A parallel two-phase flow solver on unstructured mesh in 3D

Li Luo[1,3], Qian Zhang[1], Xiao-Ping Wang[1] and Xiao-Chuan Cai[2]

The simulation of two-phase flow is important in many scientific and engineering processes, for instance, wetting, coating, painting, etc. There are many publications on phase field modelling of two-phase flows. Gao and Wang [Gao and Wang, 2014] proposed a gradient stable semi-implicit finite difference scheme in 2D and 3D by using the convex splitting method for the Cahn-Hilliard equation and a projection method for the Navier-Stokes equations. Bao et al. [Bao et al., 2012] presented a finite element method for phase field problems on 2D domains with rough boundary using unstructured meshes. The free interface problem is computationally very expensive especially in 3D; some parallelization strategies were adopted to accelerate the computation of certain two-phase flows. Shin et al. [Shin et al., 2014] presented a parallel implementation of the Level Contour Reconstruction Method (LCRM) on structured meshes for simulating the splash of a drop onto a film of liquid, in which a weak scaling efficiency of 48% on 32768 processors was reported.

In this paper, we present a new parallel finite element solver on unstructured 3D meshes and its implementation on a massively parallel computer. In order to construct a stable and efficient solver for the case of large density and viscosity ratio, we combine the stabilized schemes for the Cahn-Hilliard equation and projection-type schemes for the Navier-Stokes equations to fully decouple the phase function, the velocity, and the pressure. The resulting decoupled systems are discretized by a piecewise linear finite element method in space and solved by a Krylov subspace method. Specifically, systems arising from implicit discretization of the Cahn-Hilliard equation and the velocity equation are solved by a restricted additive Schwarz preconditioned GMRES method, and the pressure Poisson system is solved by an algebraic multigrid preconditioned CG method. We show numerically that the proposed strategy works well for 3D problems with complex geometry and is highly scalable in

[1]Department of Mathematics, The Hong Kong University of Science and Technology, Hong Kong lluoac@ust.hk · [2]Department of Computer Science, University of Colorado Boulder, Boulder, USA · [3]Shenzhen Institutes of Advanced Technology, Shenzhen, China

terms of the number of iterations and the total computing time on a supercomputer with nearly 10,000 processors.

The paper is organized as follows. In Section 1, a phase field model is described. The fully decoupled scheme with a finite element discretization is also presented in this section. The domain decomposition techniques and scalable solvers are discussed in Section 2. In Section 3, we show two numerical experiments. Performance results of the parallel implementation are also reported. The paper is concluded in Section 4.

1 Mathematical models and discretization schemes

Let Ω be a bounded domain in $\mathbb{R}^3$. The system of interest can be described by a coupled Cahn-Hilliard-Navier-Stokes equations, as follows:

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = \mathcal{L}_d \Delta \mu, \qquad \text{in} \quad \Omega, \quad (1)$$

$$\mu = -\epsilon \Delta \phi - \frac{\phi}{\epsilon} + \frac{\phi^3}{\epsilon}, \qquad \text{in} \quad \Omega, \quad (2)$$

$$Re\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} \right) = -\nabla p + \nabla \cdot (\eta D(\mathbf{u})) + \mathcal{B}\mu \nabla \phi, \qquad \text{in} \quad \Omega, \quad (3)$$

$$\nabla \cdot \mathbf{u} = 0, \qquad \text{in} \quad \Omega. \quad (4)$$

Here, a phase-field variable ϕ is introduced to describe the transition between the two homogeneous equilibrium phases $\phi_\pm = \pm 1$. μ is the chemical potential, ϵ is the ratio between interface thickness and characteristic length, and $\mu \nabla \phi$ is the capillary force. The mass density ρ and the dynamic viscosity η are interpolation functions of ϕ between fluid 1 and fluid 2, $\rho = \frac{1+\phi}{2} + \lambda_\rho \frac{1-\phi}{2}$, $\eta = \frac{1+\phi}{2} + \lambda_\eta \frac{1-\phi}{2}$, where $\lambda_\rho = \rho_2/\rho_1$ is the ratio of density between the two fluids and $\lambda_\eta = \eta_2/\eta_1$ is the ratio of viscosity. $\mathbf{u} = (u_x, u_y, u_z)$ where u_x, u_y, u_z are the velocity components along x, y, z directions, $D(\mathbf{u}) = \nabla \mathbf{u} + (\nabla \mathbf{u})^T$ is the rate of stress tensor, p is the pressure, $\mathcal{L}_d$ is the phenomenological mobility coefficient, Re is the Reynolds number and $\mathcal{B}$ measures the strength of the capillary force comparing to the Newtonian fluid stress (and $\mathcal{B}$ is inversely proportional to the capillary number). The motion of the contact line at solid boundaries Γ_w can be described by a relaxation boundary condition for the phase function and the generalized Navier boundary condition (GNBC) for velocity:

$$\frac{\partial \phi}{\partial t} + u_{\tau_1} \partial_{\tau_1} \phi + u_{\tau_2} \partial_{\tau_2} \phi = -\mathcal{V}_s L(\phi), \qquad \text{on} \quad \Gamma_w, \quad (5)$$

$$(\mathcal{L}_s l_s)^{-1} u_{\tau_1} = \mathcal{B} L(\phi) \partial_{\tau_1} \phi / \eta - \mathbf{n} \cdot D(\mathbf{u}) \cdot \boldsymbol{\tau}_1, \qquad \text{on} \quad \Gamma_w, \quad (6)$$

$$(\mathcal{L}_s l_s)^{-1} u_{\tau_2} = \mathcal{B} L(\phi) \partial_{\tau_2} \phi / \eta - \mathbf{n} \cdot D(\mathbf{u}) \cdot \boldsymbol{\tau}_2, \qquad \text{on} \quad \Gamma_w, \quad (7)$$

where $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ are two unit tangent directions that are orthogonal to each other along the solid surface, $\boldsymbol{\tau}_1 \cdot \boldsymbol{\tau}_2 = 0$. $\mathbf{n}$ is the unit outward normal direction of the solid surface. $\mathcal{V}_s$ is a phenomenological parameter. $L(\phi) = \epsilon\partial_n\phi + Q(\phi)$, $Q(\phi) = \partial\gamma_{wf}(\phi)/\partial\phi$ and $\gamma_{wf}(\phi) = -\frac{\sqrt{2}}{3}\cos\theta_s \sin(\frac{\pi}{2}\phi)$, θ_s is the static contact angle. $u_{\tau_1} = \mathbf{u}\cdot\boldsymbol{\tau}_1$ and $u_{\tau_2} = \mathbf{u}\cdot\boldsymbol{\tau}_2$. $\mathcal{L}_s$ is the slip length of liquid, $l_s = \frac{1+\phi}{2} + \lambda_{l_s}\frac{1-\phi}{2}$ is an interpolation between two different wall-fluid slip length, and $\lambda_{l_s} = l_{s2}/l_{s1}$ the ratio of slip length. In addition, the following impermeability conditions $u_n := \mathbf{u}\cdot\mathbf{n} = 0$, and $\partial_n\mu = 0$ are also imposed on the solid boundaries.

We present a semi-implicit finite element method for solving the above coupled systems on unstructured meshes in 3D. We apply a convex splitting of the free energy functional and treat the nonlinear term explicitly so that the resulting matrix does not change in time, and therefore can be pre-computed. In addition, we consider a pressure stabilized formulation [Guermond and Salgado, 2009] to decouple the Navier-Stokes equations into a convection-diffusion equation for velocity and a Poisson equation for pressure. Then, both of them can be easily approximated by the piecewise linear finite element methods.

Let Ω_h be a conforming mesh of Ω, and Γ_w^h is the solid boundary of Ω_h. In this paper, we only consider tetrahedral elements and P_1 functions. We define the following finite element spaces

$$
\begin{aligned}
W_h &= \left\{ w_h \in H^1(\Omega);\ w_h|_E \in P_1(E), \forall E \in \Omega_h \right\},\\
\mathbf{U}_h &= \left\{ \mathbf{u}_h \in \left[H^1(\Omega)\right]^3;\ \mathbf{u}_h\cdot\mathbf{n} = 0 \text{ on } \Gamma_w^h;\ \mathbf{u}_h|_E \in P_1(E)^3, \forall E \in \Omega_h \right\},\\
M_h &= \left\{ q_h \in W_h;\ \partial_n q_h = 0 \text{ on } \Gamma_w^h \right\}.
\end{aligned}
$$

We denote by $(\cdot,\cdot)$ the $L^2(\Omega_h)$-inner product and by $\langle\cdot,\cdot\rangle_{\Gamma_w^h}$ the $L^2(\Gamma_w^h)$-inner product. Next, we introduce a time step $\delta t > 0$. The first-order temporal discretization in the weak form can be described in the following four steps:

Step 1: Solve the Cahn-Hilliard equation using a convex-splitting method: find $(\phi_h^{n+1}, \mu_h^{n+1}) \in W_h \times W_h$, such that for $\forall\ w_h \in W_h$,

$$
\left(\frac{\phi_h^{n+1}-\phi_h^n}{\delta t}, w_h\right) + (\mathbf{u}_h^n\cdot\nabla\phi_h^n, w_h) = -\mathcal{L}_d(\nabla\mu_h^{n+1}, \nabla w_h), \tag{8}
$$

$$
\begin{aligned}
(\mu_h^{n+1}, w_h) = {}& \epsilon(\nabla\phi_h^{n+1}, \nabla w_h) + \frac{s}{\epsilon}(\phi_h^{n+1}, w_h) + \frac{1}{\epsilon}\left((\phi_h^n)^3 - (1+s)\phi_h^n, w_h\right)\\
&+ \left\langle \left(\frac{1}{\mathcal{V}_s}\left(\frac{\phi_h^{n+1}-\phi_h^n}{\delta t} + u_{\tau_1,h}^n\partial_{\tau_1}\phi_h^n + u_{\tau_2,h}^n\partial_{\tau_2}\phi_h^n \right) + Q\left(\phi_h^n\right) \right), w_h \right\rangle_{\Gamma_w}.
\end{aligned} \tag{9}
$$

Step 2: Update ρ_h^{n+1}, η_h^{n+1} and ${l_s}_h^{n+1} \in W_h$:

$$
(\rho_h^{n+1}, \eta_h^{n+1}, {l_s}_h^{n+1}) = \frac{1+\phi_h^{n+1}}{2} + (\lambda_\rho, \lambda_\eta, \lambda_{l_s})\frac{1-\phi_h^{n+1}}{2}. \tag{10}
$$

Step 3: Solve the velocity system of Navier-Stokes equations using a pressure stabilization scheme: find $\mathbf{u}_h^{n+1} \in \mathbf{U}_h$, such that for $\forall\ \mathbf{v}_h \in \mathbf{U}_h$,

$$
\begin{aligned}
&Re\left(\left(\frac{\frac{1}{2}(\rho_h^{n+1}+\rho_h^n)\mathbf{u}_h^{n+1}-\rho_h^n\mathbf{u}_h^n}{\delta t}+\rho_h^{n+1}(\mathbf{u}_h^n\cdot\nabla)\mathbf{u}_h^{n+1}+\frac{1}{2}\left(\nabla\cdot(\rho_h^{n+1}\mathbf{u}_h^n)\right)\mathbf{u}_h^{n+1}\right),\mathbf{v}_h\right)\\
&=-\left(\eta_h^{n+1}\left(\nabla\mathbf{u}_h^{n+1}+(\nabla\mathbf{u}_h^{n+1})^T\right),\nabla\mathbf{v}_h\right)+\mathcal{B}(\mu_h^{n+1}\nabla\phi_h^{n+1},\mathbf{v}_h)-(2\nabla p_h^n-\nabla p_h^{n-1},\mathbf{v}_h)\\
&\quad-\left\langle\eta_h^{n+1}\left(\mathcal{L}_s l_s{}_h^{n+1}\right)^{-1}u_{\tau_1,h}^{n+1},v_{\tau_1,h}\right\rangle_{\Gamma_w}-\left\langle\eta_h^{n+1}\left(\mathcal{L}_s l_s{}_h^{n+1}\right)^{-1}u_{\tau_2,h}^{n+1},v_{\tau_2,h}\right\rangle_{\Gamma_w}\\
&\quad+\mathcal{B}\left\langle\left(\epsilon\partial_n\phi_h^{n+1}+Q\left(\phi_h^{n+1}\right)\right)\partial_{\tau_1}\phi_h^{n+1},v_{\tau_1,h}\right\rangle_{\Gamma_w}\\
&\quad+\mathcal{B}\left\langle\left(\epsilon\partial_n\phi_h^{n+1}+Q\left(\phi_h^{n+1}\right)\right)\partial_{\tau_2}\phi_h^{n+1},v_{\tau_2,h}\right\rangle_{\Gamma_w}.
\end{aligned}
\tag{11}
$$

Step 4: Solve the pressure system of Navier-Stokes equations: find $p_h^{n+1} \in M_h$, such that for $\forall\ q_h \in M_h$,

$$
\left(\nabla(p_h^{n+1}-p_h^n),\nabla q_h\right)=-\frac{\bar{\rho}}{\delta t}Re(\nabla\cdot\mathbf{u}_h^{n+1},q_h). \tag{12}
$$

In the above scheme, s is a stabilization parameter. $v_{n,h}=\mathbf{v}_h\cdot\mathbf{n}$, $v_{\tau_1,h}=\mathbf{v}_h\cdot\boldsymbol{\tau}_1$, $v_{\tau_2,h}=\mathbf{v}_h\cdot\boldsymbol{\tau}_2$, and $\bar{\rho}=\min(1,\lambda_\rho)$.

Remark 1. The time discretization scheme constructed above leads to a decoupled system for the phase function, the velocity, and the pressure. At each time step, we solve a convection-diffusion equation for $\mathbf{u}$, a system of convection-diffusion/elliptic equations for (ϕ,μ), and a Poisson equation for p. The matrices from the last two equations do not change in time, and can then be pre-computed for computational efficiency.

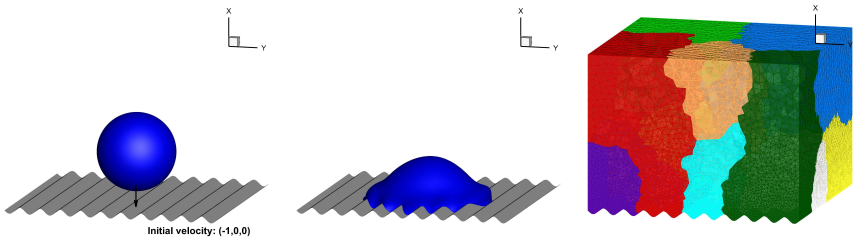
2 Scalable solvers based on domain decomposition and algebraic multigrid techniques

In the scheme formulated in the previous section, there are three linear systems of equations to be solved at each time step. For the nonsymmetric problems in Step 1 and Step 3, we employ a restricted additive Schwarz preconditioned GMRES method to solve the linear systems of phase function and velocity. The choice of subdomain solver is critical to the Schwarz preconditioner. One of the popular choices is the incomplete LU (ILU) factorization. A large number of fill-ins levels helps in reducing iterations, but leads to an expensive solver in terms of the compute time and the memory usage. The impact of these factors will be discussed in numerical experiments. To solve the symmetric positive definite problem in Step 4, we employ an algebraic multigrid (AMG) preconditioned CG method. A scalable AMG solver BoomerAMG [Henson and Yang, 2002] is used as a preconditioner to effectively solve the pressure Poisson equation.

3 Numerical experiments

In this section, we present some numerical experiments and analyze the parallel performance of the proposed algorithm. The algorithm is implemented using a finite element package libMesh [Kirk et al., 2006] for generating the stiffness matrices, and a parallel scientific computing library PETSc [Balay et al., 2016] for the preconditioned Krylov subspace solvers. The computational mesh is generated using Gmsh [Geuzaine and Remacle, 2009] and partitioned using MeTiS [Karypis and Kumar, 1995]. Two numerical experiments will be presented including a droplet spreading over a rough surface and a two-phase flow in a bumpy channel.

We first consider a droplet spreading over a rough solid surface with parallel stripped texture. Along the y-axis the bottom surface is parametrized by a wave function $x = 0.025\sin(40y)$ with $y \in [-0.025\pi, 0.5\pi]$, and along the z-axis the function is translated from $z = 0$ to $z = 0.5\pi$. The height of the domain is 1.2. A spherical drop is initially located at $(0.35, 0.2375\pi, 0.25\pi)$ with radius 0.3. The initial speed is $(-1, 0, 0)$. A nonuniform mesh is generated such that near the bottom boundary the mesh is finer. The mesh has 3,055,992 elements and 535,509 vertices. The average mesh size near the bottom surface is $h = 5.64 \times 10^{-2}$ and the time step size is $\delta t = 2 \times 10^{-4}$. Other parameters used are as follows: $\lambda_\rho = 0.001$, $\lambda_\eta = 0.1$, $\lambda_{l_s} = 1$, $Re = 1000$, $\theta_s = 50°$, $\epsilon = 0.02$, $\mathcal{B} = 12$, $\mathcal{L}_d = 5 \times 10^{-4}$, $\mathcal{V}_s = 500$, $\mathcal{L}_s = 0.038$, and $s = 1.5$. The initial condition and the droplet spreading at $t = 0.4$ as well as a sample partition are shown in Fig 1.

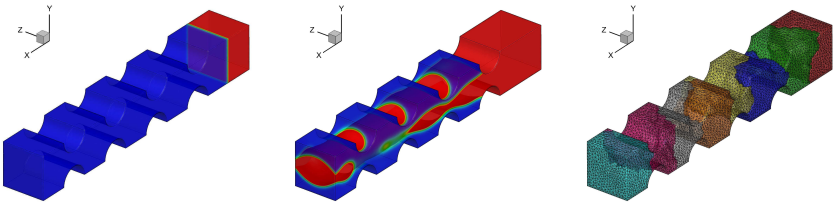


(a) (b) (c)

Fig. 1 (a) Initial condition, (b) the evolution of interface at $t = 0.4$, and (c) a sample partition into 16 subdomains for the droplet spreading case.

We next consider a flow of two immiscible fluids (red represents fluid 1 and blue represents fluid 2) in a bumpy channel is driven by the pressure gradient between the inflow boundary ($x = -0.5$, $p = 4000$) and the outflow boundary ($x = 0.5$, $p = 0$). The other boundaries are solid surfaces. The computational domain is $[-0.5, 0.5] \times [-0.075, 0.075] \times [-0.075, 0.075]$, and the radius of the cylinder bumps is 0.05. The mesh has 588,696 elements and 113,457 vertices.

The average mesh size is $h = 9.15 \times 10^{-3}$ and the time step size is $\delta t = 10^{-4}$. Other parameters are as follows: $\lambda_\rho = 0.1$, $\lambda_\eta = 0.1$, $\lambda_{l_s} = 10$, $Re = 100$, $\theta_s = 120°$, $\epsilon = 0.005$, $\mathcal{B} = 12$, $\mathcal{L}_d = 5 \times 10^{-4}$, $\mathcal{V}_s = 200$, $\mathcal{L}_s = 0.0025$, and $s = 1.5$. The initial condition and the evolution of interface at $t = 0.28$ as well as a sample partition are shown in Fig 2.



(a) (b) (c)

Fig. 2 (a) Initial condition, (b) the evolution of interface at $t = 0.28$, and (c) a sample partition into 8 subdomains for the bumpy channel flow case.

3.1 Parallel performance

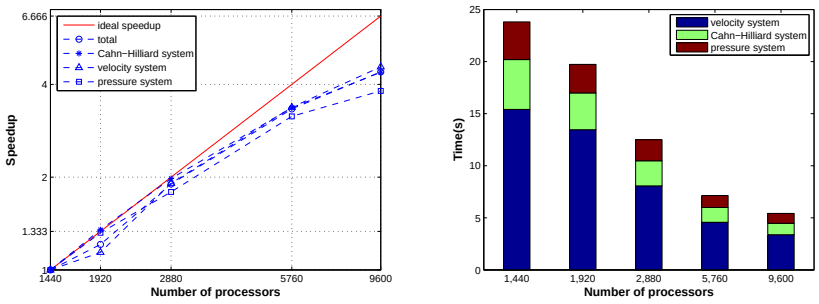
In this subsection, we focus on the bumpy channel flow case and report the parallel performance of the proposed solution algorithm. The scalability tests are performed on the Tianhe 2 supercomputer which ranks # 2 on the latest Top 500 list. Each node of Tianhe 2 has 24 processors and 64 GB memory. For the rest of the section, "*np*" denotes the number of processors, "GMRES" and "CG" denote the average number of GMRES and CG iterations per time step, respectively. "sp." represents the speedup. All timings are reported in seconds. The restart value of GMRES is fixed at 50. 10^{-6} is used as the relative stopping condition for linear solvers.

The unstructured mesh has 301,412,352 elements and 51,270,353 vertices. We focus on how different levels of ILU fill-ins in the subdomain solver of Schwarz preconditioner affect the parallel efficiency. The overlapping size is fixed to 1. The number of processors increases from np = 1,920 to 5,760 to 9,600. The results for different levels of ILU fill-ins at different np are summarized in the first 8 columns in Table 1. The results show that at least 2 levels of ILU fill-ins are needed for the Cahn-Hilliard system. Increasing the level of fill-ins helps reducing the number of GMRES iterations, this effect is more obvious for the Cahn-Hilliard system. However, higher level of fill-ins may cost more computation time. The table also suggests that ILU(3) is the best choice for the Cahn-Hilliard system and ILU(1) is the best choice for the velocity system. We have also considered the effect of varying the

Table 1 The average number of iterations, compute time per time step, and speed up for solving Cahn-Hilliard system, the velocity system, and the pressure system."-" means the case fails to converge.

		Cahn-Hilliard system #unknowns=102,540,706			velocity system #unknowns=153,811,059			pressure system #unknowns=51,270,353			
np	subsolve	GMRES	time	sp.	GMRES	time	sp.	sweep	CG	time	sp.
1,920	ILU(1)	441.4	21.36	1	35	13.72	1	1	24.1	2.74	1
1,920	ILU(2)	39.9	4.36	1	26.7	17.18	1	2	20.2	3.31	1
1,920	ILU(3)	12.7	3.60	1	17.2	25.61	1	3	19.8	3.92	1
5,760	ILU(1)	-	-	-	30	4.57	3.00	1	24.1	1.15	2.38
5,760	ILU(2)	42.2	1.80	2.42	13.1	6.06	2.83	2	20.7	1.42	1.63
5,760	ILU(3)	13.4	1.43	2.52	7	9.38	2.73	3	19.7	1.66	2.36
9,600	ILU(1)	-	-	-	29.8	3.38	4.06	1	24.8	0.95	2.88
9,600	ILU(2)	40.6	1.29	3.38	14.3	4.27	4.02	2	21	1.13	2.92
9,600	ILU(3)	13.7	1.09	3.30	9.8	6.63	3.86	3	19.9	1.34	2.93

number of sweeps of the smoother in the AMG preconditioner for solving the pressure system. The last 4 columns in Table 1 shows that the number of CG iterations seems to be independent of *np* for all cases. However, increasing the number of sweeps does not improve the convergence of the linear solver much but requires more computational time, therefore one sweep of smoother is preferable for the multigrid method. Combining the above choices, we present the speedups and computational time for each system (marked as "total" including Step 1, 3, and 4 of the algorithm) starting from 1,440 processors in Fig 3. Excellent speedup is achieved when *np* is up to 2,880 and the final speedup is 4.39 out of 6.67 on a fixed-size system which is reasonably good.



(a) (b)

Fig. 3 Speedup (a) and distribution of total compute time (b) for the two-phase flow in a bumpy channel.

4 Conclusions

In this paper we introduce a parallel finite element method on 3D unstructured meshes for the two-phase flow problem modelled by a phase-field model consisting of the coupled Cahn-Hilliard and Navier-Stokes equations. A restricted additive Schwarz preconditioned GMRES method is used to solve the systems arising from implicit discretization of the Cahn-Hilliard equation and the velocity equation, and an algebraic multigrid preconditioned CG method is used to solve the pressure Poisson system. Numerical experiments suggest that the overall algorithm scales well on unstructured meshes for problems with up to 150 millions unknowns and on machines with close to 10,000 processors.

References

S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. R., B. F. Smith, S. Zampini, and H. Zhang. PETSc Users Manual. Technical Report ANL-95/11 - Revision 3.7, Argonne National Laboratory, 2016.

K. Bao, Y. Shi, S. Sun, and X.-P. Wang. A finite element method for the numerical solution of the coupled Cahn-Hilliard and Navier-Stokes system for moving contact line problems. *J. Comput. Phys.*, 231:8083–8099, 2012.

M. Gao and X.-P. Wang. An efficient scheme for a phase field model for the moving contact line problem with variable density and viscosity. *J. Comput. Phys.*, 272:704–718, 2014.

C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *Int. J. Numer. Methods Eng.*, 79(11):1309–1331, 2009.

J.-L. Guermond and A. Salgado. A splitting method for incompressible flows with variable density based on a pressure Poisson equation. *J. Comput. Phys.*, 228:2834–2846, 2009.

V. E. Henson and U. M. Yang. BoomerAMG: a parallel algebraic multigrid solver and preconditioner. *Appl. Numer. Math.*, 41:155–177, 2002.

George Karypis and Vipin Kumar. METIS – Unstructured graph partitioning and sparse matrix ordering system, V2.0. Technical report, 1995.

B. S. Kirk, J. W. Peterson, R. H. Stogner, and G. F. Carey. `libMesh`: A C++ library for parallel adaptive mesh refinement/coarsening simulations. *Eng. Comput.*, 22(3–4):237–254, 2006.

S. Shin, J. Chergui, and D. Juric. A solver for massively parallel direct numerical simulation of three-dimensional multiphase flows. *arXiv*, 1410.8568, 2014.