

# A fully-coupled framework for solving Cahn-Hilliard Navier-Stokes equations: Second-order, energy-stable numerical methods on adaptive octree based meshes

Makrand A. Khanwale<sup>a,b</sup>, Kumar Saurabh<sup>a</sup>, Milinda Fernando<sup>c</sup>, Victor M. Calo<sup>d,e,f</sup>, James A. Rossmanith<sup>b</sup>, Hari Sundar<sup>c</sup>, Baskar Ganapathysubramanian<sup>a,\*</sup>

<sup>a</sup>*Department of Mechanical Engineering, Iowa State University, Iowa, USA 50011*

<sup>b</sup>*Department of Mathematics, Iowa State University, Iowa, USA 50011*

<sup>c</sup>*School of Computing, The University of Utah, Salt Lake City, Utah, USA 84112*

<sup>d</sup>*Department of Applied Geology, Western Australian School of Mines, Faculty of Science and Engineering, Curtin University, Kent Street, Bentley, Perth, WA 6102, Australia*

<sup>e</sup>*Mineral Resources, Commonwealth Scientific and Industrial Research Organization (CSIRO), Kensington, Perth, WA 6152, Australia*

<sup>f</sup>*Curtin Institute for Computation, Curtin University, Kent Street, Bentley, Perth, WA 6102, Australia*

---

## Abstract

We present a fully-coupled, implicit-in-time framework for solving a thermodynamically-consistent Cahn-Hilliard Navier-Stokes system that models two-phase flows. In this work, we extend the block iterative method presented in Khanwale et al. [*Simulating two-phase flows with thermodynamically consistent energy stable Cahn-Hilliard Navier-Stokes equations on parallel adaptive octree based meshes*, J. Comput. Phys. (2020)], to a fully-coupled, provably second-order accurate scheme in time, while maintaining energy-stability. The new method requires fewer matrix assemblies in each Newton iteration resulting in faster solution time. The method is based on a fully-implicit Crank-Nicolson scheme in time and a pressure stabilization for an equal order Galerkin formulation. That is, we use a conforming continuous Galerkin (cG) finite element method in space equipped with a residual-based variational multiscale (RBVMS) procedure to stabilize the pressure. We deploy this approach on a massively parallel numerical implementation using parallel octree-based adaptive meshes. We present comprehensive numerical experiments showing detailed comparisons with results from the literature for canonical cases, including the single bubble rise, Rayleigh-Taylor instability, and lid-driven cavity flow problems. We analyze in detail the scaling of our numerical implementation.

**Keywords:** two-phase flows, energy stable, adaptive finite elements, octrees, variational multiscale approach

---

## 1. Introduction

Understanding the fundamental mechanisms of phase interactions is critical to developing accurate and efficient models of two-phase flows. In particular, insights into the phase interactions may lead to accurate, low-cost coarse-scale models for large systems (e.g., chemical/biological reactors); and optimization-based design of micro-scale systems (e.g., bio-microfluidics and advanced

---

\*Corresponding author

Email addresses: khanwale@iastate.edu (Makrand A. Khanwale), maksbh@iastate.edu (Kumar Saurabh), milinda@cs.utah.edu (Milinda Fernando), Victor.Calo@Curtin.edu.au (Victor M. Calo), rossmani@iastate.edu (James A. Rossmanith), hari@cs.utah.edu (Hari Sundar), baskarg@iastate.edu (Baskar Ganapathysubramanian)

manufacturing using multiphase flows). To achieve this understanding, we need physically accurate models that fully capture the phase interactions through a consistent description of the interfacial processes and the interface evolution. Models using coupled Cahn-Hilliard Navier-Stokes (CHNS) equations bear the promise of providing such a description [1].

Similar to level set methods [2], CHNS models track the boundary between two phases using a smooth function, referred to as the *phase field* function. This approach allows for a diffuse transition between the physical properties from one phase to the other and circumvents modeling the jump discontinuities at the interface. The use of the Cahn-Hilliard equations to track the phase field offers many advantages, including mass conservation, thermodynamic consistency, and a free-energy-based description of surface tension with a well-established theory from non-equilibrium thermodynamics [3, 4]. Carefully designed numerical schemes allow the discrete numerical solutions of these CHNS models to inherit these continuous properties.

CHNS models couple the momentum equation governing an “averaged mixture velocity” with the interface-tracking Cahn-Hilliard equation. Even assuming that each fluid phase is incompressible, the resulting mixture velocity may not be solenoidal; the pointwise incompressibility depends on the averaging. Volume averaging, at least under strictly isothermal conditions, usually results in a solenoidal mixture velocity. Nevertheless, mass averaging results in a non-solenoidal mixture velocity resulting in quasi-incompressible models (e.g., Guo et al. [5], Shokrpour Roudbari et al. [6], and references therein). Thus, we use the volume averaging strategy. The resulting solenoidal mixture velocity is a useful property in the subsequent numerical method development.

In Khanwale et al. [7] we proposed an energy-stable and mass-conserving discretization of the thermodynamically-consistent CHNS model; that approach used a block iterative method for solving the two sets of equations (CH and NS). Therefore, the fully discrete system resulted in two non-linear systems of algebraic equations, one corresponding to the discretized version of the momentum equations, the other corresponding to the discretized Cahn-Hilliard equations. Both sets of equations used an implicit time-stepping strategy alongside an internal (within each block iteration) Newton’s method for solving the resulting non-linear algebraic equations. While this strategy decoupled the implementation challenges of the CH and NS systems, it required multiple matrix assemblies due to the multiple block iterations within each time-step. Xu [8] showed that the matrix assembly and preconditioner setup in a framework such as Khanwale et al. [7] can be very expensive.

In the current work we seek to improve the results of [7] in three key aspects.

1. **Second-order energy-stable scheme:** We extend the time integration scheme presented in [7] to second-order accuracy while maintaining energy-stability and mass conservation. Furthermore, to address computational efficiency concerns, we use a fully-coupled method instead of using a block-iterative approach. The coupling entails solving the fully discretized CHNS system. Therefore, within each Newton iteration, we solve a linear system with six degrees of freedom per node (three velocities, one pressure, two phase field variables).
2. **VMS-based stabilization for conforming Galerkin elements:** We extend the variational multiscale (VMS) based treatment in [7] to the fully-coupled approach with conforming Galerkin elements. We use VMS stabilization to circumvent the discrete inf-sup condition in equal order polynomial representations for velocity and pressure (e.g., Volker [9]). This especially important in the case of adaptive  $h$ -refinement [10–12], which is extensively used in this work.
3. **Scalable octree-based adaptive mesh:** We apply the proposed method to problems where we need sufficient resolution of the interfacial length scales to capture the interface dynamics accurately. To make this computationally tractable, we implement the proposed numerical scheme inside the DENDRO5 [13] adaptive mesh refinement framework, which efficiently re-

55

solves the interface dynamics in 3D systems with highly deformable interfaces. This implementation extends the previous implementation of [7], which was done in the older DENDRO4 [12] framework.

### 1.1. Second-order energy-stable scheme

There are two main approaches for CHNS modelling depending on the averaging used to define the mixture velocity. Examples of CHNS models based on volume-averaged velocity include Kim et al. [14], Feng [15], Shen and Yang [16, 17], Han and Wang [18], Chen and Shen [19], Shen et al. [20], Zhu et al. [21]. Kim et al. [14] used a strategy similar to block iteration, while Feng [15] used a fully-coupled approach similar to one we adopt in this work. Shen and Yang [16, 17] used a block-iteration strategy that reduces their discretization to a sequence of elliptic equations for the velocity and phase fields. Subsequently, Han and Wang [18] also used a block-iterative strategy with an energy-stable time scheme. Chen and Shen [19] showed an energy-stable time scheme with a fully-coupled solver. Guo et al. [5] recently reported a detailed analysis for a mass averaged mixture velocity CHNS system using a fully coupled strategy. Here, in section 3.1, we prove that a second-order extension of the time scheme presented in [7] for the fully-coupled solver is energy-stable and mass conserving. The benefit of such a time integration scheme is that it does not require storage of more than one previous time step, while still providing accuracy and ensuring energy stability.

### 1.2. VMS-based stabilization for conforming Galerkin elements

Discretizing the momentum equations in the CHNS model with solenoidal velocity requires velocity-pressure pairs that satisfy the discrete inf-sup condition (e.g., Ladyzhenskaya-Babuska-Brezzi (LBB) stable). However, we prefer to use standard conforming Galerkin finite elements to leverage parallel adaptive meshing tools. These methods circumvent the discrete inf-sup condition by using stabilization approaches such as *grad-div* stabilizations. We build such a stabilization using a VMS approach [22, 23]. However, the VMS approach as a result of elemental estimates involves the use of an adjustable constant,  $\tau$ , which requires careful design for a fully-coupled system. In this work, we extend the formulation based on the Residual-Based Variational Multiscale Method (RBVMS) [24] we presented in [7] to the fully coupled approach in section 3.2.

### 1.3. Scalable octree-based adaptive mesh generation

Adaptive spatial discretizations are popular in computational sciences [25–27] to improve efficiency and resolution quality. In some applications [28–30], an adaptive spatial discretization is the key to make those simulations feasible on modern supercomputers. In distributed-memory computations, adaptive discretizations introduce additional computational challenges such as load-balancing, low-cost mesh generation, and mesh-partitioning. The scalability of the algorithms used for mesh generation and partitioning is crucial, especially when the represented solution requires frequent re-meshing. Octrees [28, 30–32] are widely used in the community due to their simplicity and their extreme parallel scalability. In [7], we used Dendro4 [12] as the underlying parallel octree library. DENDRO4 is a parallel octree library that supports linear finite element computations on adaptive octrees. DENDRO5 [13] extends DENDRO4 and supports higher-order finite difference (FD), finite volume (FV), and finite element (FE) discretizations on fully adaptive octrees. In the present work, we use DENDRO5 [13] as our primary parallel octree mesh library. From now on, we use DENDRO to refer to DENDRO5, unless otherwise specified.

DENDRO [13] is a freely available open-source library that is currently used by several research communities to tackle problems in computational relativity [30], relativistic fluid dynamics, and other computational science applications. DENDRO octree generation and partitioning is based on

the TREESORT [33] algorithm. Octant neighborhood information is needed to perform numerical computations on topological octrees. To compute these neighborhood data structures, we use the TREESearch algorithm, which has better scalability compared to traditional binary search approaches [30]. DENDRO enforces a 2:1 balancing constraint that imposes that adjacent octants differ by at most a factor of 2 in size. This constraint imposition uses top-down and bottom-up traversals with minor modifications to the TREESORT algorithm. We detail the adaptive meshing and scalability of our framework in section 4 and section 6, respectively.

## 105 2. Governing equations

We consider a bounded domain  $\Omega \subset \mathbb{R}^d$ , for  $d = 2, 3$  containing two immiscible fluids, and the time interval,  $[0, T]$ . Let  $\rho_+(\eta_+)$  and  $\rho_-(\eta_-)$  denote the specific density (viscosity) of the two phases. Let the phase field function,  $\phi$ , be the variable that tracks the location of the phases and varies smoothly between +1 to -1. The non-dimensional density is  $\rho(\phi) = \alpha\phi + \beta$ , where  $\alpha = \frac{\rho_+ - \rho_-}{2\rho_+}$  and  $\beta = \frac{\rho_+ + \rho_-}{2\rho_+}$ ; our non-dimensional form uses the specific density/viscosity of fluid 1 as the non-dimensionalizing density/viscosity. Similarly, the non-dimensional viscosity is  $\eta(\phi) = \gamma\phi + \xi$ , where  $\gamma = \frac{\eta_+ - \eta_-}{2\eta_+}$  and  $\xi = \frac{\eta_+ + \eta_-}{2\eta_+}$ . The governing equations in their non-dimensional form are as follows:

$$\text{Momentum Eqns: } \frac{\partial(\rho(\phi)v_i)}{\partial t} + \frac{\partial(\rho(\phi)v_iv_j)}{\partial x_j} + \frac{1}{Pe} \frac{\partial(J_j v_i)}{\partial x_j} + \frac{Cn}{We} \frac{\partial}{\partial x_j} \left( \frac{\partial\phi}{\partial x_i} \frac{\partial\phi}{\partial x_j} \right) + \frac{1}{We} \frac{\partial p}{\partial x_i} - \frac{1}{Re} \frac{\partial}{\partial x_j} \left( \eta(\phi) \frac{\partial v_i}{\partial x_j} \right) - \frac{\rho(\phi)\hat{g}_i}{Fr} = 0, \quad (1)$$

$$\text{Thermo Consistency: } J_i = \frac{(\rho_- - \rho_+)}{2} m(\phi) \frac{\partial\mu}{\partial x_i}, \quad (2)$$

$$\text{Solenoidality: } \frac{\partial v_i}{\partial x_i} = 0, \quad (3)$$

$$\text{Continuity: } \frac{\partial\rho(\phi)}{\partial t} + \frac{\partial(\rho(\phi)v_i)}{\partial x_i} + \frac{1}{Pe} \frac{\partial J_i}{\partial x_i} = 0, \quad (4)$$

$$\text{Chemical Potential: } \mu = \psi'(\phi) - Cn^2 \frac{\partial}{\partial x_i} \left( \frac{\partial\phi}{\partial x_i} \right), \quad (5)$$

$$\text{Cahn-Hilliard Eqn: } \frac{\partial\phi}{\partial t} + \frac{\partial(v_i\phi)}{\partial x_i} - \frac{1}{PeCn} \frac{\partial}{\partial x_i} \left( \frac{\partial(m(\phi)\mu)}{\partial x_i} \right) = 0. \quad (6)$$

Note that we use Einstein notation throughout this work; in this notation  $v_i$  represents the  $i^{\text{th}}$  component of the vector  $\mathbf{v}$ , and any repeated index is implicitly summed over. In the above equations,  $\mathbf{v}$  is the volume averaged mixture velocity,  $p$  is the volume averaged pressure,  $\phi$  is the phase field (interface tracking variable), and  $\mu$  is the chemical potential. Mobility  $m(\phi)$  is assumed to be a constant with a value of one. The non-dimensional parameters are as follows: Peclet,  $Pe = \frac{u_r L_r^2}{m\sigma}$ ; Reynolds,  $Re = \frac{u_r L_r}{v_r}$ ; Weber,  $We = \frac{\rho_r u_r^2 L_r}{\sigma}$ ; Cahn,  $Cn = \frac{\epsilon}{L_r}$ ; and Froude,  $Fr = \frac{u_r^2}{gL_r}$ , with  $u_r$  and  $L_r$  denoting the reference velocity and length, respectively.  $\hat{\mathbf{g}}$  is a unit vector defined as  $(0, -1, 0)$  denoting the direction of gravity and  $\psi(\phi(\mathbf{x}))$  is a known free-energy function. In particular, we use the polynomial form of the free energy density defined as follows:

$$\psi(\phi) = \frac{1}{4} (\phi^2 - 1)^2 \quad \text{and} \quad \psi'(\phi) = \phi^3 - \phi. \quad (7)$$

The system of equations eq. (1) – eq. (6) has a dissipative law given by:

$$\frac{dE_{\text{tot}}}{dt} = -\frac{1}{Re} \int_{\Omega} \frac{\eta(\phi)}{2} \|\nabla \mathbf{v}\|_F^2 d\mathbf{x} - \frac{Cn}{We} \int_{\Omega} m(\phi) \|\nabla \mu\|^2 d\mathbf{x}, \quad (8)$$

where the total energy is

$$E_{\text{tot}}(\mathbf{v}, \phi, t) = \int_{\Omega} \frac{1}{2} \rho \|\mathbf{v}\|^2 d\mathbf{x} + \frac{1}{CnWe} \int_{\Omega} \left( \psi(\phi) + \frac{Cn^2}{2} \|\nabla \phi\|^2 + \frac{1}{Fr} \rho(\phi) y \right) d\mathbf{x}. \quad (9)$$

The norms used in the above expression are the Euclidean vector norm and the Frobenius matrix norm:

$$\|\mathbf{v}\|^2 := \sum_i |v_i|^2 \quad \text{and} \quad \|\nabla \mathbf{v}\|_F^2 := \sum_i \sum_j \left| \frac{\partial v_i}{\partial x_j} \right|^2. \quad (10)$$

**Remark 1.** A realistic interface thickness (parametrized by the Cahn number) is in the nanometer range. Resolving this scale is computationally intractable, as all the other scales in the problem are orders of magnitude larger. Therefore, an ansatz that diffuse interface models follow is that the solution tends to the real physics in the limit of  $Cn \rightarrow 0$ . This limiting process progressively reduces the Cahn number from large to small until the dynamics become independent of the Cahn number. However, the choice of Cahn number ( $Cn$ ) determines the Peclet number ( $Pe$ ), which is given by  $Pe = \frac{u_r l_r^2}{m \sigma}$ .  $Pe$  represents the ratio of the advection timescale to the diffuse interface relaxation time to its equilibrium tanh profile (a purely computational construct). Magaletti et al. [34] reported a careful asymptotic analysis of these timescales, which suggests a  $1/Pe = \alpha Cn^2$  scaling. We use this scaling with  $\alpha = 3$ .

### 3. Numerical method and its properties

We extend the scheme of Khanwale et al. [7] to a fully-implicit, fully-coupled Crank-Nicolson time-marching scheme for the system of eq. (1) – eq. (6). This extension delivers better accuracy and energy-stability for larger time-steps while only storing data structures of one previous time-step.

Let  $\delta t$  be a time-step; let  $t^k := k\delta t$ ; thus, we define the following time-averages:

$$\tilde{\mathbf{v}}^k := \frac{\mathbf{v}^k + \mathbf{v}^{k+1}}{2}, \quad \tilde{p}^k := \frac{p^{k+1} + p^k}{2}, \quad \tilde{\phi}^k := \frac{\phi^{k+1} + \phi^k}{2}, \quad \text{and} \quad \tilde{\mu}^k := \frac{\mu^{k+1} + \mu^k}{2}, \quad (11)$$

and the following function evaluations:

$$\tilde{\psi}^k := \psi(\tilde{\phi}^k), \quad \tilde{\psi}'^k := \psi'(\tilde{\phi}^k), \quad \tilde{\rho}^k := \rho(\tilde{\phi}^k), \quad \text{and} \quad \tilde{\eta}^k := \eta(\tilde{\phi}^k). \quad (12)$$

Using these temporal values, we define our time-discretized weak form of the Cahn-Hilliard Navier-Stokes (CNHS) equations.

**Definition 1**

Let  $(\cdot, \cdot)$  be the standard  $L^2$  inner product. We state the time-discrete variational problem as follows: find  $\mathbf{v}^{k+1}(\mathbf{x}) \in \mathbf{H}_0^1(\Omega)$ ,  $p^{k+1}(\mathbf{x})$ ,  $\phi^{k+1}(\mathbf{x})$ ,  $\mu^{k+1}(\mathbf{x}) \in H^1(\Omega)$  such that

$$\begin{aligned} \text{Momentum Eqns: } & \left( w_i, \tilde{\rho}^k \frac{v_i^{k+1} - v_i^k}{\delta t} \right) + \left( w_i, \tilde{\rho}^k \tilde{v}_j^k \frac{\partial \tilde{v}_i^k}{\partial x_j} \right) \\ & + \frac{1}{Pe} \left( w_i, \tilde{J}_j^k \frac{\partial \tilde{v}_i^k}{\partial x_j} \right) - \frac{Cn}{We} \left( \frac{\partial w_i}{\partial x_j}, \frac{\partial \tilde{\phi}^k}{\partial x_i} \frac{\partial \tilde{\phi}^k}{\partial x_j} \right) - \frac{1}{We} \left( \frac{\partial w_i}{\partial x_i}, \tilde{p}^k \right) \quad (13) \\ & + \frac{1}{Re} \left( \frac{\partial w_i}{\partial x_j}, \tilde{\eta}^k \frac{\partial \tilde{v}_i^k}{\partial x_j} \right) - \left( w_i, \frac{\tilde{\rho}^k \hat{g}_i}{Fr} \right) = 0, \end{aligned}$$

$$\text{Thermo Consistency: } \tilde{J}_i^k = \frac{(\rho_- - \rho_+)}{2} \frac{\partial \tilde{\mu}^k}{\partial x_i}, \quad (14)$$

$$\text{Solenoidality: } \left( q, \frac{\partial v_i^k}{\partial x_i} \right) = 0, \quad \left( q, \frac{\partial v_i^{k+1}}{\partial x_i} \right) = 0, \quad (15)$$

$$\text{Chemical Potential: } - \left( q, \tilde{\mu}^k \right) + \left( q, \tilde{\psi}'^k \right) + Cn^2 \left( \frac{\partial q}{\partial x_i}, \frac{\partial \tilde{\phi}^k}{\partial x_i} \right) = 0, \quad (16)$$

$$\text{Cahn-Hilliard Eqn: } \left( q, \frac{\phi^{k+1} - \phi^k}{\delta t} \right) - \left( \frac{\partial q}{\partial x_i}, \tilde{v}_i^k \tilde{\phi}^k \right) + \frac{1}{PeCn} \left( \frac{\partial q}{\partial x_i}, \frac{\partial \tilde{\mu}^k}{\partial x_i} \right) = 0, \quad (17)$$

$\forall \mathbf{w} \in \mathbf{H}_0^1(\Omega)$ ,  $\forall q \in H^1(\Omega)$ , given  $\mathbf{v}^k \in \mathbf{H}_0^1(\Omega)$ , and  $\phi^k, \mu^k \in H^1(\Omega)$ .

**Remark 2.** While  $\phi \in [-1, 1]$  in the continuous equations, the discrete  $\phi$  may violate these bounds. These bound violations may not change the dynamics of  $\phi$  adversely, but they could lose the strict positivity of some quantities which depend on  $\phi$  (e.g., mixture density  $\rho(\phi)$  and viscosity  $\eta(\phi)$ ). This effect is especially significant for high density and viscosity contrasts. We fix this issue by saturation scaling (i.e., we pull back the value of  $\phi$  only for the calculation of density and viscosity). We, therefore, define  $\phi^*$  for the mixture density and viscosity calculations, where  $\phi^*$  is:

$$\phi^* := \begin{cases} \phi, & \text{if } |\phi| \leq 1, \\ \text{sign}(\phi), & \text{otherwise.} \end{cases} \quad (18)$$

130

### 3.1. Energy-stability of the time-stepping scheme

In this subsection, we prove the energy-stability of the time time-stepping scheme. For completeness, we recall some of the crucial results from [7]. We begin with mass conservation.

**Proposition 1: Mass conservation**

The scheme of eq. (13) – eq. (17) with the following boundary conditions:

$$\frac{\partial \tilde{\mu}}{\partial x_i} \hat{n}_i \Big|_{\partial\Omega} = 0, \quad \frac{\partial \tilde{\phi}}{\partial x_i} \hat{n}_i \Big|_{\partial\Omega} = 0, \quad \tilde{\mathbf{v}}^k \Big|_{\partial\Omega} = \mathbf{0}, \quad (19)$$

where  $\hat{\mathbf{n}}$  is the outward normal to the boundary  $\partial\Omega$ , is globally mass conservative:

$$\int_{\Omega} \phi^{k+1} d\mathbf{x} = \int_{\Omega} \phi^k d\mathbf{x}. \quad (20)$$

We verify this claim numerically in sections 5.1 and 5.2.

### Lemma 1: Weak equivalence of forcing

The forcing term due to Cahn-Hilliard in the momentum equation, eq. (13), with the test function  $w_i = \delta t \tilde{v}_i^k$ , becomes

$$\frac{Cn}{We} \left( \frac{\partial}{\partial x_j} \left( \frac{\partial \tilde{\phi}^k}{\partial x_i} \frac{\partial \tilde{\phi}^k}{\partial x_j} \right), \delta t \tilde{v}_i^k \right) = \frac{\delta t}{WeCn} \left( \tilde{\phi}^k \frac{\partial \tilde{\mu}^k}{\partial x_i}, \tilde{v}_i^k \right), \quad (21)$$

$\forall \tilde{\phi}^k, \tilde{\mu}^k \in H^1(\Omega)$ , and  $\forall \tilde{\mathbf{v}}^k \in \mathbf{H}_0^1(\Omega)$ , where  $\mathbf{v}^k, \mathbf{v}^{k+1}, p^k, p^{k+1}, \phi^k, \phi^{k+1}, \mu^k, \mu^{k+1}$ , satisfy eq. (13) – eq. (17).

### Lemma 2

The following identity holds:

$$\left( \tilde{\psi}'^k, \phi^{k+1} - \phi^k \right) = \left( \psi(\phi^{k+1}) - \psi(\phi^k), 1 \right) - \left( \frac{\psi'''(\lambda)}{24}, (\phi^{k+1} - \phi^k)^3 \right), \quad (22)$$

for some  $\lambda$  between  $\phi^k$  and  $\phi^{k+1}$ .

### Lemma 3

The following estimate holds:

$$\left| \left( \frac{\psi'''(\lambda)}{24}, (\phi^{k+1} - \phi^k)^3 \right) \right| \leq C_m L^3 \delta t^3 \left\| \frac{\psi'''(\lambda)}{24} \right\|_{L^\infty(\Omega)}, \quad (23)$$

where  $L$  is a Lipschitz constant and  $C_m$  is the volume of the physical domain:

$$|\phi^{k+1} - \phi^k| \leq L \delta t \quad \text{and} \quad C_m := \int_{\Omega} d\mathbf{x}.$$

A vital difference between the time integration schemes in [7] and this work is the evaluation of mixture density and viscosity at the average of  $\phi$  ( $\tilde{\phi}^k$ ). To analyze the scheme accordingly, we present the following results.

**Lemma 4**

The variational advection term from the Cahn-Hilliard contribution in the momentum equation, eq. (13), can be written as follows:

$$\left( \tilde{\rho}^k \left( v_i^{k+1} - v_i^k \right), \tilde{v}_i^k \right) = \frac{1}{2} \int_{\Omega} \left[ \rho \left( \phi^{k+1} \right) \|\mathbf{v}^{k+1}\|^2 - \rho \left( \phi^k \right) \|\mathbf{v}^k\|^2 \right] d\mathbf{x} \quad (24)$$

$\forall \tilde{\phi}^k, \phi^k, \phi^{k+1} \in H^1(\Omega)$ , and  $\forall \mathbf{v}^k, \mathbf{v}^{k+1} \in \mathbf{H}_0^1(\Omega)$ , where  $\mathbf{v}^k, \mathbf{v}^{k+1}, \phi^k, \phi^{k+1}$  satisfy eq. (13) – eq. (17), and

$$\|\mathbf{v}\|^2 := \sum_i |v_i|^2, \quad (25)$$

*Proof.* We start with the left-hand side of eq. (24):

$$\begin{aligned} \left( \tilde{\rho}^k \left( v_i^{k+1} - v_i^k \right), \tilde{v}_i^k \right) &= \frac{1}{2} \int_{\Omega} \left[ \tilde{\rho}^k \|\mathbf{v}^{k+1}\|^2 - \tilde{\rho}^k \|\mathbf{v}^k\|^2 \right] d\mathbf{x} \\ &= \frac{1}{4} \int_{\Omega} \left[ \rho \left( \phi^{k+1} \right) \|\mathbf{v}^{k+1}\|^2 - \rho \left( \phi^{k+1} \right) \|\mathbf{v}^k\|^2 \right] d\mathbf{x} \\ &\quad + \frac{1}{4} \int_{\Omega} \left[ \rho \left( \phi^k \right) \|\mathbf{v}^{k+1}\|^2 - \rho \left( \phi^k \right) \|\mathbf{v}^k\|^2 \right] d\mathbf{x}, \end{aligned} \quad (26)$$

where we use the definitions eq. (11) and eq. (12) and the fact that  $\rho$  is an affine function of  $\phi$ . Continuing the algebraic manipulations we obtain:

$$\begin{aligned} \left( \tilde{\rho}^k \left( v_i^{k+1} - v_i^k \right), \tilde{v}_i^k \right) &= \frac{1}{4} \int_{\Omega} \left[ \rho \left( \phi^{k+1} \right) \|\mathbf{v}^{k+1}\|^2 - \rho \left( \phi^{k+1} \right) \|\mathbf{v}^k\|^2 \right] d\mathbf{x} \\ &\quad + \frac{1}{4} \int_{\Omega} \left[ \rho \left( \phi^k \right) \|\mathbf{v}^{k+1}\|^2 - \rho \left( \phi^k \right) \|\mathbf{v}^k\|^2 \right] d\mathbf{x} \\ &\quad + \frac{1}{4} \int_{\Omega} \left[ \rho \left( \phi^k \right) \|\mathbf{v}^k\|^2 - \rho \left( \phi^k \right) \|\mathbf{v}^k\|^2 \right] d\mathbf{x} \\ &\quad + \frac{1}{4} \int_{\Omega} \left[ \rho \left( \phi^{k+1} \right) \|\mathbf{v}^{k+1}\|^2 - \rho \left( \phi^{k+1} \right) \|\mathbf{v}^{k+1}\|^2 \right] d\mathbf{x}, \end{aligned} \quad (27)$$

$$\begin{aligned} \implies \left( \tilde{\rho}^k \left( v_i^{k+1} - v_i^k \right), \tilde{v}_i^k \right) &= \frac{1}{2} \int_{\Omega} \left[ \rho \left( \phi^{k+1} \right) \|\mathbf{v}^{k+1}\|^2 - \rho \left( \phi^k \right) \|\mathbf{v}^k\|^2 \right] d\mathbf{x} \\ &\quad - \frac{1}{4} \int_{\Omega} \left[ \rho \left( \phi^{k+1} \right) - \rho \left( \phi^k \right) \right] \|\mathbf{v}^k\|^2 d\mathbf{x} \\ &\quad - \frac{1}{4} \int_{\Omega} \left[ \rho \left( \phi^{k+1} \right) - \rho \left( \phi^k \right) \right] \|\mathbf{v}^{k+1}\|^2 d\mathbf{x}. \end{aligned} \quad (28)$$

The last two terms vanish using eq. (14) and eq. (17), the fact that  $\rho(\phi)$  is affine in  $\phi$ , and the trilinear identities eq. (A.5) and eq. (A.6):

$$\begin{aligned} \int_{\Omega} \left[ \rho \left( \phi^{k+1} \right) - \rho \left( \phi^k \right) \right] \|\mathbf{v}\|^2 d\mathbf{x} &= \delta t \left( \frac{\partial (v_j v_j)}{\partial x_i}, \tilde{\rho}^k \tilde{v}_i^k \right) + \frac{\delta t}{Pe} \left( \frac{\partial (v_j v_j)}{\partial x_i}, \tilde{J}_i^k \right) \\ &= 2\delta t \left( v_j \frac{\partial v_i}{\partial x_j}, \tilde{\rho}^k \tilde{v}_i^k \right) + \frac{2\delta t}{Pe} \left( v_j \frac{\partial v_i}{\partial x_j}, \tilde{J}_i^k \right) = 0. \end{aligned} \quad (29)$$

This yields the desired result. ■

**Lemma 5**

The variational advection term from the Cahn-Hilliard contribution in the momentum equation, eq. (13), becomes:

$$\begin{aligned} \frac{\delta t}{WeCn} \left( \tilde{\phi}^k \tilde{v}_i^k, \frac{\partial \tilde{\mu}^k}{\partial x_i} \right) &= -\frac{1}{2} \int_{\Omega} \left[ \rho(\phi^{k+1}) \|\mathbf{v}^{k+1}\|^2 - \rho(\phi^k) \|\mathbf{v}^k\|^2 \right] d\mathbf{x} \\ &\quad - \frac{\delta t}{Re} \left\| \sqrt{\tilde{\eta}^k} \nabla \tilde{\mathbf{v}}^k \right\|_{L^2}^2 - \frac{1}{Fr} \left( y, \rho(\phi^{k+1}) - \rho(\phi^k) \right), \end{aligned} \quad (30)$$

$\forall \tilde{\phi}^k, \phi^{k+1}, \tilde{\mu}^k \in H^1(\Omega)$ , and  $\forall \mathbf{v}^k, \mathbf{v}^{k+1} \in \mathbf{H}_0^1(\Omega)$ , where  $\mathbf{v}^k, \mathbf{v}^{k+1}, p^k, p^{k+1}, \phi^k, \phi^{k+1}, \mu^k, \mu^{k+1}$  satisfy eq. (13) – eq. (17), and

$$\begin{aligned} \|\mathbf{v}\|^2 &:= \sum_i |v_i|^2, \\ \left\| \sqrt{\tilde{\eta}^k} \nabla \tilde{\mathbf{v}}^k \right\|_{L^2}^2 &:= \int_{\Omega} \sqrt{\tilde{\eta}^k} \sum_i \sum_j \left| \frac{\partial \tilde{v}_i^k}{\partial x_j} \right|^2 d\mathbf{x} = \int_{\Omega} \sqrt{\tilde{\eta}^k} \|\nabla \tilde{\mathbf{v}}^k\|_F^2 d\mathbf{x}. \end{aligned} \quad (31)$$

145

*Proof.* The structure of this proof is similar to one presented in Khanwale et al. [7] with some changes. However, for completeness we produce the proof below. We start with momentum equation (13) with the test function  $w_i = \delta t \tilde{v}_i^k$ :

$$\begin{aligned} &\left( \tilde{\rho}^k \frac{v_i^{k+1} - v_i^k}{\delta t}, \delta t \tilde{v}_i^k \right) + \left( \tilde{\rho}^k \tilde{v}_j \frac{\partial \tilde{v}_i^k}{\partial x_j}, \delta t \tilde{v}_i^k \right) + \frac{1}{Pe} \left( \tilde{J}_j^k \frac{\partial \tilde{v}_i^k}{\partial x_j}, \delta t \tilde{v}_i^k \right) - \frac{1}{Fr} \left( \tilde{\rho}^k \hat{g}_i, \delta t \tilde{v}_i^k \right) \\ &+ \frac{Cn}{We} \left( \frac{\partial}{\partial x_j} \left( \frac{\partial \tilde{\phi}^k}{\partial x_i} \frac{\partial \tilde{\phi}^k}{\partial x_j} \right), \delta t \tilde{v}_i^k \right) + \frac{1}{We} \left( \frac{\partial \tilde{p}^k}{\partial x_i}, \delta t \tilde{v}_i^k \right) - \frac{1}{Re} \left( \frac{\partial}{\partial x_j} \left( \tilde{\eta}^k \frac{\partial \tilde{v}_i^k}{\partial x_j} \right), \delta t \tilde{v}_i^k \right) = 0. \end{aligned} \quad (32)$$

The second and third terms are in a trilinear form so from eq. (A.5) and eq. (A.6) they go to zero and we have:

$$\begin{aligned} &\frac{1}{2} \int_{\Omega} \left[ \tilde{\rho}^k \|\mathbf{v}^{k+1}\|^2 - \tilde{\rho}^k \|\mathbf{v}^k\|^2 \right] d\mathbf{x} + \frac{Cn}{We} \left( \frac{\partial}{\partial x_j} \left( \frac{\partial \tilde{\phi}^k}{\partial x_i} \frac{\partial \tilde{\phi}^k}{\partial x_j} \right), \delta t \tilde{v}_i^k \right) \\ &+ \frac{1}{We} \left( \frac{\partial \tilde{p}^k}{\partial x_i}, \delta t \tilde{v}_i^k \right) - \frac{1}{Re} \left( \frac{\partial}{\partial x_j} \left( \tilde{\eta}^k \frac{\partial \tilde{v}_i^k}{\partial x_j} \right), \delta t \tilde{v}_i^k \right) - \frac{1}{Fr} \left( \hat{g}_i, \delta t \tilde{\rho}^k \tilde{v}_i^k \right) = 0, \end{aligned} \quad (33)$$

where we made use of the fact that  $\tilde{v}_i^k = (v_i^{k+1} + v_i^k)/2$  and subsequently lemma 4. We can now use solenoidality of the velocity field to get rid of the pressure term. We can do this by integrating-by-parts on the pressure term:

$$\begin{aligned} &\frac{1}{2} \int_{\Omega} \left[ \rho(\phi^{k+1}) \|\mathbf{v}^{k+1}\|^2 - \rho(\phi^k) \|\mathbf{v}^k\|^2 \right] d\mathbf{x} + \frac{Cn}{We} \left( \frac{\partial}{\partial x_j} \left( \frac{\partial \tilde{\phi}^k}{\partial x_i} \frac{\partial \tilde{\phi}^k}{\partial x_j} \right), \delta t \tilde{v}_i^k \right) \\ &- \frac{\delta t}{We} \left( \tilde{p}^k, \frac{\partial \tilde{v}_i^k}{\partial x_i} \right) - \frac{1}{Re} \left( \frac{\partial}{\partial x_j} \left( \tilde{\eta}^k \frac{\partial \tilde{v}_i^k}{\partial x_j} \right), \delta t \tilde{v}_i^k \right) - \frac{1}{Fr} \left( \hat{g}_i, \delta t \tilde{\rho}^k \tilde{v}_i^k \right) = 0, \end{aligned} \quad (34)$$

$$\implies \frac{1}{2} \int_{\Omega} \left[ \rho(\phi^{k+1}) \|\mathbf{v}^{k+1}\|^2 - \rho(\phi^k) \|\mathbf{v}^k\|^2 \right] d\mathbf{x} + \frac{Cn}{We} \left( \frac{\partial}{\partial x_j} \left( \frac{\partial \tilde{\phi}^k}{\partial x_i} \frac{\partial \tilde{\phi}^k}{\partial x_j} \right), \delta t \tilde{v}_i^k \right) \quad (35)$$

$$- \frac{1}{Re} \left( \frac{\partial}{\partial x_j} \left( \tilde{\eta}^k \frac{\partial \tilde{v}_i^k}{\partial x_j} \right), \delta t \tilde{v}_i^k \right) - \frac{1}{Fr} \left( \hat{g}_i, \delta t \tilde{\rho}^k \tilde{v}_i^k \right) = 0,$$

$$\implies \frac{1}{2} \int_{\Omega} \left[ \rho(\phi^{k+1}) \|\mathbf{v}^{k+1}\|^2 - \rho(\phi^k) \|\mathbf{v}^k\|^2 \right] d\mathbf{x} + \frac{Cn}{We} \left( \frac{\partial}{\partial x_j} \left( \frac{\partial \tilde{\phi}^k}{\partial x_i} \frac{\partial \tilde{\phi}^k}{\partial x_j} \right), \delta t \tilde{v}_i^k \right) \quad (36)$$

$$+ \frac{\delta t}{Re} \left( \sqrt{\tilde{\eta}^k} \frac{\partial \tilde{v}_i^k}{\partial x_j}, \sqrt{\tilde{\eta}^k} \frac{\partial \tilde{v}_i^k}{\partial x_j} \right) - \frac{1}{Fr} \left( \hat{g}_i, \delta t \tilde{\rho}^k \tilde{v}_i^k \right) = 0,$$

$$\implies \frac{1}{2} \int_{\Omega} \left[ \rho(\phi^{k+1}) \|\mathbf{v}^{k+1}\|^2 - \rho(\phi^k) \|\mathbf{v}^k\|^2 \right] d\mathbf{x} + \frac{Cn}{We} \left( \frac{\partial}{\partial x_j} \left( \frac{\partial \tilde{\phi}^k}{\partial x_i} \frac{\partial \tilde{\phi}^k}{\partial x_j} \right), \delta t \tilde{v}_i^k \right) \quad (37)$$

$$+ \frac{\delta t}{Re} \left\| \sqrt{\tilde{\eta}^k} \nabla \mathbf{v}^k \right\|_{L^2}^2 - \frac{1}{Fr} \left( \hat{g}_i, \delta t \tilde{\rho}^k \tilde{v}_i^k \right) = 0.$$

Next, we invoke lemma 1 and write eq. (37) as:

$$\frac{1}{2} \int_{\Omega} \left[ \rho(\phi^{k+1}) \|\mathbf{v}^{k+1}\|^2 - \rho(\phi^k) \|\mathbf{v}^k\|^2 \right] d\mathbf{x} + \frac{\delta t}{We Cn} \left( \tilde{\phi}^k \tilde{v}_i^k, \frac{\partial \tilde{\mu}^k}{\partial x_i} \right) + \frac{\delta t}{Re} \left\| \sqrt{\tilde{\eta}^k} \nabla \mathbf{v}^k \right\|_{L^2}^2 \quad (38)$$

$$- \frac{1}{Fr} \left( \hat{g}_i, \delta t \tilde{\rho}^k \tilde{v}_i^k \right) = 0.$$

Next we simplify the gravity term noting that

$$- \frac{1}{Fr} \left( \hat{g}_i, \delta t \tilde{\rho}^k \tilde{v}_i^k \right) = - \frac{1}{Fr} \left( \frac{\partial(-y)}{\partial x_i}, \delta t \tilde{\rho}^k \tilde{v}_i^k \right) = - \frac{1}{Fr} \left( y, \delta t \frac{\partial(\tilde{\rho}^k \tilde{v}_i^k)}{\partial x_i} \right), \quad (39)$$

where  $y = x_2$  and  $\hat{\mathbf{g}} = (0, -1, 0)$ . Note that the boundary terms vanish in the process of integrating-by-parts due to the fact that  $\tilde{\mathbf{v}}^{k+1} \in \mathbf{H}_0^1(\Omega)$ . Let  $C_1 = \frac{(\rho_- - \rho_+)}{2} m(\phi)$ , then using eq. (14), the fact that  $\rho$  is affine in  $\phi$ , eq. (17), and eq. (39) we obtain:

$$\begin{aligned} - \frac{1}{Fr} \left( \hat{g}_i, \delta t \tilde{\rho}^k \tilde{v}_i^k \right) &= - \frac{1}{Fr} \left( y, \rho(\phi^{k+1}) - \rho(\phi^k) \right) - \frac{\delta t C_1}{Fr Pe} \left( y, \frac{\partial}{\partial x_i} \left( \frac{\partial \tilde{\mu}^k}{\partial x_i} \right) \right) \\ &= - \frac{1}{Fr} \left( y, \rho(\phi^{k+1}) - \rho(\phi^k) \right) + \frac{\delta t C_1}{Fr Pe} \left( \frac{\partial y}{\partial x_i}, \frac{\partial \tilde{\mu}^k}{\partial x_i} \right) \\ &= - \frac{1}{Fr} \left( y, \rho(\phi^{k+1}) - \rho(\phi^k) \right) - \frac{\delta t C_1}{Fr Pe} \left( \frac{\partial}{\partial x_i} \left( \frac{\partial y}{\partial x_i} \right), \tilde{\mu}^k \right) + \frac{\delta t C_1}{Fr Pe} \int_{\partial\Omega} \tilde{\mu}^k \left( \frac{\partial y}{\partial x_i} \right) \hat{n}_i d\mathbf{x} \quad (40) \\ &= - \frac{1}{Fr} \left( y, \rho(\phi^{k+1}) - \rho(\phi^k) \right) + \frac{\delta t C_1}{Fr Pe} \int_{\partial\Omega} \tilde{\mu}^k \hat{g}_i \hat{n}_i d\mathbf{x} \\ &= - \frac{1}{Fr} \left( y, \rho(\phi^{k+1}) - \rho(\phi^k) \right), \end{aligned}$$

where  $\hat{n}_i$  is outward normal to the boundary of the domain  $\Omega$ .

**Remark 3.** In the last line of (40) we asserted that

$$\frac{\delta t C_1}{Fr Pe} \int_{\Omega} \tilde{\mu}^k \hat{g}_i \hat{n}_i d\mathbf{x} = 0. \quad (41)$$

This is true as long as there is no three-phase contact line on any boundary on which  $\hat{n}_i \hat{g}_i$  is non-zero. For the purpose of the analysis presented here, we will assume that this is true.

Combining (40) with eq. (38) yields the desired result:

$$\begin{aligned} \frac{1}{2} \int_{\Omega} [\rho(\phi^{k+1}) \|\mathbf{v}^{k+1}\|^2 - \rho(\phi^k) \|\mathbf{v}^k\|^2] d\mathbf{x} + \frac{\delta t}{WeCn} \left( \tilde{\phi}^k \tilde{v}_i^k, \frac{\partial \tilde{\mu}^k}{\partial x_i} \right) + \frac{\delta t}{Re} \left\| \sqrt{\tilde{\eta}^k} \nabla \tilde{\mathbf{v}}^k \right\|_{L^2}^2 \\ + \frac{1}{Fr} (y, \rho(\phi^{k+1}) - \rho(\phi^k)) = 0. \end{aligned} \quad (42)$$

■

We now have all the ingredients to prove energy-stability. Our argument uses the fact that the energy functional eq. (9) is decreasing as the discrete solution is evolving in time. At the semi-discrete level, the successive decrease of the energy functional for each time step represents adherence to the second law of thermodynamics. We prove energy-stability in the following theorem.

**Theorem 1: energy-stability**

The time discretization of the Cahn-Hilliard Navier-Stokes (CHNS) equations as described by eq. (13) – eq. (17) is energy-stable and follows the following energy law:

$$\begin{aligned} E_{tot}(\mathbf{v}^{k+1}, \phi^{k+1}) - E_{tot}(\mathbf{v}^k, \phi^k) &= \frac{-\delta t}{Re} \left\| \sqrt{\tilde{\eta}^k} \tilde{\mathbf{v}}^k \right\|_{L^2}^2 - \frac{\delta t}{PeCn^2We} \|\nabla \tilde{\mu}^k\|_{L^2}^2 \\ &\quad + \frac{1}{WeCn} \left( \frac{\psi'''(\lambda)}{24}, (\phi^{k+1} - \phi^k)^3 \right), \end{aligned} \quad (43)$$

provided the following time-step restriction is observed:

$$0 \leq \delta t \leq \left( \frac{\frac{1}{Re} \left( \left\| \sqrt{\tilde{\eta}^k} \nabla \tilde{\mathbf{v}}^k \right\|_{L^2}^2 \right) + \frac{1}{PeCn^2We} \|\nabla \tilde{\mu}^k\|_{L^2}^2}{\frac{C_m L^3}{WeCn} \left\| \frac{\psi'''(\lambda)}{24} \right\|_{L^\infty(\Omega)}} \right)^{\frac{1}{2}}. \quad (44)$$

*Proof.* The proof uses  $L^2$  estimates of the semi-discrete equations to estimate of the energy change between two-time steps. If the estimate is strictly negative, we have an energy-stable scheme. We begin with eq. (17) with the test function  $q = \delta t \tilde{\mu}^k$ :

$$(\phi^{k+1} - \phi^k, \tilde{\mu}^k) = \left( \tilde{v}_i^k \tilde{\phi}^k, \delta t \frac{\partial \tilde{\mu}^k}{\partial x_i} \right) - \frac{\delta t}{PeCn} \|\nabla \tilde{\mu}^k\|_{L^2}^2. \quad (45)$$

Next, we take eq. (16) with test function  $q = \phi^{k+1} - \phi^k$ , where eq. (12) defines  $\tilde{\psi}'$ :

$$(\tilde{\mu}^k, \phi^{k+1} - \phi^k) = (\tilde{\psi}', \phi^{k+1} - \phi^k) + \frac{Cn^2}{2} \left( \|\nabla \phi^{k+1}\|_{L^2}^2 - \|\nabla \phi^k\|_{L^2}^2 \right), \quad (46)$$

where we also use the fact that  $\tilde{\phi}^k = (\phi^{k+1} + \phi^k)/2$ . The first term on right-hand side of eq. (46)

simplifies further using lemma 2:

$$\begin{aligned} \left( \tilde{\mu}^k, \phi^{k+1} - \phi^k \right) &= \left( \psi(\phi^{k+1}) - \psi(\phi^k), 1 \right) - \left( \frac{\psi'''(\lambda)}{24}, (\phi^{k+1} - \phi^k)^3 \right) \\ &\quad + \frac{Cn^2}{2} \left( \|\nabla \phi^{k+1}\|_{L^2}^2 - \|\nabla \phi^k\|_{L^2}^2 \right). \end{aligned} \quad (47)$$

Now, combining eq. (47) and eq. (45), we have:

$$\begin{aligned} \left( \psi(\phi^{k+1}) - \psi(\phi^k), 1 \right) - \left( \frac{\psi'''(\lambda)}{24}, (\phi^{k+1} - \phi^k)^3 \right) + \frac{Cn^2}{2} \left( \|\nabla \phi^{k+1}\|_{L^2}^2 - \|\nabla \phi^k\|_{L^2}^2 \right) \\ = \left( \tilde{v}_i^k \tilde{\phi}^k, \delta t \frac{\partial \tilde{\mu}^k}{\partial x_i} \right) - \frac{\delta t}{PeCn} \|\nabla \tilde{\mu}^k\|_{L^2}^2. \end{aligned} \quad (48)$$

Next, we divide eq. (48) by  $WeCn$  and from lemma 5, we can substitute the first term on the right-hand side by eq. (30):

$$\begin{aligned} \frac{1}{2} \int_{\Omega} \left[ \rho(\phi^{k+1}) \|\mathbf{v}^{k+1}\|^2 - \rho(\phi^k) \|\mathbf{v}^k\|^2 \right] d\mathbf{x} + \frac{\delta t}{Re} \left( \left\| \sqrt{\tilde{\eta}^k} \nabla \tilde{\mathbf{v}}^k \right\|_{L^2}^2 \right) \\ + \frac{1}{WeCn} \left( \psi(\phi^{k+1}) - \psi(\phi^k), 1 \right) - \frac{1}{WeCn} \left( \frac{\psi'''(\lambda)}{24}, (\phi^{k+1} - \phi^k)^3 \right) \\ + \frac{Cn}{2We} \left( \|\nabla \phi^{k+1}\|_{L^2}^2 - \|\nabla \phi^k\|_{L^2}^2 \right) + \frac{\delta t}{PeCn^2 We} \|\nabla \tilde{\mu}^k\|_{L^2}^2 \\ + \frac{1}{Fr} \left( y, \rho(\phi^{k+1}) - \rho(\phi^k) \right) = 0. \end{aligned} \quad (49)$$

Simplifying and using the definition of the energy functional, eq. (9), we obtain the energy law:

$$\begin{aligned} E_{\text{tot}}(\mathbf{v}^{k+1}, \phi^{k+1}) - E_{\text{tot}}(\mathbf{v}^k, \phi^k) &= \frac{-\delta t}{Re} \left\| \sqrt{\tilde{\eta}^k} \nabla \tilde{\mathbf{v}}^k \right\|_{L^2}^2 - \frac{\delta t}{PeCn^2 We} \|\nabla \tilde{\mu}^k\|_{L^2}^2 \\ &\quad + \frac{1}{WeCn} \left( \frac{\psi'''(\lambda)}{24}, (\phi^{k+1} - \phi^k)^3 \right). \end{aligned} \quad (50)$$

In order for this energy to be non-increasing in time, we require the following:

$$\frac{\delta t}{Re} \left\| \sqrt{\tilde{\eta}^k} \nabla \tilde{\mathbf{v}}^k \right\|_{L^2}^2 + \frac{\delta t}{PeCn^2 We} \|\nabla \tilde{\mu}^k\|_{L^2}^2 \geq \frac{1}{WeCn} \left( \frac{\psi'''(\lambda)}{24}, (\phi^{k+1} - \phi^k)^3 \right). \quad (51)$$

Using the estimate from lemma 3, we can guarantee this inequality provided that:

$$\frac{\delta t}{Re} \left\| \sqrt{\tilde{\eta}^k} \nabla \tilde{\mathbf{v}}^k \right\|_{L^2}^2 + \frac{\delta t}{PeCn^2 We} \|\nabla \tilde{\mu}^k\|_{L^2}^2 \geq \frac{1}{WeCn} \left\| \frac{\psi'''(\lambda)}{24} \right\|_{L^\infty(\Omega)} C_m L^3 \delta t^3. \quad (52)$$

This condition becomes a condition on the maximum energy-stable time-step size:

$$0 \leq \delta t \leq \left( \frac{\frac{1}{Re} \left\| \sqrt{\tilde{\eta}^k} \nabla \tilde{\mathbf{v}}^k \right\|_{L^2}^2 + \frac{1}{PeCn^2 We} \|\nabla \tilde{\mu}^k\|_{L^2}^2}{\frac{C_m L^3}{WeCn} \left\| \frac{\psi'''(\lambda)}{24} \right\|_{L^\infty(\Omega)}} \right)^{\frac{1}{2}}, \quad (53)$$

which proves the theorem. ■

**Remark 4.** Condition eq. (53) is a weak condition (satisfied by most  $\delta t$ ), as all the quantities in its statement are order one quantities. The bounds for  $\left(\frac{\psi'''(\lambda)}{24}, (\phi^{k+1} - \phi^k)^3\right)$  are the absolute worst-case scenario, which in practice rarely happen. Therefore, though we cannot claim unconditional stability for the scheme, the scheme is energy-stable for large range of  $\delta t$  values, which allows us to take large time steps.

<sup>160</sup> **Remark 5.** Our estimate for the time step size is identical to the one in [7]. Therefore, this second-order extension of the time scheme in Khanwale et al. [7] with the fully-coupled approach preserves energy-stability.

### 3.2. Spatial discretization and the variational multiscale approach

The unknowns:

$$(\phi, \mu, \mathbf{v}, p), \quad (54)$$

are all discretized in space using continuous Galerkin finite elements with piecewise polynomial approximations. It is well-known from the literature that approximating the velocity,  $\mathbf{v}$ , and the pressure,  $p$ , with the same polynomial order leads to numerical instabilities as this violates the discrete inf-sup condition or Ladyzhenskaya-Babuska-Brezzi condition (e.g., see Volker [9, page 31]). A popular method to overcome this difficulty is to add stabilization terms to the evolution equations that transform the inf-sup stability condition to a coercivity statement [35]. In particular, a large class of stabilization techniques derive from the so-called *variational multiscale* approach (VMS) [36], which generalizes the well-known SUPG/PSPG [37] method in the context of large-eddy simulation (LES) [38]. Additionally, VMS provides a natural leeway into modeling high-Reynolds number flows [24].

The VMS approach uses a direct-sum decomposition of the function spaces as follows. If  $\mathbf{v} \in \mathbf{V}$ ,  $p \in Q$ , and  $\phi \in Q$  then we decompose these spaces as:

$$\mathbf{V} = \bar{\mathbf{V}} \oplus \mathbf{V}' \quad \text{and} \quad Q = \bar{Q} \oplus Q', \quad (55)$$

where  $\bar{\mathbf{V}}$  and  $\bar{Q}$  are the cG(r) subspaces of  $\mathbf{V}$  and  $Q$ , respectively, and the primed versions are the complements of the cG(r) subspaces in  $\mathbf{V}$  and  $Q$ , respectively. We decompose the velocity and pressure as follows:

$$\mathbf{v} = \bar{\mathbf{v}} + \mathbf{v}', \quad \phi = \bar{\phi} + \phi', \quad \text{and} \quad p = \bar{p} + p', \quad (56)$$

<sup>175</sup> where the *coarse scale* components are  $\bar{\mathbf{v}} \in \bar{\mathbf{V}}$  and  $\bar{p}, \bar{\phi} \in \bar{Q}$ , and the *fine scale* components are  $\mathbf{v}' \in \mathbf{V}'$  and  $p', \phi' \in Q'$ . We define a projection operator,  $\mathcal{P} : \mathbf{V} \rightarrow \bar{\mathbf{V}}$ , such that  $\bar{\mathbf{v}} = \mathcal{P}\{\mathbf{v}\}$  and  $\mathbf{v}' = \mathbf{v} - \mathcal{P}\{\mathbf{v}\}$ . A similar operator can decompose  $p$  and  $\phi$ .

Substituting these decompositions in the original variational form in definition 1 in section 3

yields:

$$\begin{aligned}
\text{Momentum Eqns: } & \left( w_i, \rho(\phi) \frac{\partial \bar{v}_i}{\partial t} \right) + \left( w_i, \frac{\partial (\rho(\phi) v'_i)}{\partial t} \right) + \left( w_i, \rho(\phi) \bar{v}_j \frac{\partial \bar{v}_i}{\partial x_j} \right) \\
& + \left( w_i, \rho(\phi) v'_j \frac{\partial \bar{v}_i}{\partial x_j} \right) + \left( w_i, \frac{\partial (\rho(\phi) \bar{v}_j v'_i)}{\partial x_j} \right) + \left( w_i, \frac{\partial (\rho(\phi) v'_j v'_i)}{\partial x_j} \right) \\
& + \frac{1}{Pe} \left( w_i, J_j \frac{\partial \bar{v}_i}{\partial x_j} \right) + \frac{1}{Pe} \left( w_i, \frac{\partial (J_j v'_i)}{\partial x_j} \right) + \frac{Cn}{We} \left( w_i, \frac{\partial}{\partial x_j} \left( \frac{\partial \phi}{\partial x_i} \frac{\partial \phi}{\partial x_j} \right) \right) \\
& + \frac{1}{We} \left( w_i, \frac{\partial (\bar{p} + p')}{\partial x_i} \right) + \frac{1}{Re} \left( \frac{\partial w_i}{\partial x_k}, \eta(\phi) \frac{\partial (\bar{v}_i + v'_i)}{\partial x_k} \right) - \left( \frac{w_i, \rho(\phi) \hat{g}_i}{Fr} \right) \\
& + \left( q, \frac{\partial \bar{v}_i}{\partial x_i} \right) + \left( q, \frac{\partial v'_i}{\partial x_i} \right) = 0,
\end{aligned} \tag{57}$$

$$\text{Cahn-Hilliard Eqn: } \left( q, \frac{\partial (\bar{\phi} + \phi')}{\partial t} \right) - \left( \frac{\partial q}{\partial x_i}, \bar{v}_i \bar{\phi} \right) - \left( \frac{\partial q}{\partial x_i}, v'_i \bar{\phi} \right) - \left( \frac{\partial q}{\partial x_i}, \bar{v}_i \phi' \right) \tag{58}$$

$$-\frac{1}{Pe Cn} \left( \frac{\partial q}{\partial x_i}, \frac{\partial (m(\phi) \mu)}{\partial x_i} \right) = 0, \tag{59}$$

$$\text{Chemical Potential: } -(q, \mu) + \left( q, \frac{d\psi}{d\phi} \right) - Cn^2 \left( q, \frac{\partial}{\partial x_i} \left( \frac{\partial \phi}{\partial x_i} \right) \right) = 0, \tag{60}$$

where  $\mathbf{w}, \bar{\mathbf{v}}, \in \mathcal{P}\mathbf{H}^r(\Omega)$ ,  $\bar{p}, \bar{\phi} \in \mathcal{P}H^r(\Omega)$ ,  $\mathbf{v}' \in (\mathcal{I} - \mathcal{P})\mathbf{H}^r(\Omega)$ ,  $\phi', p' \in (\mathcal{I} - \mathcal{P})H^r(\Omega)$ , and  $\mu, q \in \mathcal{P}H^r(\Omega)$ . Here  $\mathcal{I}$  is the identity operator and  $\mathcal{P}$  is the projection operator. We use the residual-based approximation proposed in Bazilevs et al. [24] for the fine-scale components to close the equations:

$$\rho(\phi) v'_i = -\tau_m \mathcal{R}_m(\rho, \bar{v}_i, \bar{p}), \quad p' = -\rho(\phi) \tau_c \mathcal{R}_c(\bar{v}_i), \quad \text{and} \quad \phi' = -\tau_\phi \mathcal{R}_\phi(\bar{v}_i, \phi). \tag{61}$$

We substitute the infinite-dimensional spaces by their discrete counterparts (superscript  $h$ ) using conforming Galerkin finite elements where the trial and test functions are taken from the same spaces. Note that we only solve for the coarse-scale components. The resulting discrete variational formulation can then be defined as follows.

**Definition 2**

Find  $\bar{\mathbf{v}}^h \in \mathcal{P}\mathbf{H}_0^{r,h}(\Omega)$  and  $\bar{p}^h, \bar{\phi}^h, \mu^h \in \mathcal{P}H^{r,h}(\Omega)$  such that

$$\begin{aligned} \text{Momentum: } & \left( w_i, \bar{\rho}^h \frac{\partial \bar{v}_i^h}{\partial t} \right) + \left( w_i, \bar{\rho}^h \bar{v}_j^h \frac{\partial \bar{v}_i^h}{\partial x_j} \right) - \left( w_i, \tau_m \mathcal{R}_m \left( \bar{v}_j^h, \bar{p}^h \right) \frac{\partial \bar{v}_i^h}{\partial x_j} \right) \\ & + \left( \frac{\partial w_i}{\partial x_j}, \bar{v}_j^h \left( \tau_m \mathcal{R}_m \left( \bar{v}_i^h, \bar{p}^h \right) \right) \right) - \left( \frac{\partial w_i}{\partial x_j}, \frac{\tau_m^2}{\bar{\rho}^h} \mathcal{R}_m \left( \bar{v}_j^h, \bar{p}^h \right) \mathcal{R}_m \left( \bar{v}_i^h, \bar{p}^h \right) \right) \\ & + \frac{1}{Pe} \left( w_i, J_j^h \frac{\partial \bar{v}_i^h}{\partial x_j} \right) + \frac{1}{Pe} \left( \frac{\partial w_i}{\partial x_j}, J_j^h \frac{\tau_m}{\bar{\rho}^h} \mathcal{R}_m \left( \bar{v}_i^h, \bar{p}^h \right) \right) \\ & - \frac{Cn}{We} \left( \frac{\partial w_i}{\partial x_j}, \frac{\partial \bar{\phi}^h}{\partial x_i} \frac{\partial \bar{\phi}^h}{\partial x_j} \right) - \frac{1}{We} \left( \frac{\partial w_i}{\partial x_i}, \bar{p}^h \right) \\ & + \frac{1}{We} \left( \frac{\partial w_i}{\partial x_i}, \bar{\rho}^h \tau_c \mathcal{R}_c \left( \bar{v}_i^h \right) \right) + \frac{1}{Re} \left( \frac{\partial w_i}{\partial x_k}, \bar{\eta}^h \frac{\partial \bar{v}_i^h}{\partial x_k} \right) - \left( \frac{w_i}{Fr}, \bar{\rho}^h \hat{g}_i \right) = 0, \end{aligned} \quad (62)$$

$$\text{Thermo: } J_i^h = \frac{(\rho_- - \rho_+)}{2} \frac{\partial \mu^h}{\partial x_i}, \quad (63)$$

$$\text{Solenoidality: } \left( q, \frac{\partial \bar{v}_i^h}{\partial x_i} \right) - \left( \frac{\partial q}{\partial x_i}, \frac{\tau_m}{\bar{\rho}^h} \mathcal{R}_m \left( \bar{v}_i^h, \bar{p}^h \right) \right) = 0, \quad (64)$$

$$\begin{aligned} \text{Cahn-Hilliard: } & \left( q, \frac{\partial \bar{\phi}^h}{\partial t} \right) - \left( \frac{\partial q}{\partial x_i}, \bar{v}_i^h \bar{\phi}^h \right) + \left( \frac{\partial q}{\partial x_i}, \frac{\tau_m}{\bar{\rho}^h} \mathcal{R}_m \left( \bar{v}_i^h, \bar{p}^h \right) \bar{\phi}^h \right) \\ & + \left( \frac{\partial q}{\partial x_i}, \bar{v}_i^h \tau_\phi \mathcal{R}_\phi \left( \bar{v}_i^h, \bar{\phi}^h \right) \right) - \frac{1}{PeCn} \left( \frac{\partial q}{\partial x_i}, \frac{\partial (\bar{m}^h \mu^h)}{\partial x_i} \right) = 0, \end{aligned} \quad (65)$$

$$\text{Potential: } - \left( q, \mu^h \right) + \left( q, \frac{d\psi}{d\bar{\phi}^h} \right) + Cn^2 \left( \frac{\partial q}{\partial x_i}, \frac{\partial \bar{\phi}^h}{\partial x_i} \right) = 0, \quad (66)$$

$\forall \mathbf{w} \in \mathcal{P}\mathbf{H}_0^{r,h}(\Omega)$  and  $\forall q \in \mathcal{P}H^{r,h}(\Omega)$ .

In the above expressions, we used the following notation:

$$\bar{\rho}^h := \rho \left( \bar{\phi}^h \right), \quad \eta^h := \eta \left( \bar{\phi}^h \right), \quad \text{and} \quad \bar{m}^h := m \left( \bar{\phi}^h \right), \quad (67)$$

and the following parameter values:

$$\tau_m = \left( \frac{4}{\Delta t^2} + \bar{v}_i^h G_{ij} \bar{v}_j^h + \frac{1}{\bar{\rho}^h Pe} \bar{v}_i^h G_{ij} \bar{J}_j^h + C_I \left( \frac{\bar{\eta}^h}{\bar{\rho}^h Re} \right)^2 G_{ij} G_{ij} \right)^{-1/2}, \quad (68)$$

$$\tau_c = \frac{1}{tr(G_{ij}) \tau_m}, \quad \tau_\phi = \left( \frac{4}{\Delta t^2} + \bar{v}_i^h G_{ij} \bar{v}_j^h + C_\phi \left( \frac{1}{PeCn} \right)^2 G_{ij} G_{ij} \right)^{-1/2}.$$

Here we set  $C_I$  and  $C_\phi$  for all our simulations to 6 and the residuals are given by

$$\begin{aligned} \mathcal{R}_m \left( \bar{v}_i^h, \bar{p}^h \right) &= \bar{\rho}^h \frac{\partial \bar{v}_i^h}{\partial t} + \bar{\rho}^h \bar{v}_j^h \frac{\partial \bar{v}_i^h}{\partial x_j} + \frac{1}{Pe} J_j^h \frac{\partial \bar{v}_i^h}{\partial x_j} + \frac{Cn}{We} \frac{\partial}{\partial x_j} \left( \frac{\partial \bar{\phi}^h}{\partial x_i} \frac{\partial \bar{\phi}^h}{\partial x_j} \right) \\ &\quad + \frac{1}{We} \frac{\partial \bar{p}^h}{\partial x_i} - \frac{1}{Re} \frac{\partial}{\partial x_k} \left( \bar{\eta}^h \frac{\partial \bar{v}_i^h}{\partial x_k} \right) - \frac{\bar{\rho}^h \hat{g}}{Fr}, \\ \mathcal{R}_c \left( \bar{v}_i^h \right) &= \frac{\partial \bar{v}_i^h}{\partial x_i}, \quad \mathcal{R}_\phi \left( \bar{v}_i^h, \bar{\phi}^h, \mu^h \right) = \frac{\partial \bar{\phi}^h}{\partial t} + \frac{\partial \left( \bar{v}_i^h \bar{\phi}^h \right)}{\partial x_i} - \frac{1}{PeCn} \frac{\partial}{\partial x_i} \left( \frac{\partial \left( \bar{m}^h \mu^h \right)}{\partial x_i} \right). \end{aligned} \quad (69)$$

The following assumptions were made in the above variational problem.

1.  $\mathbf{v}' = 0$  on the boundary  $\partial\Omega$ ; similarly  $\phi' = 0$  on  $\partial\Omega$ .
2.  $\left( \frac{\partial w_i}{\partial x_k}, \bar{\eta}^h \frac{\partial \bar{v}_i^h}{\partial x_k} \right) = 0$  from the orthogonality condition of the projector. The projector utilizes the inner product that comes from the bilinear form of these viscous terms [24, 39].
3. We assume that  $\left( \frac{\partial q}{\partial x_i}, v'_i \phi' \right) = 0$  under the reasoning that fluctuations in  $\phi'$  are small compared to  $v'_i$ . This significantly simplifies the formulation.
4. We use the coarse-scale part of the VMS decomposition of  $\phi$  to compute the mixture density and mixture viscosity. We use the pulled back  $\phi^*$  (see remark 2) for this calculation, which regularizes  $\phi$  by smoothing out overshoots and undershoots. The pull back ensures  $\phi^* \in H^r$ , and its projection on the mesh  $\phi^{*,h} \in H^{r,h}$  as required for the cG formulation.

**Remark 6.** *The above formulation is written for a generic order ( $r$ ) for the interpolating polynomials (basis functions). Nevertheless, we restrict our attention to  $r = 1$  for the numerical experiments in this paper.*

Finally, the time derivative in the above expressions is still continuous. In the fully discrete numerical method we substitute the time-derivatives in the momentum and phase field equations using the time scheme presented in eq. (13) – eq. (17).

### 3.3. Handling non-linearity

The fully discrete version of eq. (13) – eq. (17) represents a non-linear system of algebraic equations that discretize the CHNS system eq. (62) – eq. (66) and that must be solved in each time-step. Symbolically, we can write this nonlinear algebraic system as

$$F_i \left( U_1^k, U_2^k, \dots, U_n^k \right) = 0, \quad (70)$$

where  $\mathbf{U}^k$  is a vector containing all of the degrees of freedom at the discrete time  $t^k$ . In order to solve this nonlinear algebraic system we make use of a Newton method, which requires us to solve the following linear system in each Newton iteration:

$$J_{ij}^{s,k} \delta U_j^{s,k} = -F_i \left( U_1^{s,k}, U_2^{s,k}, \dots, U_n^{s,k} \right), \quad J_{ij}^{s,k} := \frac{\partial}{\partial U_j} F_i \left( U_1^{s,k}, U_2^{s,k}, \dots, U_n^{s,k} \right), \quad (71)$$

where  $U_j^{s,k}$  is the vector containing all the degrees of freedom at the  $k^{\text{th}}$  time step and at the  $s^{\text{th}}$  Newton iteration.  $\delta U_j^{s,k}$  is the “perturbation” (update) vector that will be used to update the current Newton iteration guess:

$$U_j^{s+1,k} = U_j^{s,k} + \delta U_j^{s,k}. \quad (72)$$

$J_{ij}^{s,k}$  is the Jacobian matrix, which we analytically compute by calculating the variations (partial derivatives) of the operators with respect to the degrees of freedom. The calculation of  $J_{ij}^{s,k}$  is more challenging for the fully-coupled approach compared to the block iterative technique in Khanwale et al. [7]. The iterative procedure begins with an initial guess, which we simply take as the solution from the previous time-step:

$$U_i^{0,k} = U_i^{k-1}, \quad (73)$$

and ends once we reach the desired tolerance:

$$\|\delta U_j^{s,k}\| \leq \text{TOL}. \quad (74)$$

Once this tolerance is reached we set  $U_j^k = U_j^{s,k}$  and move on to the next time-step.

200 In this work we solve linear system eq. (71) in each Newton iteration on a massively parallel architecture. In particular, we make use of the PETSc library, which provides efficient parallel implementations of the above ideas along with an extensive suite of preconditioners and solvers for the linear system [40–42]. The precise choice of linear solvers and preconditioners is different for different numerical experiments; and therefore, we provide more details on these choices in the  
205 numerical experiment sections of this paper.

#### 4. Octree based domain decomposition

Octrees are widely used in the computational sciences to represent dynamically-adapted hierarchical meshes [30, 32, 33, 43]; this is largely due to their conceptual simplicity and their ability to scale across a large number of processors. Adaptivity is crucial in the computational sciences,  
210 where in many cases it reduces the overall degrees of freedoms (problem size), making these simulations feasible on currently available computers. The use of adaptive discretizations can introduce additional challenges, especially in distributed computing, such as load-balancing, low-cost scalable mesh generation, and mesh partitioning. Thus, we use Dendro, a highly scalable parallel octree library, to generate full adaptive quasi-structured octree based meshes and partitions. In the following sections, we summarize how Dendro allows us to perform our numerical simulations. The reader can find a detailed account of the algorithms used in Dendro in [30, 33].

##### 4.1. Octree construction and 2:1 balancing

Dendro refines an octant based on user-specified criteria proceeding in a top-down fashion. The user defines the refinement criteria by a function that takes the coordinates of the octant, and  
220 returns `true` or `false`. Since the refinement happens locally to the element, this step is embarrassingly parallel. In distributed-memory machines, the initial top-down tree construction enables an efficient partitioning of the domain across an arbitrary number of processes. All processes start at the root node (i.e., the cubic bounding box for the entire domain). We perform redundant computations on all processes to avoid communication during the refinement stage. Starting from the  
225 root node, all processes refine (similar to a sequential implementation) until the process produces at least  $\mathcal{O}(p)$  octants requiring further refinement. The procedure ensures that upon partitioning across  $p$  processors, each processor gets at least one octant. Then using a space-filling-curve (SFC) based partitioning, we partition the octants across  $p$  partitions [33]. Once the algorithm completes this partitioning, we can restrict the refinement criterion to a processor's partition, which we can re-distribute to ensure load-balancing. We enforce a condition in our distributed octrees that no two neighbouring octants differ in size by more than a factor of two (2:1 balancing). This ratio makes subsequent operations simpler without affecting the adaptive properties. Our balancing algorithm  
230

uses a variant of TREESORT [33] with top-down and bottom-up traversal of octrees which is different from existing approaches [32, 44, 45].

235 **4.2. SFC-based octree partition**

The refinement and subsequent two-to-one balancing of the octree procedures can result in a non-uniform distribution of elements across processes, leading to load imbalance. This imbalance is particularly challenging when meshing complex geometries. SFC induces a partial ordering operator in higher dimensional space, where TREESORT [33] performs a parallel sort operation on the 240 octants. The SFC traverses the octants in the sorted order, which reduces the partitioning problem to partitioning a 1D curve. Finally, we use a Hilbert SFC curve based partitioning compared to traditionally used Morton (Z-curve), which produces superior partitions in large scale computations [33].

245 **4.3. Mesh generation**

By meshing, we refer to the construction of the data structures required to perform numerical computations on topological octree data. DENDRO builds distributed data structures to perform finite difference (FD), finite volume (FV), and finite element (FE) computations. In this work, we use the FE data structures. One of the key steps of the mesh generation stage is to construct neighborhood information for octants, which uses two maps computed by DENDRO. The first map  $o_{20}$  determines the neighboring octants of a given octant, and the map  $o_{2N}$  computes the nodes corresponding to a given octant. We generate the  $o_{20}$  map by performing parallel searches similar to approaches described in [30]. Assuming we have  $n$  octants per partition, the search operations and building required to create the  $o_{20}$  and  $o_{2N}$  data structures have  $\mathcal{O}(n \log(n))$  and  $\mathcal{O}(n)$  complexity, respectively.

255 **4.4. Handling hanging nodes**

While the use of quasi-structured grids such as octree-grids makes parallel meshing scalable and efficient without sacrificing adaptivity, one challenge is to handle the resulting non-conformity efficiently. The resulting *hanging nodes* occur on faces/edges shared between unequal elements. These hanging nodes do not represent independent degrees of freedom. We do not store the hanging nodes in DENDRO to minimize the memory footprint and to improve the overall efficiency. The polynomial order of the elements and the free (non-hanging) nodes on the face/edge determine the value of the function at the hanging nodes. Therefore, we introduce these extra degrees of freedom as temporary variables before elemental matrix assembly or matrix-vector multiplication (for matrix-free computations) and eliminate them following the elemental operation. This virtualization of the hanging nodes is straightforward as we limit the meshes to a two-to-one balance, which limits the number of overall cases we need to consider explicitly see [30] for further details on the handling of hanging nodes in DENDRO.

260 **4.5. Re-meshing and interpolation**

An essential requirement is to adapt the spatial mesh as the fluid interface moves across the 270 domain. Figure 10 shows the adaptive mesh refinement following the deformation of the interface in a Rayleigh-Taylor instability. In distributed-memory systems, this localized meshing requires a re-partition and re-balance of the load. Thus, after a few time steps, we re-mesh. This re-meshing step is similar to the initial mesh generation and refinement. Now, the process uses the current position of the interface as well as the original geometry. The two-to-one balance enforcement and 275 meshing follow this mesh generation. We now transfer the velocity field from the old mesh to the new mesh using a simple interpolation process. That is, the grid transfer only happens between

parent and child (for coarsening and refinement) as it otherwise remains unchanged. Therefore, the transfer from the old mesh to the new one uses standard polynomial interpolation, followed by a simple re-partitioning based on the new mesh.

280 **5. Numerical experiments**

### 5.1. 2D simulations: manufactured solutions

We use the method of manufactured solutions to assess the convergence properties of our method. We select an input “solution” which is solenoidal, and substitute it in the full set of governing equations. We then use the residual as a body force on the right-hand side of eqs. (13) to (17). We choose the following “solution” with appropriate body forcing terms:

$$\begin{aligned} \mathbf{v} &= (\pi \sin^2(\pi x_1) \sin(2\pi x_2) \sin(t), -\pi \sin(2\pi x_1) \sin^2(\pi x_2) \sin(t), 0), \\ p &= \cos(\pi x_1) \sin(\pi x_2) \sin(t), \quad \phi = \mu = \cos(\pi x_1) \cos(\pi x_2) \sin(t). \end{aligned} \quad (75)$$

Our numerical experiments use the following non-dimensional parameters:  $Re = 10$ ,  $We = 1$ ,  $Cn = 1.0$ ,  $Pe = 3.0$ , and  $Fr = 1.0$ . The density ratio is set to  $\rho_-/\rho_+ = 0.85$ .

For the first experiment we use a 2D uniform mesh with  $300 \times 300$  bilinear elements (quads).

- 285 Panel (a) of fig. 1 shows the temporal convergence of the  $L^2$  errors (numerical solution compared with the manufactured solution) calculated at  $t = \pi$  to allow for one complete time period. The figure shows the evolution of the error versus time-step on a log-log scale. The errors are decreasing with a slope close to two for the phase-field function  $\phi$ , thereby demonstrating second-order convergence. For velocity the slope on the log-log scale is initially about 2.0, but then tapers off for  
290 smaller time-steps; we expect this tapering off at smaller time-steps due to the fact that we used a fixed mesh and at some point the spatial errors dominate.

We next conduct a spatial convergence study. We fix the time step at  $\delta t = 10^{-3}$ , and vary the spatial mesh resolution. Panel (b) of fig. 1 shows the spatial convergence of  $L^2$  errors (numerical solution compared with the manufactured solution) at  $t = \pi$ . We observe second order convergence  
295 for both velocity and  $\phi$ .

Panel (c) of fig. 1 shows mass conservation for an intermediate resolution simulation with  $\delta t = 10^{-3}$  and  $175 \times 175$  elements. We plot mass drift:

$$\int_{\Omega} \phi(\mathbf{x}, t) d\mathbf{x} - \int_{\Omega} \phi(\mathbf{x}, t=0) d\mathbf{x}, \quad (76)$$

and expect this value to be close to zero as per the theoretical prediction of proposition 1. We observe excellent mass conservation with fluctuations of the order of  $10^{-12}$ , which is to be expected in double precision arithmetic. Here we used a relative tolerance of  $10^{-10}$  for the Newton iteration. For the linear solves within each Newton iteration we used a relative tolerance of  $10^{-7}$ .

300 **5.2. 2D simulations: single rising bubble**

To validate the framework, we consider benchmark cases for a single rising bubble in a quiescent water channel [46–48]. We set the Froude number ( $Fr = u^2/(gD)$ ) to 1, which fixes the non-dimensional velocity scale to  $u = \sqrt{gD}$ , where  $g$  is the gravitational acceleration, and  $D$  is the diameter of the bubble. This scaling gives a Reynolds number of  $\rho_c g^{1/2} D^{3/2}/\mu_c$ , where  $\rho_c$   
305 and  $\mu_c$  are the specific density and specific viscosity of the continuous fluid, respectively. The Archimedes number,  $Ar = \rho_c g^{1/2} D^{3/2}/\mu_c$ , scales the diffusion term in the momentum equation. The Weber number here becomes  $We = \rho_c g D^2/\sigma$ . We use the density of the continuous fluid to

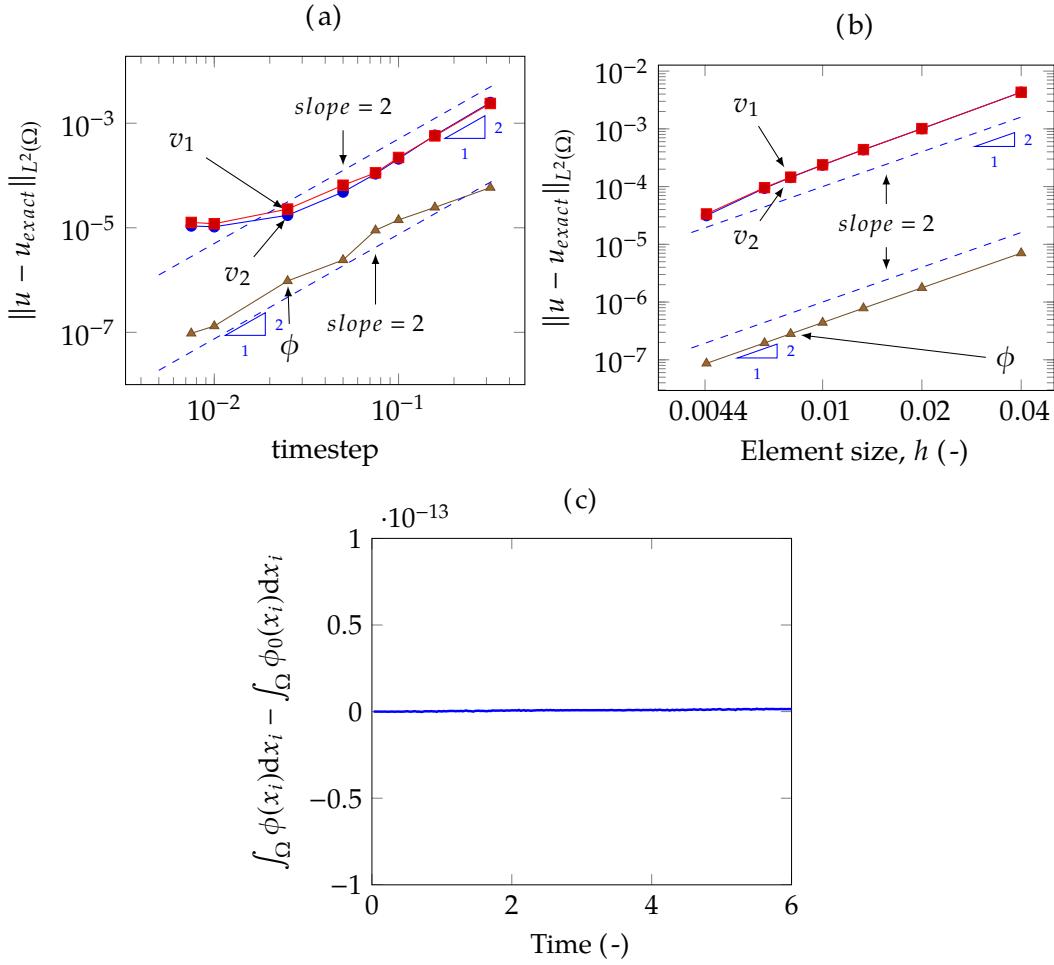


Figure 1: *Manufactured Solution Examples* (section 5.1). Shown in the Panels are (a) the temporal convergence of the numerical scheme for the case of manufactured solutions; (b) the spatial convergence of numerical scheme for the manufactured solutions with a time step of  $10^{-3}$ ; and (c) the mass conservation for the case of manufactured solutions using  $300 \times 300$  elements with time step of  $10^{-2}$ .

non-dimensionalize:  $\rho_+ = 1$ . The density and viscosity ratios are  $\rho_+/\rho_-$  and  $\nu_+/\nu_-$ , respectively. We present results for two standard benchmark cases.

Table 1 shows the parameters and the corresponding non-dimensional numbers for the two cases simulated in this work. The bubble is centered at  $(1, 1)$ , and since our scaling length scale is the bubble diameter, the bubble diameter for our simulations is 1. The domain is  $[0, 2] \times [0, 4]$ . Following the benchmark studies in the literature, we choose the top and bottom wall to have no slip boundary conditions and the side walls to have boundary conditions:  $v_1 = 0$  ( $x$ -velocity) and  $\frac{\partial v_2}{\partial x} = 0$  ( $y$ -velocity). We use the biCGstab (bcgs) linear solver from the PETSc suite along with the Additive Schwarz (ASM) preconditioner for the linear solves in the Newton iterations (see section 3.3). We use a time step of  $2.5 \times 10^{-3}$  for both the test cases. The convergence criterion for both test cases uses a relative tolerance of  $10^{-6}$  for Newton iteration and a relative tolerance of  $10^{-7}$  for the linear solves within each Newton iteration.

Test Case	$\rho_c$	$\rho_b$	$\mu_c$	$\mu_b$	$\rho_+/\rho_-$	$\nu_+/\nu_-$	$g$	$\sigma$	$Ar$	$We$	$Fr$
1	1000	100	10	1.0	10	10	0.98	24.5	35	10	1.0
2	1000	1.0	10	0.1	1000	100	0.98	1.96	35	125	1.0

Table 1: Physical parameters and corresponding non-dimensional numbers for the 2D single rising drop benchmarks considered in section 5.2.

### 5.2.1. Test case 1

This test case considers the effect of higher surface tension, and consequently less deformation of the bubble as it rises. We compare the bubble shape in fig. 2 with benchmark quantities presented in three previous studies [46–48]. We take  $Cn = 5 \times 10^{-3}$  for this case. Panel (a) of fig. 2 shows a shape comparison against benchmark studies in the literature, and we see an excellent agreement in the shape of the bubble. Panel (b) of fig. 2 shows a comparison of centroid locations with respect to time against benchmark studies in the literature; again, we see an excellent agreement. We can see from the magnified inset in panel (b) of fig. 2 that as we keep increasing the mesh resolution, the plot approaches the benchmark studies. We see an almost exact overlap between the benchmark and cases with  $h = 2/400$  and  $h = 2/600$ , where  $h$  is the size of the element, demonstrating spatial convergence.

We next check whether the numerical method follows the theoretical energy stability proved in theorem 1. We present the evolution of the energy functional defined in eq. (9) for test case 1. Panel (c) of fig. 2 shows that the energy is decreasing in accordance with the energy stability condition for all three spatial resolutions of  $h = 2.0/200$ ,  $h = 2.0/400$ , and  $h = 2.0/600$ .

Finally, we check the mass conservation. Panel (d) shows the total mass of the system minus the initial mass. At all reported spatial resolutions the change in the total mass is of the order of  $10^{-8}$ , even after 1600 time steps. The numerical method delivers excellent mass conservation for long time horizons.

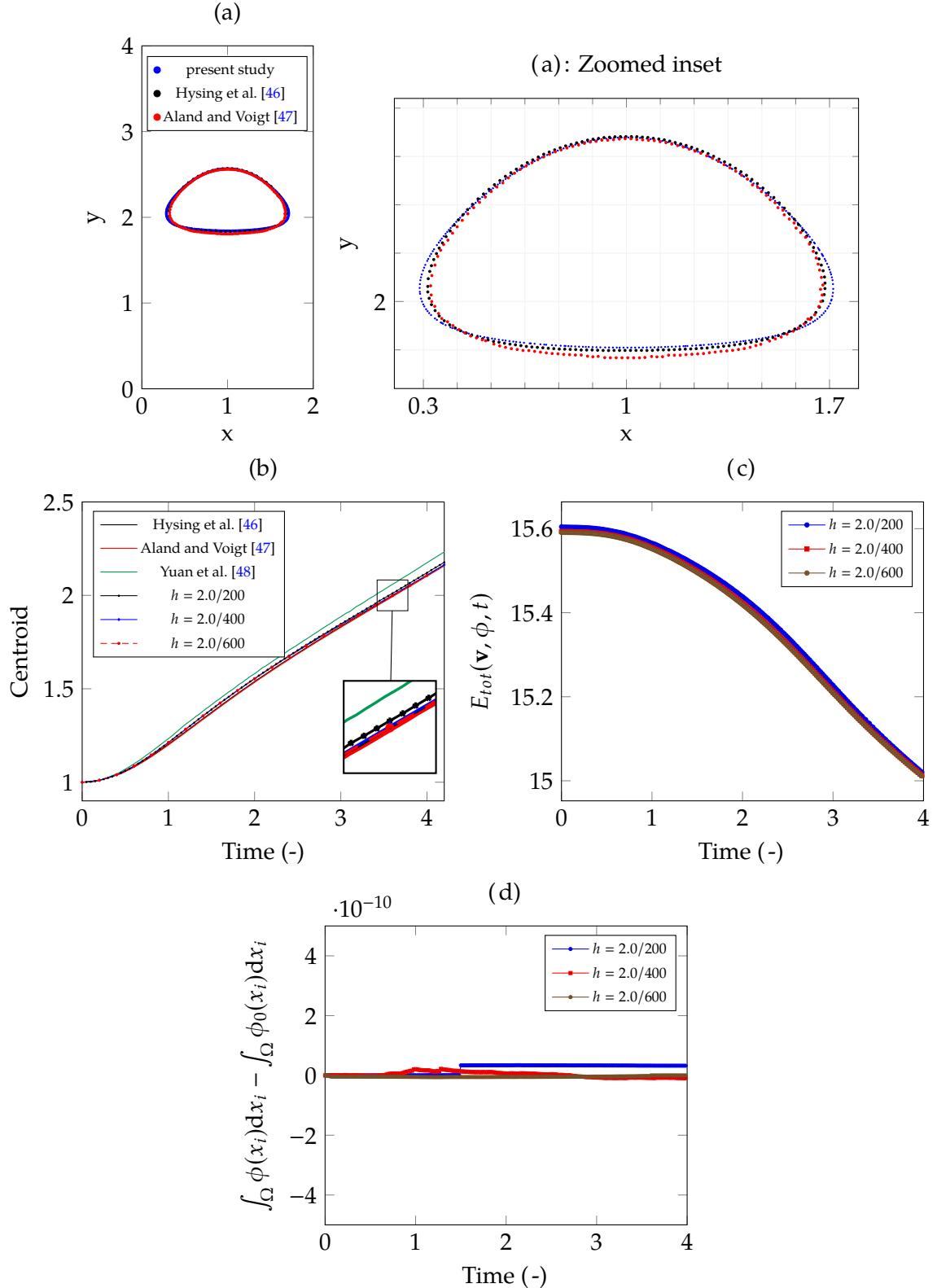


Figure 2: 2D Single Rising Drop Test Case 1 (section 5.2.1). Shown in the panels are (a) comparisons of the computed bubble shape against results from the literature at non-dimensional time  $T = 4.2$ ; (b) comparisons of the rise of the bubble centroid against results from the literature; (c) decay of the energy functional illustrating theorem 1; and (d) total mass conservation (integral of total  $\phi$ ).

### 5.2.2. Test case 2

This test case considers a lower surface tension resulting in high deformations of the bubble as it rises. As before, we compare the bubble shape in fig. 3 with benchmark quantities presented in [46–48]. Panel (a) shows the shape comparison with benchmark studies in the literature. We see an excellent agreement in the shape of the bubble. All simulations (our results and benchmarks) exhibit a skirted bubble shape. We see an excellent match in the overall bubble shape with minor differences in the dynamics of its tails. Specifically, we see that the tails of the bubble in our case pinch-off to form satellite bubbles<sup>1</sup>. We performed this simulation with a  $Cn = 0.0025$  and three different spatial resolutions. We can see in panel (a) of fig. 3 that our simulation captures well this filament pinch-off in the tails. The works of Aland and Voigt [47], Yuan et al. [48] did not observe these thin tails and pinch-offs, while Hysing et al. [46] described pinch-off of the tails and satellite bubbles.

Panel (b) of fig. 3 compares the centroid location evolution with time. Again we see an excellent agreement with all three previous benchmark studies. We can see from the magnified inset in this panel that as we increase the mesh resolution the plot approaches the benchmark studies and we see an almost exact overlap between the benchmark and cases with  $h = 2/1000$  and  $h = 2/2000$  demonstrating spatial convergence. Next, we report the evolution of the energy functional defined in eq. (9) for test case 2. Panel (c) of fig. 3 shows the decay of the total energy functional in accordance with the energy stability condition for all three spatial resolutions of  $h = 2.0/800$ ,  $h = 2.0/1000$ , and  $h = 2.0/1200$ . Finally, panel (d) of fig. 3 shows the total mass of the system in comparison with the total initial mass of the system. We can see that for all spatial resolutions, the change in the total mass against the initial total mass is of the order of  $1 \times 10^{-8}$ , which illustrates that the numerical method satisfies mass conservation over long time horizons.

---

<sup>1</sup>Such instabilities require a low  $Cn$  number, as only a thin interface can capture the dynamics of the thin tails of the bubble

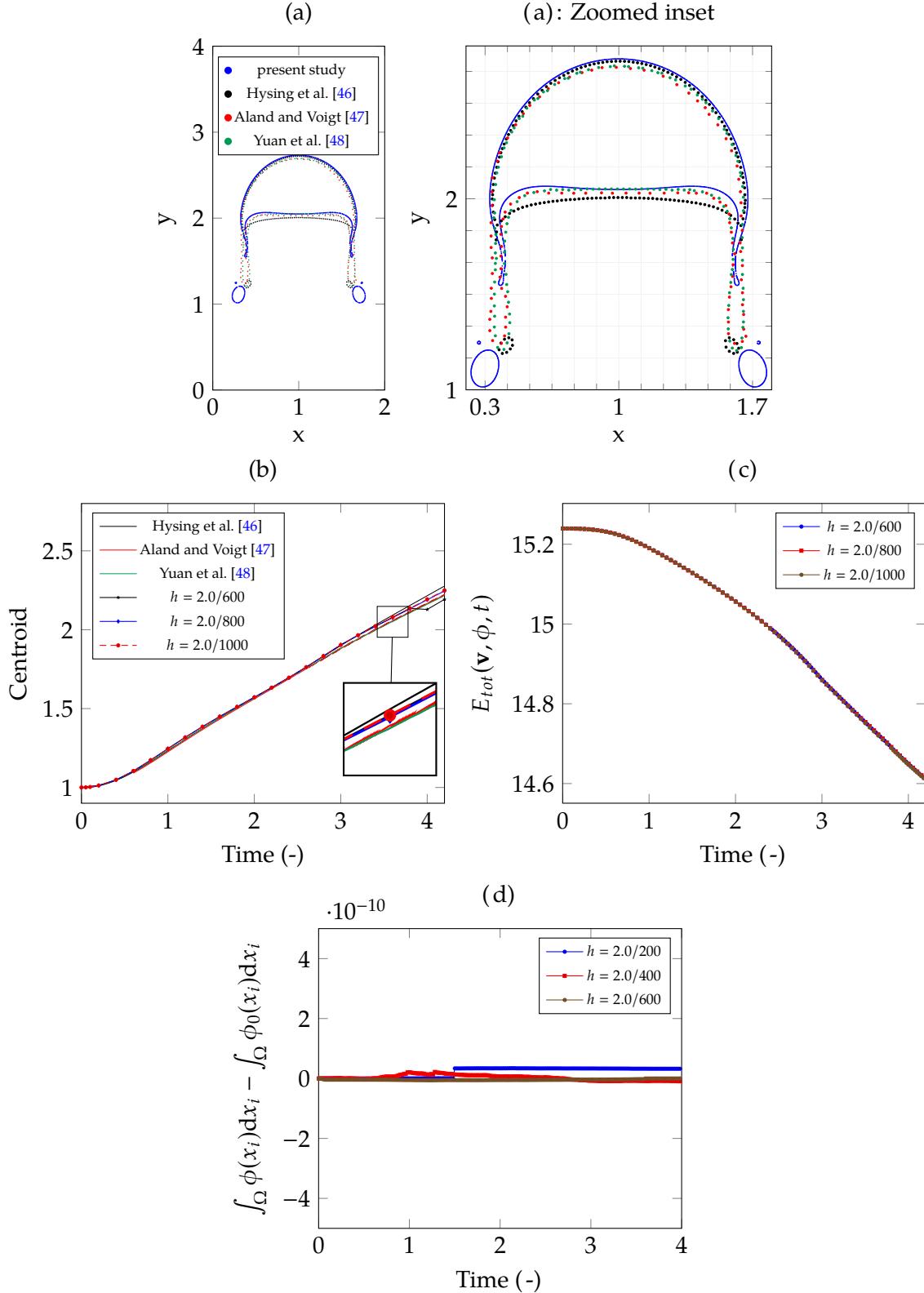


Figure 3: 2D Single Rising Drop Test Case 2 (section 5.2.2). Shown in the panels are (a) comparisons of the computed bubble shape against results from the literature at non-dimensional time  $T = 4.2$ ; (b) comparisons of the rise of the bubble centroid against results from the literature; (c) decay of the energy functional illustrating theorem 1; and (d) total mass conservation (integral of total  $\phi$ ).

### 5.3. 2d simulations: Rayleigh-Taylor instability

We now demonstrate the performance of the numerical framework with large deformation in the interface and chaotic regimes (high Reynolds numbers). While the bubble rise case in the previous sub-section is an interplay between surface tension and buoyancy, buoyancy dominates the evolution of the Rayleigh-Taylor instability. Here the choice of non-dimensional numbers ensures that the surface tension effect is small (high Weber number). In contrast, other studies switch off the surface tension forcing terms in the momentum equations (see, e.g., [49–52]).

The setup is as follows: the heavier fluid is on top of lighter fluid and the interface is perturbed. The heavier fluid on top penetrates the lighter fluid and buckles, which generates instabilities. This interface motion is challenging to track due to large changes in its topology. Additionally, the Rayleigh-Taylor instabilities generally encompass turbulent conditions that require resolving finer scales to capture the complete dynamics. We non-dimensionalize the problem by selecting the width of the channel as the characteristic length scale and the density of the lighter fluid as the characteristic specific density. Just as in the bubble rise case we use buoyancy-based scaling, setting the Froude number ( $Fr = u^2/(gD)$ ) to 1, which fixes the non-dimensional velocity scale to be  $u = \sqrt{gD}$ , where  $g$  is the gravitational acceleration, and  $D$  is the width of the channel. Using this velocity to calculate the Reynolds number, we get  $Re = \rho_L g^{1/2} D^{3/2} / \mu_L$ , where  $\rho_L$  and  $\mu_L$  are the specific density and specific viscosity of the light fluid, respectively. We set the Reynolds number to  $Re = 1000$ . These choices lead to a Weber number of  $We = \rho_c g D^2 / \sigma$ . To compare our results with previous studies, we simulate with the same initial conditions as presented in Xie et al. [49], Guermond and Quartapelle [52], Tryggvason [53], Ding et al. [54]. The  $We$  number is selected to be 1000, so that the effect of surface tension is small on the evolution of the interface.

The Atwood number ( $At$ ) is often used to parametrize the dependence on density ratio, with  $At = (\rho_+ - \rho_-) / (\rho_+ + \rho_-)$ . For the density ratios of 0.33, and 0.1, the Atwood numbers are  $At = 0.5$ , and  $At = 0.82$ , respectively. We chose specific density of the heavy fluid to non-dimensionalize, therefore  $\rho_+ = 1.0$ , and  $\rho_- = 0.33$  for  $At = 0.5$ , while  $\rho_- = 0.1$  for  $At = 0.82$ .  $\nu_+/\nu_-$  the viscosity ratio is set to 1. We use a no-slip boundary condition for velocity on all the walls along with no flux boundary conditions for  $\phi$  and  $\mu$ . The no-flux boundary condition for  $\phi$  and  $\mu$  inherently assumes a 90 degree wetting angle for both the fluids.

**Remark 7.** Weak surface tension reduces vortex roll-up in the simulations of immiscible systems, especially at lower  $At$ . Experimental results from Waddell et al. [55] show different vortex roll-up amounts for miscible and immiscible systems. This difference is analogous to the difference between zero surface tension simulations and finite surface tension simulations. This effect is irrelevant to compare front locations against the literature (short time horizons). Nevertheless, it is crucial to accurately track the long time dynamics (as smaller filaments are more stable in the non-zero surface tension case).

We run numerical experiments for  $Cn = 0.005, 0.0025, 0.00125$  with a uniform mesh of  $400 \times 3200$ ,  $400 \times 3200$ , and  $800 \times 6400$ , respectively, for two different Atwood numbers:  $At = 0.82, 0.5$ . The time step size for all the experiments is  $1.25 \times 10^{-4}$ . A carefully tuned algebraic multi-grid linear solver with successive over-relaxation is setup for the linear solves in the Newton iterations (see section 3.3). We detail the command-line arguments used in Appendix B. For the convergence criteria for both 2D Rayleigh-Taylor test cases we use a relative tolerance of  $10^{-6}$  for Newton iteration and a relative tolerance of  $10^{-7}$  for the linear solves.

Figure 4 shows the snapshots of the interface shape along with the corresponding vorticity generated as it evolves in time for  $At = 0.82$  and  $Cn = 0.00125$ . We observe the usual evolution of Rayleigh Taylor instability where the heavier fluid penetrates the light one, causing the lighter fluid to rise near the wall. The penetrating plume of the heavier fluid sheds small filaments at a non-dimensional time of  $t' = 1.358$ . The penetrating plume is symmetric at early times, with symmetry

breaking occurring at longer times. The instability further proceeds to a periodic flapping. At longer times, the instability transitions to a chaotic mixing stage.<sup>2</sup>

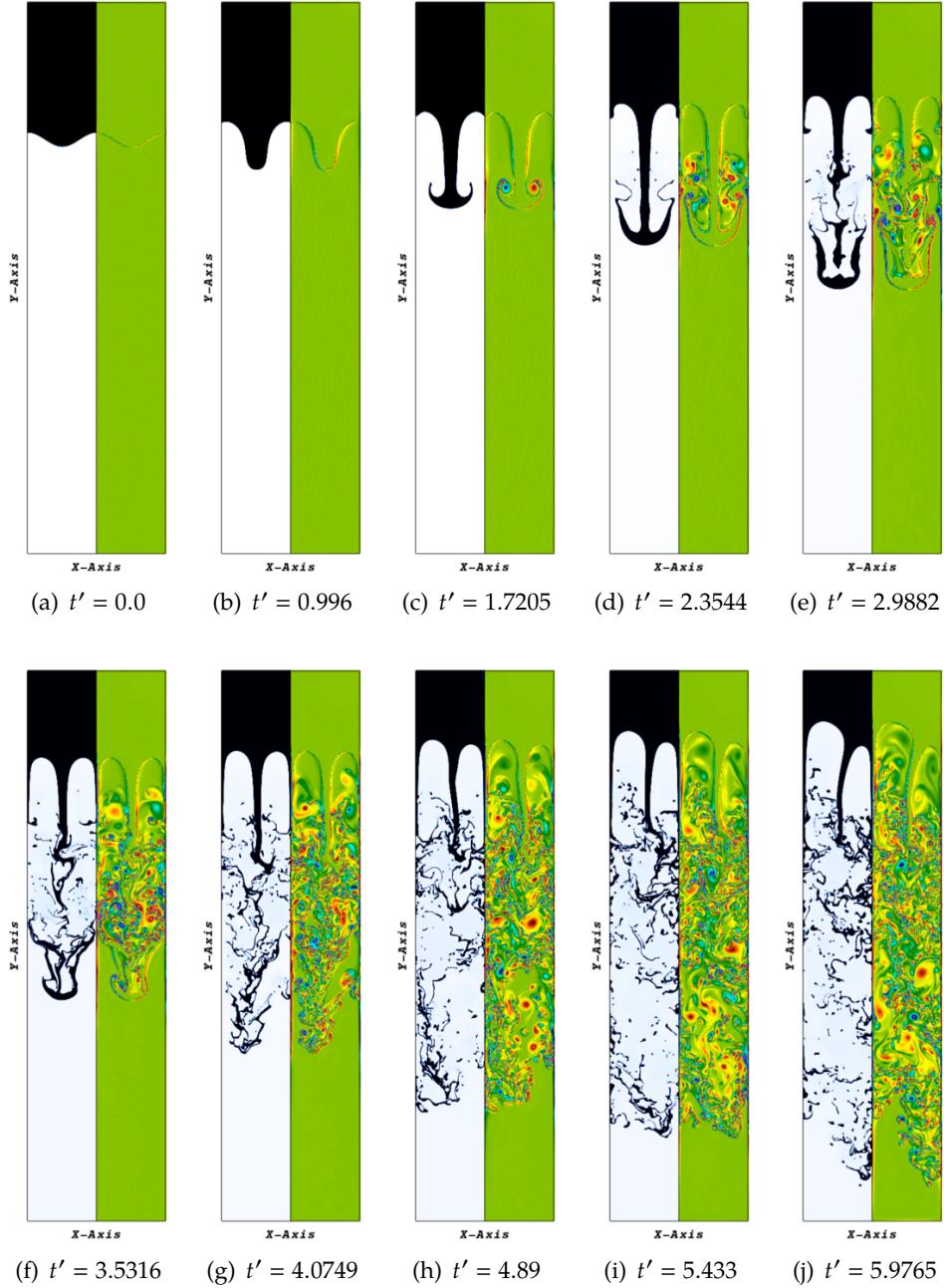


Figure 4: *Rayleigh-Taylor instability in 2D* (section 5.3): Dynamics of the interface as a function of time for  $At = 0.82$  (density ratio of 0.1). In each panel the left plot illustrates the interface, and the right plot shows corresponding vorticity. Here normalised time  $t' = t\sqrt{At}$

---

<sup>2</sup>These high-resolution simulations of Rayleigh Taylor instability over long time horizons could serve as benchmarks.

### 5.3.1. Influence of $Cn$ on long time dynamics

We observe from fig. 4 that the development of the instability (at longer times) depends on the resolution of shedding filaments. Therefore, the long-time dynamics depend on the interface thickness that the  $Cn$  number controls. To analyze the influence of  $Cn$  number on the development of the Rayleigh-Taylor instability, we perform numerical experiments with three  $Cn$  numbers. First, we compare our results with those in the literature to validate the framework. For the cases of  $At = 0.5$  and  $At = 0.82$  several previous studies presented the location of the top and bottom fronts as a function of time. Panel (a) fig. 5 compares the bottom and top front locations with previous studies [49–52]. Our current results match the previous benchmarks for all three  $Cn$  numbers.

Panels (b) and (c) from fig. 5 show the energy decay in line with theorem 1 for  $At = 0.5, 0.82$ . We observe energy stability for all the three  $Cn$  numbers.

At longer times, defining the top/bottom front becomes difficult due to filament breakup. Therefore, we plot the center of mass of the heavy fluid as a function of time. This location is a good integral metric to track coarse-scale dynamics. Panels (a) and (b) of Figure 6 show the evolution of center of mass of the heavy fluid for  $At = 0.5$  and  $At = 0.82$ , respectively. We observe a convergence of dynamics as we decrease the  $Cn$  number. There are some deviations even for small  $Cn$  numbers at longer time durations. The chaotic filament breakup and concomitant fluid features due to the relative motion of the interface cause these deviations. We explore them by visualizing the development of coherent vortices using the  $Q$ -criterion [56]. Figure 7(a) shows that  $Cn = 0.005$  under-resolves the filaments shed as the instability develops. This lack of sufficient resolution causes under-resolution of fine-scale vortices which depend on the shear instability generated by the finer filaments. We observe these finer filaments as we decrease the  $Cn$  number; fig. 7(b) shows the interface and the corresponding  $Q$ -criterion for  $Cn = 0.0025$  at the same time point as  $Cn = 0.005$ . We resolve these finer filaments better in this case. Upon further reduction of  $Cn$  to 0.00125, fig. 7(c) captures much finer flow structures compared to  $Cn = 0.0025$ . We observe that even if integral metrics like front locations and center of mass match for two  $Cn$  numbers (0.005 and 0.0025 in our case, see fig. 6), the fine-scale flow structures can be quite different due to their dependence on the resolution of the finer filaments. This fine structure resolution has a profound influence on the higher-order statistics of Rayleigh-Taylor instability. We defer a detailed analysis of higher-order statistics to future work.

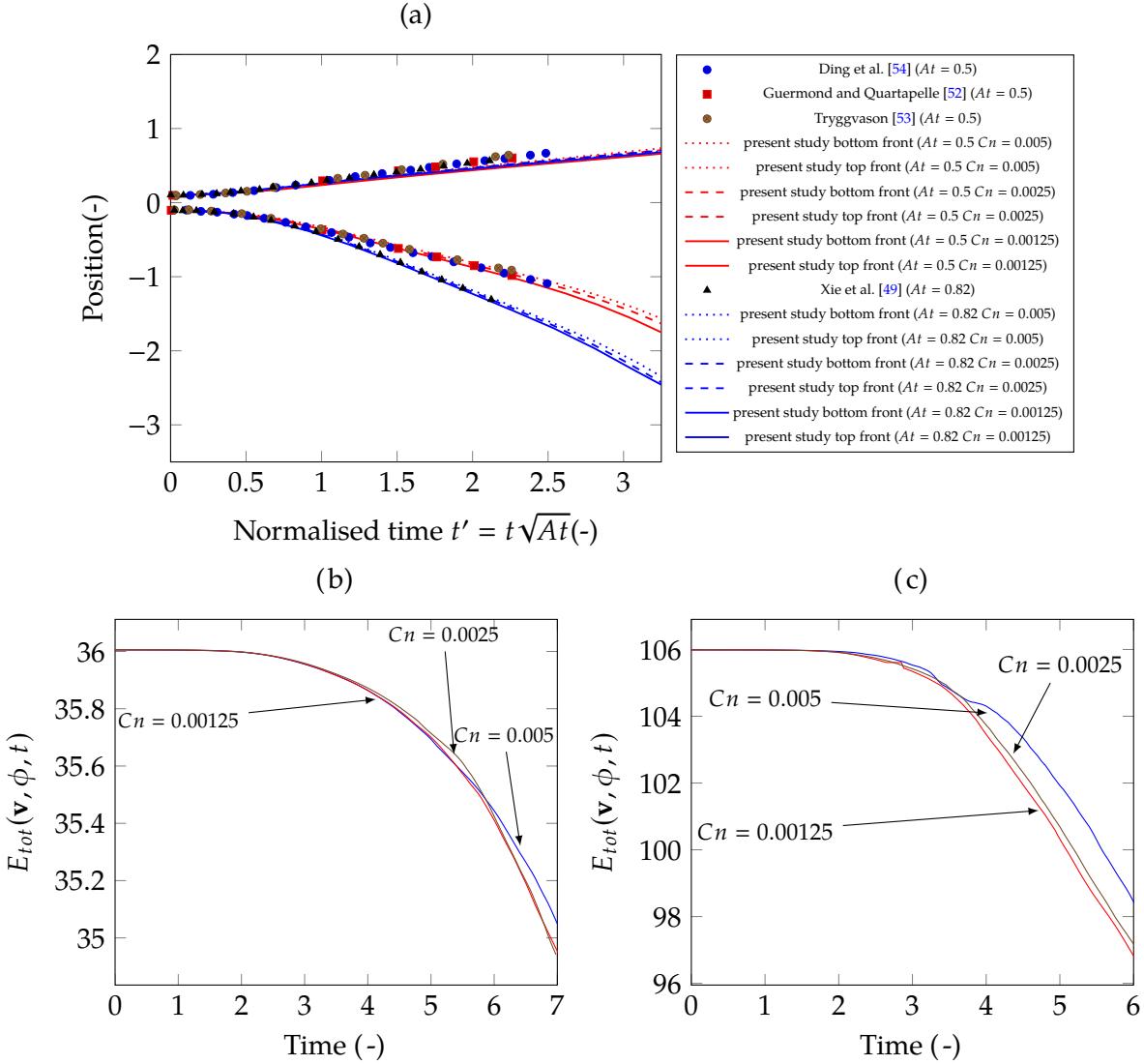


Figure 5: *Rayleigh-Taylor instability 2D* (section 5.3): (a) Comparison of positions of top and bottom front of the interface with literature; (b) decay of the energy functional illustrating theorem 1 for  $At = 0.5$ ; (c) decay of the energy functional illustrating theorem 1 for  $At = 0.82$

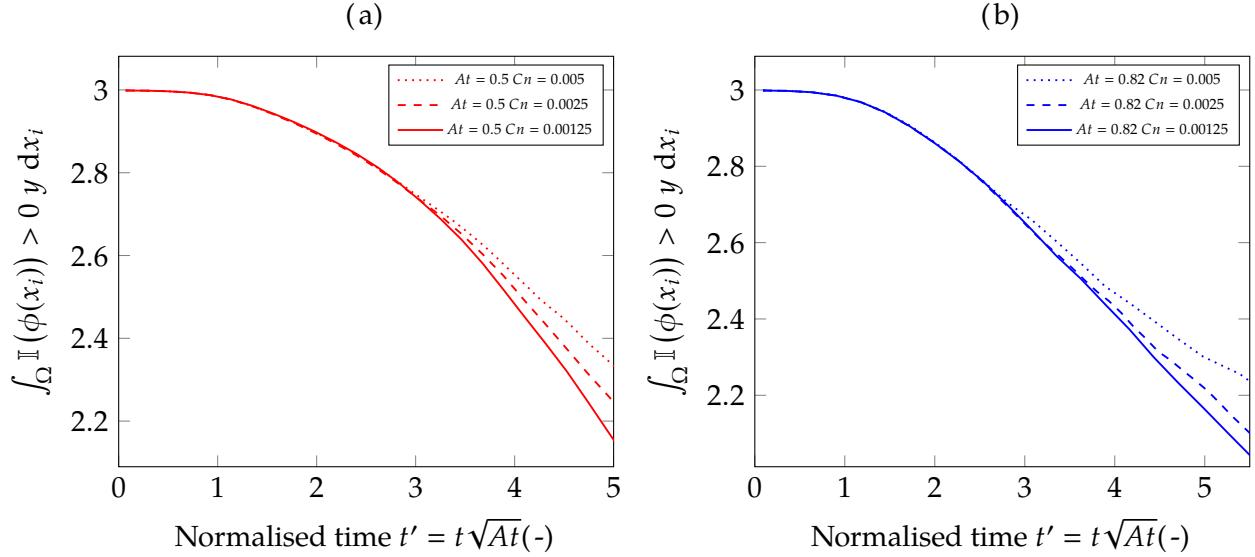


Figure 6: *Rayleigh-Taylor instability 2D* (section 5.3): Comparison of centroids of heavy fluid for different  $Cn$  numbers; (a)  $At = 0.5$ ; (b)  $At = 0.82$ .

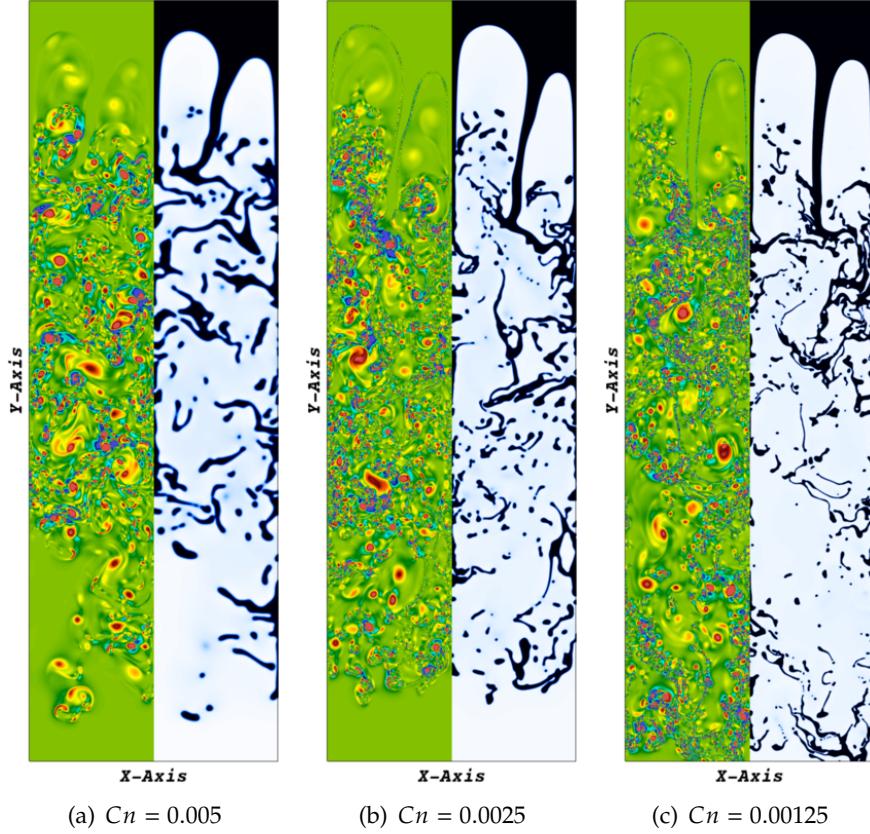


Figure 7: *Rayleigh-Taylor instability in 2D* (section 5.3):  $Q$ -criterion of Rayleigh Taylor instability for  $At = 0.82$  (density ratio of 0.1) at  $t' = t \sqrt{At} = 5.6143$ . In each panel the left plot illustrates the  $Q$ -criterion, and the right plot shows corresponding interface location. These plots are zoomed insets of the domain near the interfacial instabilities.

Finally, we report on the numerical mass conservation properties of the proposed scheme for the 2D Rayleigh-Taylor experiments. Panel (a) of fig. 8 shows the change in the total mass with respect to the initial total mass. We observe that it is of the order of  $10^{-4}$ . Therefore, we see excellent mass conservation even with a high amount of deformation of the interface over very large time horizons (over 30,000 time steps), especially in the presence of fine filaments that are clearly under-resolved for computationally tractable  $Cn$  numbers. We see that there is some deterioration in mass conservation for the smallest  $Cn$  number, again due to the under-resolution of thin filaments. We see similar behavior of mass conservation for  $At = 0.82$  for all three  $Cn$  numbers; these results are shown in Panel (b) of fig. 8.

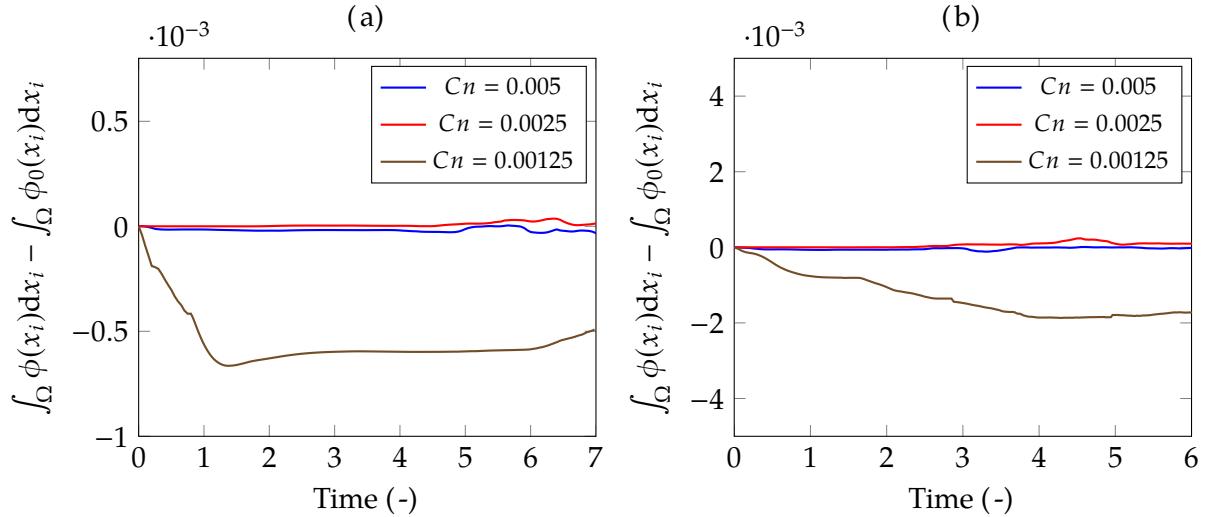


Figure 8: *Rayleigh-Taylor instability 2D* (section 5.3): (a) total mass conservation (integral of total  $\phi$ ) for  $At = 0.5$ ; (b) total mass conservation (integral of total  $\phi$ ) for  $At = 0.82$ .

#### 5.4. 3D simulations: Rayleigh-Taylor instability

Next we deploy our framework in 3D and simulate the Rayleigh-Taylor instability in 3D using adaptive octree meshes. For the 3D simulations we choose the following initial condition for  $\phi$  to describe the interface:

$$\phi(\mathbf{x}) = \tanh\left(\sqrt{2}\left[\frac{x_2 - h_0 - g(\mathbf{x})}{Cn}\right]\right), \quad (77)$$

$$g(\mathbf{x}) = 0.05 [\cos(2\pi x_1) + \cos(2\pi x_3)]. \quad (78)$$

Here  $h_0$  is the location in the vertical direction for the interface, which in this case is chosen to be twice the characteristic length from the bottom of the channel. Typical simulations in the literature choose a rectangular domain that only captures one wavelength of the initial condition (e.g., [50]). To illustrate the advantage of the adaptive octree framework, we choose to include four wavelengths in the initial condition, resulting in a larger domain. Figure 9 shows the initial condition, along with the schematics of the computational domain. We use a  $Cn = 0.0075$  and  $At = 0.15$ . For this lower  $At$  number simulation, the effect of non-zero surface tension is important. The non-dimensionalization follows the same logic as the 2D cases, with the Reynolds number set to 1000, the Weber number ( $We = \rho_c g D^2 / \sigma$ ) set to 1000, and viscosity ratio,  $\nu_+/\nu_-$ , set to 1.

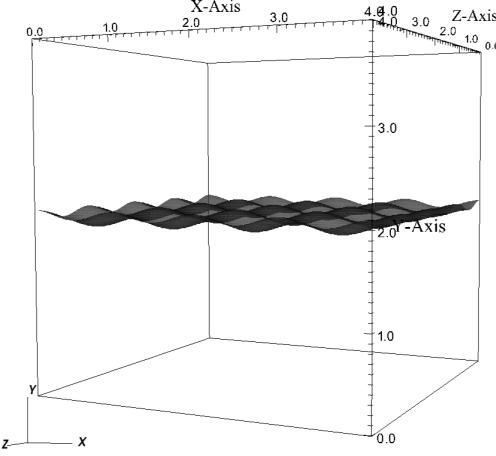


Figure 9: 3D Rayleigh-Taylor instability (section 5.4): shown here is the computational domain with the iso-surface  $\phi = 0$  showing the initial condition of the interface.

460 Due to the energy stability of the proposed numerical method we are able to take a reasonably large time-step size of  $\delta t = 0.0025$ . We refine near the interface to a level corresponding to element length of  $4/2^8$ , ensuring about three elements for resolving the diffuse interface, while the refinement away from the interface is  $4/2^4$ . Similar to the 2D cases, the boundary conditions are no-slip for velocity and no-flux for  $\phi$  and  $\mu$  on all the walls. In Appendix B we provide a detailed 465 description of the preconditioners and linear solvers along with the command-line arguments that are used. The convergence criterion for all the 3D Rayleigh-Taylor simulations use a relative tolerance of  $10^{-6}$  for Newton iteration and a relative tolerance of  $10^{-6}$  for the linear solves within each 470 Newton iteration.

475 Figure 10 shows the evolution of the interface along with the solution-adapted mesh. We color the mesh with the order parameter value (blue for the heavy fluid and white for the light fluid) to show the evolution of the system. Shown in this figure is that as the interface evolves it deforms and expands, causing the mesh density to gradually increase. This gradual growth helps with the efficiency of the simulation, since a uniform mesh for this case would be computationally intractable. At the point with the largest interfacial area, the simulation has around 95 million elements, corresponding to 1.7 billion degrees of freedom. The efficient and scalable implementation 480 of the approach allows us to run this large scale simulation on Stampede2 with 256 KNL nodes. We present the scalability of the approach later in the scaling section.

485 Figure 11 shows that the initial sinusoidal perturbation develops into penetrating plumes of heavy fluid pushing down while the lighter fluid buckles and forms bubbles. The simulation maintains symmetry until the mixing becomes chaotic at long times; this is similar to the results in the 2D case. As different parts of the interface move in opposite directions, Kelvin-Helmholtz instabilities cause the plumes to roll up, which in turn causes mushroom-like structures to develop (see fig. 11(d)). Although these two types of spikes (upwards/downwards) begin to develop in a checkerboard pattern that preserves symmetry, their dynamics are different due to the velocity differential that the two fronts face.

490 The downward spikes undergo further deformation and we see the emergence of four long filaments from the mushroom structure (see fig. 11(g)) caused by the shear generated between the fluids. Liang et al. [57] and Jain et al. [58] also report four secondary filaments in their simulations for the same  $At$  number, although their simulations were for zero surface tension (i.e., dynamics similar to miscible systems).

On the other hand, the mushroom structures from the upward spikes develop into long and

thin circular films. The upward spikes develop circular films adjacent to the wall "bubbles" (i.e., structures near the wall that the heavier fluid generates as it is displaced by the lighter one) interact with these bubbles to merge and form larger structures (see fig. 11(h)). While the central plumes have little-to-no interaction with the wall, the bubbles continue to rise and ultimately collide with the top wall.

Another important difference between the upward and downward spikes is their rate of growth. Figures 11(e) to 11(h) show that the fronts of the upward spikes move slower than the fronts of the downward spikes due to the density differential. Figure 11(i) shows that the mushroom structures collide with top and bottom walls, which then leads to further breakup that creates the conditions for chaotic mixing. To the best of our knowledge, this is the first analysis in the literature of the dynamics for multiple wave single-mode instabilities.

### 5.5. 3D simulations: lid-driven cavity

Our next example exhibits considerable deformation of the interface. The setup consists of a regularized lid-driven cavity, such that there are no corner singularities due to the imposed velocities. Half of the cubical domain contains fluid 1, the other half contains fluid 2. The fluids have finite surface tension. Figure 12 shows the domain. Initially, the interface between the two fluids is flat. We use the length of the domain as the non-dimensionalizing length scale. The key non-dimensional numbers are  $Re = 100$  (laminar),  $We = 100$ ,  $Cn = 0.005$ , and  $Pe = 13333$ . The viscosity and density ratios are 1. We adaptively refine the mesh with the finest element size near the interface being  $1/(2^8)$ , the coarsest element size away from the interface being  $1/(2^4)$ , and wall element size is  $1/(2^6)$ . Boundary conditions are no-slip for velocity on all the walls except the top wall, and no flux conditions for  $\phi$  and  $\mu$ . For the top wall boundary conditions the  $y$  and  $z$ -velocities are set to zero, while the  $x$ -velocity is given by following function:

$$v_1 = 2^4 x_1 (1 - x_1) x_3 (1 - x_3). \quad (79)$$

The velocity goes to zero on all the corners, thus avoiding corner singularities. Although, Chakravarthy and Ottino [59], Chella and Viñals [60], Park et al. [61] studied this problem in 2D, but it was not explored in 3D to date.

The dynamics of the system with  $Re = 100$  is in the laminar regime. In single-phase systems, a vortex develops near the top wall as the cavity flow evolves. This vortex forms for the two-phase case, as fig. 13 displays. The vortex in fig. 13 rolls up the flat interface up the sidewall and causes a thin wetting layer to form on the top wall. Figure 14 shows how this layer progressively moves until it rolls back into the center of the box. The interface rapidly rolls up as it moves towards the domain center; it moves away from walls since the velocity goes to zero near the wall. Chakravarthy and Ottino [59], Chella and Viñals [60], Park et al. [61] described this behavior in 2D simulations. The stability of the thin layer at the top wall depends on the surface tension strength ( $We$  number). We simulate a moderate  $We$  number which allows mixing and deformation but the film at the top is stable; for larger  $We$  numbers the film at the top starts breaking up into bubbles. We use a relative tolerance of  $10^{-6}$  for Newton iteration. For the linear solves within each Newton iteration we use a relative tolerance of  $10^{-6}$ .

The previous two examples illustrate the capability of the framework to capture fairly complex interfacial motion in 3D due to its robust and efficient implementation, which we discuss next.

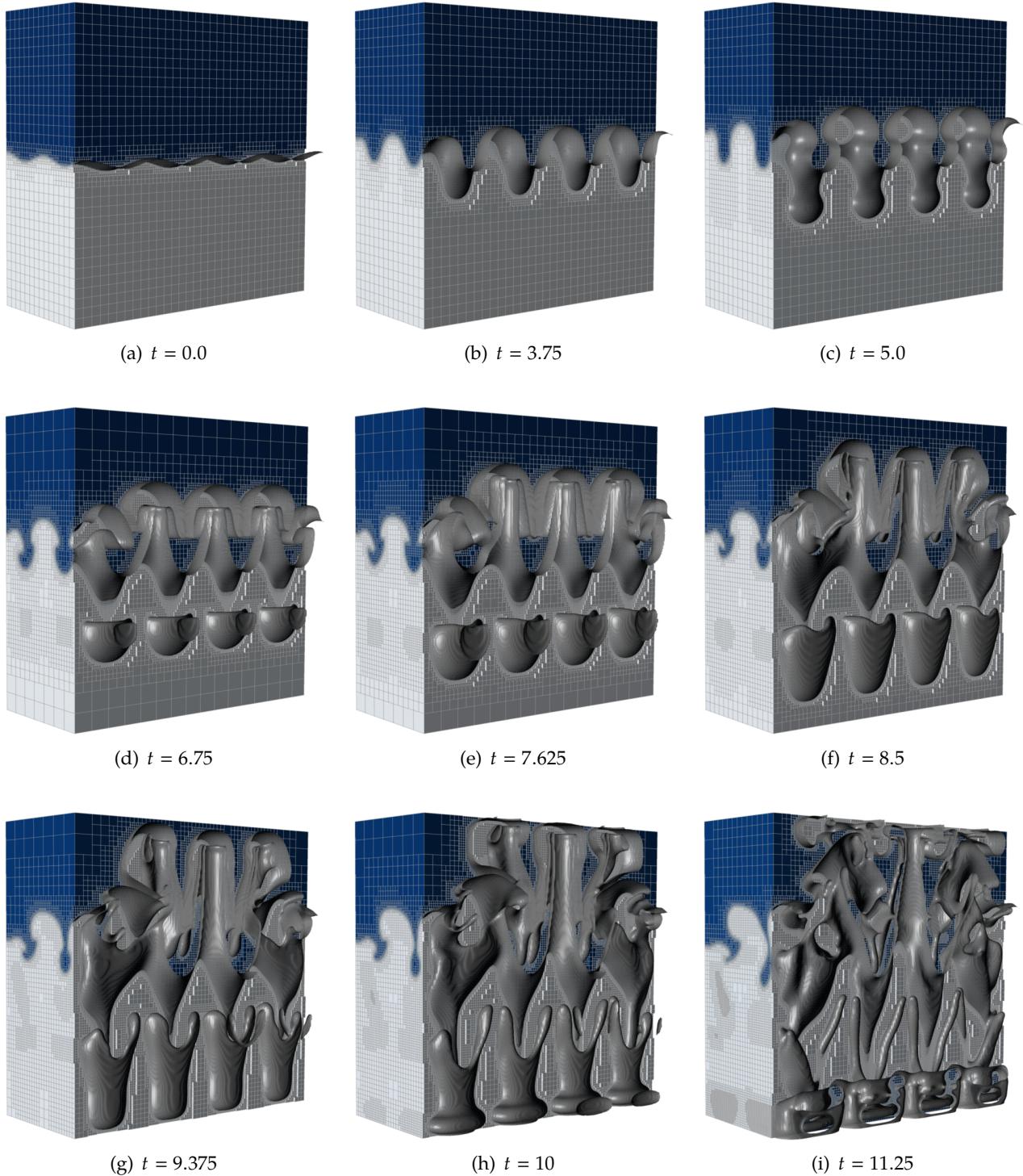


Figure 10: *Rayleigh-Taylor instability in 3D* (section 5.4): Snapshots of the mesh at various time-points in the simulation for Rayleigh-Taylor instability for  $At = 0.15$ . The figures show half of the mesh of the actual domain to illustrate the refinement around the interface of two fluids represented by the gray iso-surface of  $\phi = 0$ . The phase field  $\phi$  values color the mesh, where blue represents heavy fluid and white represents light fluid. Here  $t(-)$  is the non-dimensional time.

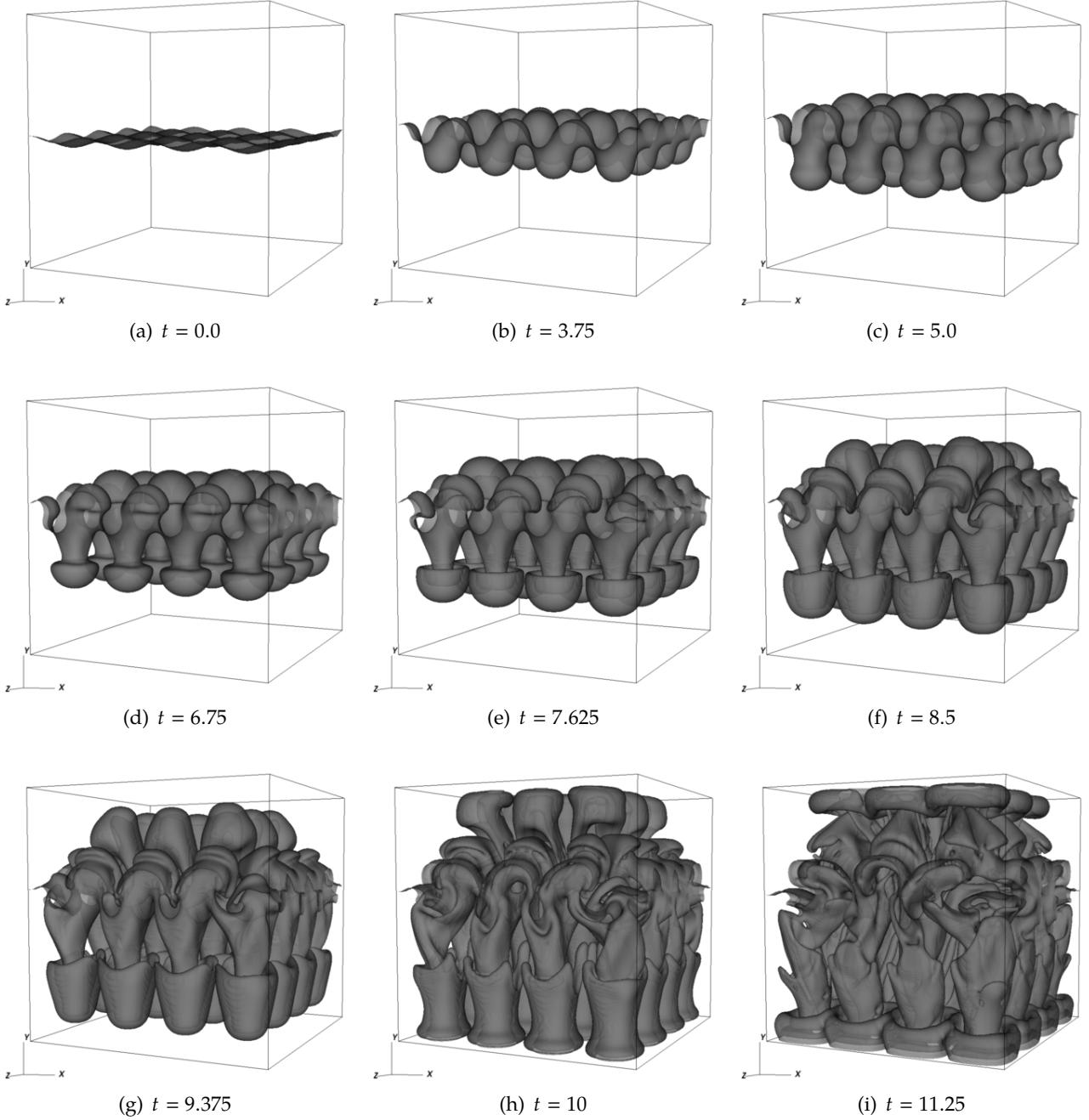


Figure 11: *Rayleigh-Taylor instability in 3D* (section 5.4): Snapshots of the zero isosurface of  $\phi$  which represents the interface at various time-points in the simulation for Rayleigh-Taylor instability for  $At = 0.15$ . Here  $t(-)$  is the non-dimensional time.

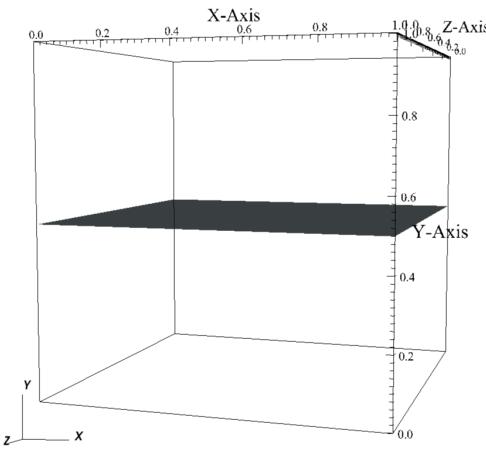


Figure 12: Schematic of the computational domain used for lid-driven cavity (section 5.5) with the iso-surface  $\phi = 0$  showing the initial condition of the interface.

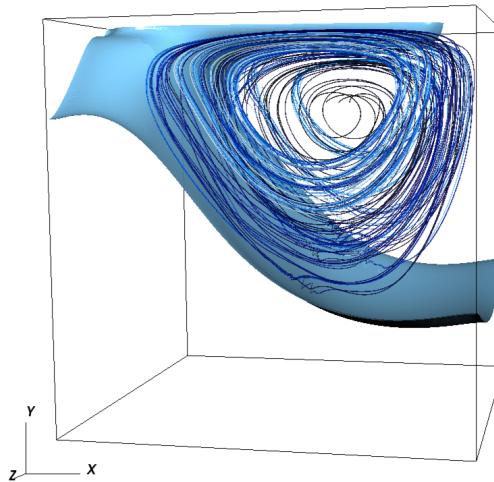


Figure 13: Streamlines of velocity overlayed on the interface for lid-driven cavity (section 5.5) at  $t = 16.5$ .

## 6. Scaling of the numerical implementation

### 6.1. Strong scaling

We perform a detailed timing analysis to demonstrate the parallel scalability of the framework. We perform all scaling tests on TACC's [Stampede2](#) Knights Landing processors ranging from 1 node to 256 nodes with 68 processors per node. We study in detail the lid-driven cavity case in section 5.5 for the scaling analysis. We run each scaling experiment for ten time steps such that the initialization and setup do not dominate the timing. Three different levels of refinement characterize each mesh: 1. background mesh refinement ( $L_{\text{bkg}}$ ); 2. wall refinement ( $L_{\text{wall}}$ ); and 3. interface refinement ( $L_{\text{interface}}$ ). As the interface evolves, the mesh is subsequently refined near the interface and coarsened away from it. Table 2 shows the level of refinement and the approximate number of total elements for each of the three different meshes used for the scaling study.

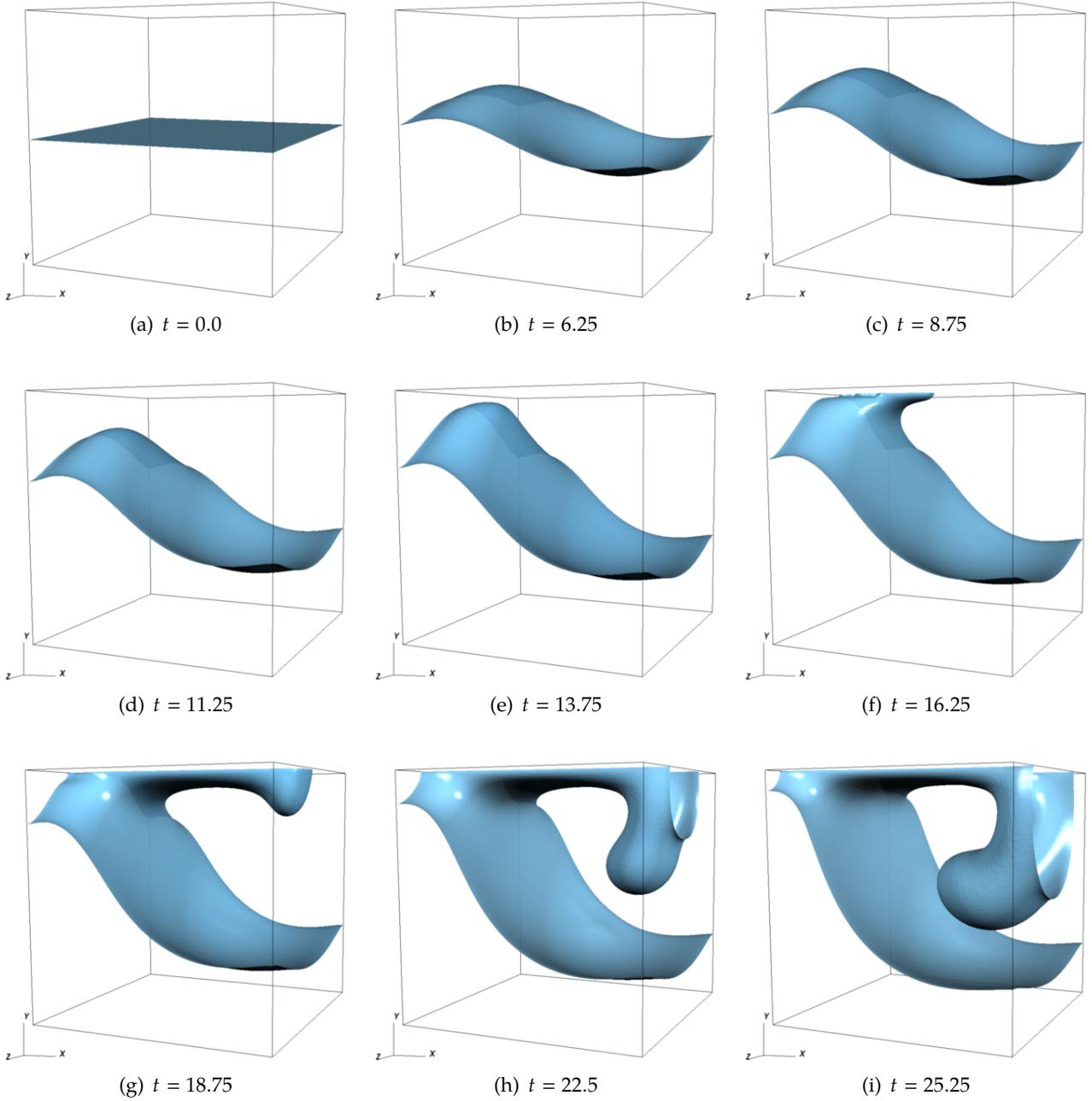


Figure 14: *Mixing of two fluids in a cubic driven cavity* (section 5.5): Snapshots of the zero iso-surface of  $\phi$  which represents the interface at various time-points in the simulation for Lid-driven cavity. Here  $t(-)$  is the non-dimensional time.

	$L_{\text{bkg}}$	$L_{\text{wall}}$	$L_{\text{interface}}$	$N_{\text{elem}}$
M1	3	5	7	280 K
M2	4	6	8	1 M
M3	5	7	9	19 M

Table 2: Level of refinements involved in different meshes used for scaling studies

Panel (a) of fig. 15 shows the result of strong scaling for three different meshes, and panel (b) shows the corresponding relative speedup. Figure 15 demonstrates that we achieve a better than ideal scaling for these cases. To analyze the reasons behind this, we analyze component-by-component framework timings and observe that the two dominant ones in the total run time are the Jacobian assembly ( $J$  in section 3) and the preconditioner setup (PC setup in PETSc), with PC setup dominating the timing (see fig. 17). Therefore, to understand the scaling behavior, we plot the strong scaling curves and corresponding relative speed up for these two components in fig. 16. Panels (a) and (b) of fig. 16 detail the scaling for Jacobian assembly, whereas, panels (c) and (d) show the scaling for PC setup. As expected, the Jacobian assembly scales almost ideally for all the cases considered from panels (a) and (b) of fig. 16. However, the PC setup has better than ideal scaling, see panels (c) and (d). As the PC setup dominates the total solution time, the overall scaling mimics this behavior.

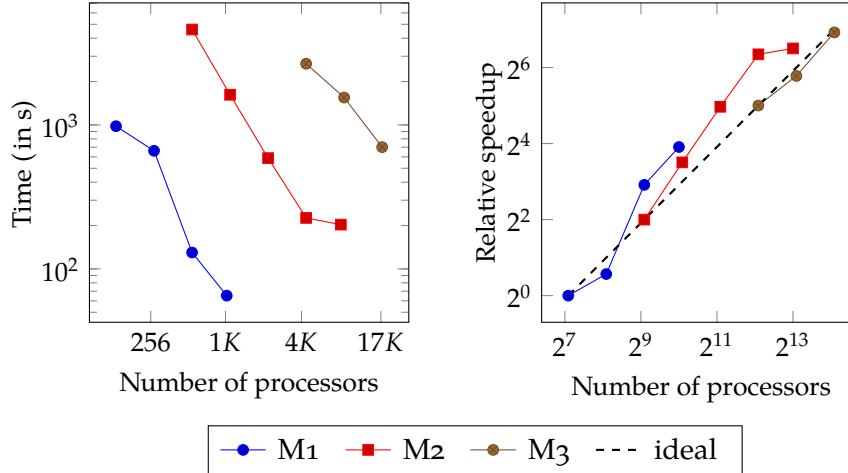


Figure 15: *Strong scaling:* Shown in the panels are (a) the strong scaling behavior and (b) the relative speedup for total solve time for our solver on Stampede2 Knights Landing processor. Three different mesh are considered: M1 with 280 K elements, M2 with 1 M elements and M3 with 19 M elements. Excellent scaling behavior is observed up to  $\mathcal{O}(17K)$  processors.

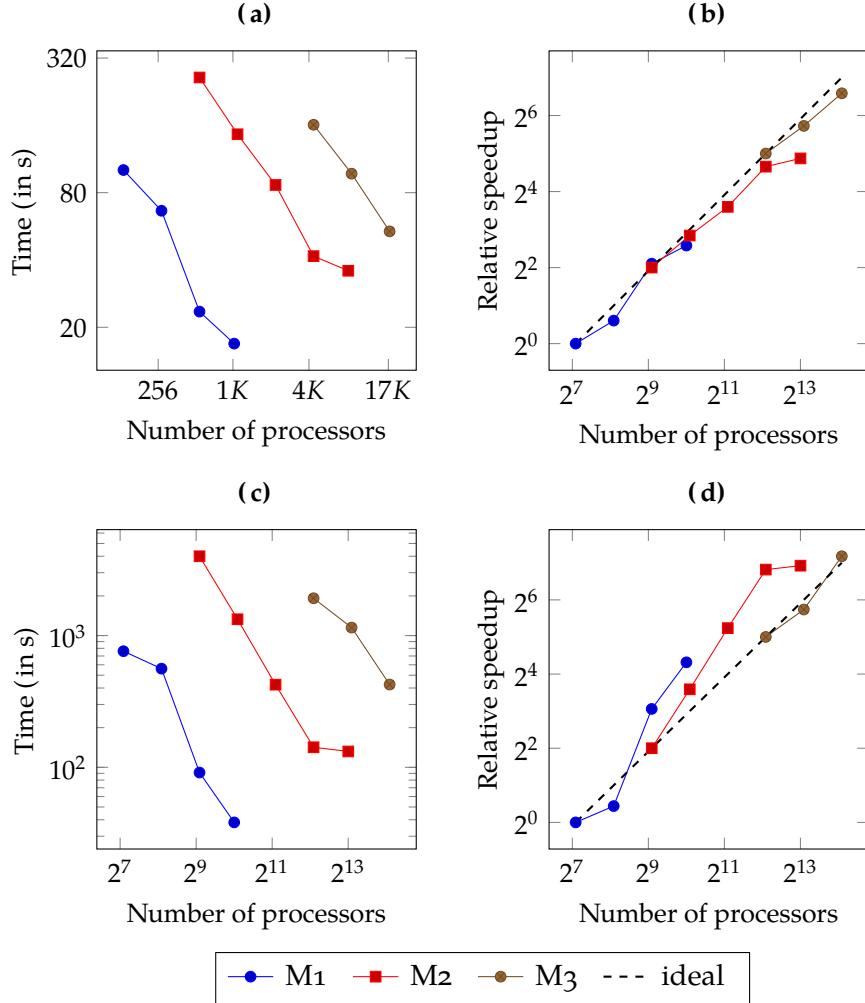


Figure 16: *Strong scaling of components*: Figure showing the segregated strong scaling behaviour of two major component of our solver on [Stampede2](#) Knights Landing processor: (a) shows Jacobian assembly strong scaling; (b) shows relative speedup for Jacobian assembly; (c) shows preconditioner setup; (d) shows relative speedup for Preconditioner setup.

The deep memory hierarchy inherent in modern-day clusters may be the cause for the better than ideal scaling behavior. We use an additive Schwarz-based block preconditioning with each block using LU factorization (-sub\_pc\_type in PETSc) (see [Appendix B](#)) to compute the factors of the block matrix. The factorized block preconditioned matrix loses its sparsity, and the resultant matrix is less sparse ( $b \times b$ ) matrix, where  $b$  is the block size of the matrix on a processor. This denser matrix can no longer fit in L1, L2, or L3 cache for a big enough problem size. As the number of processors increases, the resulting problem size in a processor diminishes. Thus, these denser blocks begin to fit in cache, and therefore, we achieve a better than ideal speedup for PC setup and subsequent solve time.

Figure 17 shows the relative fraction of time spent in different meshes. We observe two specific trends. First, as the number of processors increases such that number of elements remains constant, the relative cost of PC setup decreases. This is because of the fact that LU factorization is performed on a smaller block matrix. Also, the cost of communication while performing matrix assembly and vector assembly increases with increase in number of processor, whereas LU factorization is performed on a block matrix and thus requires no communication. Secondly, increasing the number

of elements such that the number of processor is fixed has a substantial effect on the relative cost of PC setup. LU factorization is an  $\mathcal{O}(N^3)$  operation, whereas the number of FLOPS and communication involved in other operations like matrix assembly is atmost  $\mathcal{O}(N^2)$ . This explains the increase in the PC setup cost. Future work will seek to develop schemes that will efficiently use cheaper preconditioners, whilst maintaining the energy stability and mass conservation of these schemes.

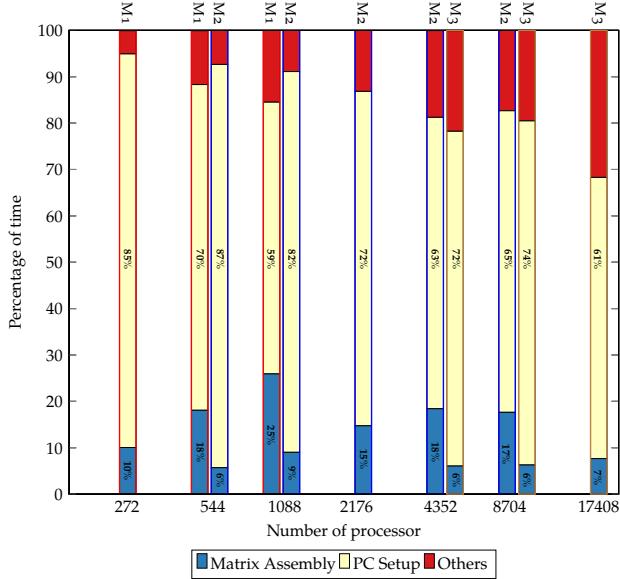


Figure 17: Percentage of time taken for individual components for three different meshes used for strong scaling case. The meshes are labelled at the top.

**Remark 8.** We use a Krylov space solver (BCGS) with ASM/LU preconditioning. Therefore, algebraic multi-grid solvers with PETSC's AMG can substantially improve the solve time. Although we had some success with this setup (e.g., the 2D Rayleigh Taylor in section 5.3 cases use this setup), the 3D examples use the ASM/LU combination.

## 6.2. Weak scaling

Performing exact weak scaling<sup>3</sup> is non-trivial due to the adaptive nature of the mesh. Therefore, we consider simulations with an approximately equal number of elements per processor to deduce the weak scaling nature of our solver. Figure 18 shows the weak scaling results for the different problem sizes. Overall, similar to the strong scaling, we see an excellent weak scaling efficiency for different problem sizes. For each of the cases, we observe a weak scaling efficiency of greater than 0.5 for a 64 fold increase in the problem size. We also analyze the weak scaling for the dominant components of the overall time, i.e. Jacobian assembly and PC setup. Figure 19 segregates the time for these components. The weak scaling efficiency ( $> 0.5$ ) for each of these individual components is excellent as we vary the problem size by two orders of magnitude.

<sup>3</sup>Weak scaling studies the variation in simulation time as we increase the number of processors with a fixed problem size per processor.

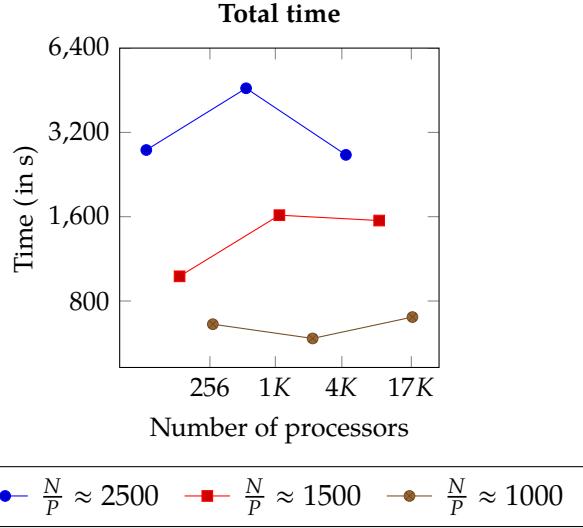


Figure 18: *Weak scaling*: Figure showing weak scaling behaviour of total solve time of our solver on Stampede2 Knights Landing processor for different number of elements per processor.

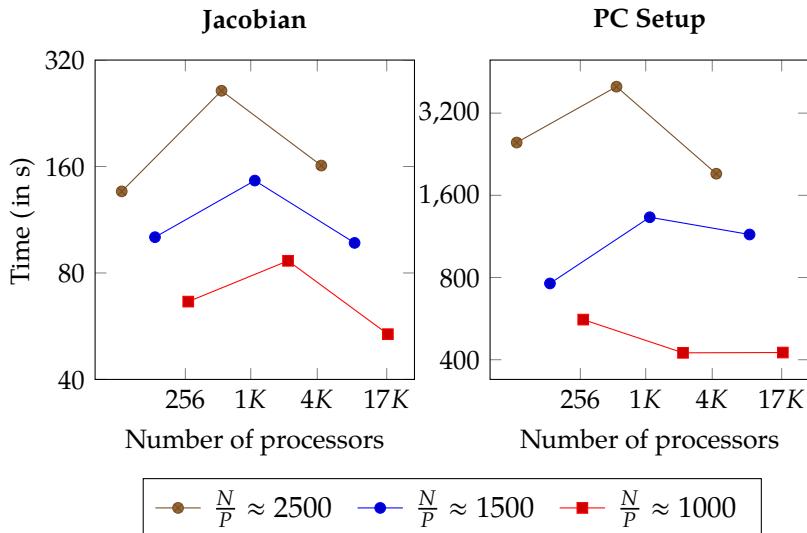


Figure 19: *Weak scaling*: Figure showing the segregated weak scaling behavior of two major components of our solver on Stampede2 Knights Landing processor for different number of elements per processor: Jacobian assembly and preconditioner setup.

The preconditioner complexity and cost challenges (as seen from fig. 17) are also present in the block iterative method of Khanwale et al. [7], where one performs these operation multiple times within one timestep (once every block iteration). This combined cost makes the block iterative method much more expensive compared to the fully coupled solver. Therefore, the current coupled framework improves the overall time to solve by reducing the number of evaluations required for Jacobian assembly and PC setup. In challenging application problems, preconditioning the linear problem (sub-iteration within each Newton solve) requires substantial computational effort; future work will seek improve the speed and efficiencies of the preconditioning.

585 **7. Conclusions and future work**

In this work we developed a numerical framework for solving the Cahn-Hilliard Navier-Stokes (CHNS) model of two-phase flows. We used a continuous Galerkin spatial discretization with linear finite elements and a second-order time-marching scheme. The newly developed scheme is a variant of the energy-stable energy block-iterative scheme used in Khanwale et al. [7], but in this work  
590 we improved both on the formal order of accuracy of the scheme (now second-order accurate) and the efficiency of the Newton iteration. We rigorously proved that the new time-marching scheme maintains energy-stability. We used a variational multiscale approach to stabilize the pressure. The resulting method was implemented using a scalable adaptive meshing framework to perform large  
595 3D simulations of complex multiphase flows. Solution-adapted meshing was accomplished using octrees in the DENDRO package. We presented a comprehensive set of numerical experiments in both 2D and 3D, which were used to validate and test our numerical framework. We also used these numerical experiments to validate our theoretical estimates for the energy-stability and mass  
600 conservation properties of the method. We ran the resulting computational code for the CHNS model on a massively parallel architecture to simulate multi-wave single-mode Rayleigh-Taylor instability. We were able to push the framework to solve up to 2 billion degrees of freedom. The same 3D framework was also used to simulate the mixing of two fluids in a cubic lid-driven cavity. We performed a detailed scaling analysis of the 3D framework and show excellent strong and weak scaling up to  $\mathcal{O}(17K)$  MPI processes.

For future work we plan to perform a detailed analysis of the performance of higher-order basis  
605 functions. As was observed in the scaling study (section 6), the preconditioner setup and Jacobian assembly are dominant parts of the overall solve time; we will aim to reduce the computational costs of these steps. We will also seek to develop projection-based methods which utilize all benefits of VMS approach, while decoupling the pressure row into a separate linear problem. Very efficient multigrid solvers can then be used for both the blocks to make them more efficient and scalable.  
610 These improved solvers will help with adapting the framework to a matrix-free approach to solve the linear problems which are very efficient at large scales as shown by Ishii et al. [62].

**8. Acknowledgements**

The authors acknowledge XSEDE grant number TG-CTS110007 for computing time on TACC Stampede2. The authors also acknowledge computing allocation through a DD award on TACC  
615 Frontera, ALCF computing resources and Iowa State University computing resources. JAR was supported in part by NSF Grant DMS-1620128. BG, KS, MAK were funded in part by NSF grant 1935255, 1855902. HS was funded in part by NSF grant 1912930. This publication was also made possible in part by the CSIRO Professorial Chair in Computational Geoscience at Curtin University and the Deep Earth Imaging Enterprise Future Science Platforms of the Commonwealth Scientific  
620 Industrial Research Organisation, CSIRO, of Australia. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 777778 (MATHROCKS). The Curtin Corrosion Centre and the Curtin Institute for Computation kindly provide ongoing support.

**References**

- 625 [1] D. M. Anderson, G. B. McFadden, A. A. Wheeler, Diffuse-interface methods in fluid mechanics, Annual Review of Fluid Mechanics 30 (1998) 139–165. doi:[doi:10.1146/annurev.fluid.30.1.139](https://doi.org/10.1146/annurev.fluid.30.1.139).

- [2] S. Osher, R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, volume 153 of *Applied Mathematical Sciences*, Springer, 2003.
- [3] D. Jacqmin, An energy approach to the continuum surface tension method, in: 34th AIAA Aerospace Sciences Meeting & Exhibit, volume AIAA96, American Institute of Aeronautics and Astronautics, Reno, Nevada, USA, 1996, p. 0858.
- [4] D. Jacqmin, Contact-line dynamics of a diffuse fluid interface, *Journal of Fluid Mechanics* 402 (2000) 57–88. doi:[doi:10.1017/S0022112099006874](https://doi.org/10.1017/S0022112099006874).
- [5] Z. Guo, P. Lin, J. Lowengrub, S. M. Wise, Mass conservative and energy stable finite difference methods for the quasi-incompressible Navier–Stokes–Cahn–Hilliard system: Primitive variable and projection-type schemes, *Computer Methods in Applied Mechanics and Engineering* 326 (2017) 144–174. doi:[doi:10.1016/j.cma.2017.08.011](https://doi.org/10.1016/j.cma.2017.08.011).
- [6] M. Shokrpour Roudbari, G. Şimşek, E. H. van Brummelen, K. G. van der Zee, Diffuse-interface two-phase flow models with different densities: A new quasi-incompressible form and a linear energy-stable method, *Mathematical Models and Methods in Applied Sciences* 28 (2018) 733–770. doi:[doi:10.1142/S0218202518500197](https://doi.org/10.1142/S0218202518500197).
- [7] M. A. Khanwale, A. D. Lofquist, H. Sundar, J. A. Rossmanith, B. Ganapathysubramanian, Simulating two-phase flows with thermodynamically consistent energy stable cahn-hilliard navier-stokes equations on parallel adaptive octree based meshes, *Journal of Computational Physics* 419 (2020) 109674. doi:[doi:https://doi.org/10.1016/j.jcp.2020.109674](https://doi.org/10.1016/j.jcp.2020.109674).
- [8] S. Xu, Buoyancy-driven flow and fluid-structure interaction with moving boundaries, Ph.D. thesis, Iowa State University, 2018.
- [9] J. Volker, *Finite Element Methods for Incompressible Flow Problems*, volume 51 of *Springer Series in Computational Mathematics Book*, Springer, 2016.
- [10] C. Burstedde, L. C. Wilcox, O. Ghattas, p4est: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees, *SIAM Journal on Scientific Computing* 33 (2011) 1103–1133.
- [11] H. Sundar, R. S. Sampath, S. S. Adavani, C. Davatzikos, G. Biros, Low-constant parallel algorithms for finite element simulations using linear octrees, in: Proceedings of the 2007 ACM/IEEE conference on Supercomputing, ACM, 2007, p. 25.
- [12] H. Sundar, R. S. Sampath, G. Biros, Bottom-up construction and 2:1 balance refinement of linear octrees in parallel, *SIAM J. Sci. Comput.* 30 (2008) 2675–2708. doi:[doi:10.1137/070681727](https://doi.org/10.1137/070681727).
- [13] M. Fernando, H. Sundar, paralab/Dendro-5.01: Parallel adaptive octree library for finite different, finite volume and finite element computations, 2020. URL: <https://doi.org/10.5281/zenodo.3879315>. doi:[doi:10.5281/zenodo.3879315](https://doi.org/10.5281/zenodo.3879315).
- [14] J. Kim, K. Kang, J. Lowengrub, Conservative multigrid methods for Cahn–Hilliard fluids, *Journal of Computational Physics* 193 (2004) 511–543. doi:[doi:10.1016/j.jcp.2003.07.035](https://doi.org/10.1016/j.jcp.2003.07.035).
- [15] X. Feng, Fully Discrete Finite Element Approximations of the Navier–Stokes–Cahn–Hilliard Diffuse Interface Model for Two-Phase Fluid Flows, *SIAM Journal on Numerical Analysis* 44 (2006) 1049–1072. doi:[doi:10.1137/050638333](https://doi.org/10.1137/050638333).

- 665 [16] J. Shen, X. Yang, A phase-field model and its numerical approximation for two-phase incompressible flows with different densities and viscosities, *SIAM Journal on Scientific Computing* 32 (2010) 1159–1179. doi:[doi:10.1137/09075860X](https://doi.org/10.1137/09075860X).
- 670 [17] J. Shen, X. Yang, Energy stable schemes for Cahn-Hilliard phase-field model of two-phase incompressible flows, *Chinese Annals of Mathematics, Series B* 31 (2010) 743–758. doi:[doi:10.1007/s11401-010-0599-y](https://doi.org/10.1007/s11401-010-0599-y).
- 675 [18] D. Han, X. Wang, A second order in time, uniquely solvable, unconditionally stable numerical scheme for Cahn–Hilliard–Navier–Stokes equation, *Journal of Computational Physics* 290 (2015) 139–156. doi:[doi:10.1016/j.jcp.2015.02.046](https://doi.org/10.1016/j.jcp.2015.02.046). arXiv:1407.7048.
- [19] Y. Chen, J. Shen, Efficient, adaptive energy stable schemes for the incompressible Cahn–Hilliard Navier–Stokes phase-field models, *Journal of Computational Physics* 308 (2016) 40–56. doi:[doi:10.1016/j.jcp.2015.12.006](https://doi.org/10.1016/j.jcp.2015.12.006).
- 680 [20] J. Shen, J. Xu, J. Yang, The scalar auxiliary variable (sav) approach for gradient flows, *Journal of Computational Physics* 353 (2018) 407–416.
- [21] G. Zhu, H. Chen, J. Yao, S. Sun, Efficient energy-stable schemes for the hydrodynamics coupled phase-field model, *Applied Mathematical Modelling* 70 (2019) 82–108.
- 685 [22] T. J. Hughes, L. Mazzei, K. E. Jansen, Large eddy simulation and the variational multiscale method, *Computing and Visualization in Science* 3 (2000) 47–59.
- [23] N. Ahmed, T. Chacón Rebollo, V. John, S. Rubino, A review of variational multiscale methods for the simulation of turbulent incompressible flows, *Archives of Computational Methods in Engineering* 24 (2017) 115–164. doi:[doi:10.1007/s11831-015-9161-0](https://doi.org/10.1007/s11831-015-9161-0).
- 690 [24] Y. Bazilevs, V. Calo, J. Cottrell, T. Hughes, A. Reali, G. Scovazzi, Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows, *Computer Methods in Applied Mechanics and Engineering* 197 (2007) 173–201.
- [25] T. Coupez, E. Hachem, Solution of high-Reynolds incompressible flow with stabilized finite element and adaptive anisotropic meshing, *Computer Methods in Applied Mechanics and Engineering* 267 (2013) 65–85. doi:[doi:10.1016/j.cma.2013.08.004](https://doi.org/10.1016/j.cma.2013.08.004).
- 695 [26] E. Hachem, S. Feghali, R. Codina, T. Coupez, Anisotropic adaptive meshing and monolithic Variational Multiscale method for fluid-structure interaction, *Computers and Structures* 122 (2013) 88–100. doi:[doi:10.1016/j.compstruc.2012.12.004](https://doi.org/10.1016/j.compstruc.2012.12.004).
- [27] E. Hachem, M. Khalloufi, J. Bruchon, R. Valette, Y. Mesri, Unified adaptive Variational MultiScale method for two phase compressible-incompressible flows, *Computer Methods in Applied Mechanics and Engineering* 308 (2016) 238–255. doi:[doi:10.1016/j.cma.2016.05.022](https://doi.org/10.1016/j.cma.2016.05.022).
- 700 [28] M. Ishii, M. Fernando, K. Saurabh, B. Khara, B. Ganapathysubramanian, H. Sundar, Solving pdes in space-time: 4d tree-based adaptivity, mesh-free and matrix-free approaches, in: SC'19: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 1–61. URL: <https://doi.org/10.1145/3295500.3356198>. doi:[doi:10.1145/3295500.3356198](https://doi.org/10.1145/3295500.3356198).

- [29] H. Bao, J. Bielak, O. Ghattas, L. F. Kallivokas, D. R. O'Hallaron, J. R. Shewchuk, J. Xu, Large-scale simulation of elastic wave propagation in heterogeneous media on parallel computers, *Computer methods in applied mechanics and engineering* 152 (1998) 85–102.
- [30] M. Fernando, D. Neilsen, H. Lim, E. Hirschmann, H. Sundar, Massively parallel simulations of binary black hole intermediate-mass-ratio inspirals, *SIAM Journal on Scientific Computing* 41 (2019) C97–C138. URL: <https://doi.org/10.1137/18M1196972>. doi:[doi:10.1137/18M1196972](https://doi.org/10.1137/18M1196972). arXiv:<https://doi.org/10.1137/18M1196972>.
- [31] H. Sundar, R. S. Sampath, G. Biros, Bottom-up construction and 2:1 balance refinement of linear octrees in parallel, *SIAM Journal on Scientific Computing* 30 (2008) 2675–2708.
- [32] C. Burstedde, L. C. Wilcox, O. Ghattas, p4est: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees, *SIAM Journal on Scientific Computing* 33 (2011) 1103–1133. doi:[doi:10.1137/100791634](https://doi.org/10.1137/100791634).
- [33] M. Fernando, D. Duplyakin, H. Sundar, Machine and application aware partitioning for adaptive mesh refinement applications, in: *Proceedings of the 26th International Symposium on High-Performance Parallel and Distributed Computing, HPDC '17*, ACM, New York, NY, USA, 2017, pp. 231–242. URL: <http://doi.acm.org/10.1145/3078597.3078610>. doi:[doi:10.1145/3078597.3078610](https://doi.org/10.1145/3078597.3078610).
- [34] F. Magaletti, F. Picano, M. Chinappi, L. Marino, C. M. Casciola, The sharp-interface limit of the Cahn–Hilliard/Navier–Stokes model for binary fluids, *Journal of Fluid Mechanics* 714 (2013) 95–126. doi:[doi:10.1017/jfm.2012.461](https://doi.org/10.1017/jfm.2012.461).
- [35] T. Tezduyar, S. Mittal, S. Ray, R. Shih, Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements, *Computer Methods in Applied Mechanics and Engineering* 95 (1992) 221–242.
- [36] T. J. Hughes, G. Scovazzi, L. P. Franca, Multiscale and stabilized methods, *Encyclopedia of Computational Mechanics Second Edition* (2018) 1–64.
- [37] A. Brooks, T. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, *Computer Methods in Applied Mechanics and Engineering* 32 (1982) 199–259.
- [38] T. J. Hughes, Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods, *Computer Methods in Applied Mechanics and Engineering* 127 (1995) 387–401.
- [39] T. J. Hughes, G. Sangalli, Variational multiscale analysis: the fine-scale green's function, projection, optimization, localization, and stabilized methods, *SIAM Journal on Numerical Analysis* 45 (2007) 539–557.
- [40] S. Balay, W. D. Gropp, L. C. McInnes, B. F. Smith, Efficient management of parallelism in object oriented numerical software libraries, in: E. Arge, A. M. Bruaset, H. P. Langtangen (Eds.), *Modern Software Tools in Scientific Computing*, Birkhäuser Press, 1997, pp. 163–202.
- [41] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. D. Gropp, D. Karpeyev, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, H. Zhang, PETSc Web page, <https://www.mcs.anl.gov/petsc>, 2019.

- 745 [42] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener,  
 V. Eijkhout, W. D. Gropp, D. Karpeyev, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes,  
 R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, H. Zhang, PETSc  
 Users Manual, Technical Report ANL-95/11 - Revision 3.11, Argonne National Laboratory,  
 2019.
- 750 [43] H. Sundar, R. Sampath, G. Biros, Bottom-up construction and 2:1 balance refinement of linear  
 octrees in parallel, *SIAM Journal on Scientific Computing* 30 (2008) 2675–2708. doi:[doi:10.1137/070681727](https://doi.org/10.1137/070681727).
- [44] M. Bern, D. Eppstein, S.-H. Teng, Parallel construction of quadtrees and quality triangulations,  
*International Journal of Computational Geometry & Applications* 9 (1999) 517–532.
- 755 [45] H. Sundar, R. S. Sampath, S. S. Adavani, C. Davatzikos, G. Biros, Low-constant parallel al-  
 gorithms for finite element simulations using linear octrees, in: SC'07: Proceedings of the  
 International Conference for High Performance Computing, Networking, Storage, and Anal-  
 ysis, ACM/IEEE, 2007, pp. 1–12.
- 760 [46] S.-R. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, L. Tobiska, Quantita-  
 tive benchmark computations of two-dimensional bubble dynamics, *International Journal for  
 Numerical Methods in Fluids* 60 (2009) 1259–1288.
- [47] S. Aland, A. Voigt, Benchmark computations of diffuse interface models for two-dimensional  
 bubble dynamics, *International Journal for Numerical Methods in Fluids* 69 (2012) 747–761.  
 URL: <http://doi.wiley.com/10.1002/fld.2611>. doi:[doi:10.1002/fld.2611](https://doi.org/10.1002/fld.2611).
- 765 [48] H. Z. Yuan, Z. Chen, C. Shu, Y. Wang, X. D. Niu, S. Shu, A free energy-based surface tension  
 force model for simulation of multiphase flows by level-set method, *Journal of Computational  
 Physics* 345 (2017) 404–426. URL: <http://dx.doi.org/10.1016/j.jcp.2017.05.020>. doi:[doi:10.1016/j.jcp.2017.05.020](https://doi.org/10.1016/j.jcp.2017.05.020).
- 770 [49] Y. Xie, O. Wodo, B. Ganapathysubramanian, Incompressible two-phase flow: Diffuse interface  
 approach for large density ratios, grid resolution study, and 3D patterned substrate wetting  
 problem, *Computers and Fluids* 141 (2015) 223–234. doi:[doi:10.1016/j.compfluid.2016.04.011](https://doi.org/10.1016/j.compfluid.2016.04.011).
- [50] G. Tryggvason, S. O. Unverdi, Computations of three-dimensional Rayleigh–Taylor instability,  
*Physics of Fluids A: Fluid Dynamics* 2 (1990) 656–659.
- 775 [51] X. Li, B. Jin, J. Glimm, Numerical study for the three-dimensional Rayleigh–Taylor instability  
 through the TVD/AC scheme and parallel computation, *Journal of Computational Physics* 126 (1996) 343–355.
- [52] J.-L. Guermond, L. Quartapelle, A projection fem for variable density incompressible flows,  
*Journal of Computational Physics* 165 (2000) 167–188.
- 780 [53] G. Tryggvason, Numerical simulations of the rayleigh-taylor instability, *Journal of Compu-  
 tational Physics* 75 (1988) 253–282.
- [54] H. Ding, P. D. Spelt, C. Shu, Diffuse interface model for incompressible two-phase flows with  
 large density ratios, *Journal of Computational Physics* 226 (2007) 2078–2095.

- [55] J. T. Waddell, C. E. Niederhaus, J. W. Jacobs, Experimental study of Rayleigh-Taylor instability: Low atwood number liquid systems with single-mode perturbations, Physics of Fluids 13 (2001) 1263–1273. doi:[doi:10.1063/1.1359762](https://doi.org/10.1063/1.1359762).
- [56] J. Hunt, A. Wray, P. Moin, Eddies, stream, and convergence zones in turbulent flows, Center for turbulence research report CTR-S88 (1988) 193–208.
- [57] H. Liang, Q. X. Li, B. C. Shi, Z. H. Chai, Lattice Boltzmann simulation of three-dimensional Rayleigh-Taylor instability, Physical Review E 93 (2016) 1–11. doi:[doi:10.1103/PhysRevE.93.033113](https://doi.org/10.1103/PhysRevE.93.033113).
- [58] S. S. Jain, A. Mani, P. Moin, A conservative diffuse-interface method for compressible two-phase flows, Journal of Computational Physics 418 (2020) 109606. URL: <https://doi.org/10.1016/j.jcp.2020.109606> <https://linkinghub.elsevier.com/retrieve/pii/S0021999120303806>. doi:[doi:10.1016/j.jcp.2020.109606](https://doi.org/10.1016/j.jcp.2020.109606). arXiv:1911.03619.
- [59] V. Chakravarthy, J. Ottino, Mixing of two viscous fluids in a rectangular cavity, Chemical Engineering Science 51 (1996) 3613–3622. URL: <https://linkinghub.elsevier.com/retrieve/pii/0009250996000073>. doi:[doi:10.1016/0009-2509\(96\)00007-3](https://doi.org/10.1016/0009-2509(96)00007-3).
- [60] R. Chella, J. Viñals, Mixing of a two-phase fluid by cavity flow, Physical Review E 53 (1996) 3832–3840. URL: <https://link.aps.org/doi/10.1103/PhysRevE.53.3832>. doi:[doi:10.1103/PhysRevE.53.3832](https://doi.org/10.1103/PhysRevE.53.3832).
- [61] K. Park, C. A. Dorao, M. Fernandino, Numerical Solution of Coupled Cahn-Hilliard and Navier-Stokes System Using the Least-Squares Spectral Element Method, in: Volume 1B, Symposia: Fluid Mechanics (Fundamental Issues and Perspectives; Industrial and Environmental Applications); Multiphase Flow and Systems (Multiscale Methods; Noninvasive Measurements; Numerical Methods; Heat Transfer; Performance); Transport Phe, American Society of Mechanical Engineers, 2016, pp. 1–9. URL: <https://asmedigitalcollection.asme.org/FEDSM/proceedings/FEDSM2016/50299/Washington,DC,USA/233438>. doi:[doi:10.1115/FEDSM2016-1008](https://doi.org/10.1115/FEDSM2016-1008).
- [62] M. Ishii, M. Fernando, K. Saurabh, B. Khara, B. Ganapathysubramanian, H. Sundar, Solving pdes in space-time: 4d tree-based adaptivity, mesh-free and matrix-free approaches, in: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2019, pp. 1–61.

## Appendix A. Some elementary propositions

For completeness we recall an important proposition and one its corollaries from [7].

### Proposition 2

The following identity holds:

$$\frac{\partial \tilde{\phi}^k}{\partial x_j} \left( \frac{\partial}{\partial x_j} \left( \frac{\partial \tilde{\phi}^k}{\partial x_i} \right) \right) = \frac{1}{2} \frac{\partial}{\partial x_j} \left( \frac{\partial \tilde{\phi}^k}{\partial x_i} \frac{\partial \tilde{\phi}^k}{\partial x_i} \right) \quad (\text{A.1})$$

$\forall \tilde{\phi}^k \in H^1(\Omega)$ , where  $\phi^k, \phi^{k+1}, \mu^k, \mu^{k+1}, \mathbf{v}^k, \mathbf{v}^{k+1}$  solves eq. (13) – eq. (17).

815

**Remark 9.** The advection term in eq. (1) can be defined in the skew-symmetric form as (see lemma 6.10 of section 6.1.2 of [9] for details):

$$B_1(v_i, v_j) = \rho v_j \frac{\partial v_i}{\partial x_j} + \frac{1}{2} \rho v_i \frac{\partial v_j}{\partial x_j} = \rho v_j \frac{\partial v_i}{\partial x_j} \quad \text{and} \quad B_2(v_i, v_j) = J_j \frac{\partial v_i}{\partial x_j} + \frac{1}{2} J_i \frac{\partial v_j}{\partial x_j} = J_j \frac{\partial v_i}{\partial x_j}, \quad (\text{A.2})$$

where we also used solenoidality of the mixture velocity eq. (4). The skew symmetric form induces a trilinear form when weakened:

$$b_1(v_i, v_j, w_i) = (B_1(v_i, v_j), w_i) = \frac{1}{2} \left( \rho v_j \frac{\partial v_i}{\partial x_j}, w_i \right) - \frac{1}{2} \left( \rho v_j \frac{\partial w_i}{\partial x_j}, v_i \right), \quad (\text{A.3})$$

$$b_2(v_i, J_j, w_i) = (B_2(v_i, J_j), w_i) = \frac{1}{2} \left( J_j \frac{\partial v_i}{\partial x_j}, w_i \right) - \frac{1}{2} \left( J_j \frac{\partial w_i}{\partial x_j}, v_i \right). \quad (\text{A.4})$$

Using the above proposition then yields:

$$b_1(v_i, v_j, v_i) = (B_1(v_i, v_j), v_i) = \frac{1}{2} \left( \rho v_j \frac{\partial v_i}{\partial x_j}, v_i \right) - \frac{1}{2} \left( \rho v_j \frac{\partial v_i}{\partial x_j}, v_i \right) = 0, \quad (\text{A.5})$$

$$b_2(v_i, J_j, v_i) = (B_2(v_i, J_j), v_i) = \frac{1}{2} \left( J_j \frac{\partial v_i}{\partial x_j}, v_i \right) - \frac{1}{2} \left( J_j \frac{\partial v_i}{\partial x_j}, v_i \right) = 0. \quad (\text{A.6})$$

### Corollary 1: Strong equivalence of forcing

If we have the following equivalence in the weak sense:

$$\frac{Cn}{We} \left( \frac{\partial}{\partial x_j} \left( \frac{\partial \tilde{\phi}^k}{\partial x_i} \frac{\partial \tilde{\phi}^k}{\partial x_j} \right), \delta t \tilde{v}_i^k \right) = \frac{\delta t}{WeCn} \left( \tilde{\phi}^k \frac{\partial \tilde{\mu}^k}{\partial x_i}, \tilde{v}_i^k \right), \quad (\text{A.7})$$

$\forall \tilde{\phi}^k, \tilde{\mu}^k \in H^1(\Omega)$ , and  $\forall \tilde{\mathbf{v}}^k \in \mathbf{H}_0^1(\Omega)$ , where  $\tilde{\mathbf{v}}^k, \tilde{\mathbf{v}}^{k+1}, p^k, p^{k+1}, \phi^k, \phi^{k+1}, \mu^k, \mu^{k+1}$  satisfy eq. (13) – eq. (17), then the following equivalence also holds in the strong sense:

$$\frac{Cn}{We} \frac{\partial}{\partial x_j} \left( \frac{\partial \tilde{\phi}^k}{\partial x_i} \frac{\partial \tilde{\phi}^k}{\partial x_j} \right) = \frac{1}{WeCn} \tilde{\phi}^k \frac{\partial \tilde{\mu}^k}{\partial x_i}, \quad (\text{A.8})$$

if  $\tilde{\phi}^k, \tilde{\mu}^k \in H^1(\Omega) \cap C_c^\infty(\Omega)$ , and  $\tilde{\mathbf{v}}^k \in \mathbf{H}_0^1(\Omega) \cap \mathbf{C}_c^\infty(\Omega)$ .

## Appendix B. Details of solver selection for the numerical experiments

For the cases presented in sections 5.2, 5.4 and 5.5 we use the BiCGStab linear solver (a Krylov space solver) with additive Schwarz-based preconditioning. For better reproduction, the command line options we provide PETSc are given below which include some commands used for printing some norms as well.

```
820 -ksp_type bcgs
-ksp_type bcgs
825 -pc_type asm
-sub_pc_type lu 0
#For monitoring residuals
-snes_monitor
-snes_converged_reason
-ksp_converged_reason
```

830 For the Rayleigh-Taylor instability case (section 5.3), we used an algebraic multigrid (AMG) linear solver with an successive over relaxation preconditioner with a GMRES at each level as a smoother. The options used for the Petscsetup is given below.

```
835 solver_options_ns = {
    snes_atol = 1e-5
    snes_rtol = 1e-6
    snes_stol = 1e-5
    snes_max_it = 40
    ksp_rtol = 1e-5
    ksp_atol = 1e-6
840    ksp_diagonal_scale = True
    ksp_diagonal_scale_fix = True

    #multigrid

845    #solver selection
    ksp_type = "fgmres"
    pc_type = "gamg"
    pc_gamg_asm_use_agg = True
    mg_levels_ksp_type = "gmres"
    mg_levels_pc_type = "sor"

850    #performance options
    mattransposematmult_via = "matmatmult"
    pc_gamg_reuse_interpolation = "True"
    mg_levels_ksp_max_it = 40
855};

};
```

The linear systems we handle are fairly ill-conditioned, therefore, the smoothers we need to use are fairly expensive. The ASM/LU based smoother is more expensive compared to other smoothers like block Jacobi, however ASM/LU is more robust (better convergence). This setup works very well 860 with a relatively constant number of Krylov iterations as the number of processes are increased in the massively parallel setting. The scaling results we present use the same setup of solvers, but there is substantial room for improvement in this area of the code where fieldsplit preconditioners using Schur complement can be used as smoothers to improve speed of the AMG solver.