



Delft University of Technology

FIPI

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

Gu, Chuan; Botto, Lorenzo

DOI

[10.1016/j.cpc.2020.107447](https://doi.org/10.1016/j.cpc.2020.107447)

Publication date

2020

Document Version

Final published version

Published in

Computer Physics Communications

Citation (APA)

Gu, C., & Botto, L. (2020). FIPI: A fast numerical method for the simulation of particle-laden fluid interfaces. *Computer Physics Communications*, 256, Article 107447. <https://doi.org/10.1016/j.cpc.2020.107447>

Important note

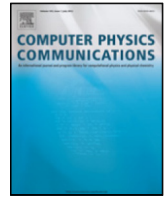
To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.



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

Chuan Gu^a, Lorenzo Botto^{a,b,*}

^a School of Engineering and Material Science, Queen Mary University of London, E1 4NS, London, United Kingdom

^b Department of Process and Energy, Delft University of Technology, 2628 CB, Delft, The Netherlands

ARTICLE INFO

Article history:

Received 25 July 2019

Received in revised form 2 June 2020

Accepted 18 June 2020

Available online 27 June 2020

Keywords:

Particle-laden flows
Liquid-liquid interfaces
Multiphase flows
Surface tension

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 100k 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, Bijelformation and other multiphase systems.

Program summary

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

© 2020 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

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 nm– 1 μ m) to the mesoscale of the bubbles, drops, or interfacial structures (~ 100 μ m– 1 cm) the particles are adsorbed to. 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.

[☆] The review of this paper was arranged by David W. Walker.

* Corresponding author at: Department of Process and Energy, Delft University of Technology, 2628 CB, Delft, The Netherlands.

E-mail address: l.botto@qmul.ac.uk (L. Botto).

Other authors have used a lattice Boltzmann approach, where the bounce-back model for resolving solid particles [14] is combined with a LB 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 [12,24]. 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 [25–27]. 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 [13,30], 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 [31–33] 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 [36–38], 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.

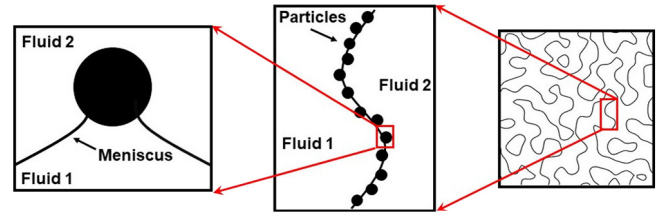


Fig. 1. A particle-laden fluid interface displays a hierarchy of length scales, from the size of the particles to the characteristic size of the mesoscale multiphase structures.

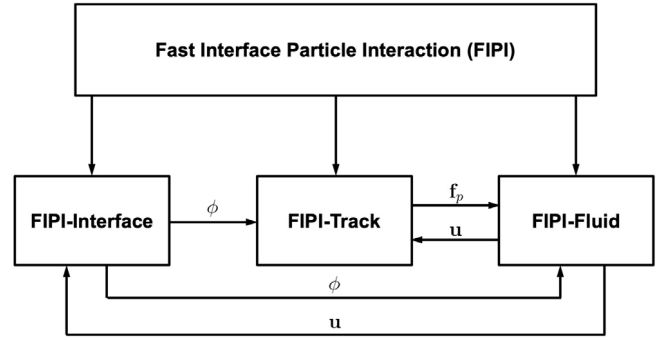


Fig. 2. Structure of the FIPI program showing the input and output relationships between each module.

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 ϕ , to the other two modules; FIPI-Track provides the forcing term \mathbf{f}_p 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 [40,41]. In the phase-field method the fluid interface has a finite thickness, across which a scalar phase field variable ϕ changes continuously. The evolution of ϕ follows the Cahn–Hilliard equation

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = \nabla \cdot M \nabla \xi, \quad (1)$$

where \mathbf{u} is the fluid velocity, M is the mobility parameter (which we take as a constant [42]), and ξ is the chemical potential. The chemical potential is defined as the functional derivative of the total free energy density:

$$\xi = \frac{\partial F}{\partial \phi}. \quad (2)$$

Following [43] the total free energy density is modelled in FIPI as

$$F(\phi, \nabla \phi) = \frac{\lambda}{4\epsilon^2} (\phi^2 - 1)^2 + \frac{1}{2} \lambda |\nabla \phi|^2, \quad (3)$$

where λ and ϵ are two constant parameters. At steady state and in the absence of convection, Eq. (1) admits the solution

$$\phi = \tanh\left(\frac{x}{\sqrt{2}\epsilon}\right), \quad (4)$$

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 ϕ given by Eq. (4), yields the surface tension [41]

$$\gamma = \int_{-\infty}^{+\infty} F(\phi, \nabla \phi) dx = \frac{2\sqrt{2}\lambda}{3\epsilon}. \quad (5)$$

This is the bare surface tension of the fluid interface. Eq. (4) shows that the numerical thickness of the fluid interface depends solely on ϵ . The ratio of λ and ϵ determines the surface tension of the fluid interface in the absence of the particle (Eq. (5)). In actual simulations ϵ and λ are chosen based on preliminary tests. The choice for ϵ is set by two constraints. The value of this parameter must be large enough so that ϕ is sufficiently well-resolved across the fluid interface (i.e., ϵ is set by the grid mesh size). On the other hand, ϵ 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 ϵ is chosen, the desired bare surface tension value is obtained by adjusting λ 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 $\mathbf{u} = \mathbf{0}$, it can be used to calculate γ when the diffusion time scale is much smaller than the convective time scale, $R/u \gg \epsilon^4/M\lambda$, with R being the characteristic radius of fluid interface curvature.

FIPI-Track. Newton's equation of motion for each particle is

$$m_p \dot{\mathbf{u}}_p = m_p \mathbf{g} + \mathbf{F} + \mathbf{F}_{pp}, \quad (6)$$

where $\dot{\mathbf{u}}_p$ is the particle velocity, m_p is the particle mass, \mathbf{F} is the force on the particle due to the surrounding fluids (including the capillary 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]

$$\mathbf{F} = \oint_{CL} \gamma \mathbf{n}_c ds + \int_{S_p} (-p\mathbf{I} + \boldsymbol{\sigma}) \cdot \mathbf{n} dA + \mathbf{F}_b, \quad (7)$$

where \mathbf{n}_c is the unit vector normal to the contact line and tangent to the fluid interface, p is the hydrodynamic pressure, $\boldsymbol{\sigma}$ 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}_c and \mathbf{n} point out of the particle. The first integral on the right hand 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

$$\mathbf{F} \approx \mathbf{F}_{pi}(\phi(\mathbf{x}_p, t)) + \mathbf{F}_h(\mathbf{x}_p, \dot{\mathbf{x}}_p, \mathbf{u}(\mathbf{x}_p, t)) \quad (8)$$

where \mathbf{F}_{pi} , \mathbf{F}_h 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, particle inertia is neglected as it is typically very small for colloidal particles in particle-laden interfaces problems.

Different closures for \mathbf{F}_{pi} and \mathbf{F}_h are possible. A suitable model should be simple (to avoid uncontrollable artefacts produced by models having too many parameters) and be based on robust physical considerations. In the following we present the models of \mathbf{F}_{pi} and \mathbf{F}_h incorporated in the current version of FIPI. These models are parameterised on local variables: the particle velocity $\dot{\mathbf{x}}_p$, the particle position \mathbf{x}_p and the local phase field variable $\phi(\mathbf{x}_p, t)$. A model for \mathbf{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 [46–49]. 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 studies examining particles moving close to a fluid interface but not touching it, several investigations have focused on the case of particles completely embedded in the fluid interface [52–55]. 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, approach.

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

$$\mathbf{F}_h = -6\pi\mu a f_d (\dot{\mathbf{x}}_p - \mathbf{u}_f(\mathbf{x}_p, t)), \quad (9)$$

where $\mathbf{u}_f(\mathbf{x}_p, t)$ 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 be made a function of 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 $\mathbf{u}_f(\mathbf{x}_p, t)$ is to be interpreted as the fluid velocity that would be present at \mathbf{x}_p if the particle was not there. For cases in which the total non-hydrodynamic force on the particles is not too large, we set $\mathbf{u}_f(\mathbf{x}_p, t) = \mathbf{u}(\mathbf{x}_p, t)$ (the issue of the self-induced fluid velocity is discussed in Section 6). Far-field hydrodynamic interactions between particles are accounted for at the point-force level, in that $\mathbf{u}(\mathbf{x}_p, t)$ 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].

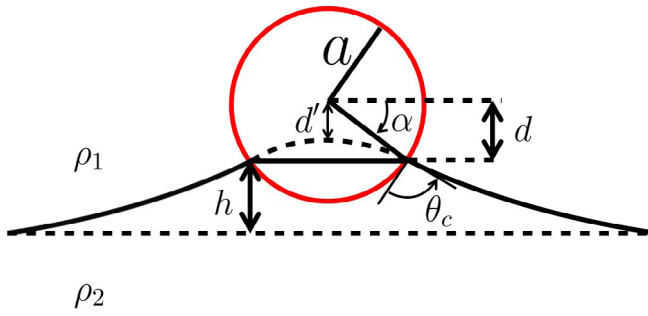


Fig. 3. Schematic illustrating the definition of contact angle θ_c , filling angle α , distortion of the fluid interface h and distance d' from the particle centre to the nearest fluid interface passing through the particle.

Capillary force model. The capillary force $\mathbf{F}_{pi} = \oint_{CL} \gamma \mathbf{n}_c ds$ exerted on a spherical particle embedded in a fluid interface results from the surface tension acting at the three-phase contact line. 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 exact expression [25]

$$F_{pi} = 2\pi\gamma a \cos \alpha \cos(\alpha + \theta_c), \quad (10)$$

where θ_c is the contact angle and α 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 [11,25].

In FIPI, the boundary conditions at particle surfaces are replaced by a force and 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 α and θ_c is not available. However, it is possible to model F_{pi} by using the fact 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

$$\mathbf{F}_{pi} = \begin{cases} f\pi\gamma\mathbf{d}', & \text{if } |\mathbf{d}'| < a \\ \mathbf{0}, & \text{if } |\mathbf{d}'| > a, \end{cases} \quad (11)$$

where \mathbf{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 θ_c . 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 $|\mathbf{d}'| < a$, the particle is considered to be adsorbed to the fluid interface and the capillary force follows a linear relationship with the distance \mathbf{d}' . When $|\mathbf{d}'| > a$, the particle is considered detached from the fluid interface and the capillary force is set to 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

FIPI, 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

$$\mathbf{F}_{pp} = \begin{cases} -k_c(r - r_c)\mathbf{n} & \text{if } r < r_c \\ \mathbf{0}, & \text{if } r > r_c, \end{cases} \quad (12)$$

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 sufficiently regular. This is the case for interfacial colloids, since in this case the dynamics is damped by viscosity and furthermore each particle is often interacting with several neighbouring particles.

FIPI-Fluid. In a fully-resolved simulation, 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 FIPI is the simulation of a large number of particles, a 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):

$$\mathbf{0} = -\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{f}_c + \mathbf{f}_p \quad (13)$$

and

$$\nabla \cdot \mathbf{u} = 0. \quad (14)$$

The particle forcing term is

$$\mathbf{f}_p = -\sum_{j=1}^N \left(\mathbf{F}_{pi}^{(j)} + \mathbf{F}_h^{(j)} \right) \delta(\mathbf{x} - \mathbf{x}_p^{(j)}). \quad (15)$$

and \mathbf{f}_c 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 runs over the N particles present in the computational domain, contains three-dimensional Dirac delta functions centred at each particle. The forces $\mathbf{F}_{pi}^{(j)}$ and $\mathbf{F}_h^{(j)}$ have the form given by Eq. (11) and Eq. (9), respectively.

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

$$\mathbf{f}_c = \xi \nabla \phi = \left(\frac{\lambda}{\epsilon^2} (\phi^3 - \phi) - \lambda \nabla^2 \phi \right) \nabla \phi. \quad (16)$$

It can be shown that as $\epsilon \rightarrow 0$, the capillary force approaches the value

$$\mathbf{f}_c = -\mathbf{n}\kappa\gamma\delta(z), \quad (17)$$

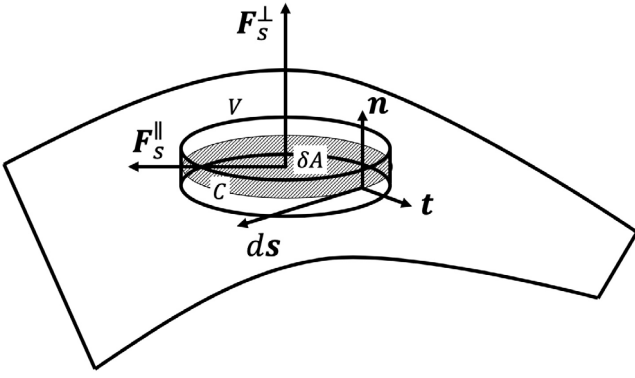


Fig. 4. Schematic of an arbitrary surface patch δA within a fluid interface S . The surface stress induced by the particles is obtained by averaging over δA , which is assumed to contain many particles.

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

2.1. Modulation of surface stresses in FIPI

By comparing Eqs. (15) and (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 across a finite thickness $O(\epsilon)$. In the actual implementation, the Dirac delta forces from the particles are smoothed out over the mesh size Δx .

In the continuum limit, when the interparticle separation is much smaller than the radius of fluid interface curvature, the formulation Eqs. (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 pillbox region V (see Fig. 4) 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.

Using the surface divergence theorem [69], the volume integral of the capillary force can be written as

$$\int_V \mathbf{f}_c dV = - \int_{\delta A} \gamma \mathbf{n} \kappa dA = \int_C \gamma \mathbf{t} ds, \quad (18)$$

where δA is the intersection of V with the fluid interface, C is the boundary of δA , \mathbf{t} is the unit vector parallel to the interface and perpendicular to δA , and \mathbf{n} is the unit normal vector perpendicular to the interface and pointing towards the convex side of the interface. Eq. (18) shows that the integral of the volume force \mathbf{f}_c , acting perpendicularly to the fluid interface, is equivalent to the line integral of a tangential force. In the case of \mathbf{f}_c this tangential force is the surface tension.

Let us now consider the effect of the particles, by taking the volume integral of the particle forcing term \mathbf{f}_p . Neglecting the particle weight for simplicity and using Eqs. (6) and (8), we can write

$$\int_V \mathbf{f}_p dV = \int_V \sum_{j=1}^N (\mathbf{F}_{pp}^{(j)} \delta(\mathbf{x} - \mathbf{x}_p^{(j)})) dV = \sum_{\alpha} \sum_{\beta \neq \alpha} \mathbf{F}_{pp}^{(\alpha\beta)}. \quad (19)$$

In this expression $\mathbf{F}_{pp}^{(\alpha\beta)}$ is the interparticle force exerted by particle β on particle α . The index α runs over the particles comprised

within δA , and the index β runs over all the particles. In analogy with Eq. (18), the integral of the normal force \mathbf{f}_p produce by the particles on the fluid interface corresponds to a tangential force. This tangential force is due to the lateral interaction between the particles in δA and all the other particles.

The comparison between Eqs. (18) and (19) suggests that it should be possible to recast the effect of interparticle interactions in terms of a line integral of a suitably defined surface stress. This can be done by noting that [70]

$$\lim_{\delta A \rightarrow 0} \frac{1}{\delta A} \sum_{\alpha} \sum_{\beta \neq \alpha} \mathbf{F}_{pp}^{(\alpha\beta)} = -\nabla_s \cdot \boldsymbol{\tau}_s \quad (20)$$

where

$$\boldsymbol{\tau}_s(\mathbf{x}_s) = \frac{1}{2} \sum_{\alpha} \sum_{\beta \neq \alpha} \mathbf{F}_{pp}^{(\alpha\beta)} (\mathbf{y}_s^{\beta} - \mathbf{y}_s^{\alpha}) G(\mathbf{x}_s - \mathbf{y}_s^{\alpha}). \quad (21)$$

is the particle-induced surface stress at \mathbf{x}_s . Here G is a 2D filter function with support on δA . The limit in Eq. (20) requires δA to include a sufficient amount of particles so that a continuum representation is approximately valid. Using the divergence theorem [71] and decomposing $\boldsymbol{\tau}_s$ into its isotropic component $-\Pi_s \mathbf{I}$ and its deviatoric component $\boldsymbol{\tau}'$, we can write

$$\int_V (\mathbf{f}_c + \mathbf{f}_p) dV = \int_C (\gamma - \Pi_s) \mathbf{t} ds + \int_C \boldsymbol{\tau}'_s \cdot \mathbf{t} ds, \quad (22)$$

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

$$\gamma_{eff} = \gamma - \Pi_s, \quad (23)$$

This expression, which is thermodynamically correct [72], can be expected from intuition. If Π_s is positive, corresponding to a repulsive interaction between the particles, the effective isotropic tension is reduced. The tensor $\boldsymbol{\tau}'_s$ 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 Π_s and $\boldsymbol{\tau}'_s$ 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 FIPI with other discretisation approaches (LBM, Finite Volume, finite difference, etc.) that do not require period boundary condition, provided sufficient resolution can be guaranteed.

With $\phi(\mathbf{x}) = \int \hat{\phi}(\mathbf{k}) e^{2\pi i \mathbf{x} \cdot \mathbf{k}} d\mathbf{k}$, Eq. (1) can be rewritten in Fourier space as

$$\frac{\partial \hat{\phi}}{\partial t} + \{\mathbf{u} \cdot \nabla \phi\}_k = -k^2 M \left\{ \frac{\lambda}{\epsilon^2} (\phi^3 - \phi) \right\}_k - k^4 M \lambda \hat{\phi} \quad (24)$$

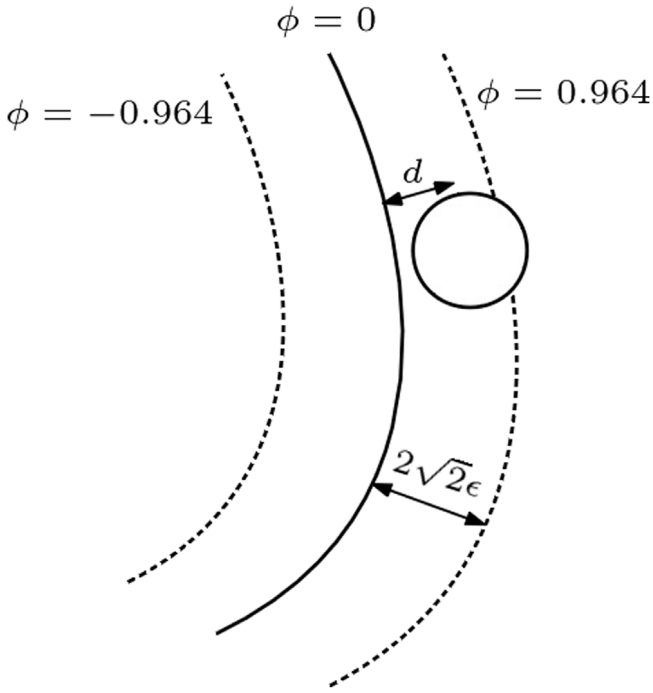


Fig. 5. In FIPI the phase field variable at the particle position $\phi(\mathbf{x}_p)$ is mapped to the particle-interface distance d .

where \mathbf{k} is the wave vector, $\hat{\phi}$ is the phase-field variable in Fourier space, $\{\}_k$ is the Fourier transform of the nonlinear term inside bracket and $k = |\mathbf{k}|$.

The time derivative term $\frac{\partial \hat{\phi}}{\partial t}$ is discretised by a second-order backward differencing formula (BDF). The fourth-order gradient term is treated implicitly while the nonlinear terms are 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 Fourier space is

$$\frac{3\hat{\phi}^{n+1} - 4\hat{\phi}^n + \hat{\phi}^{n-1}}{2\Delta t} = -(2\hat{e}^n - \hat{e}^{n-1}) - k^2 M(2\hat{f}^n - \hat{f}^{n-1}) - k^4 M \lambda \hat{\phi}^{n+1} \quad (25)$$

$$\hat{e}^n = \left\{ \mathbf{u}^n \cdot \left(-i\mathbf{k} \hat{\phi}^n \right)_{-k} \right\}_k \quad (26)$$

$$\hat{f}^n = \left\{ \frac{\lambda}{\epsilon^2} \left((\phi^n)^3 - \phi^n \right) \right\}_k \quad (27)$$

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

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

$$\mathbf{u}_p(t^n, \mathbf{x}_p^n) = \mathbf{u}_f(t^n, \mathbf{x}_p^n) + \frac{\mathbf{F}_{pp}(t^n, \mathbf{x}_p^n) + \mathbf{F}_{pi}(t^n, \mathbf{x}_p^n)}{6\pi\mu a f_d} \quad (28)$$

The particle–particle interaction force \mathbf{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 \mathbf{F}_{pi} requires knowledge of the distance \mathbf{d} from particle centre \mathbf{x}_p^n to the nearest fluid interface, according to Eq. (11). We have found that having the phase variable $\phi(t^n, \mathbf{x}_p^n)$ at our disposal is very useful in this respect. Assuming the phase field profile ϕ is close to equilibrium, we have $\phi \simeq \phi_{eq} = \tanh(\frac{x}{\sqrt{2}\epsilon})$ in the proximity of the fluid interface. An effective fluid interface region (Fig. 5) can be defined where $|\mathbf{d}| < 2\sqrt{2}\epsilon$ (corresponding

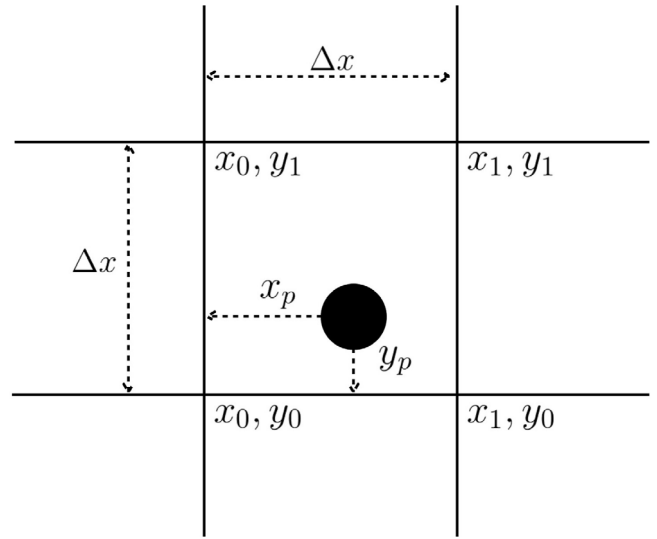


Fig. 6. Schematic of the variables used to project the delta-function forces from each particle to the surrounding Eulerian grid nodes (for the 2D case).

to $|\phi| < 0.964$). Within this region, the phase field variable at \mathbf{x}_p^n can be mapped to \mathbf{d} . The distance from the centre of particle j to the local fluid interface can be calculated as

$$d^{(j)} = \left| \frac{\sqrt{2}}{2} \epsilon \log \frac{1 + \phi^{(j)}}{1 - \phi^{(j)}} \right|. \quad (29)$$

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

One requirement for the mapping to be valid is that the particle radius a must be smaller than the half 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 $\phi^{(j)} \rightarrow \pm 1$).

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

$$\begin{aligned} \phi(x_p, y_p) &= \left(1 - \frac{x_p}{\Delta x}\right) \left(1 - \frac{y_p}{\Delta x}\right) \phi(x_0, y_0) \\ &+ \frac{x_p}{\Delta x} \left(1 - \frac{y_p}{\Delta x}\right) \phi(x_1, y_0) \\ &+ \left(1 - \frac{x_p}{\Delta x}\right) \frac{y_p}{\Delta x} \phi(x_0, y_1) \\ &+ \frac{x_p}{\Delta x} \frac{y_p}{\Delta x} \phi(x_1, y_1), \end{aligned} \quad (30)$$

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

To advance the displacement of the particle from the particle velocity, Heun’s method is adopted. This method is essentially a two-stage Runge–Kutta method for time marching. In the first stage, an estimate of the new particle position is calculated as

$$\bar{\mathbf{x}}_p^{n+1} = \mathbf{x}_p^n + \delta t \mathbf{u}_p^n(t^n, \mathbf{x}_p^n), \quad (31)$$

where δt is the time step. In the second stage, the final position of the particle is calculated with second-order accuracy using the

velocities at \mathbf{x}_p^n and $\bar{\mathbf{x}}_p^{n+1}$:

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \frac{\delta t}{2} (\mathbf{u}_p(t^n, \mathbf{x}_p^n) + \mathbf{u}_p(t^{n+1}, \bar{\mathbf{x}}_p^{n+1})). \quad (32)$$

The fluid momentum equation (13) contains two forcing terms: \mathbf{f}_c can be readily calculated from ϕ obtained in FIPI-Interface according to Eq. (16); The second term includes the hydrodynamic drag \mathbf{F}_h and the capillary force \mathbf{F}_{pi} 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 ΔV , a force $\mathbf{F}^{(j)}$ from the particle j located at $\mathbf{x}_p = (x_p, y_p)$ is extrapolated onto the nearby Eulerian grid points as

$$\begin{aligned} \mathbf{f}(x_0, y_0) &= \mathbf{F}^{(j)} \frac{1}{\Delta V} \frac{1-x_p}{\Delta x} \frac{1-y_p}{\Delta x} \\ \mathbf{f}(x_1, y_0) &= \mathbf{F}^{(j)} \frac{1}{\Delta V} \frac{x_p}{\Delta x} \frac{1-y_p}{\Delta x} \\ \mathbf{f}(x_0, y_1) &= \mathbf{F}^{(j)} \frac{1}{\Delta V} \frac{1-x_p}{\Delta x} \frac{y_p}{\Delta x} \\ \mathbf{f}(x_1, y_1) &= \mathbf{F}^{(j)} \frac{1}{\Delta V} \frac{x_p}{\Delta x} \frac{y_p}{\Delta x} \end{aligned} \quad (33)$$

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

$$\nabla p = \mu \nabla^2 \mathbf{u} + \mathbf{f}_{total} \quad (34)$$

This equation is solved by the Fourier spectral method used in solving the Cahn-Hilliard equation. Applying the divergence operator to both sides of Eq. (34) and taking account of the incompressibility condition Eq. (14) yields the Poisson equation

$$\nabla^2 p = \nabla \cdot \mathbf{f}_{total}. \quad (35)$$

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

$$-k^2 \hat{p} = i\mathbf{k} \cdot \hat{\mathbf{f}}_{total} \quad (36)$$

which leads to

$$\hat{p} = -\frac{i\mathbf{k} \cdot \hat{\mathbf{f}}_{total}}{k^2}. \quad (37)$$

Multiplying both sides of Eq. (37) by $i\mathbf{k}$ gives

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

where the left hand is the gradient of the pressure in Fourier space and the right hand is the component of $\hat{\mathbf{f}}_{total}$ parallel to the wave vector \mathbf{k} . Transforming Eq. (34) into Fourier space and combining it with Eq. (38), the fluid velocity field in Fourier space can be calculated as

$$\hat{\mathbf{u}} = \frac{1}{\mu k^2} \left(\hat{\mathbf{f}}_{total} - \frac{\mathbf{k}(\mathbf{k} \cdot \hat{\mathbf{f}}_{total})}{k^2} \right). \quad (39)$$

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

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 could 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): intermediate executable files and files generated during code compilation;
- data(folder): simulation's output files;
- conf.in(file): input file specifying the simulation parameters;
- makefile(file): makefile used for compiling the code into an executable.

The output files of the program are saved in the folder "data". Here 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 parameters:

nx : number of grid points in x direction

ny : number of grid points in y direction

nz : number of grid points in z direction

$delta$: grid size

dt : time step

t_end : total number of time steps

$gamma$: bare surface tension value

S : dimensionless mobility parameter

Np : 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

$FIPI_Interface$: switch to enable the FIPI-Interface module

$FIPI_Track$: switch to enable the FIPI-Track module

$FIPI_Fluid$: switch to enable the FIPI-Fluid module

FIPI has been developed for LINUX/UNIX environments. 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 program's root folder. The program can be started by executing "./run/fipi".

The initial conditions for the phase field ϕ 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 ϕ is initialised with a uniform probability distribution of values between -0.1 and 0.1 (this distribution models a homogeneously

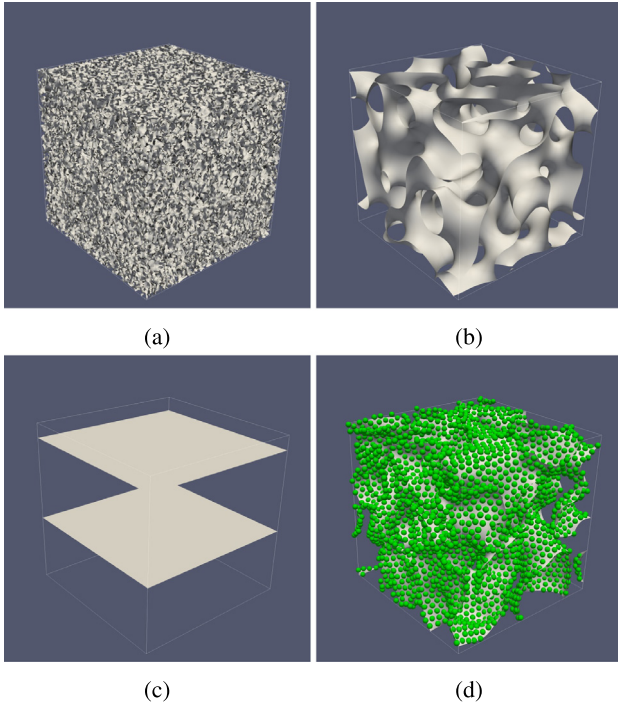


Fig. 7. Iso-surfaces of $\phi = 0$ at (a) $t = 0$ for all cases, and at the end of the simulation for (b) case 1, (c) case 2 and (d) case 3 (with particles shown).

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

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 μ 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 evident. The steady states of the three simulated 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 are completely separated 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 of 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.

The dynamics of the phase separation process can be quantified by monitoring the time evolution of the total free energy of the phase field [42]:

$$F_{tot} = \int_{\Omega} F, \quad (40)$$

where the free energy density F is defined in Eq. (3). The time history of F_{tot} for the three cases simulated is shown in Fig. 8, together with the total energy associated with the surface pressure $\Pi_s A$ produced by the particles.

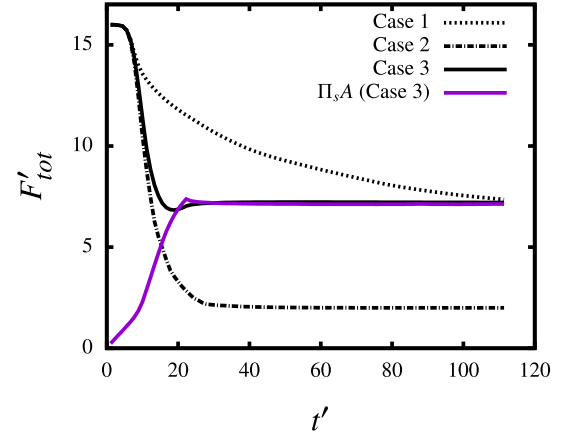


Fig. 8. Normalised total free energy versus normalised time during phase coarsening of a binary mixture laden with particles. The total free energy F_{tot} is normalised as $F'_{tot} = F_{tot}/(\gamma L^2)$. The time is normalised by the capillary time scale $t_c = \frac{L\mu}{\gamma}$ as $t' = t/t_c$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

For all three cases, F_{tot} starts to decrease after a short time from the beginning. In case 1 F_{tot} decreases much more slowly than in cases 2 and 3, because the dynamics in case 1 is only driven by the diffusion term in the 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 large [80], the simulation in case 1 was made not to proceed further.

In case 2 and 3, the module FIPI-Interface is enabled, therefore the phase separation process is further enhanced by the convective flow produced by the surface tension force. The flow due to surface tension 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 cases 2 and 3 than in case 1.

The simulations of cases 2 and 3 reach a steady state, with F_{tot} reaching a plateau 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 Π_s induced by the monolayer increases until $\Pi_s A \approx F_{tot}$ (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 ranging from 10^3 to 10^6 . The scaling of computational time t_{sim} with N is reported in Fig. 9.

For simulations with $N < 10^5$, the order of magnitude of t_{sim} is approximately constant. This indicates that when the number of particles is relatively small, the bulk of the computational cost is due to the solution of the equations in 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

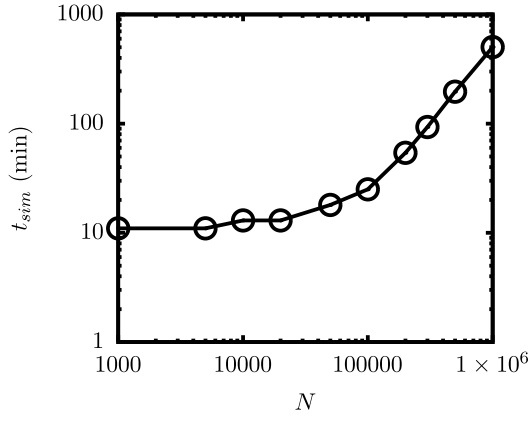


Fig. 9. Simulation time versus number of particles in the computational domain.

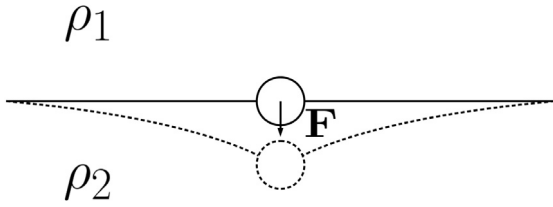


Fig. 10. Schematic of meniscus deformation upon application of an external force \mathbf{F} on a particle initially residing on a flat fluid interface.

constant at 1.38. In the current version of FIPI, the scaling of computational cost with number of particles is thus only slightly super-linear. As far as the 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 multiphase problem considered in the current paper, 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 repulsive particle-particle interaction produces a surface pressure that modify the hydrostatic pressure inside the drop. In the last study, we have reproduced in simulation a pendent drop experiment, to demonstrate that FIPI can capture the correct deformation of a drop covered by a repulsive 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]:

$$h(r) = h_i + h_o - h_{i_o}, \quad (41)$$

where

$$h_i(r) = -a \cos \alpha \cos(\alpha + \theta_c) \times$$

$$\left[\ln\left(\frac{B}{4}\right) + \ln\left(\frac{r}{a} + \sqrt{\left(\frac{r}{a}\right)^2 - \cos^2 \alpha \cos^2(\alpha + \theta_c)}\right) + \gamma_c \right], \quad (42)$$

$$h_o(r) = -a \cos \alpha \cos(\alpha + \theta_c) K_0\left(\frac{r}{l_c}\right), \quad (43)$$

$$h_{i_o}(r) = a \cos \alpha \cos(\alpha + \theta_c) \left(\ln 2 - \ln\left(\frac{r}{l_c}\right) - \gamma_c \right). \quad (44)$$

Here a is the particle radius, $l_c = \sqrt{\gamma/\Delta\rho g}$ is the capillary length, γ_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 embedded in the interface is pulled away by an external force that is perpendicular to the interface. A schematic of the problem is shown in Fig. 10. The profile of the perturbed fluid interface is measured at steady state as a function of r .

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 are 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 ρ_1 and ρ_2 respectively. A constant external force \mathbf{F} is exerted on the particle, pulling it vertically downward. After the start of the simulation, the particle begins to translate away from its equilibrium position and deforms the fluid interface. The time history of the maximum displacement of the fluid interface is shown in Fig. 11.

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]:

$$\gamma \nabla^2 h = -\Delta\rho g h - F\delta(r). \quad (45)$$

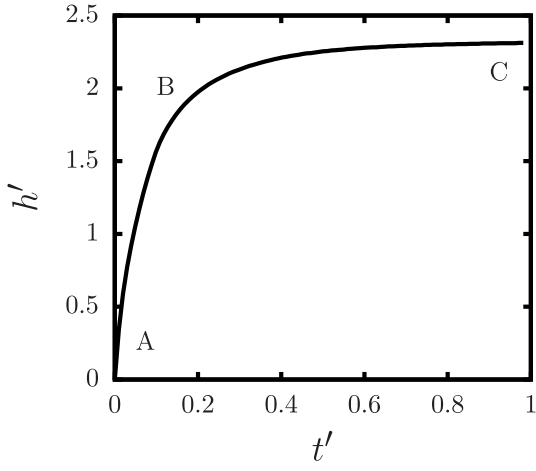
The axisymmetric solution of this equation is

$$h(r) = -\frac{F}{2\pi\gamma} K_0\left(\frac{r}{l_c}\right), \quad (46)$$

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 $h \rightarrow \infty$ as $r \rightarrow 0$. 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 Δx in FIPI. 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 ϕ across the fluid interface does not become too stretched under shearing flows and remains as close to Eq. (4) as possible.



(a)

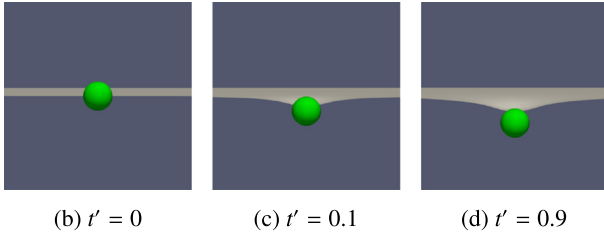


Fig. 11. (a) Time evolution of the maximum downward displacement of the fluid interface caused by exerting a downward force of magnitude F on a single particle. The displacement of the fluid interface is normalised by $h' = \frac{F}{2\pi\gamma}$ and the time by $t' = \frac{\mu}{\Delta\rho gh^*}$. The 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).

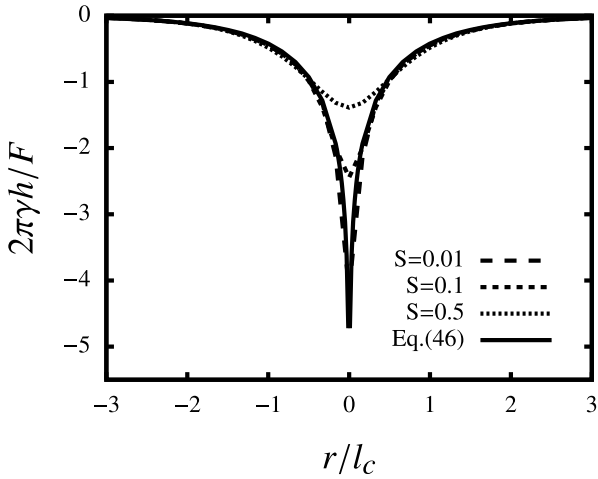


Fig. 12. Fluid interface profiles for different values of the numerical parameter S characterising the importance of the diffusion term appearing in the Cahn-Hilliard equation.

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

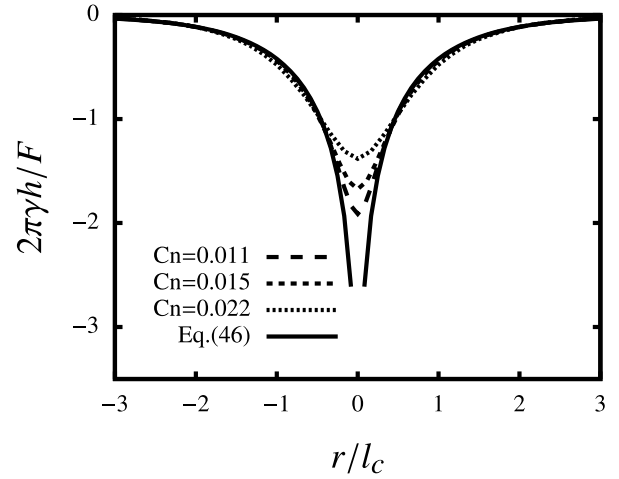


Fig. 13. Fluid interface profiles for different values of the Cahn number Cn .

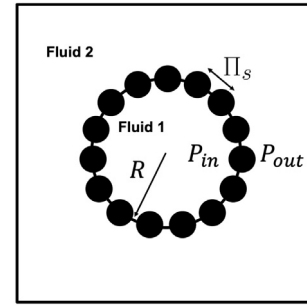


Fig. 14. Simulation set-up for measuring the Laplace pressure $\Delta p = P_{in} - P_{out}$ as a function of the surface pressure Π_s induced by a homogeneous particle monolayer.

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 (Fig. 14). 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 $\Pi_s > 0$ 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 γ by adjusting the value of k_c (in the current study, we set $r_c = 5a$ and k_c varies from 0 to 0.5γ). The corresponding pressure difference Δp 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

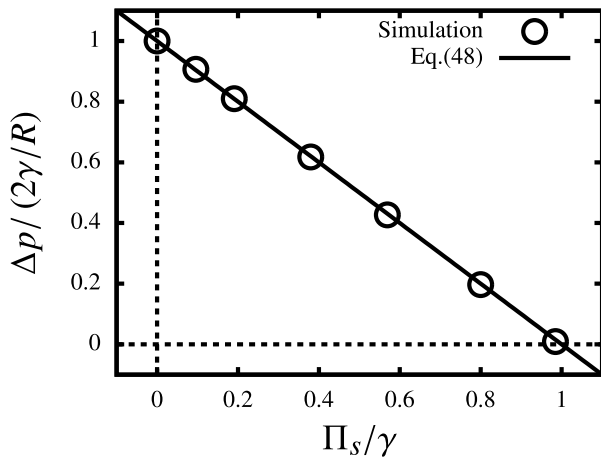


Fig. 15. Laplace pressure of a spherical drop covered by a homogeneous particle monolayer vs. particle-induced surface pressure. The solid line is Eq. (48).

force and the inevitable numerical error due to the approximation of the curvature of the fluid interface. We 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 colour 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_s^{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.

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 drop covered with a monolayer of repulsive particles. The simulation is set up in a rectangular box of size $[2\pi, 2\pi, 3\pi]$, discretised into $64 \times 64 \times 96$ nodes. Periodic boundary conditions are enforced along the three orthogonal directions. At the beginning of the simulations, a spherical drop of 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_c while keeping $r_c = 5a$ in the particle-particle interaction model, 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:

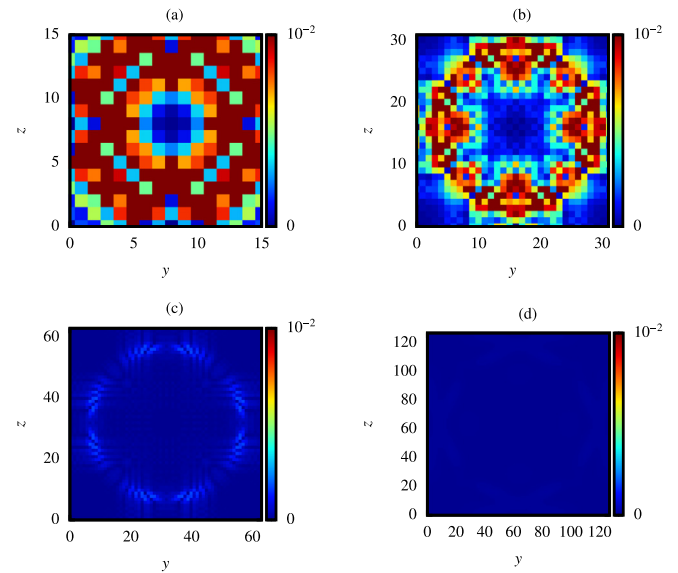


Fig. 16. Colour map of the ratio u_s/u_c of spurious velocity magnitude u_s (evaluated at the fluid interface) and capillary velocity $u_c = \gamma/\mu$ on a plane passing through the drop centre for (a) $n = 16$, (b) $n = 32$, (c) $n = 64$ and (d) $n = 128$. Here the drop is free of particles. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

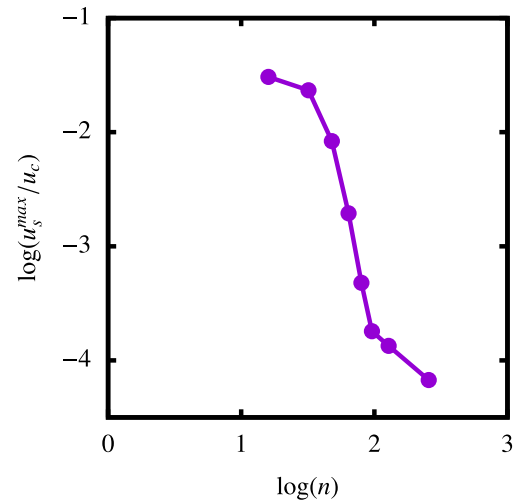


Fig. 17. Convergence of the maximum spurious fluid velocity magnitude on n , corresponding to the data of Fig. 16. The log functions are base 10.

- 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.

To illustrate the effect of the particle monolayer, the shape of the pendant drop at steady state in case 1 and 5 is shown in Fig. 18. It is evident that the drop is more stretched in case 5, due to the reduction of surface tension caused by the repulsive particles covering the drop. The effective surface tension measured by Fordham's and the direct calculation methods are reported in Table 2.

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

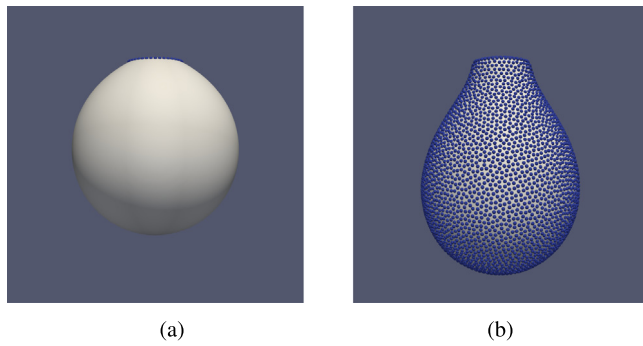


Fig. 18. Steady state shape of a pendant drop for (a) Case 1 (b) Case 5.

Table 2
Comparison of surface tension values from by Fordham's shape-fitting method and the direct calculation approach.

The effective surface tension				
Case	Strength of particle repulsion	The Shape fitting technique	The direct calculation by Eq. (23)	Relative error
1	No particles	0.0696	0.07	0.6%
2	$k_c = 0.1\gamma$	0.0543	0.056	3%
3	$k_c = 0.2\gamma$	0.0424	0.0419	1.2%
4	$k_c = 0.3\gamma$	0.0277	0.028	1.0%
5	$k_c = 0.4\gamma$	0.0175	0.0164	6.7%

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 method and the direct calculation is well within 5%, except in case 5 when the drop is near the pinch-off limit. The good comparison 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 $\mathbf{u}_f(\mathbf{x}_p, t)$ 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 $\mathcal{O}(\frac{F}{\mu\Delta x})$, where F is the magnitude of the total force on the particle minus hydrodynamics and capillary forces. Hence taking $\mathbf{u}_f(\mathbf{x}_p, t) = \mathbf{u}(\mathbf{x}_p, t)$ 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 (Δx 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 Δx .

An approximation of such velocity could be obtained by solving analytically the Stokes problem for a regularised Stokeslet centred near or in the fluid interface, with Δx as regularisation parameter (for the unregularised problem, see Ref. [53]). In the current implementation of FIPI, we assume $\mathbf{u}_f(\mathbf{x}_p, t) = \mathbf{0}$ for quasi-hydrostatic simulations and $\mathbf{u}_f(\mathbf{x}_p, t) \approx \mathbf{u}(\mathbf{x}_p, t)$ when the sum of the external and interparticle forces acting on each particle is much smaller than the capillary force γa .

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: insightful 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 drop 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 drop 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 accounted for in FIPI, 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 θ_c , a quantity that depends only on the surface energies of the liquid–solid, liquid–gas and gas–solid interfaces.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This paper is part of a project that has received funding from the European Union through Marie Curie Career Integration Grant FLOWMAT “Flow and Capillarity in Material Science” (No. 618335) and European Research Council (ERC) grant FLEX-NANOFLOW (No. 715475), under the European Union's Seventh Framework (FP7-2007-2013) and Horizon 2020 programmes.

References

- [1] B.J. Park, J.P. Pantina, E.M. Furst, M. Oettel, S. Reynaert, J. Vermant, *Langmuir* 24 (5) (2008) 1686–1694.
- [2] K. Stratford, R. Adhikari, I. Pagonabarraga, J.-C. Desplat, M.E. Cates, *Science* 309 (5744) (2005) 2198–2201.
- [3] E. Herzig, K. White, A. Schofield, W. Poon, P. Clegg, *Nat. Mater.* 6 (12) (2007) 966.
- [4] E. Dickinson, *Curr. Opin. Colloid Interface Sci.* 15 (1–2) (2010) 40–49.
- [5] B.A. Suleimanov, F. Ismailov, E. Veliyev, *J. Pet. Sci. Eng.* 78 (2) (2011) 431–437.
- [6] A. Dinsmore, M.F. Hsu, M. Nikolaidis, M. Marquez, A. Bausch, D. Weitz, *Science* 298 (5595) (2002) 1006–1009.
- [7] E.J. Stancik, M. Koughan, G.G. Fuller, *Langmuir* 20 (1) (2004) 90–94.
- [8] A.J. Mendoza, E. Guzmán, F. Martínez-Pedrero, H. Ritacco, R.G. Rubio, F. Ortega, V.M. Starov, R. Miller, *Adv. Colloid Interface Sci.* 206 (2014) 303–319.
- [9] F. Fenouillot, P. Cassagnau, J.-C. Majesté, *Polymer* 50 (6) (2009) 1333–1350.
- [10] A.B. Pawar, M. Caggioni, R. Ergun, R.W. Hartel, P.T. Spicer, *Soft Matter* 7 (17) (2011) 7710–7716.
- [11] O. Pitois, X. Chateau, *Langmuir* 18 (25) (2002) 9751–9756.
- [12] D.-G. Lee, H.-Y. Kim, *Phys. Fluids* 23 (7) (2011) 072104.
- [13] P. Singh, D. Joseph, *J. Fluid Mech.* 530 (2005) 31–80.
- [14] A.J. Ladd, *J. Fluid Mech.* 271 (1994) 285–309.
- [15] X. Shan, H. Chen, *Phys. Rev. E* 47 (3) (1993) 1815.
- [16] M.R. Swift, E. Orlandini, W. Osborn, J. Yeomans, *Phys. Rev. E* 54 (5) (1996) 5041.
- [17] A.S. Joshi, Y. Sun, *Phys. Rev. E* 82 (4) (2010) 041401.
- [18] F. Jansen, J. Harting, *Phys. Rev. E* 83 (4) (2011) 046707.
- [19] S. Aland, J. Lowengrub, A. Voigt, *Phys. Fluids* 23 (6) (2011) 062103.
- [20] N. Jaensson, M. Hulsen, P. Anderson, *Comput. & Fluids* 111 (2015) 1–17.
- [21] P.C. Millett, Y.U. Wang, *J. Colloid Interface Sci.* 353 (1) (2011) 46–51.
- [22] L. Botto, A. Prosperetti, *Phys. Fluids* 24 (1) (2012) 013303.
- [23] D. Vella, *Annu. Rev. Fluid Mech.* 47 (2015) 115–135.
- [24] N. Sinn, M. Alishahi, S. Hardt, *J. Colloid Interface Sci.* 458 (2015) 62–68.
- [25] S. O'Brien, *J. Colloid Interface Sci.* 183 (1) (1996) 51–56.
- [26] D.M. Kaz, R. McGorty, M. Mani, M.P. Brenner, V.N. Manoharan, *Nat. Mater.* 11 (2) (2012) 138–142.
- [27] Y. Sui, H. Ding, P.D. Speltz, *Annu. Rev. Fluid Mech.* 46 (2014).
- [28] L. Botto, E.P. Lewandowski, M. Cavallaro, K.J. Stebe, *Soft Matter* 8 (39) (2012) 9957–9971.
- [29] J. Onishi, A. Kawasaki, Y. Chen, H. Ohashi, *Comput. Math. Appl.* 55 (7) (2008) 1541–1553.
- [30] G. Lecrivain, R. Yamamoto, U. Hampel, T. Taniguchi, *Phys. Fluids* 28 (8) (2016) 083301.
- [31] X. Chateau, O. Pitois, *J. Colloid Interface Sci.* 259 (2) (2003) 346–353.
- [32] R. Ettelaie, S.V. Lishchuk, *Soft Matter* 11 (21) (2015) 4251–4265.
- [33] S.E. Anachkov, I. Lesov, M. Zanini, P.A. Kralchevsky, N.D. Denkov, L. Isa, *Soft Matter* 12 (36) (2016) 7632–7643.
- [34] S. Razavi, K.D. Cao, B. Lin, K.Y.C. Lee, R.S. Tu, I. Kretschmar, *Langmuir* 31 (28) (2015) 7764–7775.
- [35] O. Pitois, M. Buisson, X. Chateau, *Eur. Phys. J. E* 38 (5) (2015) 48.
- [36] D. Vella, P. Aussillous, L. Mahadevan, *Europhys. Lett.* 68 (2) (2004) 212.
- [37] K.D. Danov, R.D. Stanimirova, P.A. Kralchevsky, K.G. Marinova, N.A. Alexandrov, S.D. Stoyanov, T.B. Blijdenstein, E.G. Pelan, *J. Colloid Interface Sci.* 440 (2015) 168–178.
- [38] V. Garbin, I. Jenkins, T. Sinno, J.C. Crocker, K.J. Stebe, *Phys. Rev. Lett.* 114 (10) (2015) 108301.
- [39] S. Elghobashi, G. Truesdell, *J. Fluid Mech.* 242 (1992) 655–700.
- [40] D. Jacqmin, *J. Comput. Phys.* 155 (1) (1999) 96–127.
- [41] P. Yue, J.J. Feng, C. Liu, J. Shen, *J. Fluid Mech.* 515 (2004) 293–317.
- [42] P. Yue, C. Zhou, J.J. Feng, *J. Comput. Phys.* 223 (1) (2007) 1–9.
- [43] P. Yue, C. Zhou, J.J. Feng, C.F. Ollivier-Gooch, H.H. Hu, *J. Comput. Phys.* 219 (1) (2006) 47–67.
- [44] L.A. Lubbers, J.H. Weijs, L. Botto, S. Das, B. Andreotti, J.H. Snoeijer, *J. Fluid Mech.* 747 (2014).
- [45] H. Mehrabian, J. Harting, J.H. Snoeijer, *Soft Matter* 12 (4) (2016) 1062–1073.
- [46] J. Bławdziewicz, V. Cristini, M. Loewenberg, *Phys. Fluids* 11 (2) (1999) 251–258.
- [47] G. Wang, R. Prabhakar, E.M. Sevick, *Phys. Rev. Lett.* 103 (24) (2009) 248303.
- [48] A. Geller, S. Lee, L. Leal, *J. Fluid Mech.* 169 (1986) 27–69.
- [49] T. Bickel, *Phys. Rev. E* 75 (4) (2007) 041403.
- [50] S. Lee, R. Chadwick, L.G. Leal, *J. Fluid Mech.* 93 (4) (1979) 705–726.
- [51] S. Lee, L. Leal, *J. Fluid Mech.* 98 (1) (1980) 193–224.
- [52] J.T. Petkov, N.D. Denkov, K.D. Danov, O.D. Velev, R. Aust, F. Durst, *J. Colloid Interface Sci.* 172 (1) (1995) 147–154.
- [53] T.M. Fischer, P. Dhar, P. Heinig, *J. Fluid Mech.* 558 (2006) 451–475.
- [54] K.D. Danov, R. Dimova, B. Pouligny, *Phys. Fluids* 12 (11) (2000) 2711–2722.
- [55] C. Pozrikidis, *J. Fluid Mech.* 575 (2007) 333–357.
- [56] H. Brenner, L.G. Leal, *J. Colloid Interface Sci.* 65 (2) (1978) 191–209.
- [57] J.F. Brady, *J. Chem. Phys.* 99 (1) (1993) 567–581.
- [58] J.J. Stickel, R.L. Powell, *Annu. Rev. Fluid Mech.* 37 (2005) 129–149.
- [59] A. Vidal, L. Botto, *J. Fluid Mech.* 813 (2017) 152–174.
- [60] J. Joanny, P.-G. De Gennes, *J. Chem. Phys.* 81 (1) (1984) 552–562.
- [61] R. Aveyard, B.P. Binks, J.H. Clint, *Adv. Colloid Interface Sci.* 100 (2003) 503–546.
- [62] J.N. Israelachvili, *Intermolecular and Surface Forces*, Academic Press, 2011.
- [63] Y. Liang, N. Hilal, P. Langston, V. Starov, *Adv. Colloid Interface Sci.* 134 (2007) 151–166.
- [64] Y. Tsuji, T. Kawaguchi, T. Tanaka, *Powder Technol.* 77 (1) (1993) 79–87.
- [65] T. Mikami, H. Kamiya, M. Horio, *Chem. Eng. Sci.* 53 (10) (1998) 1927–1940.
- [66] A. Prosperetti, G. Tryggvason, *Computational Methods for Multiphase Flow*, Cambridge University Press, 2009.
- [67] A. Ferrante, S. Elghobashi, *Phys. Fluids* 15 (2) (2003) 315–329.
- [68] M. Ekiel-Jezewska, B. Metzger, E. Guazzelli, *Phys. Fluids* 18 (3) (2006) 038104.
- [69] C.E. Weatherburn, *Quart. Pure Appl. Math.* 50 (1925) 230–269.
- [70] C. Gu, L. Botto, *Soft Matter* 12 (3) (2016) 705–716.
- [71] P. Kralchevsky, J. Eriksson, S. Ljunggren, *Adv. Colloid Interface Sci.* 48 (1994) 19–59.
- [72] P.-G. De Gennes, F. Brochard-Wyart, D. Quéré, *Capillarity and Wetting Phenomena: Drops, Bubbles, Pearls, Waves*, Springer Science & Business Media, 2013.
- [73] C. Canuto, A. Quarteroni, M.Y. Hussaini, T.A. Zang, *Fundamentals of Fluid Dynamics*, Springer, 2007.
- [74] M. Frigo, S.G. Johnson, *Acoustics, Speech and Signal Processing*, 1998. Proceedings of the 1998 IEEE International Conference on, Vol. 3, IEEE, 1998, pp. 1381–1384.
- [75] M.P. Allen, D.J. Tildesley, *Computer Simulation of Liquids*, Oxford University Press, 2017.
- [76] K.D. Squires, J.K. Eaton, *Phys. Fluids A* 2 (7) (1990) 1191–1203.
- [77] E. Calzavarini, R. Volk, M. Bourgoin, E. Lévéque, J.-F. Pinton, F. Toschi, *J. Fluid Mech.* 630 (2009) 179–189.
- [78] D.W. Rouson, J.K. Eaton, *J. Fluid Mech.* 428 (2001) 149–169.
- [79] S. Sundaram, L.R. Collins, *J. Comput. Phys.* 124 (2) (1996) 337–350.
- [80] J. Zhu, L.-Q. Chen, J. Shen, V. Tikare, *Phys. Rev. E* 60 (4) (1999) 3564.
- [81] P.A. Kralchevsky, K. Nagayama, *Adv. Colloid Interface Sci.* 85 (2–3) (2000) 145–192.
- [82] Y. Rotenberg, L. Boruvka, A.W. Neumann, *J. Colloid Interface Sci.* 93 (1) (1983) 169–183.
- [83] F. Hansen, G. Rødsrud, *J. Colloid Interface Sci.* 141 (1) (1991) 1–9.
- [84] C.E. Stauffer, *J. Phys. Chem.* 69 (6) (1965) 1933–1938.
- [85] S. Fordham, *Proc. R. Soc. A* 194 (1948) 1–16.
- [86] J. Horwitz, A. Mani, *J. Comput. Phys.* 318 (2016) 85–109.
- [87] P. Saffman, *Stud. Appl. Math.* 52 (2) (1973) 115–127.
- [88] H. Ding, P.D. Speltz, C. Shu, *J. Comput. Phys.* 226 (2) (2007) 2078–2095.