distance between $\boldsymbol{x_{m'}}$ and $\boldsymbol{x_m}$, $\boldsymbol{n_{iface}}$ is the face/interface normal, and $k_{m,m'}$ is the weighted geometric average of the diffusion coefficients k_m and $k_{m'}$.

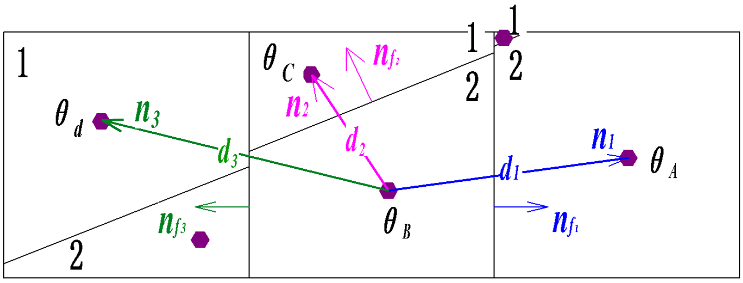

Fig. 9: Connected Centroid method for approximating the flux.
The discretized material m temperature θ_m is stored at the material m centroid $\boldsymbol{x}_m$.

3.4.3 *Linear exact algorithm for calculating the heat flux*

If an embedded interface has a Dirichlet boundary condition,

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

or a Neumann boundary condition,

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

we have designed a correction to our connected centroid flux approximation (see Section 3.4.2) which is exact for the specific cases in which the exact solution is linear in space:

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

Linear exact algorithm for interior flux approximation. Referring to Figure 10, the linear exact approximation for the material m interior flux on the left cell is:

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

Linear exact algorithm for exterior flux approximation.

Referring to Figure 10, the connected centroid flux approximation for q_{ext} is,

$$q_{\text{ext}} = k_m \frac{\theta(\boldsymbol{x}_+) - \theta(\boldsymbol{x}_-)}{||\boldsymbol{x}_+ - \boldsymbol{x}_-||} |\boldsymbol{n}' \cdot \boldsymbol{n}_{f,\text{ext}}|, \tag{19}$$

where

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

The approximation in (19) makes the assumption that

$$\theta(\boldsymbol{x}) \approx \theta(\boldsymbol{x}_-) + \frac{\theta(\boldsymbol{x}_+) - \theta(\boldsymbol{x}_-)}{||\boldsymbol{x}_+ - \boldsymbol{x}_-||} (\boldsymbol{x} - \boldsymbol{x}_-) \cdot \boldsymbol{n}' \tag{20}$$

The linear exact algorithm augments the approximation in (20) as follows,

$$\theta(\boldsymbol{x}) \approx \theta(\boldsymbol{x}_-) + \frac{\theta(\boldsymbol{x}_+) - \theta(\boldsymbol{x}_-)}{||\boldsymbol{x}_+ - \boldsymbol{x}_-||} (\boldsymbol{x} - \boldsymbol{x}_-) \cdot \boldsymbol{n}' + a_1 (\boldsymbol{x} - \boldsymbol{x}_-) \cdot \boldsymbol{t}_1 + a_2 (\boldsymbol{x} - \boldsymbol{x}_-) \cdot \boldsymbol{t}_2 \tag{21}$$

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

1. Pick the reconstructed plane with the largest surface area from the two adjoining cells; in Figure 10, the left plane has the largest surface area:
$$\Gamma_- = \left\{ \boldsymbol{x} | \boldsymbol{n}_{f,\text{int}} \cdot (\boldsymbol{x} - \boldsymbol{x}_-^I) = 0 \right\}$$
2. Determine $\boldsymbol{t}_1^I$ and $\boldsymbol{t}_2^I$ which are orthogonal to each other and orthogonal to $\boldsymbol{n}_{f,\text{int}}$.
3. Determine $\boldsymbol{x}_2^I$ and $\boldsymbol{x}_3^I$ on the plane Γ_-:
$$\boldsymbol{x}_2^I = \boldsymbol{x}_-^I + \epsilon_T ||\boldsymbol{x}_- - \boldsymbol{x}_+|| \boldsymbol{t}_1^I$$
$$\boldsymbol{x}_3^I = \boldsymbol{x}_-^I + \epsilon_T ||\boldsymbol{x}_- - \boldsymbol{x}_+|| \boldsymbol{t}_2^I$$
Here, set $\epsilon_T = \frac{1}{100}$
4. For the Dirichlet problem, define the following data points used for the least square procedure,
$$\begin{aligned}
&\boldsymbol{y}_1 \equiv \boldsymbol{x}_2^I \qquad \theta_1 \equiv g(\boldsymbol{x}_2^I) \\
&\boldsymbol{y}_2 \equiv \boldsymbol{x}_3^I \qquad \theta_2 \equiv g(\boldsymbol{x}_3^I) \\
&\boldsymbol{y}_3 \equiv \boldsymbol{x}_-^I \qquad \theta_3 \equiv g(\boldsymbol{x}_-^I) \\
&\boldsymbol{y}_4 \equiv \boldsymbol{x}_+^I \qquad \theta_4 \equiv g(\boldsymbol{x}_+^I) \qquad \text{if } \boldsymbol{x}_+^I \text{ exists.}
\end{aligned}$$
For the Neumann problem, define the following data points used for the least square procedure,
$$\begin{aligned}
&\boldsymbol{y}_1 \equiv \boldsymbol{x}_2^I \qquad \theta_1 \equiv \theta(\boldsymbol{x}_-) + ||\boldsymbol{x}_- - \boldsymbol{x}_-^I|| g(\boldsymbol{x}_2^I)/k_m \\
&\boldsymbol{y}_2 \equiv \boldsymbol{x}_3^I \qquad \theta_2 \equiv \theta(\boldsymbol{x}_-) + ||\boldsymbol{x}_- - \boldsymbol{x}_-^I|| g(\boldsymbol{x}_3^I)/k_m \\
&\boldsymbol{y}_3 \equiv \boldsymbol{x}_-^I \qquad \theta_3 \equiv \theta(\boldsymbol{x}_-) + ||\boldsymbol{x}_- - \boldsymbol{x}_-^I|| g(\boldsymbol{x}_-^I)/k_m \\
&\boldsymbol{y}_4 \equiv \boldsymbol{x}_+^I \qquad \theta_4 \equiv \theta(\boldsymbol{x}_+) + ||\boldsymbol{x}_+ - \boldsymbol{x}_+^I|| g(\boldsymbol{x}_+^I)/k_m \qquad \text{if } \boldsymbol{x}_+^I \text{ exists.}
\end{aligned}$$
5. Solve the following optimization problem:
$$\min_{a_1,a_2} \sum_{j=1}^{4} w_j ||\theta_j - \theta(\boldsymbol{y}_j)||^2. \tag{22}$$
The weights are defined as:
$$w_j = \epsilon \qquad j = 1,2 \qquad \epsilon = 1/100 \text{ for example} \tag{23}$$
$$w_3 = \frac{|\Gamma_-|}{|\Gamma_-| + |\Gamma_+|} \tag{24}$$
$$w_4 = \frac{|\Gamma_+|}{|\Gamma_-| + |\Gamma_+|} \tag{25}$$
6. The flux will be calculated as $k_m \nabla\theta \cdot \boldsymbol{n}_{f,\text{ext}}$.

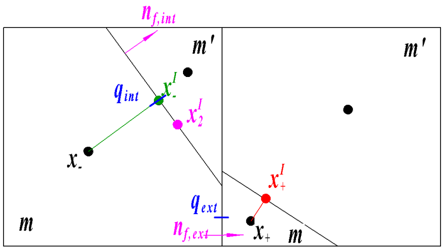

Fig. 10: 2D Linear exact algorithm example

3.4.4 Gap Error Analysis-the Threshold for Reverting to the First Order Connected Centroid Method from the Linear Exact Method

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 [20] 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 $\Delta x \to 0$, for smooth interfaces. In this section we give an alternate way to measure the gap error influence, “the area gap error”,

$$(E_{gap})_{\Omega} = \max_{\substack{all\ faces\\ of\ \Omega}} \sum_{i=1}^{nmat} \sum_{\substack{j=1,\\ j\neq i}}^{nmat} A_{i,j}. \tag{26}$$

Figure 11 shows a piece of such “area gap error” on a face of the cell.

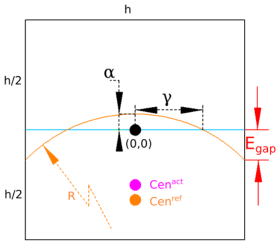

Fig. 11: Example for an “area gap error”. The real interface in the cell is an arc with radius R, the centroid of material 2 region is the purple point; the reconstructed interface is the horizontal blue line coincide with x axis centered at $(0, 0)$ dividing the cell into half. The gap error is indicated in the figure.

We use a simple case in Figure 11 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 $(0, 0)$ dividing the cell into half. Starting from representing the reference interface with a function of y, we have the following derivation:

$$(y + (R - \alpha))^2 + x^2 = R^2 \tag{27}$$

$$y(x) = \sqrt{R^2 - x^2} - R + \alpha \tag{28}$$

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

$$\int_0^{\gamma} y dx = -\int_{\gamma}^{h/2} y dx \tag{29}$$

$$\alpha = R - \sqrt{R^2 - \gamma^2} \tag{30}$$

$$\int_0^{\gamma} \sqrt{R^2 - x^2} - \sqrt{R^2 - \gamma^2} dx = \int_{\gamma}^{h/2} \sqrt{R^2 - \gamma^2} - \sqrt{R^2 - x^2} dx \tag{31}$$

$$\int_0^{h/2} \sqrt{R^2 - x^2} dx = h/2\sqrt{R^2 - \gamma^2} \tag{32}$$

Since, $\int \sqrt{a^2 - u^2} du = \frac{a^2}{2} \arcsin \frac{u}{a} + \frac{u}{2}\sqrt{a^2 - u^2} + C$ (33)

$$\frac{R^2}{2} \arcsin \frac{h/2}{R} + \frac{h}{4}\sqrt{R^2 - \frac{h^2}{4}} = h/2\sqrt{R^2 - \gamma^2} \tag{34}$$

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

$$\frac{R^2}{2} \arcsin \frac{\beta}{2} + \frac{\beta R}{4}\sqrt{R^2 - \frac{\beta^2 R^2}{4}} = (\beta R)/2\sqrt{R^2 - \sigma^2 R^2} \tag{35}$$

$$2 \arcsin \frac{\beta}{2} + \beta\sqrt{1 - \frac{\beta^2}{4}} = 2\beta\sqrt{1 - \sigma^2} \tag{36}$$

$$\sqrt{1 - \sigma^2} = \frac{2 \arcsin \frac{\beta}{2} + \beta\sqrt{1 - \frac{\beta^2}{4}}}{2\beta} \tag{37}$$

Finally, evaluate value y at $x = h/2$, and plug in Equation 37

$$E_{Gap} = R|\frac{1}{2}\sqrt{1 - \beta^2/4} - \frac{\arcsin \frac{\beta}{2}}{\beta}| \tag{38}$$

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

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

$$E_{Gap} \approx R|\frac{1}{2}(1 - \frac{\beta^2}{8}) - \frac{\frac{\beta}{2} + \frac{\beta^3}{48}}{\beta}| \tag{39}$$

$$E_{Gap} \approx \frac{\beta^2}{12} R \tag{40}$$

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,

$$E_{Gap} \approx \frac{\beta^2}{12} R = \frac{\frac{h^2}{R^2}}{12} R = \frac{h^2}{12R} = \frac{h^2}{12\eta h} = \frac{h}{12\eta} \tag{41}$$

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 18). The choice of this threshold is still quite empirical right now, and it needs more future work.

3.4.5 Paring algorithm

One of the key features that makes our multi-material supermesh method simple and distinguished from other methods is the paring algorithm presented in previous paper [20]. We have improved the paring algorithm in this paper and now it's more robust for complex geometry and also could be applied for 3D problem.

To elaborate the new paring 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 12(a). They both contain three materials marked as 1 2 3. And reconstructed interfaces are shown in the figure.

1. As in the previous paring algorithm, we start from projecting all the interfaces onto the direction that is perpendicular to the target cell face. Given an interface,

$$\varGamma = \{\boldsymbol{x} | \boldsymbol{n} \cdot (\boldsymbol{x} - \boldsymbol{x_0}) + \alpha = 0\},$$

   and a cell face,

$$\varGamma_{cell\ face} = \left\{\boldsymbol{x} | \boldsymbol{n_f} \cdot (\boldsymbol{x} - \boldsymbol{x_0}) + \alpha_f = 0\right\},$$

   find a projected interface,

$$\varGamma_{project} = \left\{\boldsymbol{x} | \boldsymbol{n}_{proj} \cdot (\boldsymbol{x} - \boldsymbol{x_0}) + \alpha_{proj} = 0\right\},$$

   which satisfies,

$$\varGamma_{project} \cap \varGamma_{cell\ face} \cap \varOmega = \varGamma \cap \varGamma_{cell\ face} \cap \varOmega$$

   and $\boldsymbol{n}_{proj} \cdot \boldsymbol{n}_f = 0$.
   In figure 12, each interface has its corresponding projected interface (dotted lines).
2. Perturb the cell faces to find the intersections.
   For example, in Figure 12 (b), the target cell face is perturbed by $\delta = 10^{-6}$ to

$$\varGamma_{pert} = \left\{\boldsymbol{x} | \boldsymbol{n_f} \cdot (\boldsymbol{x} - \boldsymbol{x_0}) + \alpha_f + \delta = 0\right\}.$$

   We divide the rectangular perturbation regions into small slabs about the intersections (points in 2D, lines in 3D) on both sides of the target cell face. Finally, for each thin rectangular slab, we get the corresponding area fraction $A_m = \dfrac{V_m(slab)}{V_{total}(slab)}$ and centroids.
3. Then we need choose either to subtract material slabs of blue region (left) from orange region (right) or subtract those of orange from blue. In Figure 13, we illustrate the algorithm with subtracting material slabs one by one of blue region from the right orange region. Note, the material order doesn't matter in this algorithm.

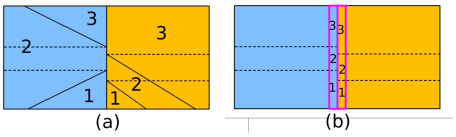

Fig. 12: Paring algorithm part 1:
interface projections are shown in (a)
and thin material slabs formation are shown in (b)

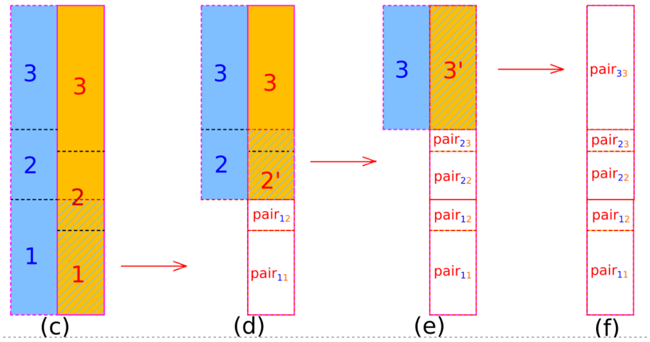

Fig. 13: Pairing algorithm part 2:
In Figure (c), we subtract the blue shaded region from right. After the cutoff left material 1 from the right, intuitively, as example shown in (d),
$pair_{1,2}$ = right volume 2 before cut − right volume 2 after cut
$pair_{1,1}$ = right volume 1 before cut − right volume 1 after cut

1 *3.4.6 The crossing-time approach (supermesh motivated)*

For ghost fluid method, if a cell center is not occupied by the destination material at T_n, but is occupied at T_{n+1}, then the crossing time approach for temporal advancing is

$$\frac{\theta^{n+1}-\theta_I}{T_{n+1}-T_{swept}} = L(\theta^{n+1}) \quad \text{, where, } T_{swept}\text{is the crossing time.} \tag{42}$$

To be specific, define the swept factor as R^{swept}, for the GFM method case:

$$F = \begin{cases} F^n, & T = T_n \\ 1/2, & T = T_{swept} \\ F^{n+1}, & T = T_{n+1} \end{cases} \tag{43}$$

$$F(T) = \frac{F^{n+1}-F^n}{T_{n+1}-T_n} * (T - T_{n+1}) + F^{n+1} \tag{44}$$

$$1/2 = F(T_{swept}) = \frac{dF}{T_{n+1}-T_n} * (T_{swept} - T_{n+1}) + F^{n+1} \tag{45}$$

$$R^{swept} = \frac{T_{swept}-T_{n+1}}{T_{n+1}-T_n} = \frac{1/2 - F^{n+1}}{dF} \tag{46}$$

2 New approach for determining the swept factor which is motivated by the supermesh is
3 developed as:

1. Given an estimate of the phase change velocity "$USTEFAN$". dT is chosen such that $USTEFAN * dT \leq dx/4$. This assures that a swept cell will NOT be full at T_{n+1}
2. (a) We have the signed distance function at T_n.
   (b) Let x^* be either (i) cell center if GFM, or (ii) cell centroid at T_{n+1} of the destination material.
   (c) If $\phi(T_n, x^*) < 0$ then find MOF reconstruction and determine if $\phi^{reconstruct}(T_{n+1}, x^*) > 0$, if yes. then the swept factor $R^{swept} = 1 - \frac{\phi_{n+1}}{\phi_{n+1} - \phi_n}$

Note this is a more general version for calculating the swept factor.

The overall strategy to apply the crossing time can be summarized as follows. Check if the new centroid $\boldsymbol{x}(im_{dest})$ is in the old destination material. If yes, then the cell centroid has not been swept and the temperature at this point needs to be interpolated from the old temperature field based on the old supermesh. If not, the cell centroid has been swept, then we can compute the swept factor and use the crossing time and temperature at the interface to perform the temporal advancing.

4 Numerical tests

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, (D) unstable solidification and (E) the filament ring growth test.

4.1 Moving Planar Interface-Sanity Check

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

The exact solution to this Stefan problem is given by

$$\theta(\boldsymbol{x}, t) = \begin{cases} 272 + e^{-V(x-0.1-Vt)}, & x > 0.1 + Vt, \\ 273, & x \leq 0.1 + Vt \end{cases} \tag{47}$$

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 1). One can observe a first order convergence of the error under grid refinement.

4.2 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: $r_0 = 0.4$;

Table 1: Convergence study of moving planar interface with the exact solution $x = 0.2$ at the end time $t = 1.0$

Space resolution	Steps	computing result	error
16×16	8	0.1963	0.0037
32×32	16	0.1980	0.0020
64×64	32	0.1989	0.0011
128×128	64	0.1995	0.005

2. Initial temperature field

$$\theta(r, 0) = \begin{cases} -4\cos(0.5\pi r/r_0) + T_{sat} & r < r_0 \\ T_{sat} & r > r_0 \end{cases} \tag{48}$$

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 2. 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.

Table 2: Convergence study of shrinking circle test. Exact radius is 0.3868

Space resolution	Time steps	Compute radius	error
64×64	5	0.3888	0.0020
128×128	10	0.3880	0.0012
256×256	20	0.3873	0.0005
512×512	40	0.3870	0.00007

In Figure 14, we provide the visualized results. From the zoom-in look of the circles in different resolutions, we can observe the convergence, as well.

4.3 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 (Figure 15) outline to nothing.

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

- $\kappa_1 = \kappa_2 = 1, L = 1$

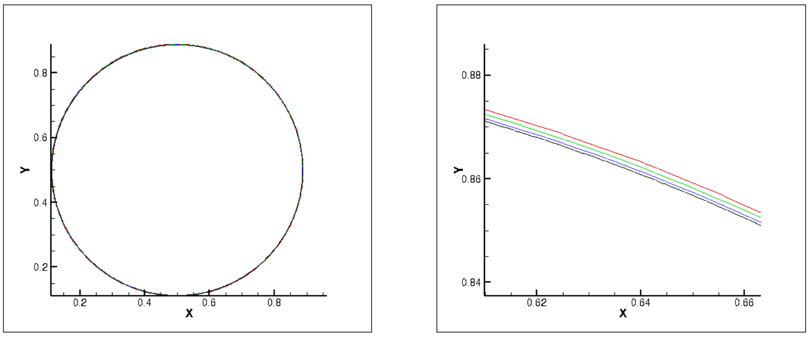

Fig. 14: Convergence study of shrinking circle at $Time = 6.25E-4$, expected radius= 0.3868.
Left: interfaces for all the resolutions overlaid on top of each other red 64×64, green 128×128, blue 256×256, black 512×512; Right: zoom-in look of the left figure.

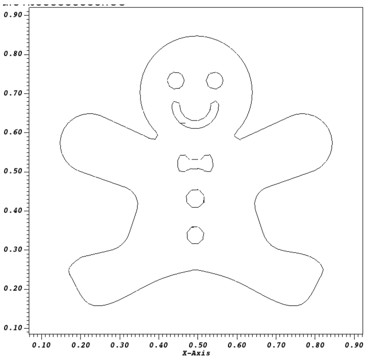

Fig. 15: Interface reconstruction for gingerbread man melting test.

- Initial temperature field

$$\theta(x,y,0) = \begin{cases} 274, & \text{if} \quad y = 0, \text{or} \quad y = 1, \\ 273, & \text{everywhere else in the domain.} \end{cases} \tag{49}$$

- 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 16 shows the temperature field and the interface evolution at two different times on 64×64 domain. For this initial temperature field, the interface is expected to be compressed in y direction.

With uniform $\varDelta t$ for each grid resolution, as shown in Figure 17, 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 3.4.3) 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 $\varDelta x$. Figure 18 shows the cells (marked as yellow) with large gap error in different time steps.

Table 3: Convergence study of gingerbread man melting. Compare the gingerbread volume at time $T = 0.5$

Space resolution	Compute radius	relative error
64 × 64	0.20152	0.00140
128 × 128	0.20012	0.00054
256 × 256	0.19958	0.00024
512 × 512	0.19934	

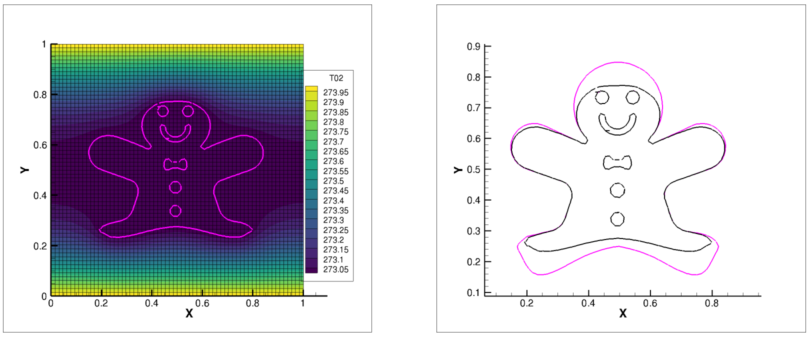

Fig. 16: Left: Temperature field in sometime during the computation; Right: Interface evolution from black $Time = 0.0$ to purple $Time = 0.02$. Both on 64×64 domain

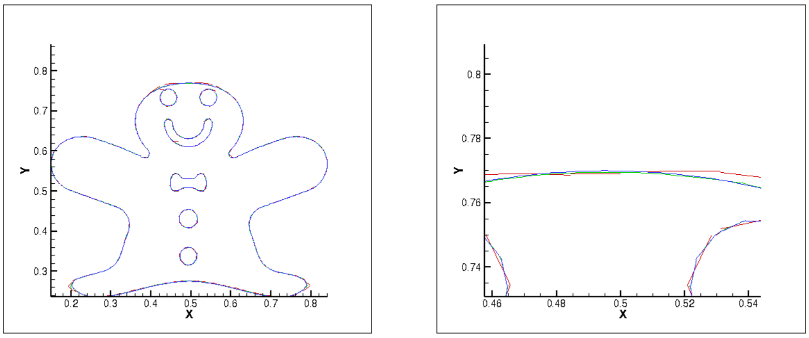

Fig. 17: Convergence observation for different resolution at $Time = 0.02$.
Red: 64×64, 50 steps.
Green: 128×128, 100 steps.
Blue: 256×256, 200 steps.

4.4 The Unstable Solidification-Dendrite Growth Test

Next, 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 [15], Chen et al[4], Gibou et al.[11] and Kaiser et al. [16], 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:

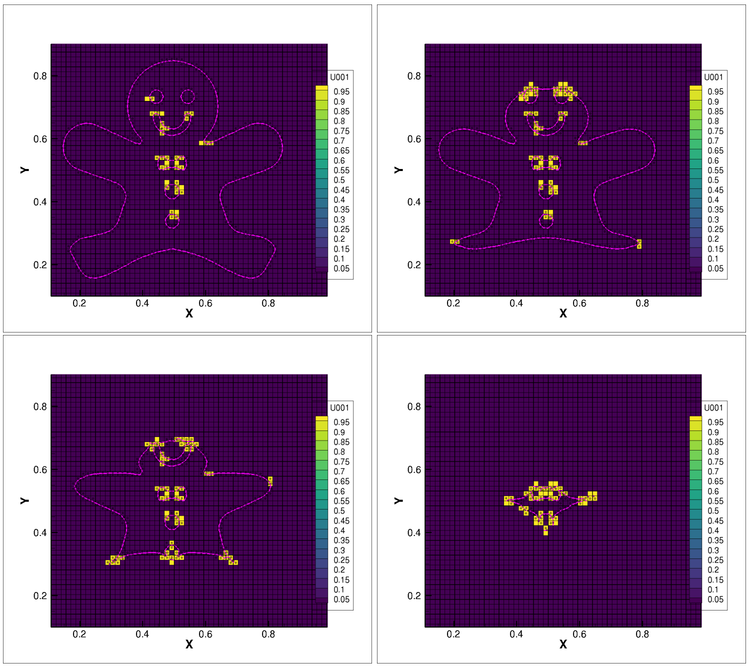

Fig. 18: Gingerbread man evolution with area gap error analysis. The yellow boxes show where the area gap error exceeds the threshold of a radius of curvature less than Δx.

1. The initial interface is defined parametrically as:

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

   Figure 19 shows the shape of the initial interface. We assume the region inside the seed is D
2. The initial temperature field is

$$\theta = \begin{cases} 273, & \text{if} \quad x \in D, \\ 272, & \text{otherwise.} \end{cases} \tag{50}$$

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 51) we use in the test is a simplified version for only the isotropic case.

$$\theta(\boldsymbol{x}, t) = -\epsilon_1 \kappa - \epsilon_2 V \tag{51}$$

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 4), 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:
   $V^{(l+1)} = \frac{[\kappa \nabla \theta^{(l)} \cdot \boldsymbol{n}]}{L}$,
   $\theta_I^{(l+1)} = \theta_I^{(l)} + \epsilon_2 V^{(l)} - \epsilon_2 V^{(l+1)}$,
   until convergence ($|V^{(l)} - V^{(l+1)}| < \epsilon$).
```

Figure 20 shows the computational result for dendrite growth at different time for 128 × 128 resolution when $\epsilon_1 = \epsilon_2 = 0$.

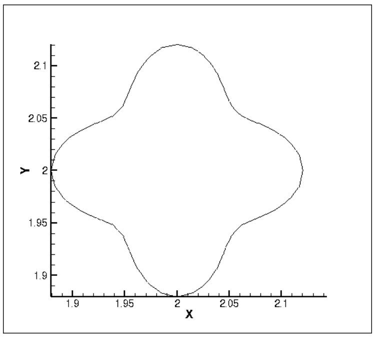

Fig. 19: Initial seed of dendrite growth test

Figure 21 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 shown in Figure 21 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.[4].

Figure 22 shows a interface evolution on 256×256 grid with different time steps after 1600 steps. The results get good agreement with the results shown in Chen et al.[4].

The supermesh algorithm we have developed is proved to handle unstable solidification problems and shows as good results as previous methods.

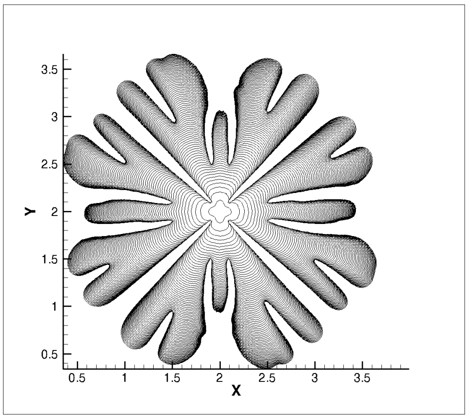

Fig. 20: Dendrite growth evolution without surface tension restriction at different time steps for resolution 128 × 128 after 1600 steps.

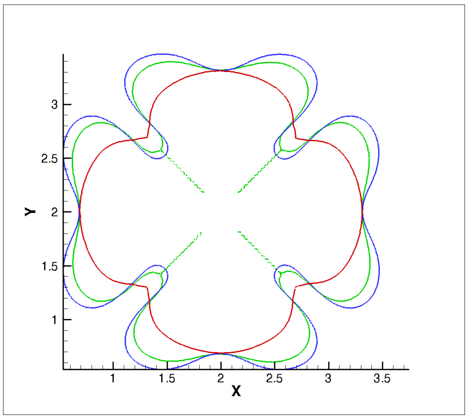

Fig. 21: Comparison of dendrite growth with surface tension restriction with different resolution at $Time = 0.32$ after 640 steps with $\epsilon_1 = 0.002, \epsilon_2 = 0.002, \Delta t = 1/2000$. Red: 64 × 64; Green: 128 × 128; Blue: 256 × 256.

4.5 The Filament Ring Growth Test

In order to provide evidence that the supermesh algorithm we have developed is able to increase the resolution potentially, we tested the algorithm with our Filament Ring Growth test.

- Physical domain and interface set-up
  The computational domain is $[0,1] \times [0,1]$. And the filament ring is defined by the following two circles ϕ_- (inner) and ϕ_+ (outer).

$$\phi_{\pm} = 0.25 \pm 0.05 - \sqrt{(x-0.5)^2 + (y-0.5)^2}$$

  So the thickness of the filament is 0.1 and it is under-resolved over 32 × 32 grid.

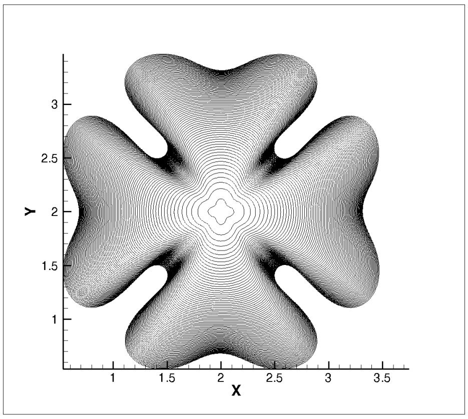

Fig. 22: Dendrite growth evolution with surface tension restriction on 256×256 grid at different time after 1600 steps with $\epsilon_1 = 0.002, \epsilon_2 = 0.002, \Delta t = 1/2000$.

- Material IDs and properties
  Material 1 is the region inside the inner circle, material 2 is the material in the ring and the rest part of the domain is material 3. We assume all three materials have the same thermal conductivity and density.
- Initial temperature and phase change set-up
  We set the initial temperature in material 2 as T_{sat} and the temperature in material 1 and 3 are both supercooled by 0.5. Both interfaces will have phase change occurred at the same time which will result in the growth of the filament ring. Material 1 and 3 will turn into material 2.

We compute the volume of all materials and measure the radius of the inner and outer circle to perform the following convergence study.

Table 4: Convergence study of gingerbread man melting. Compare the gingerbread volume at time $T = 0.5$

Space resolution	R1	R1 relative error	R2	R2 relative error
32×32	0.21408	2.6763e-5	0.32287	6.3695e-3
64×64	0.21410	1.8583e-5	0.32924	4.47633e-3
128×128	0.21412	-4.4598e-6	0.33371	2.5494e-3
256×256	0.21411		0.33627	

From the convergence study, the radius of both inner and outer circles are convergent. And the relative errors are quite small. The algorithm we have developed does improve the resolution without AMR. More results are shown in Figure 23 and 24.

5 conclusion

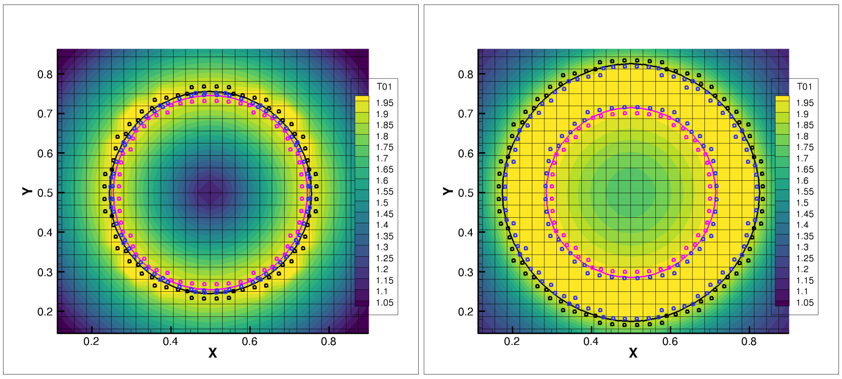

Fig. 23: Left: Temperature field, interfaces, centroids of near interfaces at time $t = 0.0$; Right: Temperature field, interfaces, centroids of near interfaces at time $t = 0.01$. The filament is under-resolved initially.

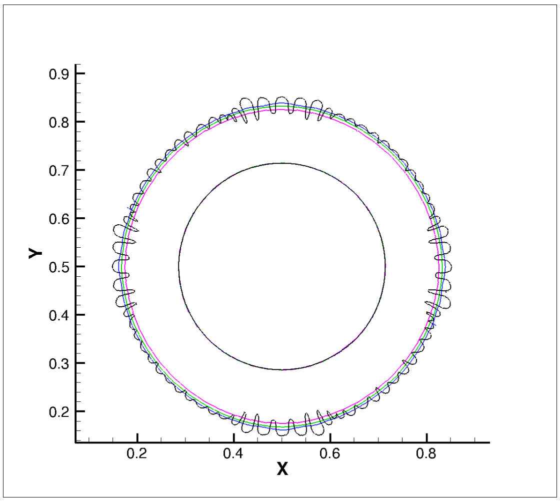

Fig. 24: Interfaces at time $t = 0.01$ with different resolutions: pink as 32×32, green as 64×64, blue as 128×128 and black as 256×256. The errors at the inner circles are very small, so they almost stack on each other. The black one shows instability at the outer circle which is expected since the very small time step causes the problem.

References

1. A Andriotis, M Gavaises, and C Arcoumanis. Vortex flow and cavitation in diesel injector nozzles. Journal of Fluid Mechanics, 610:195–215, 2008.
2. C Arcoumanis, M Gavaises, H Flora, and H Roth. Visualisation of cavitation in diesel engine injectors. Mécanique & industries, 2(5):375–381, 2001.
3. Edip Can and Andrea Prosperetti. A level set method for vapor bubble dynamics. Journal of computational physics, 231(4):1533–1552, 2012.
4. 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.

5. 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.
6. 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.
7. Vadim Dyadechko and Mikhail Shashkov. Reconstruction of multi-material interfaces from moment data. Journal of Computational Physics, 227(11):5361–5384, 2008.
8. Amir Faghri. Review and advances in heat pipe science and technology. Journal of heat transfer, 134(12), 2012.
9. 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.
10. 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.
11. 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.
12. 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.
13. C Höhmann and P Stephan. Microscale temperature measurement at an evaporating liquid meniscus. Experimental Thermal and Fluid Science, 26(2-4):157–162, 2002.
14. Matthew Jemison, Mark Sussman, and Mikhail Shashkov. Filament capturing with the multimaterial moment-of-fluid method. Journal of Computational Physics, 285:149–172, 2015.
15. Damir Juric and Grétar Tryggvason. A front-tracking method for dendritic solidification. Journal of computational physics, 123(1):127–148, 1996.
16. 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.
17. 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.
18. 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.
19. 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.
20. 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.
21. 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.
22. 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.
23. Yu F Maydanik. Loop heat pipes. Applied thermal engineering, 25(5-6):635–657, 2005.
24. Anvarbek Mukatovich Meirmanov. The Stefan Problem, volume 3. Walter de Gruyter, 2011.
25. 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.
26. 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.
27. 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.
28. 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.
29. 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.
30. Chaoxu Pei, Mark Sussman, and M Yousuff Hussaini. A hierarchical block structured space-time spectral element method for simulating complex multiphase flows.
31. Jan Pruss, Jurgen Saal, and Gieri Simonett. Existence of analytic solutions for the classical stefan problem. Mathematische Annalen, 338(3):703–756, 2007.

32. 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.
33. 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.
34. 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.
35. R Savino and D Paterna. Marangoni effect and heat pipe dry-out. Physics of Fluids, 18(11):118103, 2006.
36. 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.
37. 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.
38. 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.
39. 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.
40. 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.
41. Cornelis Vuik. Some historical notes about the stefan problem. 1993.
42. 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.
43. 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.
44. 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.
45. Li Yang and GM Homsy. Steady three-dimensional thermocapillary flows and dryout inside a v-shaped wedge. Physics of Fluids, 18(4):042107, 2006.
46. 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.