A non-intrusive numerical method for determing the most dangerous unstable modes of multiphase/multimaterial systems. *

Okojunu, Abel
Dept. of math, FL State Univ.

Nair, Unnikrishnan Sasidharan
Dept. of Mech. Engr., FL State Univ.

Sussman, Mark
Dept. of math, FL State Univ.

February 23, 2026

ABSTRACT: A linear stability analysis technique is presented for finding the most unstable modes for multiphase/multi material systems. Instead of explicitly linearizing a system of differential equations and then investigating the spectrum of the linearized equations, it is proposed to simply do a "matrix-free" power method approach. The new power method approach determines the action of the linear operator, associated with the linearized equations, on a given eigenfunction by way of strategically perturbing a base state and then applying the original, unmodified, multiphase flow algorithm. The multiphase flow algorithm is the coupled level set and "continuous MOF" method for multiphase/multimaterial problems. Since the underlying multiphase flow solver is an Eulerian interface capturing approach, solutions with complex topology can be recovered automatically without interface "surgery." Furthermore, since the proposed linear stability algorithm does not require modification of the underlying flow solver, then it is possible to find unstable modes for complex topology problems. Examples are include for Rayleigh capillary instability and a Jet in Cross Flow problem.

1 Introduction

We present an improved "Continuous Moment-of-Fluid" (CMOF) algorithm for computing solutions to multiphase (multimaterial) flows. We demonstrate the

*this material is based upon work supported by National Aeronautics and Space Administration under grant number 80NSSC20K0352

benefits of our new algorithm, beyond that of the current state-of-the-art, on these benchmark multiphase flow problems: bubble formation[23] (see Section **??**), freezing[25] (see Section **??**), liquid lens[35] (see Section **??**), and bubble dynamics in a Cryogenic fuel tank[6] (see Section **??**).

Figure **??** (LEFT figure) illustrates the Moment-of-Fluid[14, 1, 2] reconstruction of a "saw tooth" function that has a wave length of Δx. The MOF reconstruction is very "noisy" in the sense that the highest frequency Fourier coefficients of the discrete Fourier transform of the reconstruction do not decay with repeated reconstructions. In other-words the amplitudes of the high frequency Fourier modes may not decay when exposed to the repeated process of (1) MOF advection, and (2) MOF reconstruction. The presence of $O(\Delta x)$ wave length noise is a problem for applications involving surface tension driven flows. Standard Volume-of-Fluid techniques for extracting the curvature from volume fractions[52, 12] will extract a zero curvature from a "saw tooth" interface since the associated volume fraction field *and* centroid field varies in the y direction only. In other words, the underlying computational fluid dynamics surface tension force algorithm will never "see" the jagged interface and therefore will have no way for removing the noise. If there is viscosity, the presence of the noise is unphysical and the noise can lead to collateral damage (see Figure **??** and Table **??**) to the overall flow field; the noise can result in loss of accuracy.

In order to overcome the "MOF checkerboard instability" issue, we have developed the "Continuous Moment of Fluid" (CMOF) interface reconstruction algorithm[58]. Referring to Figure **??**, the repeated process of (1) CMOF advection, followed by (2) CMOF reconstruction, will quickly eliminate the noise. The difference between CMOF and MOF is that in the CMOF algorithm, the reference centroid is the material centroid relative to the encompassing $3 \times 3 \times 3$ stencil of grids cells, rather than just the center cell. See Figure **??**.

We remark that alternative approaches to using MOF for multiphase (multimaterial) flows, e.g the level set method[50, 46, 51], phase field method[27], front tracking method [60], or standard "PLIC" VOF methods[3, 44, 45], do not have the "MOF checkerboard instability" problem. These alternative approaches have other difficulties. The level set or front tracking multimaterial approaches[50, 46, 51, 60] are not volume preserving methods. The phase field method[27] smears the interface over 3 grid cells or more. The standard, second order, "PLIC" VOF methods for M material multimaterial flows[3, 44, 45] minimize a cost function which has $27(M-1)$ degrees of freedom; i.e. the control variable space is $27(M-1)$ dimensions.

Our improved CMOF method, on the other-hand, admits a sharp reconstructed interface, is volume preserving, and the control variable space has only $3(M-1)$ dimensions. The CMOF reconstruction algorithm has been improved in this article by applying "Decision Tree" machine learning techniques in order to rapidly determine the optimal "CMOF" slope. We choose the Decision Tree algorithm[9] which is a "lossless" method if one chooses to store all of the training data samples in the tree structure. In Table 1, we summarize the existing state-of-the-art for numerical methods for multiphase (multimaterial) flows with surface tension.

Author(s)	Triple point reconstruction algorithm DOF[1]	Coupled with fluid	Volume preserving	Sharp interface	Curvature discretization
Smith et al. [50]	Level set	Yes	No	Yes	Level set
Ahn and Shahkov[1]	VOF-GRAD[2] VOF-LVIRA[3] MOF 27 or 3 $\times(M-1)$	No	Yes	Yes	N/A
Kim[30]	Phase field	Yes	Yes	No	Phase field
Dyadechko and Shashkov[15]	MOF $3(M-1)$	No	Yes	Yes	N/A
Caboussat et al. [10]	VOF-IP[4] $27(M-1)$	Yes	No	Yes	Convolution/Height function[17]
Schofield et al. [44, 45]	VOF-PD[5] $27(M-1)$	No	Yes	Yes	N/A
Sijoy and Chaturvedi[49]	VOF-PLIC[6] $27(M-1)$	Yes	Yes	Yes	N/A
Kucharik et al. [31]	VOF-PLIC VOF-PD MOF 27 or 3 $\times(M-1)$	Yes	Yes	Yes	N/A
Bonhomme et al. [7]	VOF $27(M-1)$	Yes	Yes	No	VOF
Starinshak et al. [51]	Interface level set	Yes	No	Yes	N/A
Vu et al. [60]	Front tracking	Yes	No	Yes	Front tracking
Pathak and Raessi[39]	VOF-PLIC $27(M-1)$	No	Yes	Yes	N/A
Shetabivash et al. [46]	Level Set	Yes	No	Yes	Finite Difference
Ancellin et al. [3]	VOF-PLIC $27(M-1)$	No	Yes	Yes	N/A
Huang et al. [27]	Phase field	Yes	Yes	No	Phase field
Present Article, Ye et al	CMOF[7] $3(M-1)$	Yes	Yes	Yes	VOF Height function

[1]: Degrees of Freedom for 3D reconstruction; M is the number of materials
[2]: Gradient based interface reconstruction
[3]: Least squares volume-of-fluid interface reconstruction algorithm
[4]: Interior-point method for the localization of the triple point
[5]: Power diagram
[6]: Piecewise linear interface construction
[7]: Continuous Moment of Fluid construction

Table 1: Recent numerical methods of flows with triple points in chronological order.

2 Background - deforming boundary problems in fluid mechanics

In the Introduction, we briefly demonstrate the benefits of our new method over the current state-of-the-art numerical methods for multi-phase flows in which surface tension forces are important. In this section, we give an overview of numerical methods for deforming boundary problems in computational fluid dynamics in general, and show where our new method fits in.

For many deforming boundary problems in fluid dynamics, "shock capturing" numerical methods are sufficient[21, 11, 59, 22, 48, 43]. Shock capturing methods have limitations. According to "Godunov's theorem"[20], linear one-step second order accurate numerical methods for $\phi_t + c\phi_x = 0$ cannot be monotonicity preserving unless $|c|\Delta t/\Delta x$ is an integer. The concept of "monotonicity preserving" methods and its association to TVD (Total Variational Diminishing) methods was first drawn by Harten[22] who proved that (a) A monotone scheme is TVD and (b) a TVD scheme is monotonicity preserving. When Harten's theory is taken together with Godunov's theorem, one can see that "linear" second order TVD methods only exist under specialized situations. Harten's and Godunov's theory are manifested by the fact that stable shock capturing methods will invariably smear discontinuities over time.

In order to overcome the limitations of shock capturing methods, researchers have developed (i) front tracking methods[19, 57], (ii) front capturing methods[34, 38, 55, 24, 8, 1, 2, 37, 41, 26], and (iii) shock fitting methods[42].

The above methods for overcoming the the time dependent "interface smearing" problem of the shock capturing methods, have been further improved by way of hybridization: the particle level set method[16], the hybrid front tracking and level set method[47], the coupled level set and volume of fluid method[54], and the coupled level set and moment of fluid method[5, 28].

The numerical method that we introduce in this article falls in the latter "hybrid" category in which we introduce the hybrid level set and Continuous Moment of Fluid (CMOF) method for multiphase/multimaterial flows. The level set method and the continuous moment-of-fluid method are hybridized as follows: (a) the smooth level set distance function representation is used to provide an initial guess for the CMOF slope reconstruction step (see Figure **??** and Section 4.2), and (b) the level set distance functions are in turn replaced by the exact signed distance to the continuous moment-of-fluid reconstructed interface (see Figure **??** and Section 4.3). The level set and CMOF representations are synchronized every time step.

3 Mathematical model

We simulate the flow of a multiphase system consisting of M_{fluid} fluid (deforming) materials and M_{rigid} non-deforming materials. The M_{fluid} materials tessellate the computational domain and in the case that a fluid material m_{fluid} coincides with a rigid material, m_{rigid}, the rigid materials' governing equations

take precedence. Otherwise, in regions where fluid materials and rigid materials do not overlap, the fluid materials are governed by the incompressible Navier-Stokes equations of immiscible flows. We refer the reader to Figures **??**, **??**, and **??** for illustrations which distinguish between deforming (tessellating) materials and non-deforming (rigid) materials.

▷ **Material domain and interface:** Mathematically, the domain of rigid *material* m_{rigid} is the region in which $\phi_{m_{rigid}} > 0$:

$$\phi_{m_{rigid}}(\mathbf{x}, t) = \begin{cases} > 0 & \mathbf{x} \in \text{material } m_{rigid}, \\ \leq 0 & \text{otherwise}, \end{cases} \tag{1}$$

$\boldsymbol{x}$ is the position vector in space and t is time. The domain of *material* m_{fluid} is the region in which $\phi_{m_{fluid}} > 0$ and $\phi_{m_{rigid}} < 0$:

$$\phi_{m_{fluid}}(\mathbf{x}, t) = \begin{cases} > 0 & \mathbf{x} \in \text{material } m_{fluid} \cup m_{fluid,ghost}, \\ \leq 0 & \text{otherwise}, \end{cases} \tag{2}$$

The *interface* level set, $\phi_{m1,m2}$ represents the interface between materials $m1$ and $m2$

$$\phi_{m1,m2}(\mathbf{x}, t) = \begin{cases} > 0 & \mathbf{x} \in \text{material } m1, \\ < 0 & \mathbf{x} \in \text{material } m2, \\ = 0 & \mathbf{x} \text{ along } (m1, m2) \text{ interface}. \end{cases} \tag{3}$$

The normal and curvature defined based on these level set functions are:

$$\mathbf{n}_{m1,m2} = \frac{\nabla \phi_{m1,m2}}{|\nabla \phi_{m1,m2}|}, \quad \kappa_{m1,m2} = \nabla \cdot \frac{\nabla \phi_{m1,m2}}{|\nabla \phi_{m1,m2}|}. \tag{4}$$

▷ **Conservation of mass:** We assume that each fluid material, m_{fluid}, is incompressible, so that the velocity field $\boldsymbol{u} = (u, v, w)$ is divergence free within the bulk of each fluid material:

$$\nabla \cdot \boldsymbol{u} = 0. \tag{5}$$

In order to account for phase change or otherwise sources and sinks of mass, we have the following conditions on $\nabla \cdot \boldsymbol{u}$:

$$\nabla \cdot \boldsymbol{u} = \sum_{\text{sources}} \frac{\dot{m}_{\text{source}}}{\rho_{\text{source}}} \delta(\phi_{m_{\text{source}}}) - \sum_{\text{sinks}} \frac{\dot{m}_{\text{sink}}}{\rho_{\text{sink}}} \delta(\phi_{m_{\text{sink}}})$$

$H(\phi)$ is the Heaviside function which is defined as,

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

$\delta(\phi)$ is the Dirac Delta function,

$$\delta(\phi) = H'(\phi).$$

For boiling examples, $\dot{m}$ is the mass flux of boiling liquid across the liquid/vapor interface,

$$\dot{m} = \frac{k_l \nabla T_l \cdot \boldsymbol{n}_{l,v} - k_v \nabla T_v \cdot \boldsymbol{n}_{l,v}}{L},$$

where k_l and k_v are the thermal conductivities in the liquid and ambient vapor regions respectively, ρ_l and ρ_v are the densities in the liquid and ambient vapor regions respectively, L is the latent heat of vaporization, and $\boldsymbol{n}_{l,v}$ is the interface normal vector pointing from the ambient vapor region into the liquid,

$$\boldsymbol{n}_{l,v} = \frac{\nabla \phi_{l,v}}{|\nabla \phi_{l,v}|}.$$

▷ **Conservation of momentum:** The conservation of momentum for each material in its domain is given by

$$\begin{aligned} (\boldsymbol{u}\rho_m)_t + \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u}\rho_m + p_m \mathbb{I}) = \\ \nabla \cdot (2\mu_m \mathbb{D}) + \rho_m \boldsymbol{g}(1 - \alpha_m(T_m - T_{0m})) \quad \text{if} \quad \phi_m(\boldsymbol{x}, t) > 0, \end{aligned} \tag{6}$$

where p_m, T_m, α_m, and μ_m are pressure, temperature, coefficient of thermal expansion, and viscosity of material m respectively, $\mathbf{g}$ is the gravitational acceleration vector, and $\mathbb{D} = (\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T)/2$ is the rate of deformation tensor.

▷ **Conservation of energy:** The conservation of energy for each material in its domain is given by

$$(\rho_m C_{p,m} T_m)_t + \nabla \cdot (\boldsymbol{u}\rho_m C_{p,m} T_m) = \nabla \cdot (k_m \nabla T_m) \quad \text{if} \quad \phi_m(\boldsymbol{x}, t) > 0, \tag{7}$$

where $C_{p,m}$ and k_m are heat capacity and thermal conductivity of material m respectively, and T_m is the temperature.

▷ **Interfacial jump condition:** Here, we write out the general equations for a deforming interface changing phase. We define m_s and m_d as the material id's associated with a "source" material (e.g. boiling liquid or freezing liquid) and "destination" material (e.g. vapor from boiling or ice from freezing) respectively. The location of the interface separating a material m_s region from a material m_d region is governed by the level set equation,

$$\phi_{m_s,m_d,t} + \boldsymbol{u}_{m_s} \cdot \nabla \phi_{m_s,m_d,t} = -\frac{\dot{m}}{\rho_s} |\nabla \phi_{m_s,m_d,t}| \tag{8}$$

$$\phi_{m_d,m_s,t} + \boldsymbol{u}_{m_s} \cdot \nabla \phi_{m_d,m_s,t} = \frac{\dot{m}}{\rho_s} |\nabla \phi_{m_d,m_s,t}| \tag{9}$$

An equivalent expression for the level set governing equations is:

$$\phi_{m_s,m_d,t} + \boldsymbol{u}_{m_d} \cdot \nabla \phi_{m_s,m_d,t} = -\frac{\dot{m}}{\rho_d} |\nabla \phi_{m_s,m_d,t}| \tag{10}$$

$$\phi_{m_d,m_s,t} + \boldsymbol{u}_{m_d} \cdot \nabla \phi_{m_d,m_s,t} = \frac{\dot{m}}{\rho_d} |\nabla \phi_{m_d,m_s,t}| \tag{11}$$

The interface jump conditions, between two materials m_s and m_d, for the velocity, pressure, and temperature, respectively, are,

$$\boldsymbol{u}_{m_s} \cdot \boldsymbol{n}_{m_s,m_d} - \boldsymbol{u}_{m_d} \cdot \boldsymbol{n}_{m_s,m_d} = \dot{m}(\frac{1}{\rho_{m_d}} - \frac{1}{\rho_{m_s}}),$$

$$\begin{aligned}(p_{m_s}\mathbb{I} - p_{m_d}\mathbb{I}).\boldsymbol{n}_{m_s,m_d} = -\sigma_{m_s,m_d}\kappa_{m_s,m_d}\boldsymbol{n}_{m_s,m_d} + \\ (2\mu_{m_s}\mathbb{D}_{m_s} - 2\mu_{m_d}\mathbb{D}_{m_d}).\boldsymbol{n}_{m_s,m_d},\end{aligned}$$

$$T_{m_s} = T_{m_d}. \tag{12}$$

κ_{m_s,m_d} is the interface curvature and is defined as,

$$\kappa_{m_s,m_d} = \nabla \cdot \frac{\nabla \phi_{m1,m2}}{|\nabla \phi_{m1,m2}|}.$$

At a triple point junction a three-phase equilibrium known as the Neumann's triangle[13] determines the angles (see Figure **??**.a):

$$\frac{\sin(\theta_1)}{\sigma_{23}} = \frac{\sin(\theta_2)}{\sigma_{13}} = \frac{\sin(\theta_3)}{\sigma_{12}}. \tag{13}$$

4 Numerical methods and algorithms

We describe our numerical method for the 2D uniform rectangular Cartesian grid case. In the problem domain, a computational cell, $\Omega_{i,j}$, is defined as,

$$\Omega_{i,j} = \left\{ \boldsymbol{x} : x \in \left[x_i - \frac{\Delta x}{2}, x_i + \frac{\Delta x}{2} \right], \right. \tag{14}$$

$$\left. y \in \left[y_i - \frac{\Delta y}{2}, y_i + \frac{\Delta y}{2} \right] \right\} \tag{15}$$

where $\boldsymbol{x}_{i,j} = \{x_i, y_j\}$ is the center of the cell $\Omega_{i,j}$. The domain of material m in a cell at time t^n is denoted by $\Omega^n_{m,i,j}$, and the zeroth and first order moments of the mth material distribution, corresponding to the volume fraction and centroid position, are defined as,

$$F^n_{m,i,j} = \frac{\int_{\Omega^n_{m,i,j}} \mathrm{d}\Omega}{V_{i,j}}, \qquad \boldsymbol{x}^{c,n}_{m,i,j} = \frac{\int_{\Omega^n_{m,i,j}} \boldsymbol{x}\ \mathrm{d}\Omega}{V^n_{i,j,m}}, \tag{16}$$

where the computational cell volume is $V_{i,j} = \int_{\Omega_{i,j}} \mathrm{d}\Omega$, and volume of the portion for material m in a computational cell is $V^n_{i,j,m} = \int_{\Omega^n_{m,i,j}} \mathrm{d}\Omega$. The discretization for a typical point around the triple point is shown in Figure **??**.

4.1 method overview: Staggered grid Projection Method

Referring to Figure **??** and also [40, 58], we discretize the velocity on the marker-and-cell (MAC) grid and we discretize pressure, temperature, level set function(s), volume fraction(s), and centroid(s) at the center of grid cells.

An outline of our operator split method is as follows ([58]):

▷ **Calculate the time step $\Delta t = t^{n+1} - t^n$:**

$$\Delta t_1 = \min_{d=1,\ldots,\mathrm{D}} \frac{\min \Delta x_d}{2D \max |u_d^n|} \qquad \mathrm{D} = 2 \text{ or } 3$$

$$\Delta t_2 = \min_{d=1,\ldots,\mathrm{D}} \frac{\min \Delta x_d}{D \max |u_d^{\text{phase change},n-1}|}$$

$$\Delta t_3 = \min_{d=1,\ldots,D} \min_{m1,m2=1,\ldots,M} \Delta x_d^{3/2} \sqrt{\frac{\rho_{m1} + \rho_{m2}}{2\pi\sigma_{m1,m2}}} \tag{17}$$

$$\Delta t = \mathrm{CFL} \min(\Delta t_1, \Delta t_2, \Delta t_3) \qquad \mathrm{CFL} = 1/2$$

▷ **Cell Integrated Semi-Lagrangian (CISL) MAC velocity advection (see[40]):**

$$(\rho \boldsymbol{u})_t + \nabla \cdot (\rho \boldsymbol{u}\boldsymbol{u}) = 0, \tag{18}$$

▷ **CISL temperature advection (liquid and gas materials, see [58]):**

$$(\rho_m C_{p,m} T_m)_t + \nabla \cdot (\rho_m C_{p,m} \boldsymbol{u} T_m) = 0, \quad m = 1, \ldots, M \tag{19}$$

▷ **CISL level set advection:**

$$\phi_{m,t} + \boldsymbol{u} \cdot \nabla \phi_m = 0, \quad m = 1, \ldots, M$$

▷ **CISL advection of the Volume fractions and Centroids (see [28, 29, 32, 40, 58]):**

$$F_{m,t} + \nabla \cdot (\boldsymbol{u} F_m) = 0, \quad m = 1, \ldots, M \tag{20}$$

▷ **Coupling the level set functions to the volume fractions and Centroids:** The (continuous) moment-of-fluid reconstructed (see Section 4.2) slope initial guess uses the level set functions, and the level set functions are in turn replaced by the exact signed distance (see Section 4.3) to the continuous moment-of-fluid reconstructed interface.

▷ **Phase change velocity from material m_s (source) to material m_d (destination) (see [58] and [33]):**

$$\boldsymbol{u}^{\text{phase change}} = -\frac{\dot{m}}{\rho_{m_d}}\boldsymbol{n}_{m_d}, \tag{21}$$

▷ **Phase change: Level Set Advection:**

$$\phi_{m,t} + \boldsymbol{u}^{\text{phase change}} \cdot \nabla\phi_m = 0, \quad m = m_s \text{ or } m_d.$$

▷ **Phase change: unsplit CISL advection of the Volume fractions and Centroids (see [58, 33]):**

$$F_{m,t} + \nabla \cdot (\boldsymbol{u}^{\text{phase change}} F_m) = 0, \quad m = m_s \text{ or } m_d.$$

▷ **Phase change: Coupling the level set functions to the volume fractions and Centroids:** The (continuous) moment-of-fluid reconstructed (see Section 4.2) slope initial guess uses the level set functions, and the level set functions are in turn replaced by the exact signed distance (see Section 4.3) to the continuous moment-of-fluid reconstructed interface.

▷ **Phase change: Mass source redistribution:** Redistribute $\dot{m}$ to the source material, m_s, side (e.g. liquid if boiling or freezing) [58].

▷ **Thermal diffusion (see [58] for algorithmic details):**

$$\frac{(\rho C_{p,m})^{\text{mix},n+1}}{\Delta t^{\text{swept}}}(T_m^{n+1} - T_m^{\text{advection}}) = \nabla \cdot (k_m \nabla T_m^{n+1}) \tag{22}$$

▷ **Viscosity (see [40] for algorithmic details):**

$$\frac{\rho^{\text{mix},n+1}}{\Delta t}(\boldsymbol{u}^* - \boldsymbol{u}^{\text{advection}}) = \nabla \cdot (2\mu\mathbb{D}^*) - \rho^{n+1}(\alpha^{n+1}(T^{n+1} - T_0))\boldsymbol{g} \tag{23}$$

▷ **Pressure Gradient (liquid and vapor regions) (see [29, 32, 40, 58] for algorithmic details):**

$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^*}{\Delta t} = -\frac{\nabla p^{n+1}}{\rho^{\text{MAC,mix},n+1}} + \boldsymbol{g} - \frac{\sum_{m=1}^{M} \gamma_m \kappa_m \nabla H(\phi_m)}{\rho^{\text{MAC,mix},n+1}} \tag{24}$$

$$\nabla \cdot \boldsymbol{u}^{n+1} = \sum_{\text{sources}} \frac{\dot{m}_{\text{source}}}{\rho_{\text{source}}} \delta(\phi_{m_{\text{source}}}) - \sum_{\text{sinks}} \frac{\dot{m}_{\text{sink}}}{\rho_{\text{sink}}} \delta(\phi_{m_{\text{sink}}})$$

For the case in which two materials, $m1$ and $m2$, are present in a given 3×3 stencil,

$$\gamma_{m1} = \gamma_{m2} = \frac{\sigma_{m1,m2}}{2}$$

and for the case when three materials, $m1$, $m2$, $m3$, are present in a given 3×3 stencil,

$$\gamma_{m1} = \frac{\sigma_{m1,m2} + \sigma_{m1,m3} - \sigma_{m2,m3}}{2}$$

$$\gamma_{m2} = \frac{\sigma_{m1,m2} + \sigma_{m2,m3} - \sigma_{m1,m3}}{2}$$

$$\gamma_{m3} = \frac{\sigma_{m1,m3} + \sigma_{m2,m3} - \sigma_{m1,m2}}{2}$$

Remarks:

1. For step 2 "phase change" above, we use MOF interface reconstruction. Our rationale is that (1) the phase change velocity $\boldsymbol{u}^{\text{phase change}}$ is not determined from the interface curvature, so that there is no instability risk due to MOF advection here, and (2) MOF is more accurate than CMOF in this scenario.

2. We use the multigrid preconditioned conjugate gradient method [56] to solve the large sparse matrix system that results from discretizing (22). The discretization of the Dirichlet interface temperature condition due to phase change uses the second order method described in [18].

3. We also use the multigrid preconditioned conjugate gradient (MGPCG) method [56] to solve the large sparse matrix system that results from discretizing (24). In order to simulate flows in sealed tanks or flows induced by the sealing of a valve [4] we have developed a general method for enforcing the solvability condition in each "enclosed" region. We identify each "enclosed" region using a shading method developed by [53].

4. In order to preserve mass for phase change problems, we must prescribe a stringent tolerance for the pressure equation (24). In some cases, due to round off error, the MGPCG method might stall. In order to overcome the effect of round-off error, we keep track of a history of the pressure admitting the least residual, and, when the solver has stalled, we restart the MGPCG process using the previous best guess pressure and increasing the number of multigrid relaxation steps by 1.

4.2 MOF and Continuous MOF interface reconstruction methods

The original MOF reconstruction method is local to the cell and uses the reference volume fraction, $F_{\text{ref}} \equiv F^n_{m,i,j}$, and reference centroid, $\boldsymbol{x}^c_{\text{ref}} \equiv \boldsymbol{x}^n_{m,i,j}$ to find the linear (planer) interface reconstruction that has the volume fraction equal to F_{ref}, and has the least amount of error for centroid position. We find

the actual volume fraction $F_{\text{act}}(\boldsymbol{n}, b)$ and centroid $\boldsymbol{x}^c_{\text{act}}(\boldsymbol{n}, b)$ for a reconstructed line(plane) with the normal $\boldsymbol{n}$ and intercept b which minimizes

$$E_{\text{MOF}} = \|\boldsymbol{x}^c_{\text{ref}} - \boldsymbol{x}^c_{\text{act}}(\boldsymbol{n}, b)\|_2 \tag{25}$$

while $F_{\text{act}}(\boldsymbol{n}, b) = F_{\text{ref}}$ (see Figure **??**).

Starting from a whole cell and repeating this process while only considering the uncaptured regions from previous steps, we can reconstruct the material interface for each phase in a multimaterial cell (see [32] for algorithm details). This tessellating procedure generates a volume preserving reconstruction at triple-points (see Figure **??**).

The Continuous Moment-of-fluid method (CMOF) employs a similar procedure to find the interface reconstruction in a cell, but uses a different value for the reference cell centroid $\boldsymbol{x}^c_{\text{ref}}$. We define a super cell

$$\Omega^s_{i,j} = \left\{ \boldsymbol{x} : x \in \left[x_i - \frac{3\Delta x}{2}, x_i + \frac{3\Delta x}{2} \right], \right. \tag{26}$$

$$\left. y \in \left[y_i - \frac{3\Delta y}{2}, y_i + \frac{3\Delta y}{2} \right] \right\} \tag{27}$$

with volume fraction and centroid

$$F^{s,n}_{m,i,j} = \frac{\sum_{i,j} F^n_{m,i,j} V_{i,j}}{\sum_{i,j} V_{i,j}}, \quad \boldsymbol{x}^{c,s,n}_{m,i,j} = \frac{\sum_{i,j} F^n_{m,i,j} \boldsymbol{x}^{c,n}_{m,i,j} V_{i,j}}{\sum_{i,j} F^n_{m,i,j} V_{i,j}}. \tag{28}$$

For the CMOF method we find the slope $\boldsymbol{n}$ and intercept b such that $F_{\text{act}}(\boldsymbol{n}, b) = F_{\text{ref}}$, $E_{\text{MOF}} = \|\boldsymbol{x}^c_{\text{ref}} - \boldsymbol{x}^c_{\text{act}}(\boldsymbol{n}, b)\|_2$ is minimized, while $F_{\text{ref}} \equiv F^n_{m,i,j}$, $\boldsymbol{x}^c_{\text{act}}(\boldsymbol{n}, b)$ is measured relative to the super cell $\Omega^s_{i,j}$, and $\boldsymbol{x}^c_{\text{ref}} \equiv \boldsymbol{x}^{c,s,n}_{m,i,j}$. We refer the reader to Figure **??**.

The initial starting guess for finding the optimal MOF or CMOF slope is determined from the optimal choice from the following (described for the 3D case):

1. $(\Theta^{(1)}, \Phi^{(1)}) = \text{slope to angle}(\frac{\boldsymbol{x}_{ref} - \boldsymbol{x}_{uncapt}}{||\boldsymbol{x}_{ref} - \boldsymbol{x}_{uncapt}||})$

2. (first cut only)

$$(\Theta^{(2)}, \Phi^{(2)}) = \text{slope to angle}(\boldsymbol{n}^{(\text{CLSVOF})}). \tag{29}$$

   $\boldsymbol{n}^{(\text{CLSVOF})}$ is derived from the CLSVOF reconstructed slope [54].

3. (first cut only)
   $(\Theta^{(3)}, \Phi^{(3)})$ is determined by way of the following "regression" decision tree [9] machine learning process (see Figure **??**) :

   (a) at the very start of a simulation, randomly create $n_{sample} = 100^3$ sample triplets for planes cutting through a given cell:

$$(\Theta^{sample}, \Phi^{sample}, F^{sample}).$$

(b) Each sample is associated with $(\Theta^{key}, \Phi^{key}, F^{key})$ which is determined as

$$(\Theta^{key}, \Phi^{key}) = \text{slope to angle}(\frac{\boldsymbol{x}_{ref,sample} - \boldsymbol{x}_{uncapt}}{||\boldsymbol{x}_{ref,sample} - \boldsymbol{x}_{uncapt}||})$$
$$F^{key} = F^{sample}$$

(c) Each associated triplet, $(\Theta^{key}, \Phi^{key}, F^{key})$ with corresponding classification,

$$(\Theta^{classify}, \Phi^{classify}, F^{classify}) \equiv (\Theta^{sample}, \Phi^{sample}, F^{sample})$$

is added to the decision tree list.

(d) After all of the sample data has been classified, the decision tree list is split into two branches, determined by the median in the critical splitting direction (Θ, Φ, or F). See Figure **??**. The critical direction maximizes the decrease in the classification variance between the parent and its associated two branches. (each branch having its own mean).

(e) The tree is split again into 4 branches, with the next critical direction determined to maximize the classification variance reduction. Note: the critical direction is the same for each branch on a given level.

(f) the previous step is repeated recursively until each branch in the tree contains just one piece of data. The number of levels in the tree cannot exceed ceiling($\log_2 n_{sample}$).

(g) for the prediction phase, the decision tree is traversed using the key, $(\Theta^{key}, \Phi^{key}, F^{key})$ in which

$$(\Theta^{key}, \Phi^{key}) = \text{slope to angle}(\frac{\boldsymbol{x}_{ref} - \boldsymbol{x}_{uncapt}}{||\boldsymbol{x}_{ref} - \boldsymbol{x}_{uncapt}||}). \tag{30}$$
$$F^{key} = F_{ref}$$

$(\Theta^{(3)}, \Phi^{(3)})$ is the classification obtained from traversing the decision tree using the key from (30). The maximum number of decisions to be made in order to classify a key not contained in the training data set is ceiling($\log_2 n_{sample}$).

Remarks:

- ▷ In 2D, at the very start of a simulation, one randomly creates $n_{sample} = 100^2$ sample pairs for lines cutting through a given cell: $(\Theta^{sample}, F^{sample})$.
- ▷ In RZ, a 2D decision tree is created corresponding to each discrete value of $r_i = (i + 1/2)\Delta x$.
- ▷ Analytical MOF reconstruction algorithms exist [36], but not analytical CMOF reconstruction algorithms.

- ▷ for the CMOF reconstruction algorithm, $\boldsymbol{x}_{ref}$ and $\boldsymbol{x}_{uncapt}$ are measured relative to the super cell $\Omega^s_{i,j}$ (26). $\boldsymbol{x}_{uncapt}$ is the centroid of the "uncaptured" (i.e. "unreconstructed") region of $\Omega^s_{i,j}$. For the first cut, $\boldsymbol{x}_{uncapt}$ is the centroid of $\Omega^s_{i,j}$ (26).
- ▷ if the super cell, $\Omega^s_{i,j}$, contains at most two materials, then our starting guess is guaranteed to lead to a second order reconstruction, regardless of the number of ensuing optimization iterations. This is because the CLSVOF slope (29) by itself leads to a second order method.
- ▷ We have determined anecdotally that a machine learning sample size of $N^d_{1D} = 100^d$ where d is the dimension of the optimization input data and N_{1D} represents a "sample size per dimension," leads to errors comparable to iterating to convergence. We have arrived at this sample size by studying the N_{1D} dependence of the symmetric difference error for "Zalesaks" problem (see Table **??**).

4.3 Reconstructing the distance function

Here we describe a distance function reconstruction algorithm that evaluate the exact signed distance function to the reconstructed interface.

Assuming that the reconstructed interface is available on the multimaterial cells, using a MOF or CMOF reconstruction, the algorithm first initialize the distance function to a large number with correct sign. Then in a narrow band the cells tagged for involvement in the redistancing procedure, that is, contributing information to and/or getting updated distance function value through this process. In the next steps, distances are evaluated from the center of an update cell to different possible interfacial points in its neighborhood, and material, ϕ_m, and interface distance functions ,$\phi_{m1,m2}$, are updated consequently.

- ▷ Set all distance functions, ϕ_m and $\phi_{m1,m2}$, to a large negative value.
- ▷ Iterate over all cells.
  - ▷ Using the reconstructed interface, change the sign of ϕ_k and $\phi_{k,m2}$ to positive if material k is occupying the center of a cell.
  - ▷ For a cell, check the neighbors for occupying materials in a $3 \times 3 \times 3$ hypercube stencil around it. Count the materials m present if $F_m > 0.5$ in at least one of the cells in the stencil, or if a material level set changes sign between two different cells in the stencil. If more than one material are present in the stencil, this cell is a *support* cell for the redistancing algorithm. Consequently, tag cells in the $9 \times 9 \times 9$ hypercube around a support cell as *update* cells.
- ▷ Iterate over update cells.
  - ▷ Iterate over the support cells in the $3 \times 3 \times 3$ hypercube around this update cell.

- ▷ Evaluate distance to the corners, face centers, and cell center in the support cell. This procedure gives the exact distance if cell boundaries are part of the material interface (Figure **??**.a).
- ▷ Find the normal distance to each interface in the support cell. Update the distance only if the intersection point is within the support cell (Figure **??**.b).
- ▷ Find the intersection of each pair of the interfaces in the support cell, and evaluate the distance to the triple-point. Check the materials around the intersection for updating the corresponding interface distance functions (Figure **??**.c).
- ▷ Find the intersection of the interfaces with cell faces, evaluate the distance, and update the corresponding distance functions (Figure **??**.d).
- ▷ For 3D cases only, find the intersection line of each pair of interfacial planes, and find the intersection of the cell boundaries with the line. Then, evaluate the distance to the intersection points, and update the corresponding distance functions (Figure **??**).

Notes:

- In the redistancing algorithm, after finding the distance to an interface or intersection of interface with cell boundaries or other interfaces the materials around the intersection points are found by testing some points around the intersection to figure out which material they belong to. These points are picked by the combination of interface normal vectors involved in the intersection points and are shown with white diamonds in Figure **??** and Figure **??**.

- Both distance function value and normal vector is updated if a distance measurement is found with smaller magnitude compared to the stored value in ϕ_m and $\phi_{m1,m2}$.

- Applying the algorithm described above, an interface distance function $\phi_{m1,m2}$ is not well defined when the update cell belongs to neither material $m1$ nor material $m2$ (Figure **??**.a). An alternative measurement is used to extend the interface distance functions (Figure **??**.b). For a cell $\{i, j\}$ where $\phi_{m1,i,j} < 0$ and $\phi_{m2,i,j} < 0$ we have:

$$\begin{aligned} \boldsymbol{x}_{m1} &= \boldsymbol{x}_{i,j} - \phi_{m1,i,j}\boldsymbol{n}_{m1,i,j}, & (31)\\ \boldsymbol{x}_{m2} &= \boldsymbol{x}_{i,j} - \phi_{m2,i,j}\boldsymbol{n}_{m2,i,j}, & (32)\\ \boldsymbol{q} &= \boldsymbol{x}_{m1} - \boldsymbol{x}_{m2}. & (33) \end{aligned}$$

$$\begin{cases} \phi_{m1,m2} = |\boldsymbol{q}| & \text{if } |\phi_{m1,i,j}| < |\phi_{m2,i,j}| \\ \phi_{m1,m2} = -|\boldsymbol{q}| & \text{if } |\phi_{m1,i,j}| \geq |\phi_{m2,i,j}| \end{cases} \quad (34)$$

5 Numerical tests and results

References

[1] Ahn, H. T., and Shashkov, M. Multi-material interface reconstruction on generalized polyhedral meshes. *Journal of Computational Physics 226*, 2 (2007), 2096–2132.

[2] Ahn, H. T., and Shashkov, M. Adaptive moment-of-fluid method. *Journal of Computational Physics 228*, 8 (2009), 2792–2821.

[3] Ancellin, M., Després, B., and Jaouen, S. Extension of generic two-component vof interface advection schemes to an arbitrary number of components. *Journal of Computational Physics* (2022), 111721.

[4] Arienti, M., and Sussman, M. An embedded level set method for sharp-interface multiphase simulations of diesel injectors. *International Journal of Multiphase Flow 59* (2014), 1–14.

[5] Asuri Mukundan, A., Ménard, T., Brändle de Motta, J. C., and Berlemont, A. A hybrid moment of fluid–level set framework for simulating primary atomization. *Journal of Computational Physics 451* (2022), 110864.

[6] BENTZ, M., KNOLL, R., Hasan, M., and LIN, C. Low-g fluid mixing-further results from the tank pressure control experiment. In *29th Joint Propulsion Conference and Exhibit* (1993), p. 2423.

[7] Bonhomme, R., Magnaudet, J., Duval, F., and Piar, B. Inertial dynamics of air bubbles crossing a horizontal fluid–fluid interface. *Journal of Fluid Mechanics 707* (2012), 405–443.

[8] Brackbill, J. U., Kothe, D. B., and Zemach, C. A continuum method for modeling surface tension. *Journal of computational physics 100*, 2 (1992), 335–354.

[9] Breiman, L., Friedman, J., Olshen, R., and Stone, C. Classification and regression trees. monterey, ca: wadsworth & brooks/cole advanced books & software, 1984.

[10] Caboussat, A., Francois, M. M., Glowinski, R., Kothe, D. B., and Sicilian, J. M. A numerical method for interface reconstruction of triple points within a volume tracking algorithm. *Mathematical and Computer Modelling 48*, 11 (2008), 1957–1971.

[11] Colella, P., and Woodward, P. R. The piecewise parabolic method (ppm) for gas-dynamical simulations. *Journal of computational physics 54*, 1 (1984), 174–201.

[12] Cummins, S. J., Francois, M. M., and Kothe, D. B. Estimating curvature from volume fractions. *Computers & structures 83*, 6-7 (2005), 425–434.

[13] De Gennes, P.-G., Brochard-Wyart, F., and Quéré, D. *Capillarity and wetting phenomena: drops, bubbles, pearls, waves*. Springer Science & Business Media, 2013.

[14] Dyadechko, V., and Shashkov, M. Moment-of-fluid interface reconstruction. *Los Alamos report LA-UR-05-7571* (2005).

[15] Dyadechko, V., and Shashkov, M. Reconstruction of multi-material interfaces from moment data. *Journal of Computational Physics 227*, 11 (2008), 5361–5384.

[16] Enright, D., Fedkiw, R., Ferziger, J., and Mitchell, I. A hybrid particle level set method for improved interface capturing. *Journal of Computational physics 183*, 1 (2002), 83–116.

[17] Francois, M. M., Cummins, S. J., Dendy, E. D., Kothe, D. B., Sicilian, J. M., and Williams, M. W. A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. *Journal of Computational Physics 213*, 1 (2006), 141–173.

[18] Gibou, F., Fedkiw, R. P., Cheng, L.-T., and Kang, M. A second-order-accurate symmetric discretization of the poisson equation on irregular domains. *Journal of Computational Physics 176*, 1 (2002), 205–227.

[19] Glimm, J., Isaacson, E., Marchesin, D., and McBryan, O. Front tracking for hyperbolic systems. *Advances in Applied Mathematics 2*, 1 (1981), 91–119.

[20] Godunov, S. Ph.d. dissertation: Different methods for shock waves moscow state university, 1954.

[21] Godunov, S. K., and Bohachevsky, I. Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics. *Matematičeskij sbornik 47*, 3 (1959), 271–306.

[22] Harten, A. High resolution schemes for hyperbolic conservation laws. *Journal of computational physics 135*, 2 (1997), 260–278.

[23] Helsby, F., and Tuson, K. Behaviour of air bubbles in aqueous solutions. *Research 8* (1955), 270–275.

[24] Hirt, C. W., and Nichols, B. D. Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of computational physics 39*, 1 (1981), 201–225.

[25] Hu, H., and Jin, Z. An icing physics study by using lifetime-based molecular tagging thermometry technique. *International Journal of Multiphase Flow 36*, 8 (2010), 672–681.

[26] Huang, Z., Lin, G., and Ardekani, A. M. Consistent, essentially conservative and balanced-force phase-field method to model incompressible two-phase flows. *Journal of Computational Physics 406* (2020), 109192.

[27] Huang, Z., Lin, G., and Ardekani, A. M. A consistent and conservative phase-field model for thermo-gas-liquid-solid flows including liquid-solid phase change. *Journal of Computational Physics 449* (2022), 110795.

[28] Jemison, M., Loch, E., Sussman, M., Shashkov, M., Arienti, M., Ohta, M., and Wang, Y. A coupled level set-moment of fluid method for incompressible two-phase flows. *Journal of Scientific Computing 54*, 2-3 (2013), 454–491.

[29] Jemison, M., Sussman, M., and Arienti, M. Compressible, multiphase semi-implicit method with moment of fluid interface representation. *Journal of Computational Physics 279* (2014), 182–217.

[30] Kim, J. Phase field computations for ternary fluid flows. *Computer methods in applied mechanics and engineering 196*, 45 (2007), 4779–4788.

[31] Kucharik, M., Garimella, R. V., Schofield, S. P., and Shashkov, M. J. A comparative study of interface reconstruction methods for multi-material ale simulations. *Journal of Computational Physics 229*, 7 (2010), 2432–2452.

[32] Li, G., Lian, Y., Guo, Y., Jemison, M., Sussman, M., Helms, T., and Arienti, M. Incompressible multiphase flow and encapsulation simulations using the moment-of-fluid method. *International Journal for Numerical Methods in Fluids 79*, 9 (2015), 456–490.

[33] Liu, Y., Sussman, M., Lian, Y., Hussaini, M. Y., Vahab, M., and Shoele, K. A novel supermesh method for computing solutions to the multi-material stefan problem with complex deforming interfaces and microstructure. *Journal of Scientific Computing 91*, 1 (2022), 1–40.

[34] Markstein, G. Interaction of flow pulsations and flame propagation. *Journal of the Aeronautical Sciences 18*, 6 (1951), 428–429.

[35] Miao, F., Wu, B., Sun, Z., and Peng, C. Calibration method of the laser beam based on liquid lens for 3d precise measurement. *Measurement 178* (2021), 109358.

[36] Milcent, T., and Lemoine, A. Moment-of-fluid analytic reconstruction on 3d rectangular hexahedrons. *Journal of Computational Physics 409* (2020), 109346.

[37] Olsson, E., and Kreiss, G. A conservative level set method for two phase flow. *Journal of computational physics 210*, 1 (2005), 225–246.

[38] Osher, S., and Sethian, J. A. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of computational physics 79*, 1 (1988), 12–49.

[39] Pathak, A., and Raessi, M. A three-dimensional volume-of-fluid method for reconstructing and advecting three-material interfaces forming contact lines. *Journal of Computational Physics 307* (2016), 550–573.

[40] Pei, C., Vahab, M., Sussman, M., and Hussaini, M. Y. A hierarchical space-time spectral element and moment-of-fluid method for improved capturing of vortical structures in incompressible multi-phase/multi-material flows. *Journal of Scientific Computing 81*, 3 (2019), 1527–1566.

[41] Qiu, R., Huang, R., Xiao, Y., Wang, J., Zhang, Z., Yue, J., Zeng, Z., and Wang, Y. Physics-informed neural networks for phase-field method in two-phase flow. *Physics of Fluids 34*, 5 (05 2022). 052109.

[42] Salas, M. D. Shock fitting method for complicated two-dimensional supersonic flows. *AIAA Journal 14*, 5 (1976), 583–588.

[43] Saurel, R., and Abgrall, R. A multiphase godunov method for compressible multifluid and multiphase flows. *Journal of Computational Physics 150*, 2 (1999), 425–467.

[44] Schofield, S. P., Garimella, R. V., Francois, M. M., and Loubère, R. Material order-independent interface reconstruction using power diagrams. *International journal for numerical methods in fluids 56*, 6 (2008), 643.

[45] Schofield, S. P., Garimella, R. V., Francois, M. M., and Loubère, R. A second-order accurate material-order-independent interface reconstruction technique for multi-material flow simulations. *Journal of Computational Physics 228*, 3 (2009), 731–745.

[46] Shetabivash, H., Dolatabadi, A., and Paraschivoiu, M. A multiple level-set approach for modelling containerless freezing process. *Journal of Computational Physics 415* (2020), 109527.

[47] Shin, S., and Juric, D. A hybrid interface method for three-dimensional multiphase flows based on front tracking and level set techniques. *International Journal for Numerical Methods in Fluids 60*, 7 (2009), 753–778.

[48] Shu, C.-W., and Osher, S. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of computational physics 77*, 2 (1988), 439–471.

[49] Sijoy, C., and Chaturvedi, S. Volume-of-fluid algorithm with different modified dynamic material ordering methods and their comparisons. *Journal of Computational Physics 229*, 10 (2010), 3848–3863.

[50] Smith, K. A., Solis, F. J., and Chopp, D. A projection method for motion of triple junctions by level sets. *Interfaces and Free Boundaries 4*, 3 (2002), 263–276.

[51] Starinshak, D. P., Karni, S., and Roe, P. L. A new level set model for multimaterial flows. *Journal of Computational Physics 262* (2014), 1–16.

[52] Sussman, M. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. *Journal of Computational Physics 187*, 1 (2003), 110–136.

[53] Sussman, M., and Ohta, M. A stable and efficient method for treating surface tension in incompressible two-phase flow. *SIAM Journal on Scientific Computing 31*, 4 (2009), 2447–2471.

[54] Sussman, M., and Puckett, E. G. A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows. *Journal of Computational Physics 162*, 2 (2000), 301–337.

[55] Sussman, M., Smereka, P., and Osher, S. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational physics 114*, 1 (1994), 146–159.

[56] Tatebe, O. The multigrid preconditioned conjugate gradient method. In *NASA. Langley Research Center, The Sixth Copper Mountain Conference on Multigrid Methods, Part 2* (1993).

[57] Unverdi, S. O., and Tryggvason, G. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of computational physics 100*, 1 (1992), 25–37.

[58] Vahab, M., Sussman, M., and Shoele, K. Fluid-structure interaction of thin flexible bodies in multi-material multi-phase systems. *Journal of Computational Physics 429* (2021), 110008.

[59] Van Leer, B. Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov's method. *Journal of computational Physics 32*, 1 (1979), 101–136.

[60] Vu, T. V., Tryggvason, G., Homma, S., and Wells, J. C. Numerical investigations of drop solidification on a cold plate in the presence of volume change. *International Journal of Multiphase Flow 76* (2015), 73–85.