FIPI: A fast numerical method for the simulation of particle-laden fluid interfaces

Chuan Gu\textsuperscript{a}, Lorenzo Botto\textsuperscript{a,b,∗}

\textsuperscript{a}School of Engineering and Material Science, Queen Mary University of London, E1 4NS, London, United Kingdom
\textsuperscript{b}Department of Process and Energy, Delft University of Technology, 2628 CB, Delft, the Netherlands

Abstract

Solid particles can adhere to fluid interfaces, modifying interfacial properties such as the surface tension and the surface elasticity. We here describe a new simulation method, the Fast Interface Particle Interaction (FIPI) method, capable of simulating on commodity hardware up to 100\textsuperscript{k} particles interacting with fluid interfaces of complex morphology. The method is based on resolving interfacial and hydrodynamic length scales larger than the particle, and model particle-level phenomena with physically consistent algebraic models. For its ability to simulate large numbers of potentially very small particles, FIPI represents a useful tool for investigating phenomena of particle adsorption/desorption and particle-induced surface tension modification in problems of froth flotation, drop/foam stabilisation, Bijel formation, and other multiphase systems.

Program Title: FIPI
Licensing provisions: GPLv2
Programming language: C++
Nature of problem: Simulation of particle-laden fluid interfaces (3-phase flows)
Solution method: Spectral method.
Running time: Up to dozens of hours on a single processor.
Keywords: Particle-laden flows, Liquid-liquid interfaces, Multiphase flows, Surface tension

1. Introduction

Particles suspended in multiphase fluids appear in many applications, such as froth flotation [1], “Bijel” materials [2, 3], food processing [4], enhanced oil recovery [5] and encapsulation [6]. In these applications, particles are trapped in the fluid interfaces under the effect of capillarity, forming composite interfaces usually called particle-laden or particle-covered fluid interfaces [7, 8].

The interaction of particles with fluid interfaces occurs over a wide range of length scales (Fig.1), from the microscale of the particles (∼ 10\textsubscript{nm} – 1\textmu m) to the mesoscale of the bubbles, drops, or interfacial structures (∼ 100\mu m – 1cm) that the particles are adsorbed in. Developing efficient numerical methods for simulating such multiscale systems would enable the rational design of particle-level variables (particle size distribution, contact angle, etc.), flow conditions (shear rate, large-scale mixing power, etc.), or external field parameters (strength of magnetic, electric, or centrifugal forces) towards obtaining desired process outcomes or material properties.

However, simulating particle-laden fluid interfaces presents several challenges [2]. In typical applications the interface is covered with a large number of particles, so computational efficiency is important. A significant separation of scales occurs between the size of particles and the characteristic radius of the curvature of the fluid interface [9, 10]. The interaction of each particle with the interface can produce interfacial structures that are much smaller than the particles themselves [11, 12]. All these factors contribute to extremely stringent simulation requirements. One needs to use small grid sizes to resolve relevant physical features at particle-level length scales, but large computational domains to simulate relevant multi-droplet or multi-bubble systems. A computational approach that enables simulation of realistic length and time scales is needed.
In the following, the numerical methods currently available for 3-phase flows with particles are briefly reviewed. Singh and Joseph [13] carried out the first direct numerical simulation of spherical particles floating at a fluid interface. They adopted an approach combining a level-set method for capturing the fluid interfaces and a distributed Lagrangian multipliers approach for enforcing the rigid body motion inside the particles. Though limited to at most 4 particles, their studies have provided insights into the mechanics of cluster formation under capillary attraction. Other authors have used a lattice Boltzmann approach, where the bounce-back model for resolving solid particles [14] is combined with a model for multiphase fluids [15, 16]. With this approach, several phenomena have been studied, such as the formation of bicontinuous gels [2], dynamics of the particles suspended in a drying drop on a substrate [17] and the transition from bijels to Pickering emulsions [18]. Aland et al. [19] developed a fully continuum model capable of simulating colloid stabilised fluid interfaces. In their method, the surface and bulk colloid concentrations evolve according to two convection-diffusion equations while the fluid interface is captured by a phase field method. Jaensson et al. [20] used a Stokes-Cahn-Hilliard formulation to simulate particles suspended in two-phase flows. They used a boundary-fitted mesh to describe the sharp boundary between the particles and the fluid. Millett and Wang [21] presented a mesoscale simulation approach that uses a diffuse-interface method to describe the fluid phases and the solid particles. Their method was used to simulate the arrest of spinodal decomposition due to colloidal particles jamming the fluid interfaces.

Most of the simulation approaches described above allocate a large amount of resources to resolve particle-scale features. These simulations have enabled a detailed understanding of the physics of the capillary interaction between solid particles and fluid interfaces, but are either limited to a small number of particles (when highly accurate solutions are sought) or computationally expensive (the main production run in [2] takes around 1 week on a 32-processor IBM p690+ system). If large number of particles are simulated, the resolution is quite limited, even using large supercomputers (properly resolving hydrodynamics alone would require at least 8-10 nodes per particle radius [22], while most large multi-particle simulations are constrained to use a small number of grid nodes per particle radius [18]). This situation raises two questions: is it possible to devise a method that is computationally efficient while retaining essential physical features? What are these features?

An important challenge in simulating particle-laden interfaces is the need of resolving the capillary interactions between each particle and the fluid interface. A particle that is either adsorbing or desorbing from a fluid interface creates a characteristic meniscus that takes the form of a cylindrical liquid bridge extending from the particle surface to the plane of the undisturbed fluid interface [11, 23]. This bridge is responsible for the capillary force between the particle and the interface. The minimum diameter of this bridge can become much smaller than the particle size [24, 12]. Furthermore, the fluid interface moves along the surface of the particle as the particle adsorbs to or detaches from the fluid interface, bringing up the theoretical and numerical challenges of dealing with the stress singularity at the contact line [25, 26, 27]. Is it always necessary to resolve these features? When the focus is to accurately compute the capillary interaction between two or few particles [28, 29], or the interaction of one particle with the fluid interface [30, 13], the answer is certainly yes. But for simulations that require calculating collective effects of many particles on interfacial morphology, an approximated method could be a better option. In such method the attachment force exerted on the particle by the fluid interface must be approximated correctly [31, 32, 33] and the force calculated efficiently. If this force is modelled accurately, the particle will have the correct adsorption energy and will desorb at the correct value of the normal forces produced by lateral particle-particle interactions [34, 35]. If the lateral interactions are modelled accurately, the effective surface stresses (linked to surface tension and surface elasticity) will be correct [36, 37, 38], as we will confirm in the current paper.

In this article we present a simulation method for addressing problems involving particles suspended in binary fluids. We have named it the Fast Interface Particle Interaction (FIPI) method. This method resolves interfacial structures and hydrodynamics on length scales much larger than the particle size, and models particle-level physics with empirical or semi-analytical expressions. As in point-particle methods widely employed in the simulation of dispersed flows [39], the hydrodynamic forces are modelled. A new ingredient in FIPI is that the particle-interface capillary forces are modelled as well. FIPI combines an Euler-Lagrangian method for fluid-particle coupling with a diffuse-interface method for capturing the dynamics of the fluid interface. The diffuse-interface method has the benefit of handling the breakup and coalescence of fluid interface easily [40]. In addition, the phase field variable in the diffuse-interface method provides information on the local distance of the particle to the fluid interface, from which the capillary force can be calculated efficiently. In the current implementation we propose preliminary models for the hydrodynamic and capillary forces. More sophisticated models, once available, could be implemented easily within the framework of FIPI.

2. Mathematical Formulation

FIPI is composed of three modules, namely FIPI-Interface, FIPI-Track and FIPI-Fluid, related to interface capturing, particle tracking and the solution of the fluid momentum equation, respectively. The input and output of each module are illustrated in Fig.(2): FIPI-Interface provides the location of the fluid interface, whose information is contained in the phase field variable \( \phi \), to the other two modules; FIPI-Track provides the forcing term \( f_\phi \) due to particles to FIPI-Fluid; FIPI-Fluid provides the fluid velocity field \( \mathbf{u} \) to FIPI-Interface and FIPI-Track. The details of each module will be described in the follows.
**FIPI-Interface.** The FIPI-Interface module is based on a standard phase-field method [41, 40]. In the phase-field method the fluid interface has a finite thickness, across which a scalar phase field variable $\phi$ changes continuously. The evolution of $\phi$ follows the Cahn-Hilliard equation
\[
\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = \nabla \cdot (\mathbf{M} \nabla \xi),
\]
where $\mathbf{u}$ is the fluid velocity, $M$ is the mobility parameter (which we take as a constant [42]), and $\xi$ is the chemical potential. The chemical potential is defined as the functional derivative of the total free energy density:
\[
\xi = \frac{\partial F}{\partial \phi}.
\]
Following [43] the total free energy density is modelled in FIPI as
\[
F(\phi, \nabla \phi) = \frac{\lambda}{4\epsilon^2} \left( \phi^2 - 1 \right)^2 + 1/2 \lambda |\nabla \phi|^2,
\]
where $\lambda$ and $\epsilon$ are two constant parameters. At steady state and in the absence of convection, Eq.(1) admits the solution
\[
\phi = \tanh(\frac{x}{\sqrt{2}\epsilon}),
\]
(4)
where here $x$ is the coordinate along the direction perpendicular to the fluid interface. Integrating the free energy density $F$ across the fluid interface, with $\phi$ given by Eq.(4), yields the surface tension [41]
\[
\gamma = \int_{-\infty}^{\infty} F(\phi, \nabla \phi) dx = \frac{2 \sqrt{2} \lambda}{3 \epsilon}. \tag{5}
\]
This is the bare surface tension of the fluid interface. Eq.(4) shows that the numerical thickness of the fluid interface depends solely on $\epsilon$. The ratio of $\lambda$ over $\epsilon$ determines the surface tension of the fluid interface in the absence of the particle (Eq. 5). In actual simulations $\epsilon$ and $\lambda$ are chosen based on preliminary tests. The choice for $\epsilon$ is set by two constraints. The value of this parameter must be large enough so that $\phi$ is sufficiently well-resolved across the fluid interface (i.e., $\epsilon$ is set by the grid mesh size). On the other hand, $\epsilon$ must be kept as small as possible to ensure a reasonable separation of length scales between the fluid interface thickness and the length scale of the interfacial structure of interest (e.g. the radius of a drop). After $\epsilon$ is chosen, the desired bare surface tension value is obtained by adjusting $\lambda$ according to Eq. 5.

In FIPI, the fluid velocity $\mathbf{u}$ in Eq.(1) can result from not only the bare surface tension of fluid interfaces but also the reaction forces from particles. Although Eq.(4) is derived under the assumption of steady state and $\mathbf{u} = 0$, this expression is valid when the characteristic time scale of the diffusion is much smaller than that of the convection $R/\nu \gg \epsilon^4/\mathcal{M} \lambda$, with $R$ being the characteristic radius of fluid interface curvature.

**FIPI-Track.** Newton’s equation of motion for each particle is
\[
m_p \mathbf{u}_p = m_p \mathbf{g} + \mathbf{F} + \mathbf{F}_{pp}, \tag{6}
\]
where $m_p$ is the particle mass, $\mathbf{F}$ is the force on the particle due to the surrounding fluids, including the force exerted by the fluid interface, and $\mathbf{F}_{pp}$ is the force on the particle due to particle-particle interactions of non-hydrodynamic origin (e.g., due to electrostatic or contact forces). The force $\mathbf{F}$ can be decomposed as [13]
\[
\mathbf{F} = \int_{CL} \gamma \mathbf{n} ds + \int_{S_p} (-pI + \sigma^h) \cdot \mathbf{n} dA + \mathbf{F}_b, \tag{7}
\]
where $\mathbf{n}$ is the unit vector normal to the contact line and tangent to the fluid interface, $p$ is the hydrodynamic pressure, $\sigma^h$ is the viscous stress tensor, $\mathbf{n}$ is the unit vector normal to the particle surface $S_p$ and $\mathbf{F}_b$ is the buoyancy force. Both $\mathbf{n}$ and $\mathbf{F}_b$ point out of the particle. The first integral on the right hand side of Eq.(7) is the total force due to the surface tension force acting along the contact line $CL$. This term is non-zero only when the particle is embedded into the fluid interface and the fluid meniscus is curved. The second integral is the total force due to hydrodynamic stresses. Because the interface has an infinitesimal thickness, the decomposition of the total force into a capillary force and a force due to contact with the fluid molecules outside the interface is justified. At the molecular level such distinction is blurred for locations near the fluid interface [44, 45]. However, the effect of such molecular-level physics is likely to be important only for very small nanoparticles.

In FIPI, the force due to surrounding fluid $\mathbf{F}$ is modelled as
\[
\mathbf{F} \approx \mathbf{F}_{pp}(\phi(\mathbf{x}_p, t)) + \mathbf{F}_b(x_p, \mathbf{x}_p, \mathbf{u}(\mathbf{x}_p, t)) \tag{8}
\]
where $\mathbf{F}_{pp}$ and $\mathbf{F}_b$ are algebraic models for the capillary force and the hydrodynamic drag force. We have omitted the buoyancy force in the current version of FIPI but the implementation of this force should be straightforward [13]. Furthermore, the particle inertia is neglected as the particle Reynolds number is generally very small $Re_p \ll 1$ for colloidal particles in particle-laden interfaces.

Different closures for $\mathbf{F}_{pp}$ and $\mathbf{F}_b$ are admissible. A suitable model should be simple, to avoid uncontrollable artifacts.
produced by models having too many parameters, and be based on robust physical considerations. In the following we present the models of $F_{pi}$ and $F_{h}$ incorporated in the current version of FIPi. These models are local in the sense that they are parameterised on the particle velocity $\dot{x}_{p}$, particle position $x_{p}$ and local phase field variable $\phi(x_{p}, t)$. A model for $F_{pp}$ will also be described.

Hydrodynamic force model. A solid particle moving in or near a fluid interface experiences a drag force. The hydrodynamic drag on a spherical particle translating in the vicinity of a fluid interface has been studied by several authors [46, 47, 48, 49]. The drag force depends on the viscosities of the two fluids and the distance of the particle centre from the fluid interface location. Owing to the need to squeeze fluid out of the gap between the particle and the fluid interface, the force increases with respect to Stokes drag expression when the particle translates perpendicularly to the fluid interface [50, 51]. In addition to particles moving close to an interface but not touching it, several investigations have focused on the case of particles completely embedded in the fluid interface [52, 53, 54, 55]. The hydrodynamic drag force in this case depends on the ratio of the viscosities of two fluids, and the degree of immersion (which in turn can be mapped to the contact angle) [56]. For a single particle moving near or in a fluid interface, the drag force is proportional to the particle velocity. Hence, the use of a modified Stokes drag coefficient is an appropriate, although approximate, model.

In the current version of FIPi, we adopted the following simple model for the hydrodynamic drag force:

$$ F_{h} = -6\pi \mu f_{d} \left( \dot{x}_{p} - u_{f}(x_{p}, t) \right), \tag{9} $$

where $u_{f}(x_{p}, t)$ is the fluid velocity at the particle position, and $f_{d}$ is a user-specified drag coefficient. In the current implementation we choose, for simplicity, $f_{d} = 1$. However, this parameter can be easily parameterised on the contact angle, particle-interface distance, viscosity ratio, etc, choosing appropriate expressions from the literature. To model close range particle-particle interactions, the drag could also be made a function of the local solid area or volume fractions, as in bulk suspensions [57, 58]. The velocity $u_{f}(x_{p}, t)$ should be interpreted as the fluid velocity that would be present at the particle position if the particle was not there. For cases in which the total non-hydrodynamic force on the particles is not too large, we set $u_{f}(x_{p}, t) = u(x_{p}, t)$ (the issue of the self-induced fluid velocity is discussed in Sec. 6). Far-field hydrodynamic interactions between particles are accounted for at the point-force level, because $u(x_{p}, t)$ depends on the position of and forces on all the particles (which are represented as point forces in the momentum equation). For cases in which the relative inter-particle velocity is small, but the particle-laden fluid interface is sheared by a moving fluid, approximate expressions for the drag force are available for large viscosity ratios [59].

Capillary force model. The capillary force exerted on a spherical particle embedded in a fluid interface results from the surface tension acting at the three-phase contact line and is defined as $F_{pi} = \frac{\gamma}{\cos \theta_{c}} n_{p} d s$. When an adsorbed spherical particle is being displaced perpendicularly from a flat fluid interface, this force increases as the meniscus becomes curved. This capillary force obeys the expression [25]

$$ F_{pi} = 2\pi \gamma a \alpha \cos (\alpha + \theta_{c}) \cos \theta_{a}, \tag{10} $$

where $\theta_{c}$ is the contact angle and $\alpha$ is the filling angle (Fig.3). The filling angle is the angle formed between the unperturbed flat fluid interface and the line connecting the particle centre to the contact line. This angle measures the wetting area of one fluid on the surface of the particle and it increases with $h$ as the particle is being displaced from the fluid interface [25, 11].

In FIPi, the boundary conditions at particle surface are not enforced explicitly. Furthermore, the fluid interface (zero level set) is resolved also inside the volume occupied by the particle, as shown in Fig.(3). As a result, information on the values of $\alpha$ and $\theta_{c}$ is not available. It is still possible to model $F_{pi}$ knowing that the capillary force is, for small displacement, a linear function of the fluid interface distortion $h$ [60]. In FIPi the distance $d'$ from the particle centre to the nearest zero level set can be explicitly measured. Knowing that $d' \propto \alpha \propto h$ is valid for small distortion of the fluid interface, we can model $d'$ as

$$ F_{pi} = \begin{cases} 
\frac{f \pi \gamma d'}{\cos \theta_{c}}, & \text{if } |d'| < a \\
0, & \text{if } |d'| > a \end{cases}, \tag{11} $$

where $d'$ is the vector pointing from the particle centre to the fluid interface along the direction of minimal distance, and $f$ is a nondimensional parameter depending on the contact angle $\theta_{c}$ of the particle. The choice of $f$ should satisfy the condition that the maximum energy associated with the spring force $\frac{1}{2} f \pi \gamma a^{2}$ matches the minimum energy required to detach the particle from the fluid interface $\pi \gamma a^{2} (1 - \cos \theta_{c})^{2}$ [61]. For $d' < a$, the particle is considered to be adsorbed to the fluid interface and the capillary force follows a linear relationship with the distance $d'$. When $|d'| > a$, the particle is considered detached from the
fluid interface and the capillary force is zero.

Particle-particle interaction model. The interaction forces between colloidal particles in a suspension is often described by the DLVO theory [62]. The DLVO force is regarded as the sum of an attractive van der Waals force and a repulsive electric double layer force. Other non-DLVO forces exist when two particles are in close range such as the hydration force and the steric force [63]. Modelling these forces requires knowledge of the material properties of the particles and the surrounding liquid solutions, as well as other physical-chemical parameters. For the current version of FIFI, we have decided to adopt a generic linear repulsive force law to explore the effect of interparticle repulsion on the mechanical properties of a particle-laden interface, but more complicated interparticle force model can be used on an ad hoc basis. The interparticle force model is

\[ F_{pp} = \begin{cases} -k_c (r - r_c)n & \text{if } r < r_c \\ 0 & \text{if } r > r_c \end{cases} \]  

(12)

where \( k_c \) is a stiffness parameter and \( r_c \) is a cut off distance. This force can be seen as a relaxed version of the hard sphere contact force. The “softness” of the contact can be adjusted by changing the values of \( k_c \) or \( r_c \). Hard sphere contact is recovered by setting \( k_c \) to a large value and \( r_c \) to a value close to \( 2a \). As suggested in Ref.[64, 65] for discrete particle simulations, details of the particle-particle interaction model should not significantly affect the collective behaviour of the particles as long as the collisions between particles are regular enough. This is the case for a particle monolayer, since in this case each particle is constantly interacting with neighbouring particles.

FIFI-Fluid. In principle, the fluid pressure and velocity field should be calculated by solving the governing equations for the fluid subject to the no-slip boundary condition at the particle surface, with no fluid penetration and continuity of normal and tangential stresses at the fluid-fluid interface. However, this particle-resolved approach is computationally very expensive and brings about the issue of contact line motion at the surface of the particle. Accurate particle-resolved simulations are currently feasible only for a few particles and not without strong assumptions regarding the physics at the contact line [13].

Since the motivation for the development of FIFI is the simulation of a large number of particles, an one-fluid model is here adopted [66] for the interface, combined with a point-particle approach to model the effect of the particles. In essence, the one-fluid and the point-particle approaches are based on the same idea: it is possible to replace a condition at an internal boundary with a forcing term in the fluid momentum equation. Point-particle approaches have been extensively used for particle suspended in a single phase fluid [39, 67, 68].

At each time step, the following momentum and continuity equations are solved over the entire computational domain (assuming Stokes flow conditions):

\[ \mathbf{0} = -\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{f}_e + \mathbf{f}_p \]  

(13)

and

\[ \nabla \cdot \mathbf{u} = 0. \]  

(14)

The particle forcing term is

\[ \mathbf{f}_p = -\sum_{j=1}^{N} \left( F_{pp}^{(j)} + F_{hh}^{(j)} \right) \delta(\mathbf{x} - \mathbf{x}_p^{(j)}). \]  

(15)

and \( \mathbf{f}_p \) is the capillary force per unit volume exerted by the bare fluid interface on the fluid (with a surface tension value given by Eq.(5)). The summation term in Eq.(15), which loops over the \( N \) particles present in the computational domain, contains three-dimensional Dirac delta functions centred at each particle. The forces \( F_{pp}^{(j)} \) and \( F_{hh}^{(j)} \) have the form given by Eq.(11) and Eq.(9), respectively.

The following model is adopted for the volumetric capillary force [43]:

\[ \mathbf{f}_c = \kappa \nabla \phi = \left( \frac{1}{\epsilon} \delta(\phi - \phi_c) - \lambda \nabla^2 \phi \right) \nabla \phi. \]  

(16)

It can be shown that as \( \epsilon \to 0 \), the capillary force approaches to

\[ \mathbf{f}_c = -\mathbf{n} \kappa \delta(\zeta), \]  

(17)

where \( \kappa \) is the local mean curvature of the fluid interface, \( \mathbf{n} \) is the unit normal vector pointing towards the convex side, and \( \delta(\zeta) \) is the one-dimensional Dirac delta function whose argument \( \zeta \) is the coordinate normal to the fluid interface.

2.1. Modulation of surface stresses in FIFI

By comparing Eq.(15) and Eq.(17) it can be seen that the effect of the particles and the bare fluid interface are represented as forcing terms in the fluid momentum equation. The difference is that the effect of the particles is replaced by Dirac delta functions at discrete spatial locations, and the effect of the bare fluid interface is modelled as a continuous volumetric surface tension force, distributed over the finite thickness \( \delta(\epsilon) \) of the fluid interfacial region. In the actual implementation, the Dirac delta forces from the particles are smoothed out over the mesh size \( \Delta x \).

In the continuum limit, when the interparticle separation is much smaller than the radius of fluid interface curvature, the formulation Eq.(13)(15)(17) produces the correct value of the effective surface tension of the particle-laden fluid interface. To demonstrate this, we integrate the forcing terms in Eq.(13) over a thin pillow box region \( V \) surrounding the fluid interface (as done in Ref.[40]). The lateral size of this region is assumed to be sufficiently large in comparison to the interparticle separation.
In this expression this tangential force is the surface tension. A tangential force effecting interactions in terms of a line integral of a suitably defined surface stress. This can be done by noting that [70]

\[
\tau_i(x_s) = \frac{1}{2} \sum_{\beta \neq \alpha} \sum_{\beta \neq \alpha} F_{\alpha \beta}(\mathbf{y}^\alpha_s - \mathbf{y}^\beta_s) G(x_s - y_s^\beta).
\]  

(21)

is the particle-induced surface stress at \(x_s\). Here \(G\) is a 2D filter with support on \(\delta A\). The limit in Eq.(20) requires \(\delta A\) to include a sufficient amount of particles so that a continuum representation is approximately valid. Using the divergence theorem [71] and decomposing \(\tau_i\) into its isotropic component \(-\Pi, I\) and its deviatoric component \(\tau^s\), we can write

\[
\int_V (\mathbf{f}_c + \mathbf{f}_p)dV = \int_C (\gamma - \Pi)\mathbf{t}ds + \int_C \tau^s \cdot \mathbf{t}ds,
\]  

(22)

From this expression, we can see that the effective isotropic tension of the particle-laden interface is [72]

\[
\gamma_{eff} = \gamma - \Pi,\n\]

(23)

This expression, which is thermodynamically correct [72], can be expected from intuition. If \(\Pi\) is positive, corresponding to a repulsive interaction between the particles, the effective isotropic tension is reduced. The tensor \(\tau^s\) embeds information about anisotropy in the surface tension. For instance, for a pendant drop covered with particles, the surface tension in the azimuthal direction is different from the surface tension in the meridional direction [70]. This surface stress anisotropy is a manifestation of the shear elasticity of the interface. Both \(\Pi\) and \(\tau^s\) contribute to a normal force acting on the fluid interface and produce a Laplace pressure which corresponds to the integral of the pressure term in Eq.(13).

3. Numerical Implementation

An important advantage of the one-fluid approach for the fluid interface and the representation of the effect of each particle by a forcing term is computational efficiency. The solid-fluid and fluid-fluid interfaces are not treated as internal boundaries hence fast Eulerian method for single-phase flows can be employed.

A Fourier spectral method ([73]) is adopted to solve Eq.(1)(13)(14). With the readily available numerical package FFTW [74], multi-threading parallelisation of fast Fourier transform (FFT) is easily implemented. Fourier spectral method is employed because higher order derivatives (up to 4th order in the Cahn-Hilliard equation) can be computed accurately with a relatively small number of discretisation points. However, we do not see conceptual issues to implementing FPI with other discretisation approaches (LBM, Finite Volume, finite difference, etc.) that do not require period boundary condition, provided sufficient resolution can be guaranteed.

With \(\phi(x) = \int \phi(k)e^{2\pi x_k}dk\), Eq.(1) can be rewritten in Fourier space as

\[
\frac{\partial \hat{\phi}}{\partial t} + (\mathbf{u} \cdot \nabla)\phi = -k^2 M \left( \frac{\lambda}{\epsilon C} (\phi^3 - \phi) \right) - k^4 MA \phi
\]

(24)
The distance vector is calculated as Eq. (11). We have found that having the phase variable $x$ particle centre discretised Cahn-Hilliard equation in frequency space is treated explicitly by a two-step Adam-Bashforth (AB) method. The nonlinear terms are calculated first in real space. The discretised Cahn-Hilliard equation in frequency space is

$$\frac{3\hat{\phi}^{n+1} - 4\hat{\phi}^{n} + \hat{\phi}^{n-1}}{2\Delta t} = -(2\hat{\epsilon}^{n} - \hat{\epsilon}^{n-1}) - k^2 M(2\hat{\phi}^{n} - \hat{\phi}^{n-1}) - k^4 M A \delta^n$$

(25)

$$\hat{\epsilon}^{n} = \{u^n \cdot (-i k \hat{\phi}^{n})\}_{k}$$

(26)

$$\hat{\phi}^{n} = \{\frac{\partial}{\partial \phi^n}(\phi^n)^3 - \phi^n\}_{k}$$

(27)

where $()_{k}$ represents the inverse Fourier transform of the term inside the bracket.

With the models in Eq. (8)-(12), the instantaneous velocity of the particle at the time instant $n$ can be explicitly calculated from Eq. (6) as

$$u_p(t^n, x^n_p) = u_j(t^n, x^n_j) + \frac{F_{pp}(t^n, x^n_p) + F_{pp}(t^n, x^n_j)}{6\mu M f_d}.$$  

(28)

The particle-particle interaction force $F_{pp}$ is calculated in a pairwise fashion according to Eq. (12). The cell-list sorting method is used to find all the pairs within the cutoff distance $r_c$ [75].

Evaluating $F_{pi}$ requires knowledge of the distance $d$ from particle centre $x^n_p$ to the nearest fluid interface, according to Eq. (11). We have found that having the phase variable $\phi(t^n, x^n_p)$ at our disposal is very useful in this respect. Assuming the phase field profile $\phi$ is close to equilibrium, we have $\phi \approx \phi_{eq} = \tanh(\frac{-x^n_p}{2\epsilon})$ in the proximity of the fluid interface. An effective fluid interface region (Fig. 5) can be defined where $|d| < 2\sqrt{2} \epsilon$, because the distance magnitude $d^{(j)}$ diverges to infinity and $\nabla \phi$ becomes zeros as $\phi^{(j)} \rightarrow \pm 1$.

One requirement for the mapping to be valid is that the particle radius $a$ must be smaller than the half of the thickness of the interfacial region, $a < 2\sqrt{2} \epsilon$, because the distance magnitude $d^{(j)}$ diverges to infinity and $\nabla \phi$ becomes zeros as $\phi^{(j)} \rightarrow \pm 1$.

The distance vector is calculated as $d^{(j)} = d^{(j)} n^{(j)}$ with $n^{(j)} = -\frac{\phi^{(j)} \nabla \phi(x^{(j)}, t)}{|\phi^{(j)} \nabla \phi(x^{(j)}, t)|}$ being the unit normal vector.

Since $\phi$ and $\nabla \phi$ are only available at the Eulerian grid points, an interpolation scheme is required to calculate the values of $\phi(t^n, x^n_p)$ and $\nabla \phi(t^n, x^n_p)$. A trilinear interpolation scheme is adopted following Refs. [76, 77, 78]. With reference to Fig. 6, a 2D version of the interpolation scheme (bilinear) for calculating the value of $\phi(t^n, x^n_p)$ at the particle centre $x^n_p = (x_p, y_p)$ reads

$$\phi = \frac{\phi_{0,0} x_p + \phi_{1,0} x_p + \phi_{0,1} y_p + \phi_{1,1} y_p}{2}.$$  

Figure 5: In FPII the phase field variable at the particle position $\phi(x_p)$ is mapped to the particle-interface distance $d$.

Figure 6: Schematic of the variables used to project the delta-function forces from each particle onto the surrounding Eulerian grid nodes (for the 2D case).
\[ \phi(x_p, y_p) = \left(1 - \frac{x_p}{\Delta x}\right) \left(1 - \frac{y_p}{\Delta y}\right) \phi(x_0, y_0) + \frac{x_p}{\Delta x} \left(1 - \frac{y_p}{\Delta y}\right) \phi(x_1, y_0) + \left(1 - \frac{x_p}{\Delta x}\right) \frac{y_p}{\Delta y} \phi(x_0, y_1) + \frac{x_p}{\Delta x} \frac{y_p}{\Delta y} \phi(x_1, y_1), \]

where \( \Delta x \) is the length of the cell edge. The value of \( \nabla \phi(x_p, y_p) \) is interpolated in the same way.

The fluid momentum equation Eq.(13) contains two forcing terms: \( \mathbf{f} \), which can be readily calculated from \( \phi \) obtained in FIPI-Interface according to Eq.(16); The second term includes the hydrodynamic drag \( \mathbf{F}_d \) and the capillary force \( \mathbf{F}_c \) from all particles. An extrapolation scheme is required to project the forces acting at the particle centre to the nearby Eulerian grid points.

As suggested in Ref.[79], only when the weights used in the projection scheme are equal to the weights used in the interpolation scheme, the point-particle algorithm is consistent. An extrapolation scheme, which is exactly the reverse of the interpolation scheme of Fig.(6) and Eq.(30), is implemented. By regularising the Dirac delta function in Eq.(13) with the volume of a computational cell \( \Delta V \), a force \( \hat{f}_p \) from the particle \( j \) located at \( x_p = (x_p, y_p) \) is extrapolated onto the nearby Eulerian grid points as

\[
\mathbf{f}(x, y) = \mathbf{F}(\hat{f}) \left( \frac{1 - x - y}{\Delta x} \right) \left( \frac{1 - x - y}{\Delta y} \right) + \mathbf{f}(x_0, y_0) + \frac{x}{\Delta x} \mathbf{f}(x_1, y_0) + \frac{y}{\Delta y} \mathbf{f}(x_0, y_1) + \frac{x}{\Delta x} \frac{y}{\Delta y} \mathbf{f}(x_1, y_1) \right), \quad (33)
\]

We can denote all the forcing terms in Eq.(13) as \( \mathbf{f}_{\text{total}} \):

\[ \nabla^2 p = \nabla \cdot \mathbf{f}_{\text{total}}. \quad (34) \]

Transforming both \( p \) and \( \mathbf{f}_{\text{total}} \) into Fourier space, we have

\[ -k^2 \hat{p} = i \mathbf{k} \cdot \hat{\mathbf{f}}_{\text{total}} \quad (35) \]

which leads to

\[ \hat{p} = -\frac{i \mathbf{k} \cdot \hat{\mathbf{f}}_{\text{total}}}{k^2}. \quad (36) \]

Multiplying both sides of Equation (37) by \( i \mathbf{k} \) gives

\[ i \mathbf{k} \hat{p} = \frac{\mathbf{k} \cdot (i \mathbf{k} \cdot \hat{\mathbf{f}}_{\text{total}})}{k^2}. \quad (38) \]

The fluid velocity in physical space can then be calculated by performing the inverse Fourier transform of \( \hat{u} \).

When fluid Reynolds number is not small and inertia is not negligible, instead of Eq.(13), full Navier-Stokes equation with the forcing terms \( \mathbf{f}_c \) and \( \mathbf{f}_p \) can be adopted and solved in FIPI-Fluid.

4. Program Documentation

The program consists of the following sub-folders:

- src(folder): source code of the program.
- run(folder): all intermediate executable files and files generated during compiling.
- data(folder): simulation output files.
- conf.in(file): input file which specifies all the parameters.
- makefile(file): the makefile for compiling the executable.

The output files of the program are produced in the folder “data”. Here both the Eulerian fields (phase field and fluid velocity field) at different time steps, and the files describing the location and size of the particles are recorded in legacy VTK format, to be readable with the open-source software “VTK - The Visualisation Toolkit”. The input parameters are set via the script file “conf.in”. This file allows the user to set the following param-
eters:

\( n_x \) : number of grid points in x direction
\( n_y \) : number of grid points in y direction
\( n_z \) : number of grid points in z direction
\( \delta \) : grid size
\( \Delta t \) : time step
\( t_{\text{end}} \) : total number of time steps
\( \gamma \) : bare surface tension value
\( S \) : dimensionless mobility parameter
\( N_p \) : total number of particles
\( A \) : prefactor of the particle-interface capillary force \( F_{pi} \) (see Eq.(11))
\( r_p \) : particle radius
\( r_c \) : cutoff distance of the particle-particle interaction
\( k_n \) : stiffness constant of the particle-particle interaction
\( \text{FIPI}\_\text{Interface} \) : the switch for FIPI-Interface
\( \text{FIPI}\_\text{Track} \) : the switch for FIPI-Track
\( \text{FIPI}\_\text{Fluid} \) : the switch for FIPI-Fluid

FIPI has been developed for LINUX/UNIX environment. The program should be compiled with the Make build automation tool. The program requires the Fast Fourier Transform library FFTW which should be built locally, with the path of “include” and “lib” specified in the makefile. To compile the program, the user should run the command “make” in the programs root folder. The program can be started by executing “./run/fipi”.

The initial conditions for the phase field \( \phi \) and the particle configuration can be specified in the constructors of “Field” class (located in “src\_field.cpp”) and “Group” class (located in “src\_group.cpp”). For numerical stability, it is recommended to choose a time step \( \Delta t < \epsilon^3/(3M\gamma) \).

4.1. Program demonstration

For an immiscible binary fluid that is initially homogeneously mixed, the mixture will spontaneously undergo phase separation [80]. The presence of solid particles adsorbed on the fluid interface can effectively slow down or even stop the phase separation process [2, 18]. Several simulations are shown here to demonstrate that FIPI can capture this physical phenomenon.

The simulation is set up as follows: within a cubic computational domain of length \( L = 2\pi \), the phase field variable \( \phi \) is initialised using a random uniform distribution of values between \(-0.1 \) and \(0.1 \) (this distribution models a homogeneously mixed binary fluid). The Cahn number \( Cn = \epsilon/L \) is set to 0.017. The total simulation time is non-dimensionalised by the characteristic time \( t_c = L\mu/\gamma \), where \( \mu \) is the viscosity. Periodic boundary condition is enforced in all directions.

Three cases listed in Table.(1) using different FIPI components have been simulated. The isocontours of \( \phi = 0 \) at the beginning of the simulations are shown in Fig.(7)(a). Once the simulation starts, the mixture undergoes phase separation and the structure of the fluid interface becomes more clear. The steady states of the three cases are shown in Fig.(7)(b)-(d). For case 1, a highly interconnected fluid interface structure is created at the end of the simulation. For case 2, the two fluid phases get separated completely and two planar fluid interfaces are formed. For case 3, 7000 particles with radius \( a = 0.016L \) are uniformly distributed into the computational domain at the beginning and the simulation, corresponding to a particle volume fraction of 12.2%. The simulation in case 3 is done using all the FIPI modules, resulting in a stable interconnected fluid structure similar to case 1.

### Table 1: Modules enabled in the simulations of the phase separation process.

| Case | Modules enabled          |
|------|--------------------------|
| 1    | FIPI-Interface           |
| 2    | FIPI-Interface + FIPI-Fluid |
| 3    | Full FIPI                |

Figure 7: Iso-surfaces of \( \phi = 0 \) at (a) \( t = 0 \) for all cases, and at the end of the simulation for (b) case 1, (c) case 2 and (d) case 3 (with particles shown).

The dynamics of the phase separation process can be quantitatively studied by monitoring the time evolution of the total free energy of the phase field inside the computational domain [42]

\[
F_{\text{tot}} = \int_{\Omega} F, \tag{40}
\]

where the free energy density \( F \) is defined in Eq.(3). The time history of \( F_{\text{tot}} \) for the three cases simulated is shown in Fig.(8),
with the total energy associated with the surface pressure $\Pi_A$ created by particles in case 3 as comparison.

![Figure 8: Normalised total free energy versus normalised time during phase coarsening of a binary mixture laden with particles. The total free energy $F_{tot}$ is normalised as $F_{tot}' = F_{tot}/\gamma L^2$. The time is normalised by the capillary time scale $t_c = \frac{L^2}{\gamma}$ as $t' = t/t_c$.](image)

For all three cases, $F_{tot}$ starts to decrease after a short time from the beginning. In case 1, $F_{tot}$ decreases much slower than case 2 and 3, due to that the dynamics is only driven by the diffusion in Cahn-Hilliard equation. By the end of the simulation (Fig. (7)(b)) $F_{tot}$ did not reach a steady state value. As the time scale of the simulations with only the diffusion is significantly longer [80], the simulation in case 1 was made not to proceed further.

In case 2 and 3, FIPI-Interface is included therefore the phase separation process is further enhanced by the fluid flows created by the surface tension force. The surface tension flow is essentially a mean curvature flow that minimises the interfacial area, contributing to the reduction of the total free energy. This is evidenced by faster decline rates of $F_{tot}'$ for $t' > 10$ in case 2 and 3.

Both the simulations of case 2 and 3 reach steady state, with $F_{tot}'$ remaining at a higher value in case 3. It is worth to note that $F_{tot}' \approx 2$ at the end of the simulation in case 2, exactly corresponding to the surface free energy of the two planar fluid interfaces shown in Fig.(7)(c). In case 3, the particle monolayer formed at the fluid interface starts to jam the fluid interface after the interface area decreases below a critical value. The jamming effect induced by the particles effectively freezes the phase separation process. On the other hand, as the particles on the fluid interface become more compressed due to the decreasing interface area, the surface pressure $\Pi_A$ induced by the monolayer increases until $\Pi_A \approx F_{tot}'$ (the purple line in Fig.(8)), indicating that the surface tension of the bare fluid interface is balanced by the surface pressure induced by the particle monolayer. As a result the capillary force driving the fluid-fluid demixing is reduced to negligible values.

To test the efficiency of the FIPI, the simulation of the phase separation of a binary fluid was repeated particles of radius $a = 0.004L$. The simulations were performed on a laptop with i5-6200U processor for 1000 time steps and the number of particles $N$ ranges from $10^3$ to $10^8$. The scaling of computational time $t_{sim}$ with $N$ is reported in Fig.(9).

![Figure 9: Simulation time (in minutes) versus number of particles in the computational domain.](image)

For simulations with $N < 10^5$, the order of magnitude of $t_{sim}$ is approximately constant. This indicates that when the solid particles are relatively sparse, most of the computational cost is attributed to the solvers of FIPI-Fluid and FIPI-Interface. When $N > 10^5$, the magnitude of $t_{sim}$ increase dramatically. The slope of the log $t_{sim}$ versus log $N$ remains approximately constant at 1.38. In the current version of FIPI, the scaling of computational cost with number of particles is thus only slightly superlinear. As far as particle-tracking code is concerned, this good performance is mostly due to the implementation of a cell-list sorting algorithm.

5. Validation

Due to the complexity of the physical problems, few validation cases involving many particles and fluid interfaces can be found in literature. We have carried out three validation studies. In the first study, we have simulated the shape of an initially flat fluid interface perturbed by pulling a single adsorbed particle with a constant force acting perpendicularly to the fluid interface. The simulation is expected to produce the analytical solution for this problem. In the second study, the Laplace pressure of a static spherical drop covered with repulsive particles is measured in our simulations. We expect that a sorbed particle with a constant force acting perpendicularly to the fluid interface. The simulation is expected to produce the correct deformation of the drop that is covered by a particle monolayer.
5.1. Distortion of a Flat Fluid Interface by a Single Particle

A particle sitting on an initially flat fluid interface will deform the interface if the weight of the particle is not negligible. The profile \( h(r) \) of the fluid interface deformed by an attached spherical particle can be obtained analytically by solving the non-linear Young-Laplace equation using a perturbation technique [25]:

\[
    h(r) = h_i + h_o - h_{io},
\]

where

\[
    h_i(r) = -a \cos \alpha \cos(\alpha + \theta_c) \times \left[ \ln \left( \frac{B}{4} \right) + \ln \left( \frac{L}{a} + \sqrt{\left( \frac{L}{a} \right)^2 - \cos^2 \alpha \cos^2(\alpha + \theta_c)} \right) + \gamma_c \right],
\]

\[
    h_o(r) = -a \cos \alpha \cos(\alpha + \theta_c) K_0 \left( \frac{r}{\ell_c} \right),
\]

\[
    h_{io}(r) = a \cos \alpha \cos(\alpha + \theta_c) \left( \ln 2 - \ln \left( \frac{r}{\ell_c} \right) - \gamma_c \right).
\]

Here \( a \) is the particle radius, \( \ell_c = \sqrt{\gamma/\Delta \rho g} \) is the capillary length, \( \gamma_c \) is the Euler constant, \( B = a/\sqrt{\gamma/\Delta \rho g} \) is the Bond number and \( K_0 \) is the modified Bessel function of zero order.

Here we simulate the deformation of a flat, horizontal fluid interface when a solid particle is pulled away by an external force that is perpendicular to the interface when a solid particle embedded in the interface is

\[
    \rho_1
\]

\[
    \rho_2
\]

| \( F \) |
|---|

Figure 10: Schematic of meniscus deformation when applying an external force \( F \) to a particle initially residing on a flat fluid interface.

The computational domain is a cubic box of side \( L = 2\pi \). A horizontal flat fluid interface is positioned at the centre of the domain. Periodic boundary condition is enforced along three orthogonal directions. A neutrally buoyant spherical particle of radius \( a = \pi/32 \) is placed on the fluid interface at the centre of the domain. The densities of the fluid above and below the interface are \( \rho_1 \) and \( \rho_2 \) respectively. A constant external force \( F \) is exerted on the particle, pulling it vertically downward. After the simulation started, the particle began to move away from its equilibrium position and deform the fluid interface. The time history of the maximum displacement of the fluid interface is shown in Fig.(11).

\[
    \rho_1
\]

\[
    \rho_2
\]

| \( \gamma \) |
|---|

Figure 11: (a) Time evolution of the maximum downward displacement of the fluid interface caused by exerting a downward force of magnitude \( F \) on the particle. The displacement of the fluid interface is normalised by \( h^* = \frac{h}{\pi \gamma} \) and the time by \( t^* = \frac{t}{\pi \gamma} \). The shape of the particle laden interface at three different time instants is shown in (b)-(d), corresponding to the time instants A,B,C in (a).

For the interfacial region far from the particle \( r/l_c \gg 1 \), the gradient of deformation caused by the particle can be assumed to be relatively small \( \nabla h^2 \ll 1 \), and the linearised form of the Laplace equation can be obtained [81]:

\[
    \gamma \nabla^2 h = -\Delta \rho gh - F \delta(r).
\]

The axisymmetric solution of this equation is

\[
    h(r) = \frac{F}{2\pi\gamma} K_0 \left( \frac{r}{\ell_c} \right).
\]

This solution is exactly the outer solution given by Eq.(43) considering the expression of \( F \) given by Eq.(10). Since in FIPI particles are represented by singular forces, the simulated deformed fluid interface profile is expected to match Eq.(46) at large distance from the particle.

At steady state, the profile of the fluid interface was examined as a function of two dimensionless numbers: the mobility parameter \( S = \sqrt{M \mu/\epsilon} \), which defines the strength of the diffusion in Cahn-Hilliard equation Eq.(1) and the Cahn number \( Cn = \epsilon/L \), which measures the relative thickness of the fluid
interface.

In Fig.(12), the interface profiles produced by the simulations are plotted against the solution given by Eq.(46). We have kept $Cn = 0.022$ constant and varied $S$ by changing the mobility parameter $M$. The theoretical solution given by Eq.(46) suggests that $h \to \infty$ as $r \to 0$. For all values of $S$, the simulation results match Eq.(46) very well except near the location of the particle. This is expected as the singular force from the particle is regularised by the finite grid size $\Delta x$ in FIP1. The solution given by Eq.(46) is asymptotically approached by decreasing the value of $S$. It is evident that a smaller $S$ makes the fluid interface become more “flexible”, while a larger $S$ can smear out the deformation of the fluid interface. It is not advised to reduce $S$ excessively [40], as a finite value of $S$ is necessary to ensure that the profile of $\phi$ across the fluid interface does not become too stretched under shearing flows and remains as close to Eq.(4) as possible.

On the other hand, to make the fluid interface profile produced by simulations approach Eq.(46), $Cn$ must be reduced towards zero. A small $Cn$ imposes a challenging requirement on spatial resolution. Because the fluid interface region has to be resolved with a minimal of $3 \sim 4$ grid points with a phase field method, the grid spacing has to be reduced together with $Cn$. In Fig.(13), we have shown the fluid interface profile from simulations with different values of $Cn$ while keeping $S = 0.1$. The perturbed fluid interface is clearly sharper for smaller values of $Cn$.

### 5.2. Laplace Pressure of a Particle-Covered Drop

Consider a spherical drop of radius $R$ in the absence of interfacial particles. The Laplace pressure, i.e. the pressure difference between the inside and outside of the drop, is given by

$$\Delta p = \frac{2\gamma}{R}. \quad (47)$$

Introducing particles on the drop surface will change the surface tension of the drop because the repulsion between the particles induces a surface pressure $\Pi_s$ opposing the bare surface tension. Considering Eq.(23), the modified Young-Laplace equation is

$$\Delta p = \frac{2 (\gamma - \Pi_s)}{R}. \quad (48)$$

We induce a surface pressure of magnitude ranging from $0$ to $\gamma$ by adjusting the value of $k_c$ (in the current study, we set $r_c = 5a$ and $k_c$ varies from $0$ to $0.5\gamma$). The corresponding pressure difference $\Delta p$ across the surface of the drop - the pressure difference is approximately uniform because of the uniform particle distribution - is measured in the simulations. The results are reported in Fig.(15). The measured pressure drop is in excellent agreement with Eq.(48).
It is known that spurious velocities near the fluid interface are produced due an imbalance of the discretised surface tension force and the inevitable numerical error due to the approximation of the curvature of the fluid interface. We also report here a grid convergence study in which we measured the maximum magnitude of the spurious fluid velocity near a static drop without particles as a function of the number of discretisation points. A spherical drop with radius \( R = 0.39L \) is placed at the centre of a cubic domain of side length \( L \). For this test we set \( Cn = 0.0156 \). The magnitude of the fluid velocity on a plane cutting through the drop is shown as a color plot in Fig.16 for different values of the number of discretisation points \( n \) (here \( n \) refers to the number of nodes along each orthogonal direction). When \( n = 16 \) the drop interface is not accurately resolved and the spurious fluid velocity is significant. The magnitude of the spurious fluid velocity becomes a negligible fraction of the capillary velocity \( u_c = \gamma/\mu \) for \( n = 64 \). The ratio \( u_{\text{max}}/u_c \) of maximum velocity magnitude and capillary velocity is plotted against \( n \) in Fig.17. The fastest convergence is achieved for \( n < 200 \) approximately, while a smaller convergence rate is obtained for larger values of \( n \). For a reasonably accurate simulation it is necessary to limit the magnitude of spurious velocity to avoid the particles being convected out of the fluid interface.

Figure 16: Color map of the ratio \( u_{\text{max}}/u_c \) of spurious velocity magnitude \( u_s \) (evaluated at the interface) and capillary velocity \( u_c = \gamma/\mu \) on a plane plane passing through the drop centre for (a) \( n = 16 \), (b) \( n = 32 \), (c) \( n = 64 \) and (d) \( n = 128 \). Here the drop is free of particles.

Figure 17: Convergence of the maximum spurious fluid velocity magnitude on \( n \), corresponding to the data of Fig. 16. The log functions are base 10.

5.3. Effective Surface Tension of a Pendant Drop

Pendant-drop tensiometry is a standard method for measuring the surface tension of surfactant-laden or particle-laden drops [82, 83]. In this study, a pendant-drop tensiometry experiment is reproduced in simulation, using a pendant drop covered with a monolayer of repulsive particles. The simulation is set up in a rectangular box of \([2\pi, 2\pi, 3\pi]\), discretised into \( 64 \times 64 \times 96 \) nodes. Periodic boundary conditions are enforced.
along the three orthogonal axis. At the beginning of the simulations, a spherical drop with an initial radius \( R = 0.8\pi \) is positioned at the centre of domain. The drop is held at the top by a ring of stationary solid particles. The Bond number for the drop \( Bo = \Delta \rho g R^2 / \gamma \), calculated by using the surface tension of the bare fluid interface, is set to 0.1. At the beginning of the simulations, 5000 solid particles with radius \( a = 0.02R \) are uniformly distributed across the surface of the drop. The surface pressure is modulated by adjusting \( k_e \) while keeping \( r_e = 5a \) in the particle-particle interaction model given by Eq.(12).

Once the simulation starts, the spherical drop starts to deform under the effect of gravity, before reaching a steady state in which gravity and the effective surface tension balance each other. At steady state, we have calculated the effective surface tension of the drop by two methods:

- The shape fitting technique of Fordham [84, 85].
- A direct calculation using Eq.(23) [70].

We performed 1 simulation of a clean pendant drop and 4 simulations of a particle-covered pendant drop with different strengths of interparticle repulsion.

![Figure 18: Steady state shape of a pendant drop for (a) Case 1 (b) Case 5.](image)

In the case 1, the drop surface is free of particles therefore we expect \( \Pi_s = 0 \) and \( \gamma_{eff} = \gamma \). The simulation error is negligibly small, confirming that the combination of FIPI-Fluid and FIPI-Interface is capable of producing the drop shape consistent with the prescribed surface tension value. For case 2 to 5, repulsive particles fully covering the drop are considered in the simulation. The relative error between Fordham’s method and the direct calculation of surface pressure indicates that FIPI captures the correct deformation of a particle-laden drop with the modified surface tension of the fluid interface. Note that the time-dependent deformation of a pendant drop simulated with FIPI has been analysed in one of our previous publications [70].

### 6. Unperturbed fluid velocity in the calculation of the hydrodynamic drag force

As for other point-particle models, a significant challenge in FIPI is the prescription of fluid velocity \( \mathbf{u}_f(x_p, t) \) appearing in Eq.(9). This quantity should be interpreted as the fluid velocity at the location of the particle if the particle was not present. However, due to two-way coupling between fluid and particles, the fluid velocity available at the location of the particle during a FIPI simulation includes the flow disturbance caused by the particle itself. With the current FIPI implementation, the inclusion of the self-induced fluid velocity induces an error of order \( \mathcal{O}(\frac{F}{\rho_{\text{min}}}) \), where \( F \) is the magnitude of the total force on the particle minus hydrodynamics and capillary forces. Hence taking \( \mathbf{u}_f(x_p, t) = \mathbf{u}(x_p, t) \) is a reasonably good approximation provided inter-particle and external forces on the particles are relatively small. Increasing the length scale over which the delta forcing is discretised (\( \Delta x \) in the current implementation) can alleviate this problem but not completely overcome it.

For particles causing strong disturbances to the fluid, a correction would be needed to subtract the self-induced velocity [86]. In an unbounded single-phase domain, one can obtain the correct undisturbed fluid velocity by subtracting the contribution from the local Stokeslet [87]. However, in our case, one should remove the velocity field induced by a single particle residing on or near fluid interface when simulated on a...
grid of size \( \Delta x \). An approximation of such velocity could be obtained by solving analytically the Stokes problem for a regularised Stokeslet centered near or in the fluid interface, with \( \Delta x \) as regularisation parameter (for the unregularised problem, see Ref. [53]). In the current implementation of FIPI, we assume \( u_j(x_p,t) = 0 \) for quasi-hydrostatic simulations and \( u_j(x_p,t) \approx u(x_p,t) \) when the sum of the external and interparticle forces acting on each particle is much smaller than the capillary force \( \gamma a \).

7. Conclusion

We have developed the FIPI method for simulating problems involving particles suspended in two-phase liquid-liquid systems and interacting with fluid interfaces of complex morphology. The method is extremely fast, and meaningful simulations can be run on a common desktop or laptop computer in a matter of a few hours.

The simulations with the FIPI method are validated against three cases. In the first case, the meniscus deformation due to a single particle embedded in an otherwise flat fluid interface and acted upon by an external force is simulated and the results validated against an analytical solution. The meniscus shape is accurately captured at a sufficient distance from the particle centre. In the second test, the pressure difference across a spherical droplet covered with repulsive particles is computed. The results compare well with an expression that includes the surface pressure induced by the particles. In the last test, we simulate a pendant drop tensiometry configuration. The effective surface tension given by FIPI is compared against tabulated values due to Fordham [85] obtained by fitting the shape of a droplet having uniform surface tension to a numerical solution of the non-linear Young-Laplace equation. The comparison is satisfactory, suggesting that the shape of the drop simulated by FIPI is quantitatively correct.

A limitation that we discuss is the conceptual challenge of prescribing the unperturbed fluid velocity at the particle position when calculating the drag force. While this limitation is not a characteristic of FIPI (all point particle models share this issue), we believe that future improvements of FIPI should address this aspect.

The FIPI method is highly extensible, and provides an efficient platform for the study of Pickering emulsions, froth flotation problems, liquid marbles, etc. While in the current paper we have focused on monodispersed particle systems, a distribution of particle sizes can be easily accounted for. Polydispersity in contact angles can also be factored into FIPI method by associating in Eq.(11) a different parameter \( f \) to each particle. Gas-liquid systems could also be simulated by adopting a phase-field method that can handle large density ratios [88].

For a gas-liquid system, the drag force on each particle should be reduced by an amount that depends on the filling angle [13]. As a practical approximation, the filling angle could be well approximated by the assigned contact angle \( \theta_c \), a quantity that depends only on the surface energies of the liquid-solid, liquid-gas and gas-solid interfaces.

Acknowledgement

Financial support from the European Community through Career Integration Grant FLOWMAT Flow and Capillarity in Material Science (No. 618335) and ERC grant FlexNanoFlow (No. 715475) are acknowledged.

[1] B. J. Park, J. P. Pantina, E. M. Furst, M. Oettel, S. Reynaert, J. Vermant, Direct measurements of the effects of salt and surfactant on interaction forces between colloidal particles at water-oil interfaces, Langmuir 24 (5) (2008) 1686–1694.
[2] K. Stratford, R. Adhikari, I. Pagonabarraga, J.-C. Desplat, M. E. Cates, Colloidal jamming at interfaces: A route to fluid-bicontinuous gels, Science 309 (5744) (2005) 2198–2201.
[3] E. Herzig, K. White, A. Schofield, W. Poon, P. Clegg, Bicontinuous emulsions stabilized solely by colloidal particles, Nature materials 6 (12) (2007) 966.
[4] E. Dickinson, Food emulsions and foams: stabilization by particles, Current Opinion in Colloid & Interface Science 15 (1-2) (2010) 40–49.
[5] B. A. Saleimanov, F. Ismailov, E. Veliyev, Nanofluid for enhanced oil recovery, Journal of Petroleum Science and Engineering 78 (2) (2011) 431–437.
[6] A. Dinsmore, M. F. Hu, M. Nikolaides, M. Marquez, A. Bausch, D. Weitz, Colloidosomes: selectively permeable capsules composed of colloidal particles, Science 298 (5595) (2002) 1006–1009.
[7] E. J. Stancik, M. Kouhkan, G. G. Fuller, Coalescence of particle-laden fluid interfaces, Langmuir 20 (1) (2004) 90–94.
[8] A. J. Mendoza, E. Guzmán, F. Martínez-Pedroso, H. Ritacco, R. G. Rubio, F. Ortega, V. M. Starov, R. Miller, Particle laden fluid interfaces: dynamics and interfacial rheology, Advances in colloid and interface science 206 (2014) 303–319.
[9] F. Fenouillot, P. Cassagnau, J.-C. Majesté, Uneven distribution of nanoparticles in immiscible fluids: morphology development in polymer blends, Polymer 50 (6) (2009) 1333–1350.
[10] A. B. Pawar, M. Caggioni, R. Ergun, R. W. Hartel, P. T. Spicer, Arrested coalescence in pickering emulsions, Soft Matter 7 (17) (2011) 7710–7716.
[11] A. J. Ladd, Numerical simulations of particulate suspensions via a discretized Boltzmann equation, part 1. Theoretical foundation, Journal of fluid mechanics 271 (1994) 285–309.
[12] X. Shan, H. Chen, Lattice boltzmann model for simulating flows with multiple phases and components, Physical Review E 47 (3) (1993) 1815.
[13] M. R. Swift, E. Orlandini, W. Osborn, J. Yeomans, Lattice boltzmann simulations of liquid-gas and binary fluid systems, Physical Review E 54 (5) (1996) 5041.
[14] A. S. Joshi, Y. Sun, Wetting dynamics and particle deposition for an evaporating colloidal drop: A lattice boltzmann study, Physical Review E 82 (4) (2010) 041401.
[15] F. Jansen, J. Harting, From bijels to pickering emulsions: A lattice boltzmann study, Physical Review E 83 (4) (2011) 046707.
[16] S. Aland, J. Lowengrub, A. Voigt, A continuum model of colloid-stabilized interfaces, Physics of Fluids 23 (6) (2011) 062103.
[17] N. Jaensson, M. Hulsen, P. Anderson, Stokes–cahn–hilliard formulations and simulations of two-phase flows with suspended rigid particles, Computers & Fluids 111 (2015) 1–17.
[18] P. C. Millett, Y. U. Wang, Diffuse-interface field approach to modeling arbitrarily-shaped particles at fluid–fluid interfaces, Journal of colloidal and interface science 353 (1) (2011) 46–51.
[19] L. Botto, A. Prosperetti, A fully resolved numerical simulation of turbulent flow past one or several spherical particles, Physics of fluids 24 (1) (2012) 013303.
[20] D. Vella, Floating versus sinking, Annual Review of Fluid Mechanics 47 (2015) 115–135.
in isotropic turbulence, Physics of Fluids A: Fluid Dynamics 2 (7) (1990) 1191–1203.

[77] E. Calzavarini, R. Volk, M. Bourgoin, E. Lévéque, J.-F. Pinton, F. Toschi, Acceleration statistics of finite-sized particles in turbulent flow: the role of faxén forces, Journal of Fluid Mechanics 630 (2009) 179–189.

[78] D. W. Rouson, J. K. Eaton, On the preferential concentration of solid particles in turbulent channel flow, Journal of Fluid Mechanics 428 (2001) 149–169.

[79] S. Sundaram, L. R. Collins, Numerical considerations in simulating a turbulent suspension of finite-volume particles, Journal of Computational Physics 124 (2) (1996) 337–350.

[80] J. Zhu, L.-Q. Chen, J. Shen, V. Tikare, Coarsening kinetics from a variable-mobility cahn-hilliard equation: Application of a semi-implicit fourier spectral method, Physical Review E 60 (4) (1999) 3564.

[81] P. A. Kralchevsky, K. Nagayama, Capillary interactions between particles bound to interfaces, liquid films and biomembranes, Advances in colloid and interface science 85 (2-3) (2000) 145–192.

[82] Y. Rotenberg, L. Boruvka, A. Neumann, Determination of surface tension and contact angle from the shapes of axisymmetric fluid interfaces, Journal of colloid and interface science 93 (1) (1983) 169–183.

[83] F. Hansen, G. Rødsrud, Surface tension by pendant drop: I. a fast standard instrument using computer image analysis, Journal of colloid and interface science 141 (1) (1991) 1–9.

[84] C. E. Staufer, The measurement of surface tension by the pendant drop technique, The journal of physical chemistry 69 (6) (1965) 1933–1938.

[85] S. Fordham, On the calculation of surface tension from measurements of pendant drops, in: Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, Vol. 194, The Royal Society, 1948, pp. 1–16.

[86] J. Horwitz, A. Mani, Accurate calculation of stokes drag for point-particle tracking in two-way coupled flows, Journal of Computational Physics 318 (2016) 85–109.

[87] P. Saffman, On the settling speed of free and fixed suspensions, Studies in Applied Mathematics 52 (2) (1973) 115–127.

[88] H. Ding, P. D. Spelt, C. Shu, Diffuse interface model for incompressible two-phase flows with large density ratios, Journal of Computational Physics 226 (2) (2007) 2078–2095.