Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

A fast, decomposed pressure correction method for an intrusive stochastic multiphase flow solver

Brian Turnquist*, Mark Owkes

Montana State University, P.O. Box 173800, Bozeman, 59717 MT, United States

ARTICLE INFO

Article history:
Received 15 October 2020
Revised 10 February 2021
Accepted 10 March 2021
Available online 13 March 2021

Keywords:
Stochastic
Polynomial chaos
Pressure Poisson
Gas-liquid flows
Variable density Navier–Stokes
Uncertainty quantification

ABSTRACT

Solution of the pressure Poisson equation is often the most expensive aspect of solving the incompressible form of Navier–Stokes. For a single phase deterministic model the pressure calculation is costly. Expanded to an intrusive stochastic multiphase framework, the simulation expense grows dramatically due to coupling between the stochastic pressure field and stochastic density. To address this issue in a deterministic framework, Dodd and Ferrante ("A fast pressure-correction method for incompressible two-fluid flows" Journal of Computational Physics, 273, 416–434, 2014) discuss a decomposed pressure correction method which utilizes an estimated pressure field and constant density to modify the standard pressure correction method. The resulting method is useful for improving computational cost for one-fluid formulations of multiphase flow calculations. In this paper, we extend the decomposed pressure correction method to intrusive uncertainty quantification of multiphase flows. The work improves upon the original formulation by modifying the estimated pressure field. The new method is assessed in terms of accuracy and reduction in computational cost with oscillating droplet, damped surface wave, and atomizing jet test cases where we find convergence of results with the proposed method to those of a traditional pressure correction method and analytic solutions, where appropriate.

1. Introduction

Methods of uncertainty quantification (UQ) can be placed into two categories: intrusive and non-intrusive. The non-intrusive category includes approaches such as Monte Carlo [1], collocation methods [2], and non-intrusive polynomial chaos (PC) [3]. The latter two essentially improve on a Monte Carlo by presenting a better way to select the input values so not as many simulations need to be run. In all cases a standard solver can be utilized, which is run many times with parameters selected from a distribution of inputs to compile a database of simulation results. This database is then used to calculate useful statistics of the system in question.

Unlike non-intrusive flow solvers, intrusive UQ methods require a change in the fundamental structure of the solver resulting from modified equations created by the inclusion of stochastic (random) variables. An intrusive solver is created with stochastic variables which store information as a function of added uncertainty dimensions. Stochastic variables developed as a function of uncertainty may take several forms, including PC [4] and Karhunen–Loeve expansions [5,6]. Each of these methods offer their own advantages. The advantage of PC lies in the ability to utilize any number of uncertainty dimensions, the availability of a number of orthogonal basis function families, and the straightforward integration of continuous PC variables into many systems of differential equations. Assuming stable intrusive and non-intrusive UQ schemes, computational cost comparisons are based on the time it takes to generate a reliable source of statistical information on the system being modeled.

Considering UQ applications to multiphase flow dynamics, application of intrusive UQ methods to gas-liquid flows is a developing field. Le Maître et al. [7,8] first developed the stochastic Navier–Stokes equations for single-phase incompressible flows utilizing a PC expansion. Since these works, several studies have implemented a PC-based approach to single-phase flows for a variety of test cases [9–11]. Previous work by Turnquist and Owkes [12] provided the first intrusive UQ method for gas-liquid multiphase flows named the multiUQ framework. The current work builds on this previous work by reducing the computational cost.

In either intrusive or non-intrusive UQ methodology, much computational expense is devoted to solving the pressure Poisson equation. To numerically solve the incompressible Navier–Stokes equations, the standard pressure correction method (SPCM), first introduced by Chorin [13], is a commonly used approach. With the SPCM, time is discretized so that at every time step the convective, viscosity, and any source terms are evaluated and used to predict

[*] Corresponding author.
E-mail address: brturnquist@gmail.com (B. Turnquist).

https://doi.org/10.1016/j.compfluid.2021.104930
0045-7930/© 2021 Elsevier Ltd. All rights reserved.

the velocity without the pressure term. Continuity (or mass conservation) is then used to enforce a divergence free condition at the next time step, while also creating an elliptic Poisson equation to solve for pressure. The approach makes it possible to solve the Navier–Stokes equations with imposed boundary conditions at reasonable computational expense. Further work has been done to expand the method, including improvements to order [14] and application to unstructured grids [15]. When using the SPCM in a multiphase scenario, the pressure Poisson equation becomes coupled to density, which adds computational cost and limits the possible algorithms used to solve. In an effort to counter this cost and following the work of Dong and Shen [16], Dodd and Ferrante [17] proposed a decomposed pressure correction method (DPCM) which would allow for using a fast Fourier transform (FFT) based solver. While numerical errors are added to the model, computational cost is reduced; certainly the trade off is worth consideration.

Given the computational cost improvements of a DPCM in the deterministic setting, it seemed reasonable to apply this methodology to the multiUQ framework [12]. Because of the coupled nature of non-linear terms in the stochastic Navier–Stokes equations due to the use of PC variables, the simulation expense grows at an exponential rate. This so-called curse of dimensionality increases the computational cost very rapidly for intrusive UQ. However, the same curse also affects non-intrusive methods. For example, the use of a Monte-Carlo [1] approach with two or more uncertain variables requires a way to compare the effect of one uncertain variable on another, compounding the number of simulations run to get convergent statistics. Due to this problem, understanding the interaction in uncertainty between multiple variables in a multiphase system is extremely expensive. Use of accurate and cost effective numerical techniques will bring these analyses within reach.

This narrative seeks to develop a more efficient pressure correction approach for stochastic multiphase flows by applying the DPCM to the multiUQ framework outlined in Turnquist and Owkes [12]. A mathematical development of the stochastic DPCM is introduced, followed by a derivation of the numerical methods. We then present test cases which illustrate the computational improvement over previously published methods and the error associated with the density decoupled approach. Finally, we close with a summary of the results and a discussion of where this work will fit in moving forward.

2. Mathematical development

Since the focus of this work is to develop an efficient pressure solver for stochastic multiphase flows, we begin with a development of the stochastic equations for fluid motion. Assuming the fluids are incompressible, this motion can be explained by the Navier–Stokes equations, where

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} = -\eta\nabla P + \eta\nabla\cdot\left[\mu\left(\nabla\boldsymbol{u} + \nabla^{\mathrm{T}}\boldsymbol{u}\right)\right] + \eta\boldsymbol{f}_{\sigma}\delta_{s} \tag{1}$$

for velocity $\boldsymbol{u}$, time t, specific volume $\eta = 1/\rho$ (for density ρ), pressure P, dynamic viscosity μ, and surface tension force $\boldsymbol{f}_{\sigma} = \sigma\kappa\boldsymbol{n}$, where $\boldsymbol{n}$ is the interface normal vector, σ is the surface tension coefficient, and κ is the curvature of the interface between the two fluids. The surface tension force is only applied at the interface boundary, which is denoted by the Dirac delta function δ_s. Additionally, conservation of mass for the incompressible form of Navier–Stokes is accomplished with the continuity equation,

$$\nabla\cdot\boldsymbol{u} = 0. \tag{2}$$

To represent the stochastic variables, this work utilizes the PC expansion as developed by Wiener [4], where some variable ψ, which varies in space $\boldsymbol{x}$ and time t, may be allowed to vary in any number of uncertainty dimensions $\boldsymbol{\zeta}$ such that

$$\psi(\boldsymbol{x},t,\boldsymbol{\zeta}) = \sum_{k=0}^{N}\psi_k(\boldsymbol{x},t)\phi_k(\boldsymbol{\zeta}) = \psi_k(\boldsymbol{x},t)\phi_k(\boldsymbol{\zeta}), \tag{3}$$

for basis weights $\psi_k(\boldsymbol{x},t)$. $\phi_k(\boldsymbol{\zeta})$ for $k = 0,\dots,N$ is a set of $N+1$ orthogonal basis functions upon which the variable is projected. Any of several sets of orthogonal polynomials may be used, including Legendre, Hermite, Laguerre, and Chebychev, though it has been shown each works well for certain function behaviors. For example, Wiener [4] showed the Hermite polynomials can represent Gaussian distributions with a small number of basis functions.

Allowing uncertainty to exist about all variables (except time) in Eq. (1), we substitute the stochastic velocity $\boldsymbol{u}_k\phi_k$, specific volume $\eta_k\phi_k$, pressure $P_k\phi_k$, viscosity $\mu_k\phi_k$, and surface tension $\boldsymbol{f}_{\sigma:k}\phi_k$. Utilizing different free-indices k, l, m for each multiplied variable (i.e. $\eta_k\phi_k$, $P_l\phi_l$, etc.), the result is

$$\begin{aligned}\frac{\partial \boldsymbol{u}_k\phi_k}{\partial t} + \boldsymbol{u}_k\phi_k\cdot\nabla\boldsymbol{u}_l\phi_l &= -\eta_k\phi_k\nabla P_l\phi_l\\ &+\eta_k\phi_k\nabla\cdot\left[\mu_l\phi_l\left(\nabla\boldsymbol{u}_m\phi_m + \nabla^{\mathrm{T}}\boldsymbol{u}_m\phi_m\right)\right]\\ &+\eta_k\phi_k\boldsymbol{f}_{\sigma:l}\phi_l,\end{aligned} \tag{4}$$

and is one form of the stochastic Navier–Stokes equations. We also have the stochastic continuity equation

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

Because this form is difficult to work with and the real values of interest are the basis weights, we utilize the property of orthogonality inherent in the basis functions. For the Legendre polynomials

$$\int_{-1}^{1}\phi_k\phi_b\delta\zeta = \begin{cases}\langle\phi_k\phi_b\rangle & k = b\\ 0 & k \neq b\end{cases}. \tag{6}$$

To this end, we first multiply Eq. (4) by a test function ϕ_b, resulting in

$$\begin{aligned}\frac{\partial \boldsymbol{u}_k\phi_k}{\partial t}\phi_b + \boldsymbol{u}_k\phi_k\cdot\nabla\boldsymbol{u}_l\phi_l\phi_b &= -\eta_k\phi_k\nabla P_l\phi_l\phi_b\\ &+\eta_k\phi_k\nabla\cdot\left[\mu_l\phi_l\left(\nabla\boldsymbol{u}_m\phi_m + \nabla^{\mathrm{T}}\boldsymbol{u}_m\phi_m\right)\right]\phi_b\\ &+\eta_k\phi_k\boldsymbol{f}_{\sigma:l}\phi_l\phi_b.\end{aligned} \tag{7}$$

To leverage the property of orthogonality, we then integrate over the region of orthogonality, $[-1, 1]$,

$$\begin{aligned}\int_{-1}^{1}\frac{\partial \boldsymbol{u}_k\phi_k}{\partial t}\phi_b + \boldsymbol{u}_k\phi_k\cdot\nabla\boldsymbol{u}_l\phi_l\phi_b d\zeta &= \int_{-1}^{1} -\eta_k\phi_k\nabla P_l\phi_l\phi_b\\ &+\eta_k\phi_k\nabla\cdot\left[\mu_l\phi_l\left(\nabla\boldsymbol{u}_m\phi_m + \nabla^{\mathrm{T}}\boldsymbol{u}_m\phi_m\right)\right]\phi_b\\ &+\eta_k\phi_k\boldsymbol{f}_{\sigma:l}\phi_l\phi_b d\zeta,\end{aligned} \tag{8}$$

and divide through by the integral $\langle\phi_b\phi_b\rangle$ to get a more useful form of the stochastic Navier–Stokes equations,

$$\begin{aligned}\frac{\partial \boldsymbol{u}_b}{\partial t} + \boldsymbol{u}_k\cdot\nabla\boldsymbol{u}_l C_{klb} &= -\eta_k\nabla P_l C_{klb}\\ &+\eta_k\nabla\cdot\left[\mu_l\left(\nabla\boldsymbol{u}_m + \nabla^{\mathrm{T}}\boldsymbol{u}_m\right)\right]C_{klmb} + \eta_k\boldsymbol{f}_{\sigma:l}C_{klb},\end{aligned} \tag{9}$$

for $b = 0,\dots,N$ where

$$C_{klb} = \frac{\int_{-1}^{1}\phi_k\phi_l\phi_b d\zeta}{\int_{-1}^{1}\phi_b\phi_b d\zeta} = \frac{\langle\phi_k\phi_l\phi_b\rangle}{\langle\phi_b\phi_b\rangle} \tag{10}$$

and

$$C_{klmb} = \frac{\langle\phi_k\phi_l\phi_m\phi_b\rangle}{\langle\phi_b\phi_b\rangle} \tag{11}$$

are 3rd and 4th order multiplication tensors containing the constants of integration. Additionally, the stochastic continuity equations are

$$\nabla\cdot\boldsymbol{u}_b = 0 \tag{12}$$

for $b = 0, \ldots, N$.

Calculation of the surface tension force is not trivial. Eqs. (9) and (12) are implemented in an incompressible two-phase system utilizing a conservative level set interface capturing scheme as outlined in Turnquist and Owkes [12]. The surface tension is then found by way of a continuum surface force method, as first described by Brackbill et al. [18]. Tryggvason et al. [19] discuss smoothing the interface over a color function, in this case the conservative level set ψ, such that $\nabla\psi \approx \nabla H = \boldsymbol{n}\delta_s$. This methodology operates as a smoothed Heaviside function, $H(\boldsymbol{x})$, in which the stochastic implementation also deviates in the uncertainty domain, i.e. $\psi(\boldsymbol{x}, \boldsymbol{\zeta}) \approx H(\boldsymbol{x}, \boldsymbol{\zeta})$.

A deterministic curvature is calculated by way of the level set, utilizing

$$\kappa = -\nabla \cdot \boldsymbol{n}, \tag{13}$$

for a unit normal about the interface, $\boldsymbol{n} = \nabla\psi/|\nabla\psi|$. Allowing uncertainty to exist about the level set, and thus the unit normal vectors and curvature, we then have a stochastic curvature

$$\kappa_b = \frac{1}{\langle\phi_b\phi_b\rangle}\int_{-1}^{1}\frac{\nabla\psi_k\phi_k}{|\nabla\psi_l\phi_l|}\phi_b d\zeta, \tag{14}$$

which is calculated with a Gaussian quadrature. To avoid projecting discontinuous unit normal vectors onto continuous basis functions, a unit normal is calculated at each quadrature point, thus projecting curvature κ onto the selected basis functions ϕ_k. We then calculate a stochastic surface tension force as

$$\boldsymbol{f}_{\sigma:b} = \sigma_k\kappa_l(\nabla\psi_m)C_{klmb}. \tag{15}$$

3. Numerical methodology

Computations are done on a two-dimensional rectangular domain with a structured Cartesian mesh. Scalar values such as pressure P, level set ψ, density ρ, and viscosity μ are held at the cell center. Subscripts on $P^n_{i,j}$ denote discrete spatial indexing in the x and y directions, respectively, while superscripts denote time discretization. Vector components of velocity $\boldsymbol{u}$, surface tension $\boldsymbol{f}_\sigma$, and continuous normal vector $\boldsymbol{r}$ are held at the cell walls. Second-order finite difference operators are used for spatial derivatives unless otherwise noted.

3.1. Stochastic standard pressure correction

In the standard pressure correction method, the Navier–Stokes equations are discretized in time such that

$$\frac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\Delta t} = -\boldsymbol{u}^n \cdot \nabla\boldsymbol{u}^n + \eta^n\nabla\cdot\left[\mu^n\left(\nabla\boldsymbol{u}^n + \nabla^{\mathrm{T}}\boldsymbol{u}^n\right)\right] + \eta^n\boldsymbol{f}^n_\sigma\delta_s \tag{16}$$

$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^*}{\Delta t} = -\eta^{n+1}\nabla P^{n+1}, \tag{17}$$

where the superscript indicates the time level, i.e., $\Delta t = t^{n+1} - t^n$. As shown, a predicted velocity field $\boldsymbol{u}^*$ is calculated without the pressure field. Spatial discretization for $\nabla\boldsymbol{u}^n$ in the convective term is accomplished with a basic first order upwinding scheme for stability. Focusing on Eq. (16), we expand a deterministic equation to the stochastic realm by substitution of stochastic variables, multiplication of a test function ϕ_b, and integration over ζ to arrive at

$$\begin{aligned}\frac{\boldsymbol{u}^*_b - \boldsymbol{u}^n_b}{\Delta t} &= -\boldsymbol{u}^n_k \cdot \nabla\boldsymbol{u}^n_l C_{klb}\\ &\quad + \eta^n_k\nabla\cdot\left[\mu^n_l\left(\nabla\boldsymbol{u}^n_m + \nabla^{\mathrm{T}}\boldsymbol{u}^n_m\right)\right]C_{klmb} + \eta^n_k\boldsymbol{f}^n_{\sigma:l}C_{klb},\end{aligned} \tag{18}$$

for calculation of stochastic predicted velocity field weights $\boldsymbol{u}^*_b$.

We then take the divergence of Eq. (17) to find pressure P^{n+1}, while enforcing the incompressible constraint $\nabla\cdot\boldsymbol{u}^{n+1} = 0$. In a multiphase system the density field varies and leads to a coupling of density to pressure in the elliptic pressure Poisson equation

$$\nabla^2 P^{n+1} = \frac{\rho^{n+1}\nabla\cdot\boldsymbol{u}^*}{\Delta t} - \rho^{n+1}\nabla\eta^{n+1}\cdot\nabla P^{n+1}. \tag{19}$$

In a deterministic model Eq. (19) is one of the most computationally expensive pieces of the solver. When building an intrusive UQ system, computational expense is compounded. Expanding Eq. (19) into a stochastic regime we have

$$\nabla^2 P^{n+1}_b = \frac{\rho^{n+1}_k\nabla\cdot\boldsymbol{u}^*_l}{\Delta t}C_{klb} - \rho^{n+1}_k\nabla\eta^{n+1}_l\cdot\nabla P^{n+1}_m C_{klmb} \tag{20}$$

for $b = 0, \ldots, N$, resulting in $N+1$ *coupled* pressure Poisson equations, i.e. P^{n+1} exists on both sides of the equation. The pressure Poisson equations are coupled through the 4th-order tensor C_{klmb} from the multiplication of density, specific volume, pressure gradient and test function ϕ_b, which results in a very expensive step to compute the pressure. While the multiplication tensor can be reduced by dropping all zero values, the equation becomes increasingly expensive as more basis functions are required (i.e. increasing N).

3.2. Stochastic decomposed pressure correction

To reduce the compounding computational cost of calculating a stochastic pressure field, it is useful to decompose the density and pressure into constant and variable components. What is beneficial for a deterministic multiphase system is increasingly more beneficial in a stochastic system as the number of basis functions and uncertain variables used is increased. As mentioned previously, Dodd and Ferrante [17] discussed a modification of the SPCM where the pressure-density term of Eq. (17) is modified such that

$$\eta^{n+1}\nabla P^{n+1} \approx \eta_0\nabla P^{n+1} + \left(\eta^{n+1} - \eta_0\right)\nabla\hat{P} \tag{21}$$

for some estimated pressure field $\hat{P}$ and constant specific volume $\eta_0 = 1/\rho_0$. As shown, this substitution couples the needed pressure field P^{n+1} to a constant density term ρ_0, of which the gradient is zero. More specifically, as the estimated pressure field $\hat{P}$ approaches the new pressure field P^{n+1}, the constant density term is canceled out, such that

$$\lim_{\hat{P}\to P^{n+1}}\left[\eta_0\nabla P^{n+1} + \left(\eta^{n+1} - \eta_0\right)\nabla\hat{P}\right] = \eta^{n+1}\nabla P^{n+1}. \tag{22}$$

While the constant density term is essentially arbitrary, for numerical stability, $\rho_0 = \min(\rho_1, \rho_2)$ [17]. Substitution of Eq. (21) into the resulting system of equations for the modified pressure correction method is then

$$\frac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\Delta t} = -\boldsymbol{u}^n \cdot \nabla\boldsymbol{u}^n + \eta^n\nabla\cdot\left[\mu^n\left(\nabla\boldsymbol{u}^n + \nabla^{\mathrm{T}}\boldsymbol{u}^n\right)\right] + \eta^n\boldsymbol{f}^n_\sigma\delta_s \tag{23}$$

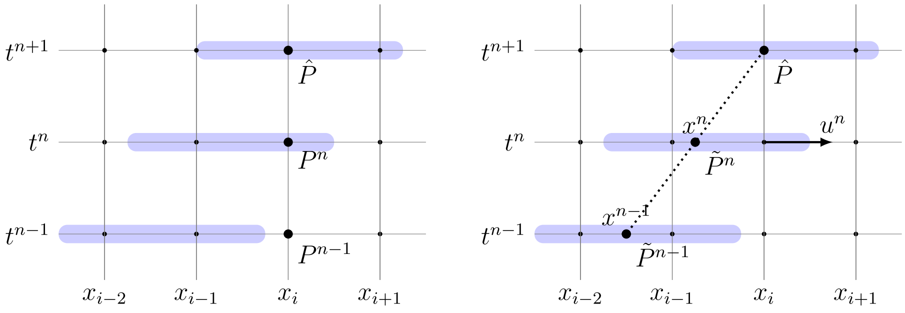
$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^*}{\Delta t} = -\eta_0\nabla P^{n+1} - \left(\eta^{n+1} - \eta_0\right)\nabla\hat{P}. \tag{24}$$

A pressure Poisson equation is found by taking the divergence of Eq. (24) and enforcing $\nabla\cdot\boldsymbol{u}^{n+1} = 0$, leaving

$$\nabla^2 P^{n+1} = \rho_0\frac{\nabla\cdot\boldsymbol{u}^*}{\Delta t} - \rho_0\nabla\cdot\left(\eta^{n+1} - \eta_0\right)\nabla\hat{P}. \tag{25}$$

This equation is linear and can be easily calculated by a number of linear solution algorithms that exist. Additionally, expanding Eq. (25) for stochastic use by substitution of PC variables and integration over ζ we have

$$\nabla^2 P^{n+1}_b = \rho_0\frac{\nabla\cdot\boldsymbol{u}^*_b}{\Delta t} - \rho_0\nabla\cdot\left(\eta^{n+1}_k - \eta_0\right)\nabla\hat{P}_l C_{klb}, \tag{26}$$



(a) Projection (Eq. 29)

(b) Semi-Lagrangian (Eq. 35)

Fig. 1. Illustration of the differences in the projection and semi-Lagrangian approaches to estimation of $\hat{P}$. Blue lines indicate the position of a one-dimensional liquid droplet. At left we see that the projection approach utilizes pressures P^n and P^{n-1} that are within the liquid and gas phases, respectively, to estimate $\hat{P}$. At right we see that the semi-Lagrangian approach utilizes pressures P^n and P^{n-1} that are both in the liquid phase to estimate $\hat{P}$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

which are $N+1$ *decoupled* pressure Poisson equations, i.e. P^{n+1} exists only on the left hand side. There are two efficiency benefits: (1) the right hand side now only requires 3rd-order tensor C_{klb} from the multiplication of specific volume, pressure gradient, and test function ϕ_b (since ρ_0 is constant) and (2) the right hand side is constant. Each basis weight P_b^{n+1} can be calculated by use of any solution algorithm, such as those contained within the HYPRE [20] package maintained by Lawrence Livermore National Lab, which was utilized for this study. This is significantly less costly than the stochastic SPCM, where the coupling of pressure and density required looping over the multiplication tensor C_{klmb} for each iteration of the pressure solver.

To complete the stochastic Navier–Stokes equations with the DPCM, we also include stochastic variables in Eqs. (23) and (24) leading to

$$\frac{\boldsymbol{u}_b^* - \boldsymbol{u}_b^n}{\Delta t} = -\boldsymbol{u}_k^n \cdot \nabla \boldsymbol{u}_l^n C_{klb} + \eta_k^n \nabla \cdot \left[\mu_l^n \left(\nabla \boldsymbol{u}_m^n + \nabla^{\mathrm{T}} \boldsymbol{u}_m^n\right)\right] C_{klmb} + \eta_k^n \boldsymbol{f}_{\sigma:l}^n C_{klb} \tag{27}$$

$$\frac{\boldsymbol{u}_b^{n+1} - \boldsymbol{u}_b^*}{\Delta t} = -\eta_0 \nabla P_b^{n+1} - \left(\eta_k^{n+1} - \eta_0\right) \nabla \hat{P}_l C_{klb}. \tag{28}$$

3.2.1. Estimation of $\hat{P}$

Calculation of P^{n+1} relies on $\hat{P}$, thus the accuracy of the final pressure field is a function of $\left(\hat{P} - P^{n+1}\right)$. Dodd and Ferrante [17] showed constant extrapolation was significantly less accurate than linear extrapolation, where

$$\hat{P} = 2P^n - P^{n-1}. \tag{29}$$

However, this simple approximation is subject to limitations, as discussed in the example of a rising air bubble in water by Dong and Shen [16], as well as Cifani [21], who made efforts to improve it. In a multiphase flow, the passage of the interface over a given location results in a pressure jump. Utilizing Eq. (29), at a grid point where the interface has just arrived, the estimate of $\hat{P}$ for the next time step would be for another pressure jump, as illustrated in Fig. 1(a), where there is a pressure jump between points P^{n-1} and P^n, which will be extrapolated through Eq. (29) causing another pressure jump. Inversely, at points where the interface has just left, the estimate of $\hat{P}$ would be for another pressure drop. A double jump exists at the front and back of a moving droplet for $\hat{P}$. This jump requires multiple time steps to clear away from any given cell, but exists about the interface throughout the simulation. While the DPCM does maintain a divergence free velocity field, the difference $\left(\hat{P} - P^{n+1}\right)$ near the interface creates pressure fluctuations which are non-physical and introduce error to the simulation, as discussed by Cifani [21], and will be discussed in Section 4.1.3 of the present work.

Alternatively, to avoid the issue caused by a simple linear projection, a semi-Lagrangian interpolation method is proposed where the location of each grid point are transported backwards with the velocity field, and the pressure $\tilde{P}$ at that point is found with bilinear interpolation (for a 2-D system). More specifically, a particle $\boldsymbol{x}_i$ located at the cell center at t^{n+1} can be traced back to its previous location using the differential equation

$$\frac{\partial \boldsymbol{x}}{\partial t} = \boldsymbol{u}, \tag{30}$$

discretized with an explicit Euler approach, e.g., $\boldsymbol{x}^n = \boldsymbol{x}_i - \boldsymbol{u}^n \Delta t$ and $\boldsymbol{x}^{n-1} = \boldsymbol{x}_i - 2\boldsymbol{u}^n \Delta t$. Once $\boldsymbol{x}^n$ and $\boldsymbol{x}^{n-1}$ are located, the pressure is found via bi-linear interpolation of the nearest grid points. Due to the stochastic velocity field, the previous locations $\boldsymbol{x}^n$ and $\boldsymbol{x}^{n-1}$ are uncertain. We defined a particle $\boldsymbol{x}_i$ at the cell center and track it back through an uncertain velocity field. Following substitution of stochastic velocity $\boldsymbol{u}_k \phi_k$, multiplication by a test function, and integration over ζ, we discretize Eq. (30) to find the previous particle locations

$$\begin{aligned} \boldsymbol{x}_b^n &= \tfrac{\langle \phi_b \rangle}{\langle \phi_b \phi_b \rangle} \boldsymbol{x}_i - \boldsymbol{u}_b^n \Delta t \quad \text{and} \\ \boldsymbol{x}_b^{n-1} &= \tfrac{\langle \phi_b \rangle}{\langle \phi_b \phi_b \rangle} \boldsymbol{x}_i - 2\boldsymbol{u}_b^n \Delta t. \end{aligned} \tag{31}$$

Even in a deterministic setting, depending on the location of this particle, the nearest neighbors may be shifted to one of the four quadrants surrounding the central pressure $P_{i,j}$. For explanation, assuming the particle to be in the third quadrant, the semi-Lagrangian pressure $\tilde{P}^n$ at (x, y) may be found with

$$\tilde{P}_{i,j}^n = \frac{1}{\Delta x \Delta y} \begin{bmatrix} x_{i+1} - x^n & x^n - x_i \end{bmatrix} \begin{bmatrix} P_{i+1,j}^{n-1} & P_{i+1,j-1}^{n-1} \\ P_{i,j}^{n-1} & P_{i,j-1}^{n-1} \end{bmatrix} \begin{bmatrix} y_j - y^n \\ y^n - y_{j-1} \end{bmatrix}. \tag{32}$$

Rather than utilize a quadrature, the average particle location $\boldsymbol{x}_0^n$ is used (i.e. $\phi_0 = 1$). Thus, the four nearest neighbors to $\boldsymbol{x}_0^n$ (based on the quadrant it falls in) are used to calculate the pressure field $\tilde{P}^n$ utilizing bi-linear interpolation. Importing uncertain variables, multiplying by a test function and integrating, we then find a stochastic bi-linear interpolation

$$\begin{aligned} \tilde{P}_{b:i,j}^n = \frac{1}{\langle \phi_b \phi_b \rangle} \int_{-1}^{1} \frac{\phi_b}{\Delta x \Delta y} \cdot & \begin{bmatrix} x_{i+1} - x_0^n \phi_0 & x_0^n \phi_0 - x_i \end{bmatrix} \begin{bmatrix} P_{l:i+1,j}^n \phi_l & P_{l:i+1,j-1}^n \phi_l \\ P_{l:i,j}^n \phi_l & P_{l:i,j-1}^n \phi_l \end{bmatrix} \\ & \begin{bmatrix} y_j - y_0^n \phi_0 \\ y_0^n \phi_0 - y_{j-1} \end{bmatrix} d\zeta. \end{aligned} \tag{33}$$

Performing the integration over ζ, and noting the simplification by utilizing the average previous location, we have

$$\tilde{P}^n_{b:i,j} = \frac{1}{\Delta x \Delta y}\begin{bmatrix} x_{i+1} - x_0^{n-1} & x_0^{n-1} - x_i \end{bmatrix} \begin{bmatrix} P^n_{b:i+1,j} & P^n_{b:i+1,j-1} \\ P^n_{b:i,j} & P^n_{b:i,j-1} \end{bmatrix} \begin{bmatrix} y_j - y_0^{n-1} \\ y_0^{n-1} - y_{j-1} \end{bmatrix}. \tag{34}$$

Once this interpolated pressure $\tilde{P}^n$ has been calculated, and utilizing the interpolated pressure from the previous time step $\tilde{P}^{n-1}$, we can again extrapolate to the next time for our estimate

$$\hat{P} = 2\tilde{P}^n - \tilde{P}^{n-1}. \tag{35}$$

The difference here is that we are tracking the interface as it arrives and moves through any given grid point, as illustrated in Fig. 1(b). While this provides a reasonable and bounded estimate of $\hat{P}$, which avoids a double jump, this estimate can be further improved using an iterative approach, where each successive iteration can utilize an improved estimation of the pressure $\hat{P}$ as described in the next section.

3.3. Iterative midpoint method

Given that the updated velocity field $\boldsymbol{u}^{n+1}$ is calculated using both the calculated pressure field P^{n+1} and $\hat{P}$, the best calculation of the next time iteration occurs when $\hat{P} = P^{n+1}$. To implement this, aniterative midpoint scheme is used. For the first iteration, we utilize Eq. (23) to find $\boldsymbol{u}^*$. To get an initial calculation of $\boldsymbol{u}^{n+1}$ (Eq. (24)) the estimated pressure field is calculated using either the linear projection method (Eq. (29)) or the semi-Lagrangian approach (Eq. (35)). On subsequent iterations we then modify Eq. (23) with midpoint information such that

$$\frac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\Delta t} = -\boldsymbol{u}^{n+1/2} \cdot \nabla \boldsymbol{u}^{n+1/2} + \eta^{n+1/2} \nabla \cdot \left[\mu^{n+1/2}\left(\nabla \boldsymbol{u}^{n+1/2} + \nabla^{\mathrm{T}} \boldsymbol{u}^{n+1/2}\right)\right] + \eta^{n+1/2} \boldsymbol{f}_\sigma^{n+1/2} \delta_s, \tag{36}$$

using $\boldsymbol{u}^{n+1/2} = (\boldsymbol{u}^n + \boldsymbol{u}^{n+1})/2$. Interface transport is also accomplished with this midpoint scheme for consistent time location of variables. Thus, we calculate midpoint values of the level set $\psi^{n+1/2}$, which we use to find specific volume $\eta^{n+1/2}$, viscosity $\mu^{n+1/2}$, and surface tension $\boldsymbol{f}_\sigma^{n+1/2}$. For pressure correction (Eq. (24)) of subsequent iterations we use the previous calculation of P^{n+1} as $\hat{P}$, converging to the best estimate of P^{n+1}, i.e. $\hat{P} \rightarrow P^{n+1}$. Furthermore, within each midpoint step, while the right hand side of Navier–Stokes is constant, it is possible to further iterate over the pressure solver, which improves the estimate of P^{n+1} while bypassing the expense of solving the rest of the Navier–Stokes equations.

As shown in Fig. 2, the estimate of the pressure field $\hat{P}$ is continually updated throughout the iterative process, converging to P^{n+1} and reducing the error inherent in the method. The maximum number of midpoint (N_m) and pressure iterations (N_P) can be reduced if some convergence criterion is reached (variable converge). Note, there is a compounding effect of updating $\hat{P}$ inside the midpoint loop. For low density ratios (≤ 100) and/or slow velocity fields it is sufficient to loop over the pressure solver once (i.e. $N_p = 1$). However, at high density ratios and/or rapidly evolving systems it is necessary to improve our estimate of the next pressure field, increasing our value of N_p. This can be done in lieu of reducing the time step size, or CFL value, as suggested by Dodd and Ferrante [17].

To obtain second-order accuracy in the time marching scheme, it is only necessary to perform two midpoint iterations (i.e. $N_m = 2$). Given that updating the midpoint loop runs through all calculations of the level set, velocity, and pressure solver, it makes sense to reduce this number and iterate over the pressure solver to converge the estimate of $\hat{P}$ to P^{n+1}. However, in practice it is not this straightforward, since the first step of the midpoint method is essentially an explicit Euler prediction, which is then improved by another iteration. Thus, our first estimates of $\hat{P}$ are based on a first-order accurate prediction. Through testing we have found that increasing N_m helps to converge $\hat{P}$, resulting in fewer overall iterations of the pressure Poisson equation per time step, as discussed in more detail in Section 4.1.1.

It also makes sense to discuss the oddity of looping several times over the most expensive part of the simulation in an attempt to improve computational efficiency. The improvement of efficiency comes from a combination of the ability to use more efficient solution methods and the uncoupled nature of the pressure Poisson equation. When using the SPCM, convergence of P^{n+1} is slow, and due to the coupling of density and pressure each basis weight P_b^{n+1} is linked to all others. With the DPCM each basis weight is solved separately, significantly reducing the number of calculations at each time step due to not looping over 4th-order multiplication tensor C_{klmb}. The decoupling reduces the computational cost growth as basis functions are added, i.e.the computational cost savings grows with N.

4. Test cases and computational assessment

Two test cases are used to evaluate the accuracy and efficiency of the proposed method, while a third is used to test the method on a more complicated scenario. First, an oscillating droplet case is used as there exists an analytic solution providing the oscillation period [22]. This case tests the ability of the surface tension force to drive flow. Second, a damped surface wave, which also has an analytic solution [23], is used to judge the accuracy of the interplay between viscosity and surface tension. Finally, the third test case is an atomizing jet, which demonstrates the ability of the method to resolve difficult physical situations. For each case, the semi-Langrangian projection method of Eq. (35) is utilized for best results, with comparison to Eq. (29) in a high density ratio scenario. These three tests focus on how the modified pressure projection method effects the solution and computational cost.

4.1. Oscillating droplet

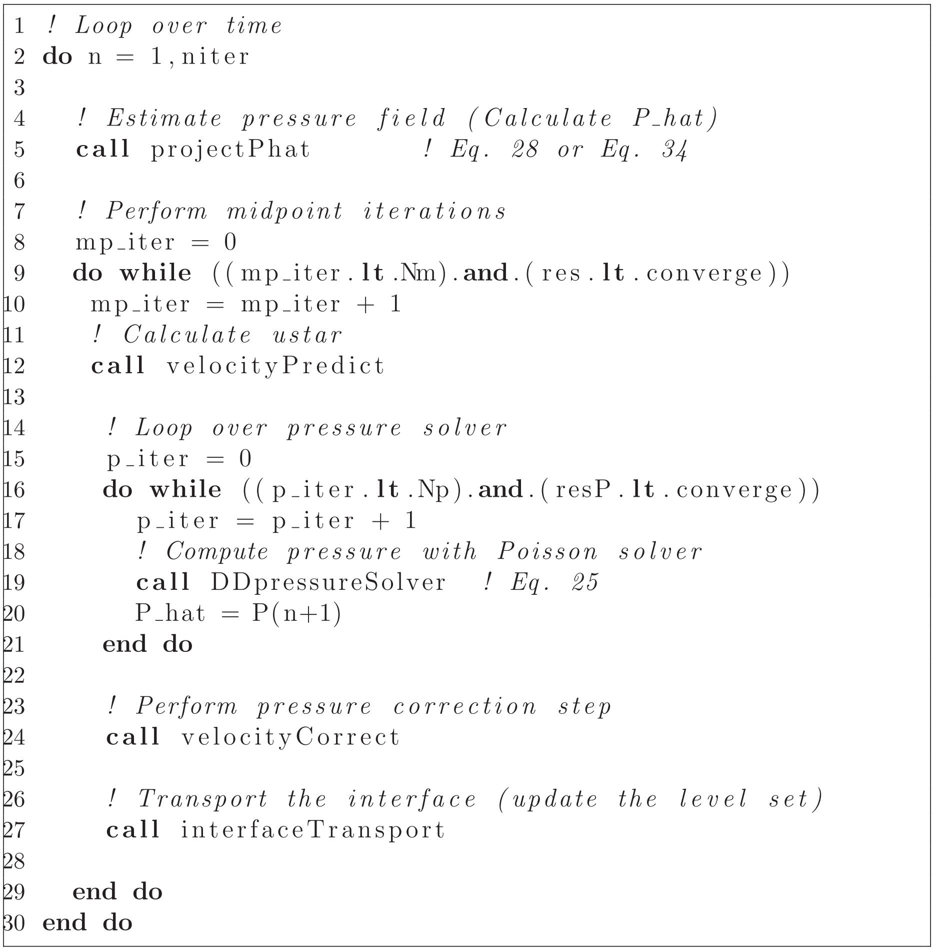
The oscillating droplet is a common benchmark test for validating the accuracy and abilities of a multiphase solver [24–27]. Rather than inducing flow from the boundary, this flow is driven by the surface tension force, which allows for an indirect test of the accuracy of the numerical method to calculate the surface tension by comparison of the analytic oscillation period to that determined from the simulation. As mentioned, the period of oscillation was defined by Lord Rayleigh [22] and described in 2-D by Fyfe et al. [28], where

$$\tau = 2\pi\sqrt{\frac{(\rho_1 + \rho_2)R^3}{6\sigma}} \tag{37}$$

for period τ and unperturbed radius $R = \sqrt{AB}$ of an ellipse described by $x^2/A^2 + y^2/B^2 = 1$, with semi-major and semi-minor axes A and B, respectively.

There are peculiarities to this case, some discussed by Salih and Ghosh Moulic [29]. First, the period of oscillation as defined by Lord Rayleigh [22] assumed an inviscid system, thus viscosity is not present in Eq. (37) and cannot affect the period of oscillation. As discussed by Salih and Ghosh Moulic [29], it is found that viscosity serves to dampen the amplitude of oscillations. Second, the period of oscillation of the simulation is found indirectly by computing the total kinetic energy of the system,

$$\mathrm{KE}_g = \frac{1}{2}\int_V \rho \boldsymbol{u} \cdot \boldsymbol{u} dV. \tag{38}$$



```
! Loop over time
do n = 1,niter

  ! Estimate pressure field (Calculate P_hat)
  call projectPhat        ! Eq. 28 or Eq. 34

  ! Perform midpoint iterations
  mp_iter = 0
  do while ((mp_iter.lt.Nm).and.(res.lt.converge))
   mp_iter = mp_iter + 1
   ! Calculate ustar
   call velocityPredict

    ! Loop over pressure solver
    p_iter = 0
    do while ((p_iter.lt.Np).and.(resP.lt.converge))
      p_iter = p_iter + 1
      ! Compute pressure with Poisson solver
      call DDpressureSolver  ! Eq. 25
      P_hat = P(n+1)
    end do

    ! Perform pressure correction step
    call velocityCorrect

    ! Transport the interface (update the level set)
    call interfaceTransport

  end do
end do
```

Fig. 2. Pseudo code describing the procedural order for an iterative approach to the fast pressure solver. While loops are performed until a small change is reached (i.e. res.lt.converge) or to a max number of iterations.

This works in part because the system begins as a static ellipse with only the potential energy of surface tension imposed. Kinetic energy then should grow and peak, but slowly diminish as the surface tension force then counteracts the movement of the fluid, at some point reaching the maximum ellipsoid shape about the 2nd dimension, when a minima of movement is taking place.

We look at variations of the oscillating droplet case with an initial ellipse of $A = 0.25$ cm and $B = 0.15$ cm centered in a $[0, 2]^2$ domain. Liquid properties are first set to $\nu_1 = \nu_2 = 0$ for viscosity (in keeping with the original analytic result) while $\rho_1 = 1.0$ g/cm^3 and $\rho_2 = 0.01$ g/cm^3 are set for density. Surface tension coefficient is set at $\sigma = 72.8$ g/s^2 for all simulations.

4.1.1. Comparison to traditional pressure correction

In the implementation of the DPCM, there are two parameters, N_c and N_P, that can be used to improve the predicted pressure $\hat{P}$. The effect of the number of midpoint iterations N_c is first investigated to determine if simply increasing N_m to improve the velocity, pressure, and level set transport will be enough to converge $\hat{P}$. With this approach to the numerical scheme, the magnitude of kinetic energy was found to be largely affected by the number of midpoint iterations, while the period is shifted slightly. Fig. 3(a) illustrates this effect for a range of N_m and mesh sizes. This is due to the continually improved estimate of $\hat{P}$ with each iteration. Also shown is the more rapid decay of kinetic energy with smaller N_m.

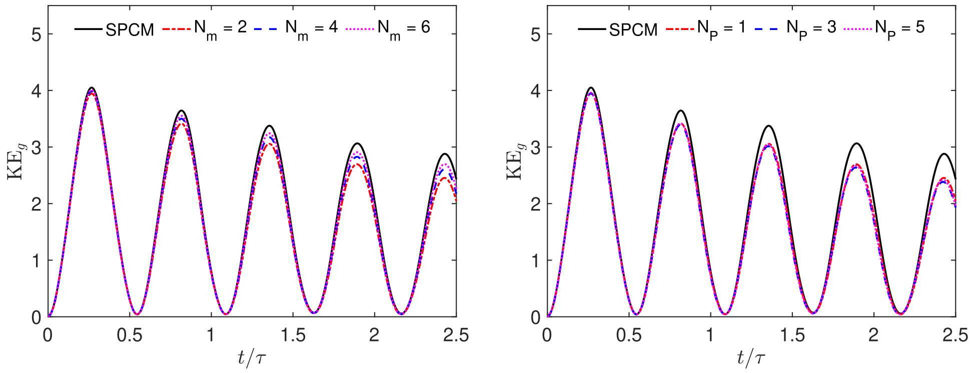
We must also determine if it is sufficient to run only two midpoint iterations ($N_m = 2$) to achieve 2nd-order accuracy in time, but loop over the pressure solver multiple times. Fig. 3(b) illustrates the solution tendency to this approach over a range of N_P with $N_m = 2$. We find that while this approach also appears to be convergent, there is much slower improvement of kinetic energy magnitude. Again, as the estimate of $\hat{P}$ converges to P^{n+1}, the difference in the velocity field between the two correction methods shrinks.

4.1.2. Determination of N_m and N_P

We can see from Fig. 3 that finding a proper $\hat{P}$ is necessary, and that it is likely better to iterate over both the midpoint scheme and the pressure solver. It is thus useful to find the most efficient combination of midpoint and pressure iterations in hopes of achieving an acceptable level of error while keeping computational expense relatively low. To determine the most efficient possible simulation layout, 42 simulations were run on a 64 × 64 mesh with the oscillating droplet method, using the SPCM results as a baseline. Tests were run over a range of values, $N_m = [2, 7]$ and $N_P = [1, 7]$, with the goal of minimizing both the error of DPCM results and total computational time. Both models were allowed to run to a given divergence level (i.e. $\nabla \cdot \boldsymbol{u} < 1 \times 10^{-14}$) with a set time step. Error was then described as the difference in kinetic energy between the two methods,

$$E_{\text{diff}} = \text{KE}_{g:\text{SPCM}} - \text{KE}_{g:\text{DPCM}}, \tag{39}$$

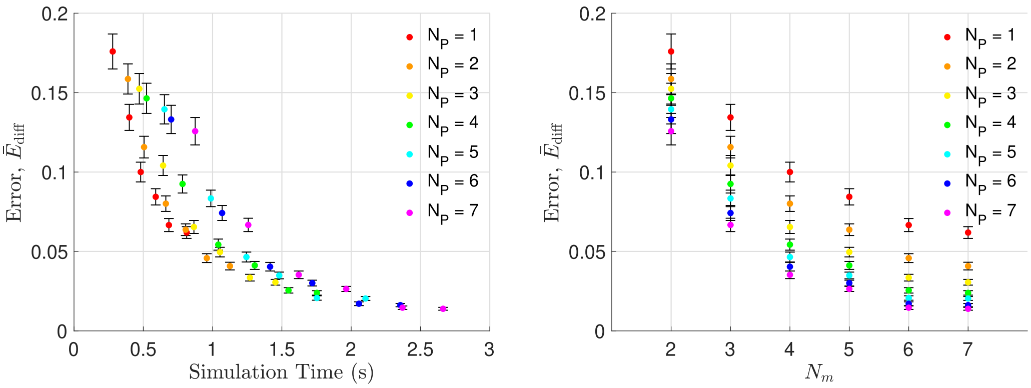
at each time step. An average error $\bar{E}_{\text{diff}}$ was calculated for each simulation, as well as an associated standard error to calculate a 95% confidence interval about the average.



(a) $N_P = 1$

(b) $N_m = 2$

Fig. 3. Global kinetic energy of an oscillating droplet for a range of midpoint iterations (N_m) and a range of pressure iterations (N_P) on a 128×128 mesh. At left, the pressure is only calculated once per midpoint iteration, i.e. $N_P = 1$. At right, simulations were run with the minimum number of midpoint iterations, i.e. $N_m = 2$.



(a) Error reduction by simulation time.

(b) Effect of N_m & N_P on error.

Fig. 4. Relationship of simulation time on the error $\bar{E}_{\text{diff}}$ between solutions of the DPCM and SPCM (left) and the impact of midpoint iterations on error $\bar{E}_{\text{diff}}$ (right). Additionally, error bars in represent a 95% confidence interval about $\bar{E}_{\text{diff}}$.

Table 1
Simulation run times of several combinations for DPCM. Reported error is the average difference between DPCM and SPCM over three oscillations.

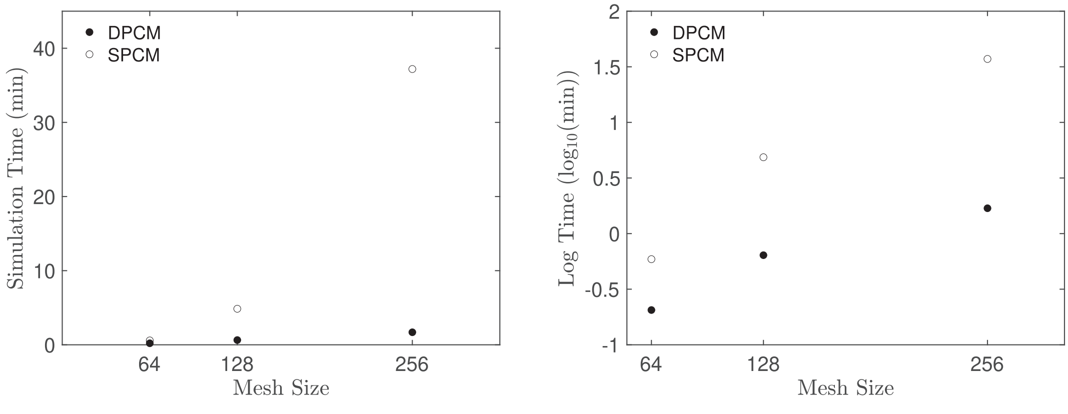
N_m	N_P	Time (min)	Error ($\bar{\boldsymbol{E}}_{\text{diff}}$)	95% CI
2	SPCM	3.769	–	–
5	4	1.304	0.041	0.0025
6	3	1.269	0.034	0.0021
6	4	1.548	0.026	0.0017
6	5	1.752	0.021	0.0014
7	2	1.125	0.041	0.0024

The results of these 42 tests are illustrated in Fig. 4, which describe the relationship between simulation time and error $\bar{E}_{\text{diff}}$ to the SPCM. The error can be continually reduced, but with growing expense. We can see convergence of the error trending to zero as more iterations are added, shown in Fig. 4(b). Also shown in the reduction in the confidence interval as N_m increases. While there is no clear combination to choose, it appears several combinations along the bottom on the curve in Fig. 4(a) offer a balance of relatively small error and computational efficiency. These combinations are highlighted in Table 1, and further show the trade-off between computational effort and accuracy.

Given that several observations appear reasonable, the combination of $N_m = 6$ and $N_P = 3$ was chosen for its relatively low error and computational expense. The choice of $N_m = 6$ and $N_P = 4$ reduces error by 24% but increased simulation time by 22%, i.e. there are diminishing returns. However, we note even the most expensive case tested with the DPCM, where $N_m = 7$ and $N_P = 7$ had a run time of 2.664 min compared to 3.769 min for the SPCM, offering an $\approx 30\%$ speedup. To test the chosen combination of $N_m = 6$ and $N_P = 3$ for mesh convergence we compared the difference in the total simulation run time for 100 iterations (using four Intel Core i7 2.5 Ghz processors) between standard and decomposed pressure correction methods, as shown in Fig. 5. As seen, we have a significant speedup with the DPCM, which grows with mesh refinement.

4.1.3. Methods for calculating $\hat{P}$

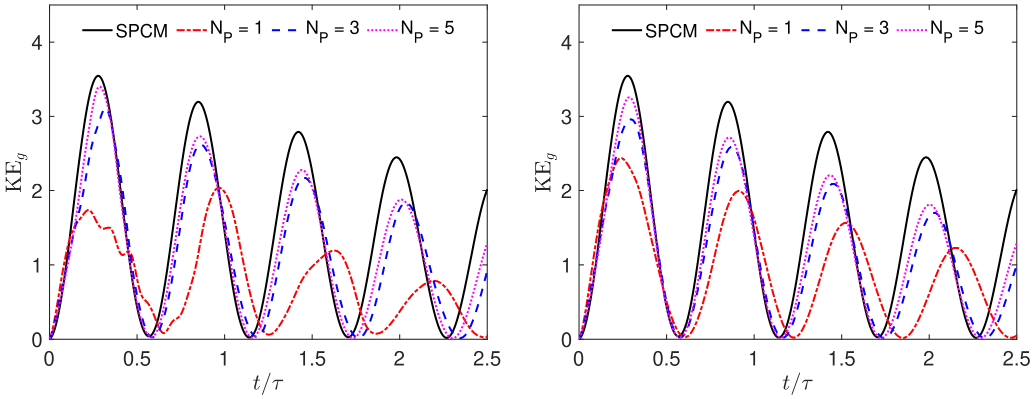
An oscillating droplet with a high density ratio (e.g. an air and water system) is more numerically difficult to resolve. In the DPCM framework, it is even more important to have a reasonable estimate of $\hat{P}$ to reduce computational errors. This is because the constant density term ρ_0 is then orders of magnitude different from the highest density fluid. This exacerbates the errors caused by the difference between $\hat{P}$ and P^{n+1}, and thus the errors in the resulting velocity field. Given this issue, we also consider the effect of a simple linear approximation (Eq. (29)), which creates unrealistic and non-physical jumps near the interface. This causes huge variations in the velocity field near the interface, and can cause simulation instabilities leading to failure. We compare this with the proposed semi-Lagrangian extrapolation which bounds $\hat{P}$ by the local maxima and minima of the current pressure field while tracking



(a) Time scale

(b) Log time scale

Fig. 5. Computational time of oscillating droplet simulations over a range of mesh sizes for a set number of iterations.



(a) Linear approximation (Eq. 29)

(b) Semi-Lagrangian approximation (Eq. 35)

Fig. 6. Effect of the estimation method used to calculate $\hat{P}$ on solutions of a high density ratio oscillating droplet for a 64×64 mesh. In each case $N_m = 2$.

movement of the interface. This provides a stable first estimate which does avoids a double jump and maintains a more stable estimate of $\hat{P}$.

Fig. 6 directly compares the two estimation schemes. As shown, simply utilizing the basic linear projection method (Eq. (29)) is a very poor estimate without iterating the pressure solver multiple times (i.e. the case of $N_P = 1$). By comparison, the proposed semi-Lagrangian method (Eq. (35)) has a more stable oscillation when $N_P = 1$. Additionally, the method converges more rapidly to the SPCM when the estimate of $\hat{P}$ is improved by iterating the pressure solver.

4.1.4. Translating droplet

To further consider the affect of the estimation scheme on $\hat{P}$ we consider the case of an oscillating droplet placed in a moving velocity field. We again utilize the fluid properties of air and water, with an initial ellipse the same as in previous cases. The only difference is that the initial velocity field is set to $u = 10$ cm/s everywhere, with a continuous inflow of $u_{\text{in}} = 10$ throughout the simulation. The goal is to exacerbate the issue caused by the double jump present in the linear estimation of $\hat{P}$.

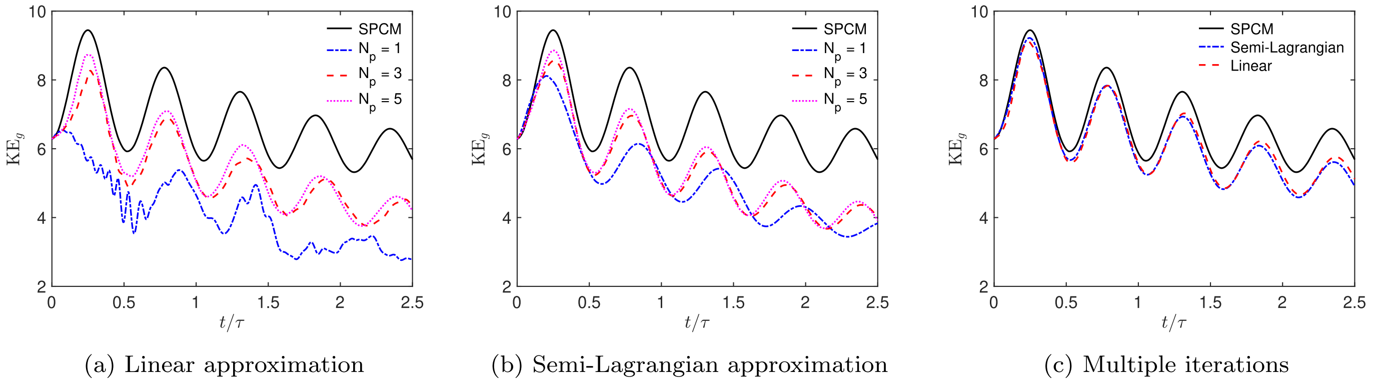
Fig. 7 displays the results of several simulations utilizing either the linear or semi-Lagragian estimate of $\hat{P}$. We see that without iteration, the linear estimate is unstable, while the semi-Langragian estimate remains stable. Iteration over either method quickly improves stability, converging to the results found with the SPCM, which we can see in Fig. 7(c) for simulations with $N_m = 6$ and $N_P = 3$. Given a large number of iterations, either method is useful. However, the semi-Lagangian method is most applicable and perhaps necessary for implementations of the DPCM that are not iterative, or in situations where fast results can be useful (i.e. running the iterative scheme with only 1 pressure iteration) for preliminary findings.

4.1.5. Stochastic viscosity simulations

Given a stochastic multiphase solver, it is possible to look at the effect of viscosity in multiphase flow dynamics. Continuing with the example of the oscillating droplet, we now impose uncertainty about the viscosity of both fluids in the system, holding all other variables equal from the deterministic simulations shown previously. To impose uncertainty about viscosity we utilize a uniform distribution for each phase, i.e.,

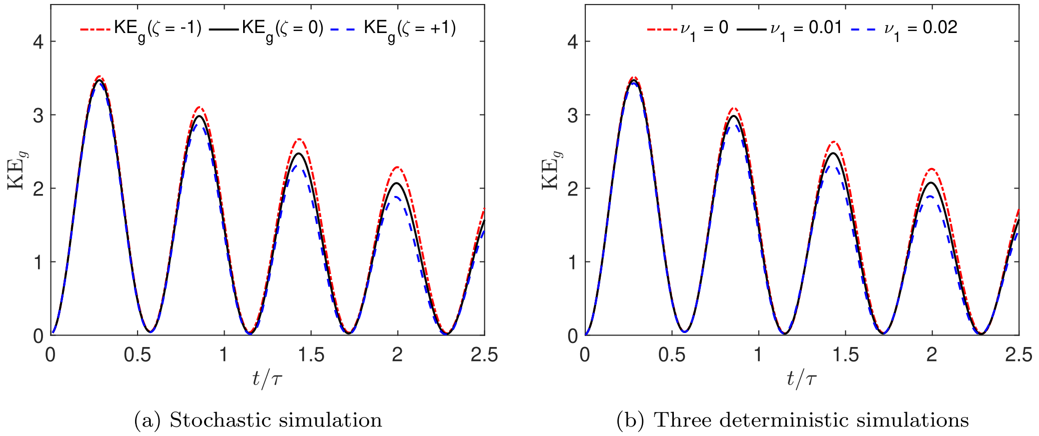
$$\nu_1 = 0.01 + 0.01\zeta \quad \text{and} \qquad \nu_2 = 0.15 + 0.15\zeta. \tag{40}$$

This allows us to test a range of cases from an inviscid situation to one where viscosity is double that of air and water. Fig. 8(a) displays the results of this test case. We can see the increase in viscosity causes added decay of the kinetic energy in the system. Additionally, we see the viscosity has a negligible impact on the period of oscillation, suggesting the inviscid assumption made by Lord Rayleigh [22] is reasonable. While these three solutions were all created by one stochastic simulation (in addition to an infinite number of others), it is necessary to run three deterministic simulations to compare. Fig. 8 shows the results of the UQ simulation closely match that of the three deterministic simulations.



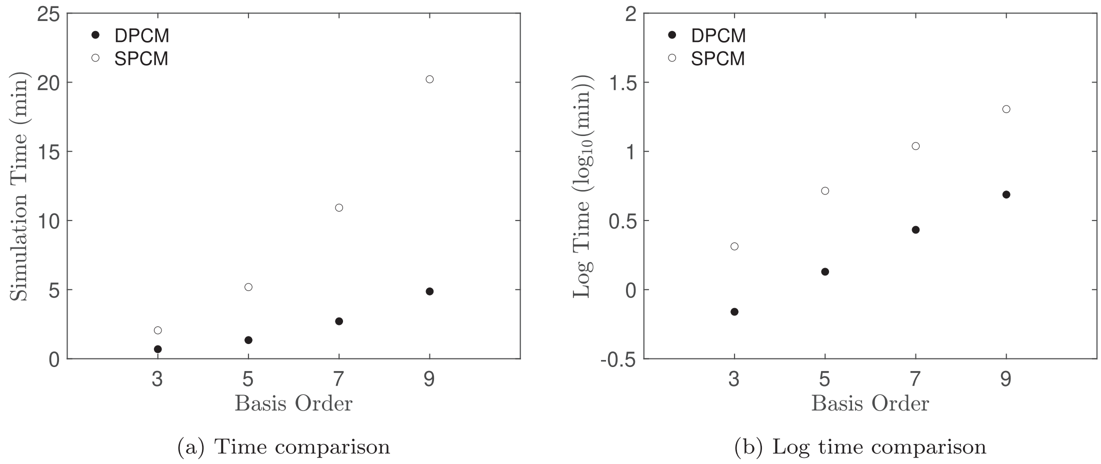
(a) Linear approximation (b) Semi-Lagrangian approximation (c) Multiple iterations

Fig. 7. Results of a translating oscillating droplet for two methods of estimating $\hat{P}$. The left two plots depict simulations using the linear and semi-Lagrangian estimation methods, respectively, over a range of N_P with $N_m = 2$. The plot at right compares the two methods with the settled upon combination of $N_m = 6$ and $N_P = 3$.



(a) Stochastic simulation (b) Three deterministic simulations

Fig. 8. Results of a stochastic oscillating droplet with variability about viscosity on a 64 × 64 mesh. At left are results from a single stochastic simulation, while at right are results from three deterministic simulations for comparison.



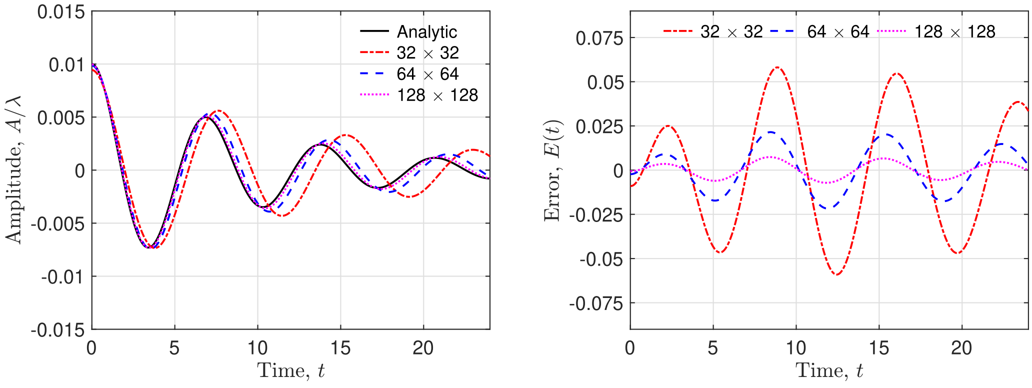
(a) Time comparison (b) Log time comparison

Fig. 9. Computational time for a set number of iterations over a range of basis functions using two types of pressure correction methods in a stochastic simulation.

Since the stochastic simulations with the proposed DPCM method can reproduce the results from these three deterministic simulations, it is useful to compare the computational cost of the SPCM method and the DPCM method. For this comparison, we are interested in determining the effect of basis order N on the computational time, and the efficiency gained by implementing the DPCM. Timing tests were run using 100 time iterations with a convergence level of 10^{-12} for both the traditional and proposed pressure correction methods. All computations are performed on four 2.5 GHz Intel Core i7 processors with a 64^2 mesh. Fig. 9 illustrates the time taken to run these iterations over a range of basis functions. Given the large difference in computational time, a semi-log plot is also presented. As shown, the cost of the SPCM grows much more rapidly than that of the DPCM as basis order is increased.

4.2. Damped surface wave

We now look at the interaction between viscosity and surface tension as outlined by Prosperetti [23] and utilized by Herrmann [30] among others [25,31,32]. This case consists of two viscous flu-



(a) Amplitude

(b) Numerical Error

Fig. 10. Amplitude of a damped surface wave with a density ratio of $\rho_1/\rho_2 = 1$ for three different mesh sizes. Analytic solution described by Prosperetti [23] shown with the solid black line.

ids separated by a horizontal interface initialized with a slight perturbation. This is described by the signed distance function $g(\boldsymbol{x}, t)$ used to initialize the level set, where

$$g(\boldsymbol{x}, t=0) = y - y_o + A_o \cos\left(x - \frac{h_g}{2}\right), \tag{41}$$

with $y_o = \pi$, $A_o = 0.01\lambda$, wavelength $\lambda = 2\pi$, and $h_g = \lambda/N_x$ for the number of mesh points in the x-direction N_x. Simulations are performed in a $[0, 2\pi] \times [0, 2\pi]$ domain. Numerical results of the wave amplitude are compared to the expected theoretic amplitude suggested by Prosperetti [23], in the case of two fluids with equal kinematic viscosity, where

$$\begin{aligned} A_{\exp}(t) = {} & \tfrac{4(1-4\beta)\nu^2 A_o}{8(1-4\beta)\nu^2+\omega_o^2}\mathrm{erfc}\big(\sqrt{\nu t}\big) \\ & + \sum_{i=1}^{4} \frac{z_i \omega_o^2 A_o}{Z_i\big(z_i^2 - \nu\big)} \exp\big[\big(z_i^2 - \nu\big)t\big]\mathrm{erfc}\big(z_i\sqrt{t}\big) \end{aligned} \tag{42}$$

and z_i are the roots of

$$\begin{aligned} & z^4 - 4\beta\sqrt{\nu}z^3 + 2(1-6\beta)\nu z^2 + 4(1-3\beta)\nu^{3/2}z \\ & + (1-4\beta)\nu^2 + \omega_o^2 = 0. \end{aligned} \tag{43}$$

The inviscid oscillation frequency is given by $\omega_o = \sqrt{\sigma/(\rho_1+\rho_2)}$, with parameter $\beta = \rho_1\rho_2/(\rho_1+\rho_2)^2$, and $Z_i = \prod_{j=1, j\neq i}^{4}\big(z_j - z_i\big)$. The amplitude of the numerical model is found by monitoring the height of the wave which is located at the center of the domain.

4.2.1. Deterministic simulations

Following the test cases of Herrmann [30], two deterministic cases are explored for a surface tension of $\sigma = 2\ \frac{\mathrm{g}}{\mathrm{s}^2}$. In the first case we use densities of $\rho_1 = \rho_2 = 1\ \frac{\mathrm{g}}{\mathrm{cm}^3}$ with a viscosity of $\nu = 0.064720863\ \frac{\mathrm{cm}^2}{\mathrm{s}}$. In the second case a density ratio of 1000 is used, with $\rho_1 = 1000\ \frac{\mathrm{g}}{\mathrm{cm}^3}$ and $\rho_2 = 1\ \frac{\mathrm{g}}{\mathrm{cm}^3}$, and both fluids have a viscosity of $\nu = 0.0064720863\ \frac{\mathrm{cm}^2}{\mathrm{s}}$. In all cases the DPCM method is used with $N_m = 6$ and $N_P = 3$.

Results of the case of density ratio one are shown in Fig. 10, which displays a convergence of results with mesh refinement. Additionally, the error of the simulation is shown, where $E(t) = (A(t) - A_{\exp})/A_o$. It is useful to point out that in this simulation, where the two fluids have equal density, the use of the DPCM introduces very minor errors into the model as $\rho_o = \rho_1 = \rho_2$. Because of this we can quickly find $\hat{P} = P^{n+1}$, through iterative convergence. In fact, it was found that during the simulation, convergence was reached after 2 pressure iterations.

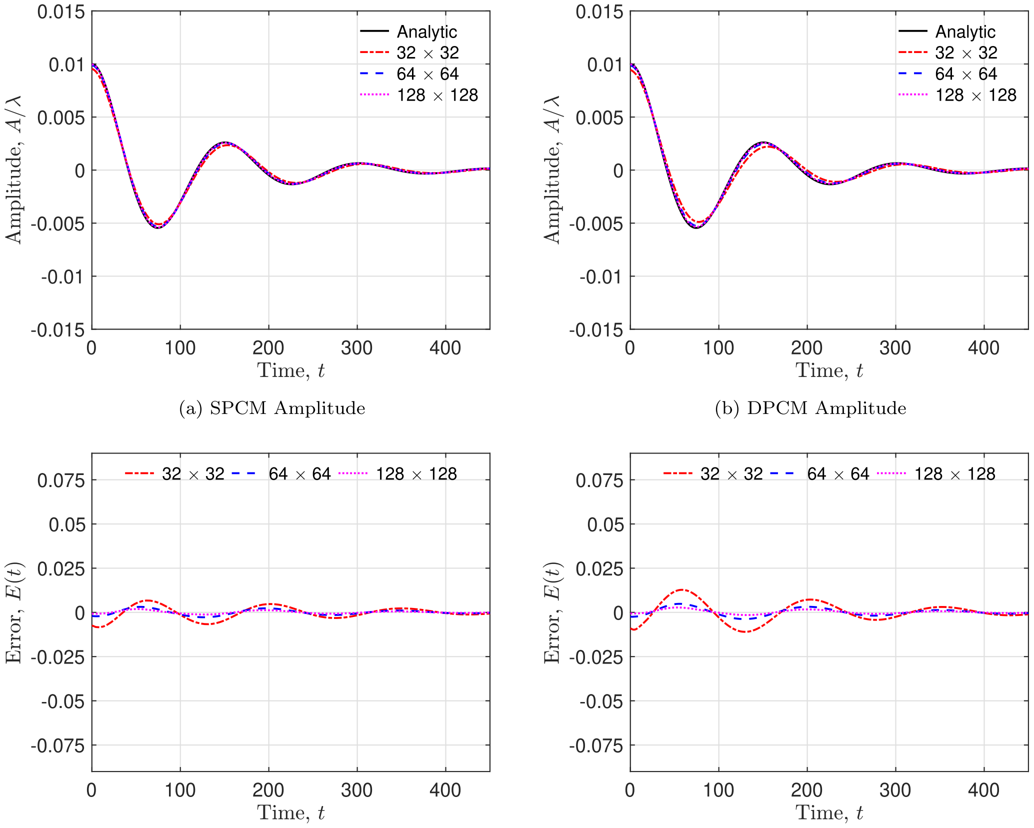
Results for the high density ratio case, where $\rho_1/\rho_2 = 1000$, are shown in Fig. 11. In this simulation we see even better agreement with analytic results than in the previous case. With mesh refinement, the solution rapidly converges to that of Eq. (42). Looking also at the comparison to the SPCM, we see they are somewhat different as the SPCM has a maximum error of $E(t) \approx 0.0084$, while for the DPCM $E(t) \approx 0.0128$ for the 32×32 mesh. Of course, this solution difference could be minimized further by increasing the number of iterations. But again, the trade-off is at a greater computational expense. It is perhaps more useful to refine the mesh for a better solution given the convergence offerings (e.g. $E(t) \approx 0.0028$ for the 128×128 mesh with DPCM vs. $E(t) \approx 0.0019$ with SPCM).

4.2.2. Stochastic surface wave

The purpose of the damped surface wave case is to test the ability of the solver to accurately predict situations where viscosity and surface tension forces interact. To test the ability of the stochastic solver to handle this interaction with uncertainty present, we will now assume there is some uncertainty about the viscosity of the fluids. As the analytic solution assumes equal viscosity fluids, we will maintain that assumption, but impose a uniform distribution. Again we will look at both a unity and high density ratio test case to determine the ability of the solver. For the equal density case we define the kinematic viscosity with $\nu(\zeta) = 0.064720863 + 0.0323604315\zeta$. In the high density ratio case we use $\nu(\zeta) = 0.0064720863 + 0.00323604315\zeta$.

Fig. 12 displays the expected and numerical results of the stochastic equal density case. This simulation was run with order $N = 5$ given the low amount of movement about the interface. For reference, the case of $A(t, \zeta = 0)$ is the same case presented in the deterministic section, where viscosity $\nu(\zeta = 0) = 0.064720863$. For this particular solution, we see very similar results to those found by the deterministic model, and it appears much of the error results from a phase difference. Note, all amplitude values are found during the course of a single simulation of the stochastic solver.

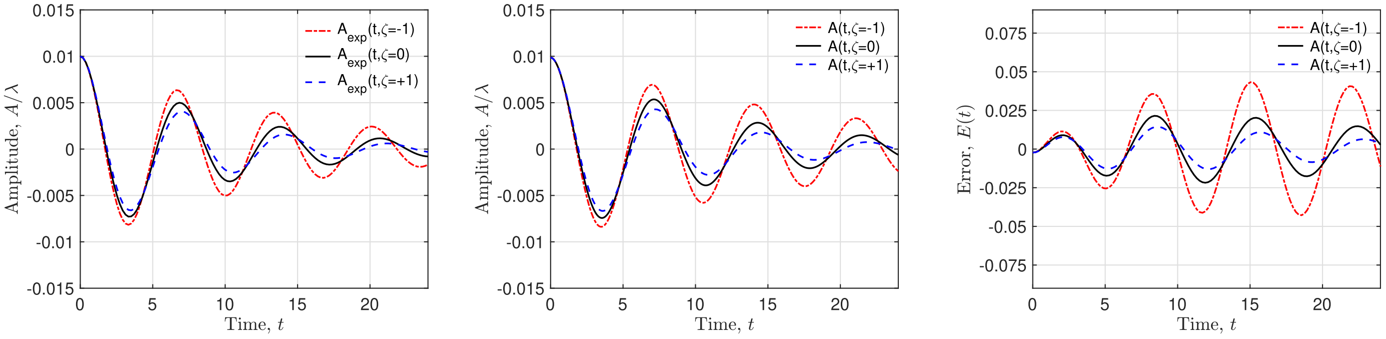
Applying an uncertain viscosity to the case of the high density ratio ($\rho_1/\rho_2 = 1000$) we find similar behavior to that shown in the deterministic model runs. Fig. 13 displays expected and numerical results for a UQ case with $N_m = 6$, $N_P = 3$, and $N = 5$. As with the deterministic solutions, the DPCM methodology appears to be more closely aligned to analytic results than the unity density ratio case. Additionally, it appears the case of the largest viscosity has the smallest deviation from analytic expectations.



(a) SPCM Amplitude (b) DPCM Amplitude

(c) SPCM Error (d) DPCM Error

Fig. 11. Amplitude of a damped surface wave with a density ratio of $\rho_1/\rho_2 = 1000$ for three different mesh sizes with both pressure correction methods. Iteration counts for DPCM solutions are $N_m = 6$ and $N_P = 3$. Analytic solution described by Prosperetti [23] shown with the solid black line.



(a) Analytic Amplitude (b) Numerical Amplitude (c) Error

Fig. 12. Amplitude of a damped surface wave with a density ratio of $\rho_1/\rho_2 = 1$ for a 64×64 mesh with $N_m = 6$, $N_P = 3$, and $N = 5$. Analytic solutions for 3 possible viscosities as described by Prosperetti [23] shown at left.

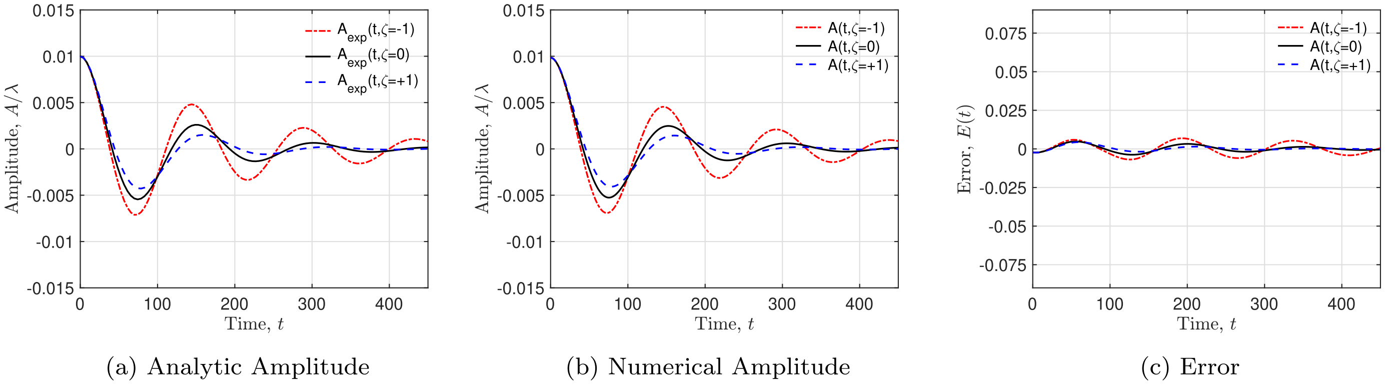
4.3. Atomizing jet

To highlight the ability of the proposed solver to solve complex and potentially real-world situations, we present the results of 2-D deterministic and stochastic atomizing jets. We compare the velocity field found with the DPCM to that of the standard method, the location and shape of the interface, as well as the difference between the pressure fields found by each.

For the deterministic jet, the fluid characteristics include kinematic viscosities of $\nu_1 = 0.01\ \frac{\text{cm}^2}{\text{s}}$ and $\nu_2 = 0.15\ \frac{\text{cm}^2}{\text{s}}$, densities of $\rho_1 = 1.0\ \frac{\text{g}}{\text{cm}^3}$ and $\rho_2 = 0.001\ \frac{\text{g}}{\text{cm}^3}$, a surface tension coefficient of $\sigma = 72.8\ \frac{\text{g}}{\text{s}^2}$, and an incoming velocity of $1000\ \frac{\text{cm}}{\text{s}}$ occurring in a 1 cm^2 domain. Given the incoming diameter of the inlet tube is $D_H = 0.1$ cm, the incoming Reynold's number is $\text{Re}_D = 1 \times 10^4$.

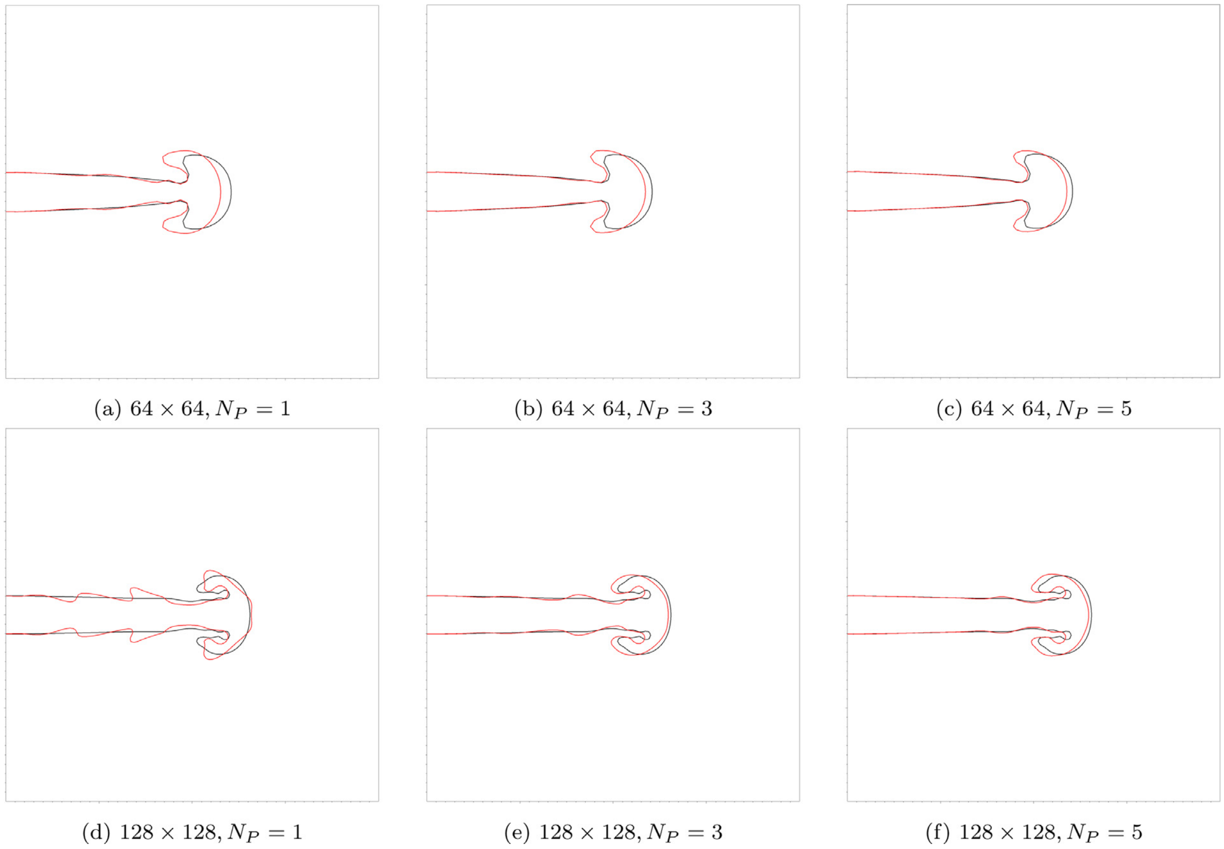
4.3.1. Comparison to traditional pressure correction

We first compare the interface locations of the two methods through time for a 2-D atomizing jet. As previously mentioned, a reasonable method for improving the estimate of the pressure field $\hat{P}$ is to iterate over the pressure Poisson equation. For this scenario, we have set CFL= 0.75 to maintain stability. As shown in Fig. 14, the location of the interface can be significantly different when $\hat{P} \neq P^{n+1}$. As this quantity is improved by increasing N_P, the interface location converges to the traditional pressure correction method. However, we also see the diminishing gains by only increasing N_P.



(a) Analytic Amplitude (b) Numerical Amplitude (c) Error

Fig. 13. Amplitude of a damped surface wave with a density ratio of $\rho_1/\rho_2 = 1000$ for a 64 × 64 mesh with $N_m = 6$, $N_P = 3$, and $N = 5$. Analytic solutions for 3 possible viscosities as described by Prosperetti [23] shown at left.



(a) $64 \times 64, N_P = 1$ (b) $64 \times 64, N_P = 3$ (c) $64 \times 64, N_P = 5$

(d) $128 \times 128, N_P = 1$ (e) $128 \times 128, N_P = 3$ (f) $128 \times 128, N_P = 5$

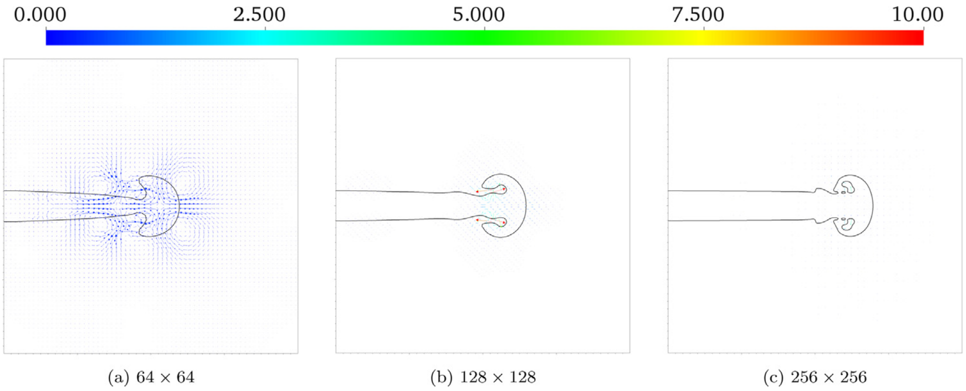
Fig. 14. Interface location as a function of the number of pressure iterations for 2 different mesh sizes of a 2-D jet at time $t = 0.0015$ s. Black and red lines indicate the interface location for the traditional and density decoupled correction methods, respectively. All simulations are run with $N_m = 2$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Next, utilizing the combination $N_m = 6$ and $N_P = 3$ suggested previously, we look at a direct comparison of the difference between the velocity fields of the two pressure solution methods, where $\boldsymbol{u}_{\text{diff}} = \boldsymbol{u}_{\text{SPCM}} - \boldsymbol{u}_{\text{DPCM}}$ in a deterministic simulation. Fig. 15 displays the difference in the velocity fields between SPCM and DPCM. Peak values occur in the coarsest mesh at a few cell locations, with a difference of ≈ 1%. This error diminishes as the mesh is refined, as shown in the figure.

4.3.2. Stochastic jet

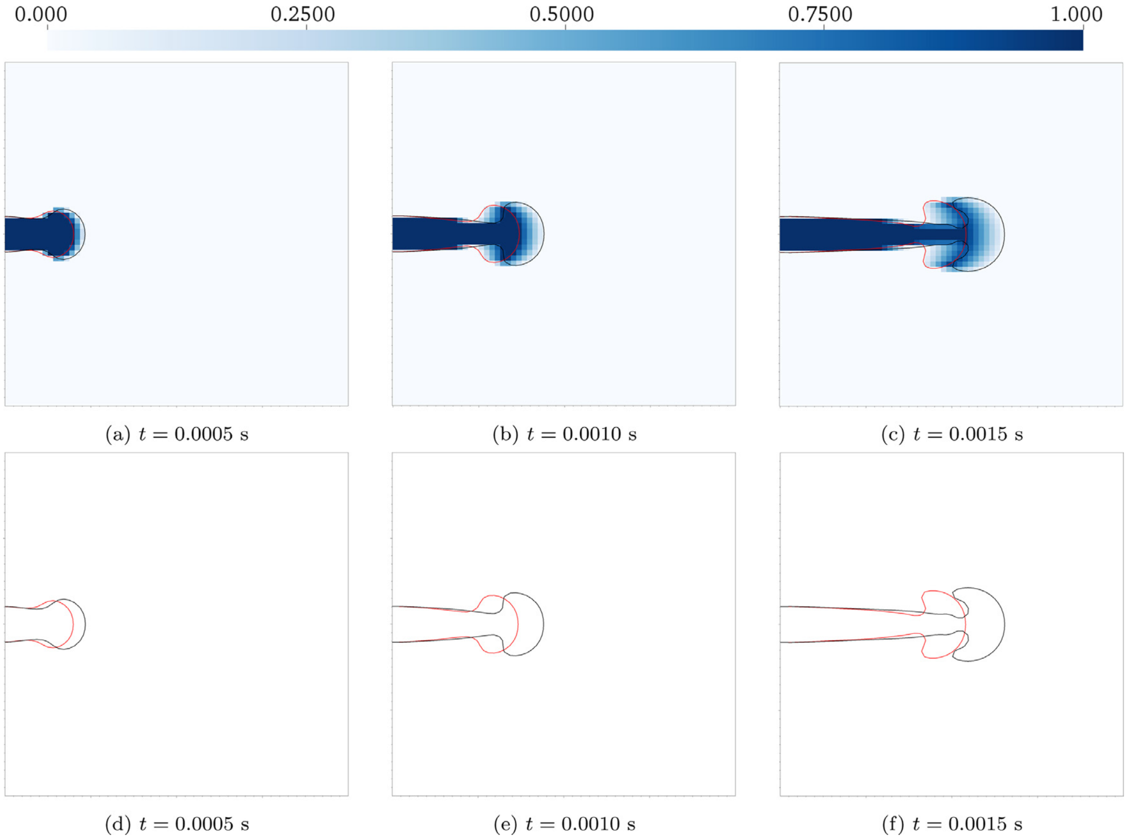
Since the goal of utilizing the DPCM is to improve the efficiency of stochastic models, we now look at a case involving uncertainty. To showcase the ability of the method to resolve a system with a great deal of uncertainty, we present a case with uncertainty about the velocity of an incoming jet. For simplicity, we utilize again a uniform distribution, where $u_{\text{in}} = (1000 + 100\zeta)$ cm/s, or a variation of ±10%. All other fluid parameters remain the same.

Fig. 16 depicts the range of solutions given this degree of uncertainty about the incoming fluid. We also see the results of two deterministic solutions for comparison. As shown, the effect of only a 10% variation in fluid velocity results in a rather dramatic difference in the possible solutions. Also shown is the probability of being in the liquid phase. With this map we can see regions where the fluid phase is known, and others where the interface is uncertain.



(a) 64 × 64 (b) 128 × 128 (c) 256 × 256

Fig. 15. Velocity field difference, $\boldsymbol{u}_{\text{diff}}$, between traditional and proposed pressure solver for 3 different mesh sizes of a 2-D atomizing jet at time $t = 0.0005$ s.



(a) $t = 0.0005$ s (b) $t = 0.0010$ s (c) $t = 0.0015$ s

(d) $t = 0.0005$ s (e) $t = 0.0010$ s (f) $t = 0.0015$ s

Fig. 16. At top is the solution of a stochastic atomizing jet with uncertainty about incoming velocity on a 64^2 grid over time. Bounding solutions are shown for fastest (black line) and slowest (red dashed) incoming velocities. Color map indicates the probability of being in the liquid phase. At bottom are results of two deterministic simulations for comparison. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5. Conclusions

We have presented a modified pressure correction method, extending the density decoupled approach of Dodd and Ferrante [17] to an intrusive stochastic multiphase solver. Deviations from the standard pressure correction method arise due to differences between the estimated pressure $\hat{P}$ and the actual pressure P^{n+1}. We propose a method to reduce these deviations by imposing a semi-Lagrangian extrapolation method for a better initial estimate of $\hat{P}$, then further improve $\hat{P}$ by iterating over the pressure solver. Results from an oscillating droplet, surface wave, and atomizing jet show convergence of the decomposed pressure correction approach to that of the standard pressure correction method.

Given convergence of the DPCM to SPCM, we discussed an efficient combination of pressure and midpoint iterations which significantly reduced computational cost over the SPCM with little added error. Due to the iterative approach to convergence, some deviation threshold must be decided to limit the number of iterations taken. We then balance accuracy and computational cost, which is typical of numerical solutions in any system of differential equations. Other possibilities are also reported which significantly speed up simulation times while shrinking deviations from the traditional pressure correction as mesh is refined.

The decomposed pressure correction methodology for application to stochastic gas-liquid multiphase systems reduces computational cost dramatically. Decoupling the pressure Poisson equations allows for the use of more sophisticated linear solvers. As a result, we see growing computational cost improvement as the basis order, mesh, and uncertainty increase.

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.

CRediT authorship contribution statement

Brian Turnquist: Writing - original draft, Writing - review & editing, Methodology, Software, Conceptualization. **Mark Owkes:** Writing - review & editing, Supervision, Funding acquisition, Conceptualization.

Acknowledgments

This material is based upon work supported by the National Science Foundation under grant nos. 1511325 and 1749779. Computational efforts were performed on the Hyalite High-Performance Computing System, operated and supported by University Information Technology Research Cyberinfrastructure at Montana State University.

Appendix A. Computational cost

The computational cost was assessed with the oscillating droplet test on a 64^2 mesh over a range of midpoint and pressure iterations. A single run of the SPCM was used to test against for average kinetic energy error, $\bar{E}_{\mathrm{diff}}$. All tests were run out to 3 oscillations for tracking of numerical dissipation. Total pressure steps taken throughout the simulation were also tracked, as well as the average number of pressure iterations per time step taken to solve the pressure Poisson equation. The results of the efficiency study are outlined in Table A.2 below.

Table A2
Simulation run times and associated metrics for finding the best combination of N_c and N_P. The error is the average kinetic energy difference between a DPCM simulation and the SPCM. A 95% confidence interval is calculated about the error i.e. ($\bar{\boldsymbol{E}}_{\mathrm{diff}} \pm \mathrm{CI}$). Highlighted rows indicate best possibilities.

N_m	$\boldsymbol{N_P}$	Simulation time (min)	Total pressure iterations	Average iterations/step	Error ($\bar{\boldsymbol{E}}_{\mathrm{diff}}$)	95% Confidence interval
2	1	0.279	40,610	54	0.176	0.0110
3	1	0.399	60,932	81	0.134	0.0082
4	1	0.481	81,219	108	0.100	0.0062
5	1	0.590	101,489	135	0.084	0.0051
6	1	0.685	121,777	162	0.067	0.0041
7	1	0.813	142,039	189	0.062	0.0037
2	2	0.389	81,287	108	0.159	0.0095
3	2	0.506	121,865	162	0.116	0.0068
4	2	0.663	162,400	217	0.080	0.0049
5	2	0.806	202,945	271	0.064	0.0037
6	2	0.958	243,450	325	0.046	0.0028
7	2	1.125	283,963	379	0.041	0.0024
2	3	0.471	121,894	163	0.152	0.0096
3	3	0.643	182,767	244	0.104	0.0063
4	3	0.865	243,456	325	0.065	0.0041
5	3	1.053	304,247	406	0.050	0.0030
6	3	1.269	364,985	487	0.034	0.0021
7	3	1.453	425,795	568	0.031	0.0018
2	4	0.524	162,594	217	0.146	0.0095
3	4	0.783	243,558	325	0.093	0.0057
4	4	1.041	324,518	433	0.054	0.0035
5	4	1.304	405,577	541	0.041	0.0025
6	4	1.548	486,559	649	0.026	0.0017
7	4	1.753	567,620	757	0.024	0.0014
2	5	0.653	203,240	271	0.140	0.0092
3	5	0.987	304,346	406	0.083	0.0052
4	5	1.244	405,613	541	0.047	0.0031
5	5	1.481	506,926	676	0.035	0.0021
6	5	1.752	608,150	811	0.021	0.0014
7	5	2.103	709,467	946	0.020	0.0012
2	6	0.701	243,772	325	0.133	0.0089
3	6	1.069	365,160	487	0.074	0.0047
4	6	1.415	486,678	649	0.040	0.0027
5	6	1.719	608,191	811	0.030	0.0018
6	6	2.056	729,727	973	0.017	0.0012
7	6	2.355	851,252	1135	0.016	0.0010
2	7	0.874	284,329	379	0.126	0.0086
3	7	1.257	425,972	568	0.067	0.0042
4	7	1.622	567,710	757	0.035	0.0024
5	7	1.964	709,466	946	0.026	0.0016
6	7	2.370	851,239	1135	0.015	0.0011
7	7	2.664	993,080	1324	0.014	0.0009

References

[1] Metropolis N, Ulam S. The Monte Carlo method. J Am Stat Assoc 1949;44(247):335–41.

[2] Malik M, Zang T, Hussaini M. A spectral collocation method for the Navier–Stokes equations. J Comput Phys 1985;61(1):64–88. doi:10.1016/0021-9991(85)90061-0.

[3] Hosder S, Walters R, Perez R. A non-intrusive polynomial chaos method for uncertainty propagation in CFD simulations. 44th AIAA Aerospace sciences meeting and exhibit. aerospace sciences meetings. American Institute of Aeronautics and Astronautics; 2006. doi:102514/62006-891.

[4] Wiener N. The homogeneous chaos. Am J Math 1938;60(4):897–936. doi:10.2307/2371268.

[5] Karhunen K. Über lineare methoden in der wahrscheinlichkeitsrechnung, 37. Sana; 1947.

[6] Loeve M.. Probability theory: foundations, random sequences1955.

[7] Le Maétre O, Knio OM, Najm HN, Ghanem RG. A stochastic projection method for fluid flow. J Comput Phys 2001;173(2):481–511. doi:10.1006/jcph.2001.6889.

[8] Le Maétre O, Reagan MT, Najm HN, Ghanem RG, Knio OM. A stochastic projection method for fluid flow: II. Random process.. J Comput Phys 2002;181(1):9. http://search.ebscohost.com.proxybz.lib.montana.edu/login.aspx?direct=true&db=a9h&AN=8513921&login.asp&site=ehost-live

[9] Xiu D, Karniadakis GE. Modeling uncertainty in flow simulations via generalized polynomial chaos. J Comput Phys 2003;187(1):137–67. doi:10.1016/j.advwatres.2013.10.003.

[10] Sochala P, Le Maître O. Polynomial chaos expansion for subsurface flows with uncertain soil parameters. Adv Water Resour 2013;62:139–54. doi:10.1016/j.advwatres.2013.10.003.

[11] El-Beltagy MA, Wafa MI. Stochastic 2D incompressible Navier–Stokes solver using the vorticity-stream function formulation.. J Appl Math 2013:1–14. http://search.ebscohost.com.proxybz.lib.montana.edu/login.aspx?direct=true&db=a9h&AN=95251122&site=ehost-live

[12] Turnquist B, Owkes M. multiUQ: an intrusive uncertainty quantification tool for gas-liquid multiphase flows. J Comput Phys 2019;399:108951. doi:10.1016/j.jcp.2019.108951.

[13] Chorin AJ. Numerical solution of the Navier–Stokes equations. Math Comput 1968;22(104):745–62.

[14] Van Kan J. A second-order accurate pressure-correction scheme for viscous incompressible flow. SIAM J Sci Stat Comput 1986;7(3):870–91.

[15] Thomadakis M, Leschziner M. A Pressure-correction method for the solution of incompressible viscous flows on unstructured grids. Int J Numer Methods Fluids 1996;22(7):581–601. doi:10.1002/(SICI)1097-0363(19960415)22:7<581::AID-FLD365>3.0.CO;2-R.

[16] Dong S, Shen J. A time-stepping scheme involving constant coefficient matrices for phase-field simulations of two-phase incompressible flows with large density ratios. J Comput Phys 2012;231(17):5788–804. doi:10.1016/j.jcp.2012.04.041.

[17] Dodd MS, Ferrante A. A fast pressure-correction method for incompressible two-fluid flows. J Comput Phys 2014;273:416–34. doi:10.1016/j.jcp.2014.05.024.

[18] Brackbill J, Kothe D, Zemach C. A continuum method for modeling surface tension. J Comput Phys 1992;100(2):335–54. doi:10.1016/0021-9991(92)90240-Y.

[19] Tryggvason G, Scardovelli R, Zaleski S. Direct numerical simulations of gas–liquid multiphase flows. Cambridge University Press; 2011.

[20] Chow E, Cleary A, Falgout R. Design of the Hypre Preconditioner Library. Tech. Rep.. Lawrence Livermore National Lab, CA (US); 1998.

[21] Cifani P. Analysis of a constant-coefficient pressure equation method for fast computations of two-phase flows at high density ratios. J Comput Phys 2019;398:108904. doi:10.1016/j.jcp.2019.108904.

[22] Lord Rayleigh FRS. VI. On the capillary phenomena of jets. Proc R Soc Lond 1879;29(196–199):71–97. doi:10.1098/rspl.1879.0015.

[23] Prosperetti A. Motion of two superposed viscous fluids. Phys Fluids 1981;24(7):1217–23. doi:10.1063/1.863522.

[24] Desjardins O, Moureau V, Pitsch H. An accurate conservative level set/ghost fluid method for simulating turbulent atomization. J Comput Phys 2008;227(18):8395–416. doi:10.1016/j.jcp.2008.05.027.

[25] Owkes M, Desjardins O. A discontinuous Galerkin conservative level set scheme for interface capturing in multiphase flows. J Comput Phys 2013;249:275–302. doi:10.1016/j.jcp.2013.04.036.

[26] Garrick DP, Owkes M, Regele JD. A finite-volume HLLC-based scheme for compressible interfacial flows with surface tension. J Comput Phys 2017;339:46–67. doi:10.1016/j.jcp.2017.03.007.

[27] Bellotti T, Theillard M. A coupled level-set and reference map method for interface representation with applications to two-phase flows simulation. J Comput Phys 2019;392:266–90. doi:10.1016/j.jcp.2019.05.003.

[28] Fyfe D, Oran E, Fritts M. Surface tension and viscosity with lagrangian hydrodynamics on a triangular mesh. J Comput Phys 1988;76(2):349–84. doi:10.1016/0021-9991(88)90147-7.

[29] Salih A., Ghosh Moulic S.. Oscillation of a Liquid Drop in a Zero-Gravity Environment - A Benchmark Problem for Two-Phase Flow Computations. Roorkee, India; 2002,.

[30] Herrmann M. A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids. J Comput Phys 2008;227(4):2674–706. doi:10.1016/j.jcp.2007.11.002.

[31] Popinet S, Zaleski S. A front-tracking algorithm for accurate representation of surface tension. Int J Numer Methods Fluids 1999;30(6):775–93. doi:10.1002/(SICI)1097-0363(19990730)30:6⟨775::AID-FLD864⟩3.0.CO;2-#.

[32] McCaslin JO, Desjardins O. A localized re-initialization equation for the conservative level set method. J Comput Phys 2014;262:408–26. doi:10.1016/j.jcp.2014.01.017.