Journal of Computational Physics 343 (2017) 186–200

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A reformulation of the conservative level set reinitialization equation for accurate and robust simulation of complex multiphase flows

Robert Chiodi *, Olivier Desjardins

Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853, USA

ARTICLE INFO

Article history:
Received 22 July 2016
Received in revised form 6 March 2017
Accepted 20 April 2017
Available online 3 May 2017

Keywords:
Multiphase flows
Incompressible flow
Conservative level set
Reinitialization
Accurate conservative level set

ABSTRACT

This paper presents an accurate and robust reinitialization equation for the conservative level set that does not significantly deform stationary surfaces. The compression and diffusion term of the reinitialization equation are reformulated to use a distance level set directly mapped from the conservative level set. The normals are calculated using a distance level set reconstructed from the interface using a fast marching method, increasing robustness and allowing the use of high order, non-TVD transport schemes. Using this new reinitialization equation, we present results for canonical test cases, such as Zalesak's disk and spurious currents, which show significant improvement. A simulation of a liquid–gas jet with $Re = 5000$ is also presented to demonstrate the volume conservation properties of the method in more complex flows.

1. Introduction

Multiphase flows are ubiquitous in nature and exist in many engineering applications, making their simulation very important. However, there are also many numerical difficulties associated with simulating multiphase flows that are not present in single fluid flows. With the presence of multiple phases comes the requirement to track the interface between the phases. Across this interface exists discontinuities in the physical properties of the phases and a singular force due to surface tension, both of which require special consideration when solving the equations in a discrete manner.

Level set methods [1] are often used to track this interface by representing it with a scalar field. The original level set method used a signed distance field, $\phi(\mathbf{x}, t)$, defined as the distance from $\mathbf{x}$ to the nearest point on the interface Γ at time t. This scalar field is advected with the flow velocity, $\mathbf{u}$, to effectively transport the interface with

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0 \,. \tag{1}$$

Through choosing a sign for the distance based on the local phase, the interface becomes implicitly represented as the zero iso-contour of ϕ. The transport will lead to errors in ϕ, requiring a reinitialization equation to reshape ϕ into a proper distance function, for which $|\nabla \phi| = 1$. Neither the transport, Eq. (1), nor the reinitialization equation, are conservative, which leads to significant volume conservation errors, requiring additional attention and techniques to improve volume conservation. This is a key issue with the original level set methods, preventing their use for a wide class of problems, such

* Corresponding author.
E-mail address: rmc298@cornell.edu (R. Chiodi).

http://dx.doi.org/10.1016/j.jcp.2017.04.053
0021-9991/

as turbulent atomization. For the remainder of this paper, these original level set methods, which transport and reinitialize the signed distance field $\phi(\mathbf{x}, t)$, will be referred to as distance level set (DLS) methods.

In order to better conserve the enclosed volume, Olsson and Kreiss [2] and Olsson et al. [3] proposed using a conservative level set (CLS) function, instead of ϕ, to implicitly track the interface. This function is defined as

$$\psi(\mathbf{x}, t) = \frac{1}{2}\left(\tanh\left(\frac{\phi(\mathbf{x}, t)}{2\epsilon}\right) + 1\right), \tag{2}$$

where ψ is the CLS and ϵ is a free parameter that determines the interface thickness. With this mapping from ϕ, the interface is now represented by the 0.5 iso-contour of ψ.

The conservative level set function, Eq. (2), can be seen as a smooth approximation of the sharp Heaviside function, $H(\phi)$, which has a value of 1 in one phase ($\phi > 0$), and 0 in the other ($\phi < 0$). As is clear, the sharper the hyperbolic tangent profile, i.e., the smaller the value of ϵ, the better the approximation of $H(\phi)$. The exact enclosed volume is given by the integration of the Heaviside function,

$$\mathcal{V}_l = \int H(\phi) d\mathcal{V} = \int H\left(\psi - \frac{1}{2}\right) d\mathcal{V}. \tag{3}$$

Similarly, the approximation of the enclosed volume using the conservative level set function is given by

$$\tilde{\mathcal{V}}_l = \int \psi d\mathcal{V}, \tag{4}$$

with $\lim_{\epsilon \to 0} \tilde{\mathcal{V}}_l = \mathcal{V}_l$. A conservative discretization of the ψ equations (transport and reinitialization) leads to exact conservation of $\tilde{\mathcal{V}}_l$, but only approximate conservation of $\mathcal{V}_l$. The capability of the method to conserve enclosed volume, i.e. the accuracy with which $\tilde{\mathcal{V}}_l$ approximates $\mathcal{V}_l$, is a central focus of this paper.

Assuming incompressibility, ψ is transported, using a conservative finite volume discretization, as

$$\frac{\partial \psi}{\partial t} + \nabla \cdot (\mathbf{u}\psi) = 0, \tag{5}$$

and, after transporting, reinitialized back to its proper hyperbolic tangent profile, with a conservative finite volume discretization, through

$$\frac{\partial \psi}{\partial \tau} = \nabla \cdot (\epsilon (\nabla \psi \cdot \mathbf{n}) \mathbf{n} - \psi (1 - \psi) \mathbf{n}), \tag{6}$$

where $\mathbf{n}$ is the interface normal vector calculated using

$$\mathbf{n} = \frac{\nabla \psi}{|\nabla \psi|}. \tag{7}$$

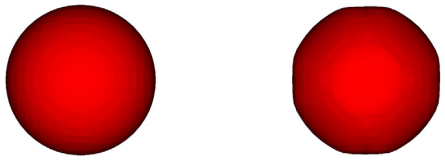
With ϵ controlling the interface profile thickness, and a smaller value of ϵ resulting in better volume conservation properties, it is desirable to set ϵ to a small value, typically $\epsilon = \Delta x/2$ [2]. This causes sharp gradients, which can produce overshoots and undershoots of ψ, causing $\mathbf{n}$ to switch directions when calculated with Eq. (7). Olsson and Kreiss [2] handled this problem through the use of low order, total variation diminishing (TVD) schemes for transport to prevent the creation of new local extrema, however, this significantly reduces the overall accuracy of the method. This method will be referred to as the conservative level set (CLS) method.

In order to alleviate the need to use TVD transport schemes, the accurate conservative level set (ACLS) method [4] uses a fast marching method (FMM) [5] to reconstruct a signed distance function, ϕ_{FMM}, from the 0.5 iso-contour of ψ, and calculates the normals using ϕ_{FMM} as

$$\mathbf{n} = \frac{\nabla \phi_{\text{FMM}}}{|\nabla \phi_{\text{FMM}}|}. \tag{8}$$

This prevents oscillatory transport errors in ψ from spuriously impacting the orientation of normals, removing the prior need to use TVD transport schemes. Because of this, inexpensive and high order accurate schemes can be used. At the same resolution, the ACLS method can therefore track finer interfacial structures than the original CLS method. Additionally, it was found that the use of high order transport schemes significantly improves volume conservation [4].

The reinitialization of ψ using Eq. (6), which is used in both the CLS method [2,3] and the ACLS method [4], can still be a significant source of numerical errors. As an example, Fig. 1 shows a spherical drop resolved by $D/\Delta x = 20$, reinitialized in place with Eq. (6), updating the normals after each iteration. Without numerical dissipation from transport, errors from the reinitialization equation accumulate and lead to significant deformation of the interface, as demonstrated in Fig. 1(b), showing the deformation of the sphere after 1000 iterations. The obvious solution would be to reduce or eliminate



(a) Interface of initialized sphere. (b) After 1000 iterations of reinitialization.

Fig. 1. Example of the deformation on stationary interfaces due to reinitialization errors.

reinitialization, however, this is not an appropriate response for simulations with interfaces that have both active (moving and deforming) and inactive (static) regions.

In order to prevent accumulating errors on inactive regions of the interface while fully reinitializing active regions, McCaslin and Desjardins [6] proposed a modification to the reinitialization equation that locally scales the amount of reinitialization based on the local strain rate tensor and an estimate of numerical diffusion derived from the interface transport velocity. While effectively reducing reinitialization errors on stationary interfaces, and retaining the conservative discretization of Eq. (6), the implementation of their method leads to a more complex reinitialization equation that introduces two new free parameters.

An alternate solution is to directly reduce the amount of errors due to reinitialization. Shukla et al. [7] use a gradient form of the reinitialization equation,

$$\frac{\partial \psi}{\partial \tau} = \mathbf{n} \cdot \nabla \left(\epsilon |\nabla \psi| - \psi (1 - \psi) \right) , \tag{9}$$

and a direct mapping of ϕ,

$$\phi_{\text{map}} = \epsilon \ln \left(\frac{\psi}{1 - \psi} \right) \tag{10}$$

to show that classical distance level set reinitialization [8] can replace the conservative level set reinitialization equation. Please note, to ensure Eq. (10) does not take values of ∞ or $-\infty$, ψ is limited to be in the exclusive range $(0, 1)$. Solving the classical distance level set reinitialization equation with the smoother ϕ_{map}, for which many methods exist, greatly reduces errors and can then be mapped back to ψ by inverting Eq. (10), however, this approach does not have a conservative discretization and, consequently, poorly conserves the initially enclosed volume.

Recently, Wacławczyk [9] proposed a similar idea to Shukla et al. [7] in the context of the divergence form of the reinitialization equation, which we prefer due to its superior conservation properties. Using Eq. (10) to obtain ϕ_{map} from ψ, Wacławczyk [9] rewrites the diffusion term from Eq. (6), $\nabla \cdot (\epsilon (\nabla \psi \cdot \mathbf{n}) \mathbf{n})$. The CLS reinitialization equation then takes the new form of

$$\frac{\partial \psi}{\partial \tau} = \nabla \cdot \left[\psi (1 - \psi) \left(\left| \nabla \phi_{\text{map}} \cdot \mathbf{n} \right| - 1 \right) \mathbf{n} \right] , \tag{11}$$

with $\mathbf{n}$ calculated as

$$\mathbf{n} = \frac{\nabla \phi_{\text{map}}}{|\nabla \phi_{\text{map}}|} . \tag{12}$$

In our experience, however, we have found that this approach suffers from the same sensitivity to oscillations during transport as the original CLS method, and therefore requires TVD transport.

The purpose of this paper is to modify the reinitialization equation presented in Wacławczyk [9] to create a robust and accurate reinitialization equation in the context of the ACLS method. First, we will present our modifications to Eq. (11) that greatly improve its robustness while retaining the low levels of error. Results from typical interface tracking test cases such as Zalesak's disk [10], the deformation of a circle [11], and a testing of spurious currents will be shown. Lastly, the new form of the reinitialization equation will be used alongside our flow solver to simulate the injection of a turbulent liquid jet into quiescent air.

2. Modifications to the reinitialization equation

As stated before, the reinitialization equation presented by Wacławczyk [9], Eq. (11), is not robust when using high order non-TVD transport schemes to transport ψ. To demonstrate, we will use the Zalesak disk test case [10], with ψ transported using a 5th order high order upstream central (HOUC-5) scheme to evaluate the face flux values [12]. This scheme is the optimal estimate from a 5th order weighted essentially non-oscillatory (WENO-5) stencil, used to evaluate ψ at the cell faces.

(a) Initial domain (b) One iteration of reinitialization (c) Five iterations of reinitialization

Fig. 2. Zalesak disk test case, showing results after 10 rotations with shading indicating the value of ψ.

In a $[-0.5, 0.5]^2$ domain, we initialize a slotted disk with a radius of 0.15, notch width, $w = 0.05$, and notch height of 0.25, centered at $(x, y) = (0.0, 0.25)$. The domain is resolved with a 100^2 uniform Cartesian mesh, leading to $w/\Delta x = 5$. This slotted disk is then rotated in the velocity field $(u, v) = (-y, x)$ for 10 rotations around the midpoint of the domain at a convective CFL of 0.48. The reinitialization equation is advanced in pseudo-time with $\Delta\tau = 1.5 \times 10^{-3}$. The direct use of Eq. (11), performing one iteration of reinitialization per transport step, leads to spurious disconnected volumes with $\psi > 0$ in the domain, as shown in Fig. 2(b). This is a direct consequence of errors from reinitialization caused by spurious normal orientation. If instead the reinitialization equation is iterated five times per transport step, the error is seen to worsen significantly in Fig. 2(c). In order to ensure ψ remains bounded and Eq. (11) remains stable, flux limiting is necessary to enforce $\psi(1-\psi) \geq 0$.

To address these issues, after transporting ψ and calculating ϕ_{map} with Eq. (10), we use a FMM [13] to solve the static Eikonal equation for the distance ϕ_{FMM} from the zero iso-contour of ϕ_{map}. The normals are then calculated with Eq. (8) instead of Eq. (12). With the decoupling of the normals from ψ through the use of ϕ_{FMM}, high accuracy non-TVD schemes can now be used, however, the boundedness of ψ is no longer guaranteed. A consequence of this is the possibility of the compressive fluxes, $\psi(1-\psi)$, switching signs. This could be prevented by using flux limiting of $\psi(1-\psi)$. Instead, we propose to reformulate the compressive fluxes to use ϕ_{map}, which during its calculation similarly limits ψ to be in the exclusive range $(0, 1)$. This can be accomplished by using hyperbolic trigonometric identities to rewrite $\psi(1-\psi)$ as

$$\psi(1-\psi) = \frac{1}{4\cosh^2\left(\frac{\phi_{\text{map}}}{2\epsilon}\right)}. \tag{13}$$

While Eq. (13) is an equality, once discretized, each side constitutes a different numerical approximation of the flux, leading to different levels of error. Calculating the compressive flux as $\psi(1-\psi)$ requires the interpolation of ψ to the cell faces. Since ψ is a relatively sharp function with a hyperbolic tangent profile, this interpolation can lead to significant error. By using ϕ_{map} instead, which is a linear and smoothly varying function, the error introduced through interpolation to the cell face is greatly reduced.

Including the remapping of the compressive flux, our proposed form of the reinitialization equation is

$$\frac{\partial\psi}{\partial\tau} = \nabla\cdot\left[\frac{1}{4\cosh^2\left(\frac{\phi_{\text{map}}}{2\epsilon}\right)}\left(\left|\nabla\phi_{\text{map}}\cdot\mathbf{n}\right| - 1\right)\mathbf{n}\right], \tag{14}$$

where $\mathbf{n}$ is calculated using Eq. (8), with the discretization as follows. To present the discretization, we will borrow notation from Desjardins et al. [14], where ζ_i represents the computational space, related to physical space x_i through $h_i = dx_i/d\zeta_i$ and $J = \Pi_{i=1}^{3} h_i$. In Cartesian coordinates, the three components 1, 2, and 3 represent x, y, and z, respectively. We will also define the x_j component of the second order finite difference x_i face gradient of a scalar quantity a as

$$\left(\nabla^{x_i} a\right)_j = \begin{cases} \dfrac{\delta_{2nd} a}{\delta_{2nd} x_j} & \text{for } i = j \\ \dfrac{\delta_{2nd} \bar{a}^{2nd\,x_i}}{\delta_{2nd} x_j} & \text{for } i \neq j \end{cases} \tag{15}$$

where $\delta_{2nd} a/\delta_{2nd} x_j$ is a second order differentiation of variable a in physical space, and $\bar{a}^{2nd\,x_i}$ is the second order interpolation of a in physical space to the x_i face. To calculate the normal, a dynamic optimal stencil is selected to minimize $1 - |\nabla\phi_{\text{FMM}}|$ [15]. For each component of the normal, a 2nd order central, 1st order forward, and 1st order backward difference calculation of $\nabla\phi_{\text{FMM}}$ is performed. The combination among the 27 possibilities (three potential stencils for the three separate components) that minimizes $1 - |\nabla\phi_{\text{FMM}}|$ is then selected to compute the normal. This differentiation with the optimal stencil will be represented as

$$\left(\nabla^{x_i,\mathrm{opt}}a\right)_j = \begin{cases} \dfrac{\delta_{\mathrm{opt}}a}{\delta_{\mathrm{opt}}x_j} & \text{for } i = j \\ \dfrac{\delta_{\mathrm{opt}}\bar{a}^{2nd\,x_i}}{\delta_{\mathrm{opt}}x_j} & \text{for } i \neq j \end{cases} \tag{16}$$

where $\delta_{\mathrm{opt}}a/\delta_{\mathrm{opt}}x_j$ is the differentiation of the scalar a with the found optimal stencil. Using this notation, the normal at each cell face, x_i, with component j, is calculated by

$$n_j^{x_i} = \left(\mathbf{n}^{x_i}\right)_j = \left(\nabla^{x_i,\mathrm{opt}}\phi_{\mathrm{FMM}}\right)_j . \tag{17}$$

This results in an "H" shaped stencil for calculating cell face normals, further discussed in Section 3.4 of Desjardins et al. [4].

Table 1 summarizes the differences between the newly presented method, where the ACLS method [4] is reinitialized by Eq. (14), and other CLS based methods. Using this new form of the reinitialization equation, the same Zalesak disk test as before is performed, with the results shown in Fig. 3 absent of spurious behavior in ψ. Note that motion of the interface will happen with this conservative reinitialization equation, however, this is a desired feature since this is what allows for better volume conservation, i.e., the conservative level set reinitialization equation puts volume conservation ahead of preserving the iso-surface location.

2.1. Implemented reinitialization algorithm

To aid in implementation of this method by others, we have included a basic algorithm for using the ACLS method with the presented form of the reinitialization equation. While this is written for use with a staggered grid finite-volume code, implementation for other frameworks should be straightforward. We have also included a visual schematic of the algorithm in Fig. 4 to aid in understanding the order of steps. Each step is performed once per advection step.

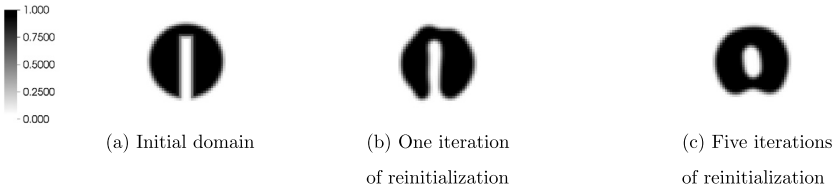
- Calculate the maximum allowable $\Delta\tau$ from the CFL constraint,

$$\mathrm{CFL}_{\mathrm{reinit}} = \max\left(\frac{\Delta\tau_{\mathrm{reinit}}}{\Delta x}, \frac{4\epsilon\,\Delta\tau_{\mathrm{reinit}}}{\Delta x^2}\right) \tag{18}$$

- Solve the transport equation, Eq. (5), updating ψ. We use HOUC-5 [12] in all presented cases for an accurate and efficient transport scheme.
- Map ψ to ϕ_{map} using Eq. (10).
- Use a FMM, such as [13], to solve the static Eikonal equation for the distance ϕ_{FMM} from the zero iso-contour of ϕ_{map}, which implicitly represents the interface. Note that we find the zero iso-contour of ϕ_{map} through linear interpolation of the discrete cell centered ϕ_{map} values.
- Calculate the cell face normals with ϕ_{FMM} and Eq. (8), as discussed in Section 2.
- Update ψ using the reinitialization equation, Eq. (14), the discretization of which is given next in Section 2.1.1. For all remaining results presented in this paper, we use one iteration of the reinitialization equation at $\mathrm{CFL}_{\mathrm{reinit}} = 0.5$. This has been found to be sufficient, as transport at a convection CFL ≤ 1 moves the interface and deforms the profile by at most one cell width [4].

Table 1
Comparison of the method presented in this paper to other methods that use the conservative level set, ψ.

Method	TVD transport	Reinitialization equation	**n** equation
CLS [2,3]	Yes	$\frac{\partial\psi}{\partial\tau} = \nabla\cdot\left(\epsilon\left(\nabla\psi\cdot\mathbf{n}\right)\mathbf{n} - \psi\left(1-\psi\right)\mathbf{n}\right)$	$\frac{\nabla\psi}{\lvert\nabla\psi\rvert}$
ACLS [4]	No	$\frac{\partial\psi}{\partial\tau} = \nabla\cdot\left(\epsilon\left(\nabla\psi\cdot\mathbf{n}\right)\mathbf{n} - \psi\left(1-\psi\right)\mathbf{n}\right)$	$\frac{\nabla\phi_{\mathrm{FMM}}}{\lvert\nabla\phi_{\mathrm{FMM}}\rvert}$
Wacławczyk CLS [9]	Yes	$\frac{\partial\psi}{\partial\tau} = \nabla\cdot\left[\psi(1-\psi)\left(\lvert\nabla\phi_{\mathrm{map}}\cdot\mathbf{n}\rvert - 1\right)\mathbf{n}\right]$	$\frac{\nabla\phi_{\mathrm{map}}}{\lvert\nabla\phi_{\mathrm{map}}\rvert}$
New method	No	$\frac{\partial\psi}{\partial\tau} = \nabla\cdot\left[\frac{1}{4\cosh^2\left(\frac{\phi_{\mathrm{map}}}{2\epsilon}\right)}\left(\lvert\nabla\phi_{\mathrm{map}}\cdot\mathbf{n}\rvert - 1\right)\mathbf{n}\right]$	$\frac{\nabla\phi_{\mathrm{FMM}}}{\lvert\nabla\phi_{\mathrm{FMM}}\rvert}$



(a) Initial domain (b) One iteration of reinitialization (c) Five iterations of reinitialization

Fig. 3. Zalesak disk test case, showing results after 10 rotations with shading indicating the value of ψ.

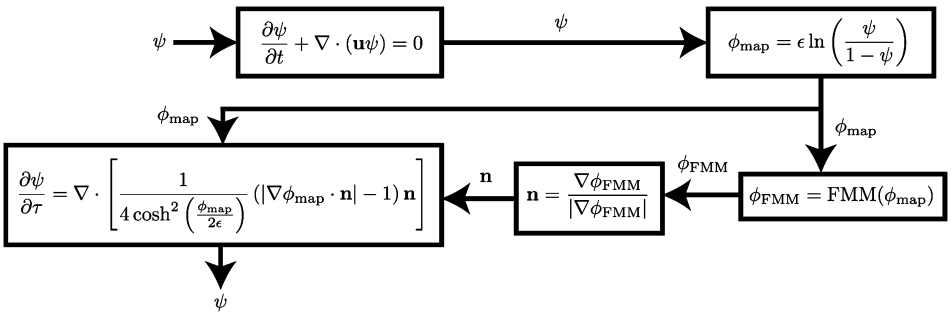


Fig. 4. A visual diagram of the algorithm presented in Section 2.1.

2.1.1. Discretization of the reinitialization equation

The discretization is an important part of the reformulation of the reinitialization equation. To solve Eq. (14) in a finite-volume framework, the flux normal to the cell faces is needed. We will denote this normal flux as F_i. Note that this flux lives on the x_i face of the cell, and is calculated as

$$F_i = \frac{1}{4\cosh^2\left(\frac{\tilde{\phi}_{\text{map}}^{2nd\,x_i}}{2\epsilon}\right)}\left(|(\nabla^{x_i}\phi_{\text{map}})_j n_j^{x_i}| - 1\right) n_i^{x_i}\,, \tag{19}$$

with Einstein notation used only for subscripts. Using these normal fluxes and the notation presented earlier, ψ is updated in pseudo-time, τ, as

$$\psi^{n+1} = \Delta\tau \sum_{i=1}^{3}\left(\frac{1}{J}\frac{\delta_{2nd}}{\delta_{2nd}\zeta_i}\left[\frac{J}{h_i}F_i\right]\right) + \psi^{\star}\,, \tag{20}$$

where $\psi^{\star}$ denotes the CLS field taken after transport but prior to reinitialization.

3. Mathematical formulation of flow solver

3.1. Governing equations

Our in-house flow solver, NGA [14,16], is used for all testing presented in this paper. For the simulations, we use a 2nd order, finite-volume, staggered-grid discretization with semi-implicit Crank–Nicolson time advancement. The momentum is calculated using the incompressible form of Navier–Stokes,

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla p + \nabla \cdot \left(\mu[\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{t}}]\right) + \rho \mathbf{g} \tag{21}$$

where $\mathbf{u}$ is the velocity, ρ is density, p is pressure, μ is dynamic viscosity, and $\mathbf{g}$ is the acceleration due to gravity. With the assumption of incompressibility, the continuity equation is reduced to

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = \frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho = 0\,. \tag{22}$$

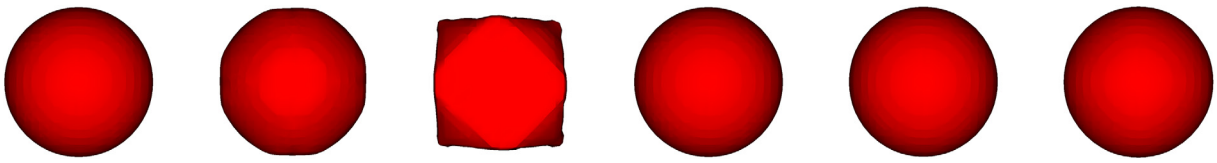
With two phases separated by an interface Γ, the density and viscosity are functions of space and time. Across this interface there is also a discontinuity in density, viscosity, and pressure. The jumps in density and viscosity can be written as $[\rho]_\Gamma = \rho_l - \rho_g$ and $[\mu]_\Gamma = \mu_l - \mu_g$, respectively, where the subscript l and g denote whether it is a liquid or gas property. The pressure jump is written as

$$[p]_\Gamma = \sigma\kappa + 2[\mu]_\Gamma \mathbf{n}^{\mathrm{t}} \cdot \nabla \mathbf{u} \cdot \mathbf{n}\,, \tag{23}$$

where σ is the surface tension coefficient, κ is the interface curvature, and $\mathbf{n}$ is the interface normal. The volume fractions of liquid and gas in each computational cell are used to calculate the density and viscosity for the flow solver through linear and harmonic averaging, respectively.

3.2. Discretization of equations

A detailed overview of the implementation and discretization of Eq. (21) and Eq. (22) in NGA is available in Desjardins et al. [16]. With the ability to maintain an accurate interface due to the new form of the reinitialization equation, the



(a) Eq. 6, Initial droplet | (b) Eq. 6, 10^3 iterations | (c) Eq. 6, 10^4 iterations | (d) Eq. 14, Initial droplet | (e) Eq. 14, 10^3 iterations | (f) Eq. 14, 10^4 iterations

Fig. 5. The droplet interface, resolved by $D/\Delta x = 20$, at initialization, after 10^3 and after 10^4 iterations of the reinitialization equation.

discretization of Eq. (23) and calculation of curvature becomes important, and as such will be described in greater detail here.

We choose to calculate curvature using a quadratic least squares approach, as presented in Marchandise et al. [17]. To do this, we construct an overdetermined quadratic linear system with 10 polynomials considering 125 points from a $5 \times 5 \times 5$ stencil. In practice, we find solving this system in a least squares sense delivers converging curvatures [4]. To localize the system of equations around the point of interest, we use a weighting vector, where each entry is

$$w_i = \exp\left(-\frac{\Delta x_i^2 + \Delta y_i^2 + \Delta z_i^2}{\frac{9}{4}h^2}\right), \tag{24}$$

with h being the characteristic mesh size and Δx_i, Δy_i, Δz_i being the displacement from the point of interest to the current point in the stencil, i. The overdetermined linear system is then solved through least squares with the form $\mathbf{wA^tAx} = \mathbf{wA^tb}$, with the contents of $\mathbf{A}$, $\mathbf{x}$, and $\mathbf{b}$ thoroughly explained in [17]. This can then be solved to obtain $\mathbf{x}$ and correspondingly compute the curvature as

$$\kappa_i = \frac{\mathrm{tr}(\mathbf{H}) - \mathbf{n}^\mathrm{t} \cdot \mathbf{H} \cdot \mathbf{n}}{|\nabla \phi_{\mathrm{FMM}}|}, \tag{25}$$

where $\mathbf{H}$ is the Hessian matrix, defined as $\mathbf{H} = \nabla\nabla\phi_{\mathrm{FMM}}$.

A known problem is that the calculated curvature will be at the cell center, while the curvature is needed at the interface to compute the source term for the pressure Poisson equation [18]. We use harmonic interpolation weighted by ϕ_{FMM} to evaluate the curvature at the interface. Using this ϕ_{FMM} based interpolation alongside the ghost fluid method (GFM) [4,19], allows a compact and accurate calculation of the pressure jump. Within the context of the ACLS method, this amounts to harmonically interpolating the curvature to a zero iso-surface of ϕ_{FMM} that lies between cells $i-1$ and i as

$$\kappa_{\mathrm{interface}} = \kappa_{i-1}\kappa_i \frac{\left(\phi_{\mathrm{FMM},i} - \phi_{\mathrm{FMM},i-1}\right)}{\phi_{\mathrm{FMM},i}\kappa_i - \phi_{\mathrm{FMM},i-1}\kappa_{i-1}}. \tag{26}$$

For the case where $\kappa_i\kappa_{i-1} \leq 0$, we resort to a linear interpolation onto the interface,

$$\kappa_{\mathrm{interface}} = \frac{\kappa_{i-1}\phi_{\mathrm{FMM},i} - \kappa_i\phi_{\mathrm{FMM},i-1}}{\phi_{\mathrm{FMM},i} - \phi_{\mathrm{FMM},i-1}}, \tag{27}$$

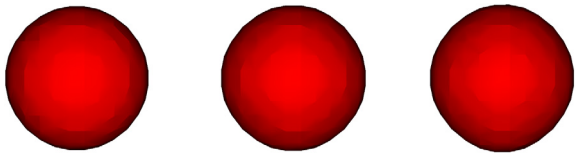
to avoid an erroneous interpolated curvature when the denominator of Eq. (26) approaches zero.

4. Results

4.1. Reinitialization of a stationary droplet

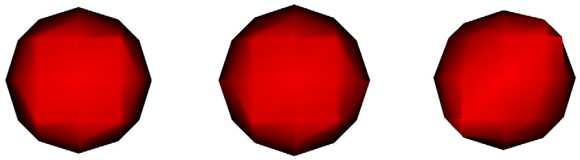
As shown in Fig. 1 in the introduction, the classical form of the reinitialization equation leads to substantial deformation of a stationary droplet. In order to confirm that the new formulation of the reinitialization equation does not suffer from this issue, a stationary droplet with a resolution of $D/\Delta x = 20$ is once again reinitialized in place. Even after 10000 iterations, the new form of the reinitialization equation introduces an insignificant amount of error, with the droplet in Fig. 5(f) only slightly deformed from the initialized droplet shown in Fig. 5(d). This is not the case for the classical form of the reinitialization equation, with Fig. 5(b) noticeably deformed from Fig. 5(a) after 1000 iterations of the reinitialization equation, and severely deformed after 10000 iterations.

In an effort to better understand the reformulated reinitialization equation's performance at various mesh resolutions, the same test is repeated using the reformulated reinitialization equation and a sphere resolved by half the number of cells, with $D/\Delta x = 10$. The result, shown in Fig. 6, displays nearly no deformation after 10000 iterations of the reinitialization equation. Lastly, we test the reformulated reinitialization equation at the low resolution of $D/\Delta x = 4$. Once again, we reinitialize the droplet for 10000 iterations, with the results shown in Fig. 7. At this level of resolution, some slight deformation is noticeable, however, it is still far below that seen for the original reinitialization equation on the five times finer mesh.



(a) $D/\Delta x = 10$, Initial droplet (b) $D/\Delta x = 10$, 10^3 iterations (c) $D/\Delta x = 10$, 10^4 iterations

Fig. 6. Deformation occurring while using the reformulated reinitialization equation on droplet resolved by $D/\Delta x = 10$.



(a) $D/\Delta x = 4$, Initial droplet (b) $D/\Delta x = 4$, 10^3 iterations (c) $D/\Delta x = 4$, 10^4 iterations

Fig. 7. Deformation occurring while using the reformulated reinitialization equation on a minimally resolved droplet, $D/\Delta x = 4$.

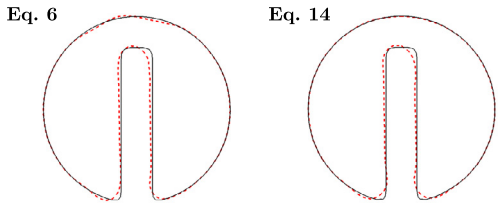


Fig. 8. Comparison of Zalesak disk problem using Eq. (6) or Eq. (14) for the reinitialization equation with a 100^2 mesh. The black line is the initial interface, the dashed red line is after one rotation.

Table 2
The L_1 and L_2 error norms of the ψ field for the Zalesak disk on a 100^2 mesh ($w/\Delta x = 5$) after one rotation, compared to the initial profile. The percent change in enclosed volume is also given.

Reinitialization	Eq. (6)	Eq. (14)
L_1 error	0.0032	0.0022
L_2 error	0.021	0.016
% Vol. change	−0.16	0.020

4.2. Zalesak's disk

The Zalesak's disk test case [10] is used to assess both forms of the reinitialization equations' ability to preserve sharp edges and corners. The parameters used for these simulations are the same as those presented earlier in Section 2, except with $\Delta\tau$ set to result in $\text{CFL}_{\text{reinit}} = 0.5$ according to Eq. (18).

First, one rotation of the disk on a 100^2 uniform Cartesian mesh ($w/\Delta x = 5$) at a CFL of 0.48 is performed. Throughout the entire rotation, both disks are able to preserve the shape of the disk, as shown in Fig. 8. Table 2 shows the percent change in volume enclosed by the interface after one rotation, as well as the L_1 and L_2 errors, where the L_1 and L_2 errors are calculated using all computational cells, N, as

$$L_1 = \frac{1}{\mathcal{V}_D}\sum_{i=1}^{N}\left(\psi_{i,\text{initial}} - \psi_{i,\text{final}}\right)\mathcal{V}_c\,, \tag{28}$$

$$L_2 = \sqrt{\frac{1}{\mathcal{V}_D}\sum_{i=1}^{N}(\psi_{i,\text{initial}} - \psi_{i,\text{final}})^2\mathcal{V}_c}\,, \tag{29}$$

where $\mathcal{V}_D$ is the volume of the domain and $\mathcal{V}_c$ is the volume of a cell. Both the L_1 and L_2 error are similar after one rotation, however, the new form of the reinitialization equation leads to over an order of magnitude better volume conservation.

To study the effect of the reinitialization over long times as a function of mesh resolution, we simulate 10 rotations of the disk at a CFL of 0.48 on meshes of 50^2 ($w/\Delta x = 2.5$), 100^2 ($w/\Delta x = 5$), and 200^2 ($w/\Delta x = 10$). Fig. 9 shows the results

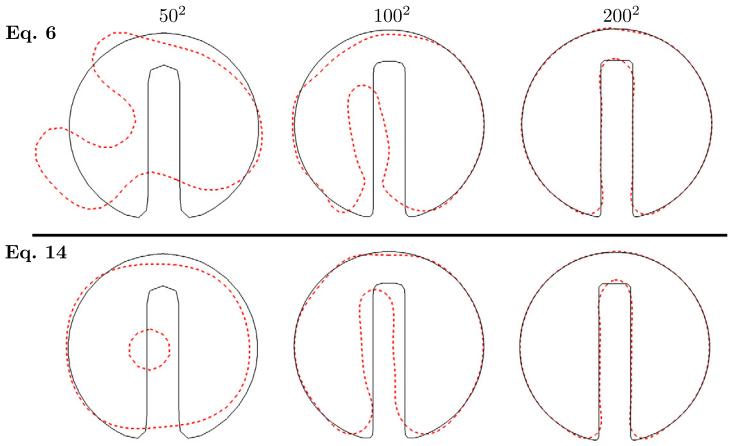


Fig. 9. Comparison of Zalesak disk problem using Eq. (6) or Eq. (14) for the reinitialization equation at three levels of resolution. The black line is the initial interface, the dashed red line is after 10 rotations.

Table 3
The L_1 and L_2 error norms of the ψ field for the Zalesak disk after ten rotations, compared to the initial profile. The percent change in enclosed volume is also given.

Reinitialization	Eq. (6)			Eq. (14)		
Resolution	50^2	100^2	200^2	50^2	100^2	200^2
L_1 error	0.034	0.017	0.0016	0.018	0.0090	0.0015
L_2 error	0.14	0.10	0.016	0.081	0.061	0.017
% Vol. change	−0.96	−0.14	−0.078	2.9	0.064	−0.0014

when using the classical CLS reinitialization equation, Eq. (6), versus the reformulated reinitialization equation, Eq. (14). Additionally, quantitative measures of error in the form of L_1 and L_2 error norms for ψ, and the percent change of enclosed volume, are available in Table 3.

At the lowest resolution, the classical reinitialization equation has difficulty maintaining the circular shape of the disk, and as such, sees a loss in enclosed volume. The reformulated reinitialization equation, on the other hand, maintains the circular disk, however, cannot preserve the notch, and consequently gains 2.9% of its initial volume. At the middle resolution, the reformulated reinitialization equation continues to display a somewhat improved ability in maintaining the circular disk and correct notch placement over long times, leading to lower L_1 and L_2 errors, along with better conservation of its initial volume. For the highest resolution, both reinitialization equations accurately track the interface for all 10 rotations.

If the lowest resolution simulation is disregarded due to the inability to preserve the notch, test cases with Eq. (14) hint at near second order convergence in all displayed error metrics. The new form of the reinitialization equation also displays much better boundedness than the classical reinitialization equation, with maximum ψ values at the three resolutions of (1.00025, 1.00061, 1.00048) versus maximum ψ values of (1.00820, 1.00991, 1.00929) for the classical reinitialization equation.

We are also interested in the efficiency of this method; how quickly the transport and reinitialization algorithms are performed while maintaining a reasonable degree of accuracy. In order to get some indication of this, we will compare results from the reformulated reinitialization equation with ACLS to those from a recent, formally second-order, exactly mass conserving, unsplit semi-Lagrangian flux based volume of fluid (VOF) method, published in Section 4.1 of Owkes and Desjardins [20]. The L_1 error, referred to as E_{shape} in [20], the change in enclosed volume, and the computation time required per timestep are compared. These simulations are performed using one compute node consisting of a dual 6-core X5670 3 GHz CPUs with 48 GB of RAM, the same used in [20].

In order to match the conditions given in [20], the Zalesak disk is only rotated once. The other conditions are the same as those discussed before. Please note, that while we will use a CFL of 0.48, a CFL is not provided for the Zalesak test cases presented in [20]. The L_1 error norm, change in enclosed volume, and required time per timestep is listed alongside the results from [20] in Table 4, with the number of cells used to resolve the domain ranging from 25^2 to 800^2 ($w/\Delta x = 1.5$ to $w/\Delta x = 40$). It is important to note, "Vol. change" in Table 4 is not the percent volume change, but the dimensional amount of volume lost, in order to match the reported results in [20]. The L_1 error is plotted in Fig. 10 for both the presented method and the VOF method, with both displaying between first and second order convergence.

The ACLS method with the reformulated reinitialization equation can be seen to deliver comparable accuracy to the unsplit, semi-Lagrangian flux based VOF method, especially at moderate and high resolutions. While it does not conserve enclosed volume to machine precision, it does exhibit strong volume conservation properties. Most importantly, the ACLS

Table 4
Comparison between the L_1 error, volume change, and required time per timestep for the ACLS method with the reformulated reinitialization equation and the unsplit, semi-Lagrangian flux based VOF method proposed in Owkes and Desjardins [20].

Resolution	L_1 error		Vol. change		Time/timestep (s)	
	Current	[20]	Current	[20]	Current	[20]
25^2	2.316e−02	1.526e−02	−1.469e−04	4.629e−18	0.01	0.07
50^2	8.010e−03	4.066e−03	−2.865e−05	3.011e−17	0.02	0.16
100^2	2.193e−03	1.257e−03	−1.160e−07	4.409e−18	0.04	0.31
200^2	5.474e−04	5.684e−04	−6.692e−10	3.705e−18	0.11	0.66
400^2	2.287e−04	2.348e−04	−1.960e−09	2.317e−18	0.37	1.47
800^2	1.002e−04	9.221e−05	−6.534e−11	1.937e−17	1.47	3.38

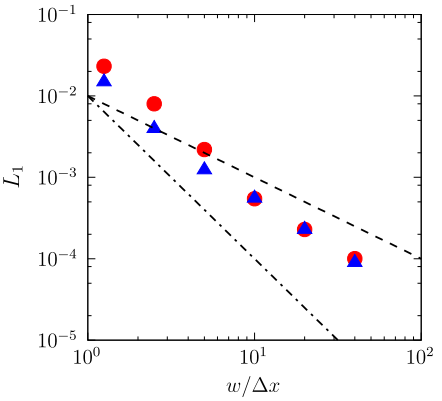


Fig. 10. Plot of the L_1 error after rotating the Zalesak disk once, where $w/\Delta x$ is the number of cells across the disk's notch. Presented method (●); results from [20] (▲); 1st order convergence (– – –); 2nd order convergence (– · – · –).

method with the reformulated reinitialization equation delivers this comparable accuracy in a fraction of the time, with the ACLS method completing the simulation up to 8 times faster than the VOF method from [20].

4.3. Deformation of a circle

In order to test volume conservation properties when structures drop below the minimum resolvable thickness, a 2D circle deformation test, originally proposed by Leveque [11], is performed. In this test, a 0.3 diameter circle is initialized with its center at $(x, y) = (0.0, 0.25)$, in a square $[-0.5, 0.5]^2$ domain. The velocity field as a function of time, t, is prescribed as

$$u = -2\sin^2(\pi x)\sin(\pi y)\cos(\pi y)\cos(\pi t/8)\,, \tag{30a}$$

$$v = 2\sin^2(\pi y)\sin(\pi x)\cos(\pi x)\cos(\pi t/8)\,, \tag{30b}$$

which will stretch the circle into a spiral until $t = 4$, at which point the flow will reverse and the interface should return to its initial circle at $t = 8$. Uniform Cartesian meshes of 64^2, 128^2, and 256^2 are used for this study in order to understand the volume conservation properties at various levels of resolution.

Fig. 11 shows the interface at maximal deformation ($t = 4$) and at the end of the cycle ($t = 8$), as well as the normalized volume enclosed by the interface, for both reinitialization equations when run with an initial CFL of 0.64. The image of the interface at $t = 4$ is superimposed over results from a simulation using a 512^2 uniform Cartesian mesh, while the interface at $t = 8$ is compared to the initial circle. The lowest resolution is not capable of preserving the fine ligaments that occur at $t = 4$, causing a loss in volume and significant topology change. As the flow reverses, segmented volumes of ψ recombine and become resolved enough to reform an interface, eventually returning close to its initial enclosed volume. This does not lead to a correct interface though, with the interface at $t = 8$ being far from the initial 0.3 diameter circle. As the mesh resolution increases, finer structures are maintained throughout the deformation and the minimum normalized enclosed volume approaches 1.

4.4. Curvature accuracy for a sphere

Prior to performing a test on spurious currents, we will study the error in calculating the curvature of a sphere ($D = 0.4$), which directly influences spurious currents. The curvature error is only calculated for faces where ϕ_{FMM} switches signs, since only these faces will contain a surface tension source term while using the GFM.

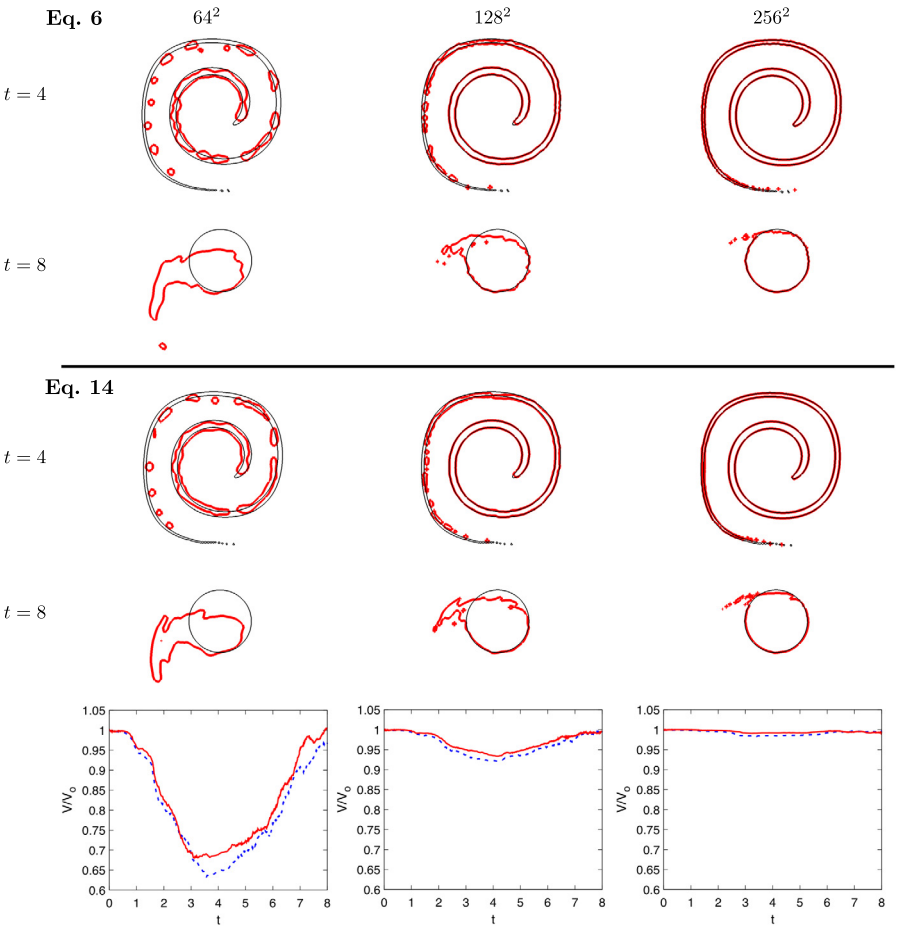


Fig. 11. Interface at $t = 4.0$ and $t = 8.0$, as well as the time history of normalized volume enclosed by the interface, for both forms of the reinitialization equation. Eq. (6) (- - -); Eq. (14) (——).

To quantitatively measure the level of errors, the L_2 and L_∞ curvature error norms are used, calculated as

$$L_2 = \frac{\sqrt{\frac{1}{N_{\text{interface}}}\sum_{i=1}^{N_{\text{interface}}}(\kappa_{\text{exact}} - \kappa_{i,\text{sim}})^2}}{\kappa_{\text{exact}}}, \tag{31}$$

$$L_\infty = \frac{\max(\kappa_{\text{exact}} - \kappa_{\text{sim}})}{\kappa_{\text{exact}}}, \tag{32}$$

where $N_{\text{interface}}$ is the number of cells in the domain containing interface. These error measures are shown for several sphere resolutions ranging from $D/\Delta x = 4$ to $D/\Delta x = 128$ in Fig. 12, and display second order convergence in the L_2 error norm and first order convergence in the L_∞ error norm.

We believe the curvature calculation using a least squares fitting of ϕ_{FMM} is an accurate and robust way to calculate curvature, however, ϕ_{FMM} itself is of a low order of accuracy. To determine if the use of the FMM to solve the static Eikonal equation has a significant impact on the calculated curvature, we rerun the simulations presented above with ϕ specified as $\phi = D/2 - \sqrt{\mathbf{x}^t\mathbf{x}}$, where $\mathbf{x} = (x, y, z)$ for the cell centers where ϕ is stored. A dramatic reduction of errors is seen in both error norms, indicating that a more accurate method of solving the static Eikonal equation is needed to further decrease the error in curvature. One possibility would be to use a Hamilton–Jacobi type reinitialization equation to obtain a more accurate ϕ field, however, this is not addressed in this paper.

The $5 \times 5 \times 5$ stencil is quite large, leading to a relatively expensive curvature calculation. In order to understand the effect of going to a smaller, $3 \times 3 \times 3$ stencil, the same cases are once again simulated and the results plotted in Fig. 12. The $3 \times 3 \times 3$ stencil can be seen to deliver similar levels of error at low resolutions, however, converges more slowly for moderate to high resolutions when using ϕ_{FMM}, which contains errors. Unless otherwise noted, the $5 \times 5 \times 5$ stencil will be used for calculating curvature.

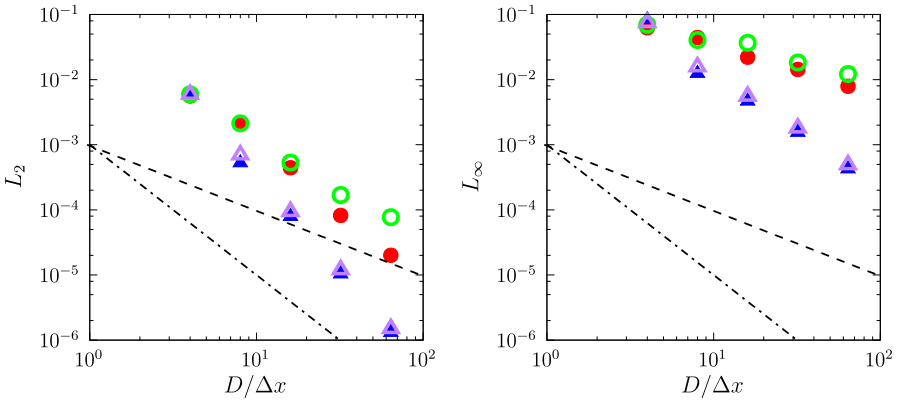


Fig. 12. L_2 (a) and L_∞ (b) curvature error norms for a sphere with ϕ_{FMM} or an exact ϕ, using a $5 \times 5 \times 5$ or $3 \times 3 \times 3$ stencil. $5 \times 5 \times 5$ stencil, ϕ_{FMM} (●); $5 \times 5 \times 5$ stencil, exact ϕ (▲); $3 \times 3 \times 3$ stencil, ϕ_{FMM} (○); $3 \times 3 \times 3$ stencil, exact ϕ (△); 1st order convergence (- - -); 2nd order convergence (· - · -).

4.5. 3D spurious currents

Due to inaccuracies in the curvature calculation and its imposition to the momentum solver through the pressure jump, spurious velocities can appear at the interface. One of the advantages of the new form of the reinitialization equation lies in its reduction of error for stationary objects. This should reduce spurious currents through the removal of interface shape errors usually introduced during reinitialization.

Since we are ultimately concerned about atomizing turbulent flows, we have conducted a three dimensional spurious currents test case using a sphere. A 0.4 diameter sphere is initialized in the center of a cubic unit domain with a uniform mesh of 40^3. A range of Laplace numbers ($La = \sigma\rho D/\mu^2$) from 120 to 1200000 are covered through simultaneously increasing the liquid and gas densities. Throughout all testing, both dynamic viscosities are kept constant at 0.1, and the surface tension is held at 1. The time step is also increased along with the Laplace number to scale with the surface tension time step restriction. The resulting steady state capillary numbers ($Ca = |u_{\max}|\mu/\sigma$) can be seen in Table 5. Here, $|u_{\max}|$ is the maximum steady state velocity, where steady state is deemed reached when the maximum velocity in the domain changes by less than 1% for 25 consecutive iterations. We have found this to be a true indication of when the system has reached steady state, which will occur when the reinitialization errors, curvature errors, and induced flow equilibrate. Defining a non-dimensional time unit as $t^\star = t\sigma/(\mu D)$, the minimum time a simulation takes to reach steady state is $t^\star = 9.975$, while the maximum time is $t^\star = 4035$. In order to study the change in spurious currents at various mesh resolutions, the Laplace number is held at $La = 12000$ and the mesh is varied from 20^3 to 80^3. The results can be seen in Table 6.

As the Laplace number increases, more iterations are required to reach a steady state, leading to an accumulation of error when using Eq. (6) and a corresponding increase in steady state capillary number at high Laplace numbers. The reformulated reinitialization equation does not suffer from this, and the steady state capillary number decreases as surface tension becomes less significant. Additionally, with a more accurate capturing of the interface, the steady state capillary number is decreased by nearly an order of magnitude across all Laplace numbers.

A significant reduction in steady state capillary number is also seen for the mesh convergence study presented in Table 6, with the reformulated reinitialization equation showing low steady state capillary numbers, although still larger than those seen from some DLS methods [21]. The classical reinitialization equation exhibits a near constant level of capillary number across the three resolutions, while Eq. (14) shows a continual decrease. This indicates that use of the classical reinitialization equation, Eq. (6), results in a leading source of error.

Table 5
Steady state capillary number for each Laplace number using the two different forms of the reinitialization equation.

La	1.2×10^2	1.2×10^3	1.2×10^4	1.2×10^5	1.2×10^6
Ca (Eq. (6))	7.4×10^{-4}	1.7×10^{-4}	1.3×10^{-4}	3.2×10^{-4}	2.7×10^{-4}
Ca (Eq. (14))	6.8×10^{-5}	2.9×10^{-5}	2.4×10^{-5}	2.3×10^{-5}	1.9×10^{-5}

Table 6
Steady state capillary number for a Laplace number of 12000 with three mesh resolutions using the two different forms of the reinitialization equation.

Resolution	20^3	40^3	80^3
Ca (Eq. (6))	4.5×10^{-4}	1.3×10^{-4}	3.0×10^{-4}
Ca (Eq. (14))	1.5×10^{-4}	2.4×10^{-5}	2.3×10^{-5}

Table 7
Physical properties used for the spurious currents test case of a spherical water drop in air.

Phase	ρ (kg/m^3)	μ (Pa s)	σ (N/m)
Water	1000.0	8.9×10^{-4}	0.07286
Air	1.225	1.81×10^{-5}	–

Table 8
Steady state Ca for water–air spurious currents test cases with the reformulated reinitialization equation. Both La and Ca are calculated using liquid properties.

D (μm)	La	Ca
10	920	3.0×10^{-4}
100	9200	1.8×10^{-3}
1000	92000	1.1×10^{-3}

Using the reformulated reinitialization equation, we also perform more physically relevant spurious currents test cases consisting of a spherical water droplet with a diameter of $D = 10$ μm, $D = 100$ μm, or $D = 1000$ μm, in a quiescent air cubic domain. A uniform Cartesian mesh with 40^3 cells is used with the domain size selected to resolve the droplet by $D/\Delta x = 16$, the same as the previous cases. The physical properties used for the water and air are given in Table 7. Once again the cases were run to steady state, as previously defined, and the resulting capillary numbers, shown in Table 8, are found to be of a reasonable magnitude.

5. Atomizing turbulent liquid jet

In order to demonstrate the ability of the ACLS method with the reformulated reinitialization equation in the context of more complex flows, we will simulate a turbulent liquid jet being injected into quiescent gas.

5.1. Simulation configuration

The simulation is performed using our in-house CFD flow solver, NGA [14,16], described previously in Section 3. In order to generate the jet inflow profile, a turbulent pipe flow simulation is run a priori and the velocity along a plane is extracted. To perform the atomizing turbulent jet simulation, this inflow is then used as the velocity profile for the liquid jet while it is injected into the domain of quiescent gas. An initial liquid hemisphere is also placed in the domain to represent the parabolic entering front of the liquid jet.

The jet of diameter D_j is injected into a $15D_j \times 6D_j \times 6D_j$ domain with a uniform Cartesian mesh of $1000 \times 400 \times 400$ cells. The domain itself is decomposed into $4 \times 24 \times 18$ sub-domains using 1728 processors on XSEDE's SDSC Comet supercomputer [22]. The non-dimensional properties governing the physics of the liquid jet are given in Table 9. Although the mesh used might not capture all of the small scales present in the turbulent flow, no sub-grid scale model is used. This should not hinder the main goals of this simulation to demonstrate the capabilities of the presented ACLS method.

5.2. Simulation results

A rendering of the liquid–gas interface at four different times separated by $t^\star = t\overline{U}_j/D_j = 3$, where $\overline{U}_j$ is the mean jet velocity, can be seen in Fig. 13. The actual enclosed volume in the simulation from $t^\star = 0$ to $t^\star = 13.5$, when liquid begins to exit the domain, is also compared to the exact value, calculated as

$$V_e = A_{\mathrm{CS,in}}\overline{U}_j t^\star + V_o \,, \tag{33}$$

where $A_{\mathrm{CS,in}}$ is the cross-sectional area of the liquid inflow and V_o is the initialized liquid volume of the hemispherical jet cap. This measurement of conservation is plotted over time in Fig. 14. Lastly, the percentage of time spent in each portion of the code is shown in Fig. 15, with the time required per step in each portion given in Table 10.

Table 9
Non-dimensional parameters for the liquid jet. A subscript j denotes a jet property, while the subscript o denotes a property of the ambient quiescent gas.

r	m	Re_D	We_D
$\frac{\rho_j}{\rho_o}$	$\frac{\mu_j}{\mu_o}$	$\frac{\rho_j \overline{U}_j D_j}{\mu_j}$	$\frac{\rho_j \overline{U}_j^2 D_j}{\sigma}$
40	40	5000	10000

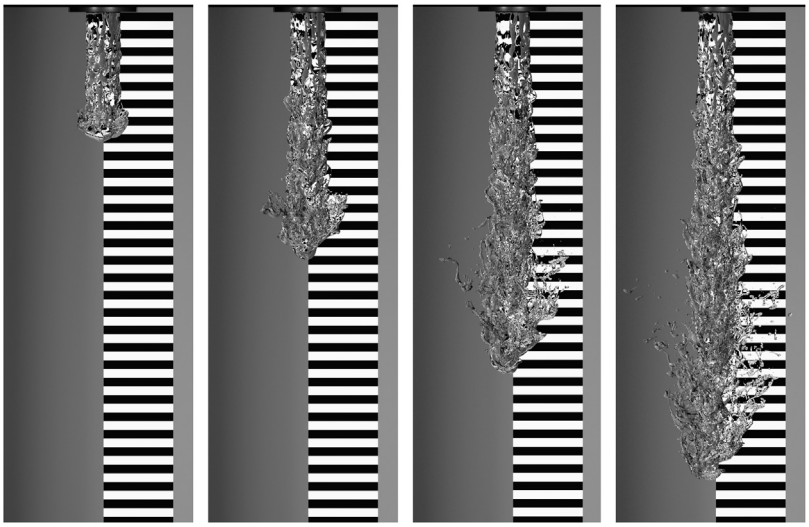


Fig. 13. Renderings of the liquid–gas interface at $t^* = 3, 6, 9, 12$.

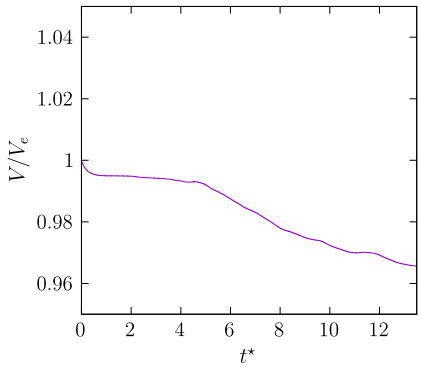


Fig. 14. Volume conservation of enclosed volume for the liquid–gas jet.

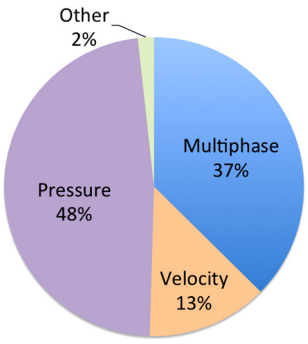


Fig. 15. Percentage of time spent in each category for the multiphase jet simulation.

Table 10
Time required per step for the major sections of the code.

Routine	Pressure	Multiphase	Velocity	Other
Time per step (CPU-hours)	2.81	2.20	0.767	0.102

The ACLS method with the reformulated reinitialization equation shows good conservation properties, with a maximum volume loss of 3.5% over the $\Delta t^\star = 13.5$ time span. Note that the multiphase portion of the timing presented in Fig. 15 and Table 10 represents the cumulative time taken each timestep for interface tracking, reinitialization, the FMM to obtain ϕ_{FMM}, and calculating the curvature. Of these, calculating the curvature is a significant fraction. If the stencil used to construct the least squares linear system is reduced to a $3 \times 3 \times 3$ stencil, the total time in multiphase routines is reduced by 30%.

6. Conclusion

In this paper, we present a reformulation of the reinitialization equation that improves its accuracy yet remains robust for simulating complex multiphase flows. Through basing both the compression and diffusion terms in the reinitialization equation on ϕ_{map} from Eq. (10), differentiation errors are reduced. To calculate the normals, a FMM is used to obtain ϕ_{FMM} from the implicit interface in ϕ_{map}, which is shown to increase robustness by providing consistent normals, preventing the formulation of spurious volumes of ψ and allowing the use of high order, non-TVD transport schemes. This new form of the reinitialization equation is then shown to reduce interface errors and spurious currents, especially for stationary surfaces. Lastly, the ACLS method with the reformulated reinitialization equation is used to simulate an atomizing liquid jet, exhibiting excellent mass conservation properties.

Acknowledgements

This work was supported by the National Science Foundation through CAREER award CBET-1351545 and computing resources on SDSC Comet.

References

[1] S. Osher, J. Sethian, Front propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.
[2] E. Olsson, G. Kreiss, A conservative level set method for two phase flow, J. Comput. Phys. 210 (1) (2005) 225–246, http://dx.doi.org/10.1016/j.jcp.2005.04.007.
[3] E. Olsson, G. Kreiss, S. Zahedi, A conservative level set method for two phase flow II, J. Comput. Phys. 225 (1) (2007) 785–807, http://dx.doi.org/10.1016/j.jcp.2006.12.027.
[4] O. Desjardins, V. Moureau, H. Pitsch, An accurate conservative level set/ghost fluid method for simulating turbulent atomization, J. Comput. Phys. 227 (18) (2008) 8395–8416, http://dx.doi.org/10.1016/j.jcp.2008.05.027.
[5] J. Tsitsiklis, Efficient algorithms for globally optimal trajectories, IEEE Trans. Autom. Control 40 (9) (1995) 1528–1538.
[6] J.O. McCaslin, O. Desjardins, A localized re-initialization equation for the conservative level set method, J. Comput. Phys. 262 (2014) 408–426, http://dx.doi.org/10.1016/j.jcp.2014.01.017.
[7] R.K. Shukla, Nonlinear preconditioning for efficient and accurate interface capturing in simulation of multicomponent compressible flows, J. Comput. Phys. 276 (2014) 508–540, http://dx.doi.org/10.1016/j.jcp.2014.07.034.
[8] S. Osher, C.-W. Shu, High-order essentially nonoscillatory schemes for Hamilton–Jacobi equations, SIAM J. Numer. Anal. 28 (4) (1991) 907–922, http://dx.doi.org/10.1137/0728049.
[9] T. Wacławczyk, A consistent solution of the re-initialization equation in the conservative level-set method, J. Comput. Phys. 299 (2015) 487–525, http://dx.doi.org/10.1016/j.jcp.2015.06.029.
[10] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, J. Comput. Phys. 362 (1979) 335–362, http://dx.doi.org/10.1016/0021-9991(79)90051-2.
[11] R.J. LeVeque, High-resolution conservative algorithms for advection in incompressible flow, SIAM J. Numer. Anal. 33 (2) (1996) 627–665, http://dx.doi.org/10.1137/0733033.
[12] R.R. Nourgaliev, T.G. Theofanous, High-fidelity interface tracking in compressible flows: unlimited anchored adaptive level set, J. Comput. Phys. 224 (2) (2007) 836–866, http://dx.doi.org/10.1016/j.jcp.2006.10.031.
[13] J.O. McCaslin, É. Courtine, O. Desjardins, A fast marching approach to multidimensional extrapolation, J. Comput. Phys. 274 (2014) 393–412, http://dx.doi.org/10.1016/j.jcp.2014.06.023.
[14] O. Desjardins, G. Blanquart, G. Balarac, H. Pitsch, High order conservative finite difference scheme for variable density low Mach number turbulent flows, J. Comput. Phys. 227 (15) (2008) 7125–7159, http://dx.doi.org/10.1016/j.jcp.2008.03.027.
[15] Z. Wang, J. Yang, F. Stern, A new volume-of-fluid method with a constructed distance function on general structured grids, J. Comput. Phys. 231 (9) (2012) 3703–3722, http://dx.doi.org/10.1016/j.jcp.2012.01.022.
[16] O. Desjardins, J. McCaslin, M. Owkes, P. Brady, Direct numerical and large-eddy simulation of primary atomization in complex geometries, At. Sprays 23 (11) (2013) 1001–1048, http://dx.doi.org/10.1615/AtomizSpr.2013007679.
[17] E. Marchandise, P. Geuzaine, N. Chevaugeon, J.-F. Remacle, A stabilized finite element method using a discontinuous level set approach for the computation of bubble dynamics, J. Comput. Phys. 225 (1) (2007) 949–974, http://dx.doi.org/10.1016/j.jcp.2007.01.005.
[18] M. Coquerelle, S. Glockner, A fourth-order accurate curvature computation in a level set framework for two-phase flows subjected to surface tension forces, J. Comput. Phys. 305 (2016) 838–876, http://dx.doi.org/10.1016/j.jcp.2015.11.014.
[19] R. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (2) (1999) 457–492, http://dx.doi.org/10.1006/jcph.1999.6236.
[20] M. Owkes, O. Desjardins, A computational framework for conservative, three-dimensional, unsplit, geometric transport with application to the volume-of-fluid (VOF) method, J. Comput. Phys. 270 (2014) 587–612, http://dx.doi.org/10.1016/j.jcp.2014.04.022.
[21] M. Herrmann, A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids, J. Comput. Phys. 227 (4) (2008) 2674–2706, http://dx.doi.org/10.1016/j.jcp.2007.11.002.
[22] San Diego Supercomputer Center, Comet User Guide, http://www.sdsc.edu/support/user_guides/comet.html, July 2016.