

# A localized re-initialization equation for the conservative level set method



Jeremy O. McCaslin\*, Olivier Desjardins

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

## ARTICLE INFO

### Article history:

Received 9 April 2013  
Received in revised form 4 October 2013  
Accepted 10 January 2014  
Available online 16 January 2014

### Keywords:

Multiphase flow  
Interface capture  
Conservative level set  
Local re-initialization  
Eikonal equation  
Flow kinematics  
Numerical diffusion

## ABSTRACT

The conservative level set methodology for interface transport is modified to allow for localized level set re-initialization. This approach is suitable to applications in which there is a significant amount of spatial variability in level set transport. The steady-state solution of the modified re-initialization equation matches that of the original conservative level set provided an additional Eikonal equation is solved, which can be done efficiently through a fast marching method (FMM). Implemented within the context of the accurate conservative level set method (ACLS) (Desjardins et al., 2008, [6]), the FMM solution of this Eikonal equation comes at no additional cost. A metric for the appropriate amount of local re-initialization is proposed based on estimates of local flow deformation and numerical diffusion. The method is compared to standard global re-initialization for two test cases, yielding the expected results that minor differences are observed for Zalesak's disk, and improvements in both mass conservation and interface topology are seen for a drop deforming in a vortex. Finally, the method is applied to simulation of a viscously damped standing wave and a three-dimensional drop impacting on a shallow pool. Negligible differences are observed for the standing wave, as expected. For the last case, results suggest that spatially varying re-initialization provides a reduction in spurious interfacial corrugations, improvements in the prediction of radial growth of the splashing lamella, and a reduction in conservation errors, as well as a reduction in overall computational cost that comes from improved conditioning of the pressure Poisson equation due to the removal of spurious corrugations.

© 2014 Elsevier Inc. All rights reserved.

## 1. Introduction

Identification of propagating fronts is a necessity within many fields of computational science. In order to capture fronts, level set methods were introduced by Osher and Sethian [19] by implicitly defining the interface as an iso-surface of a signed distance function. Integral to the level set approach is re-initialization of the distance function, which is necessary to maintain a unity gradient within the computational domain and avoid numerical instabilities. Re-initialization of the distance function is typically done by solving a Hamilton–Jacobi equation, as has been applied in the context of crack propagation on nonuniform grids [15,10], flame propagation [46,25], the transport of the phase interface for multiphase flow simulations [38,22], and in a variety of particle level set applications [7,9,8,12]. Methods have also been developed for rapid re-distancing based on closest-point considerations [39] and quadtree meshes and triangulation [35,34,36,37]. One fast and efficient alternative to the Hamilton–Jacobi approach, known as the fast marching method (FMM), was introduced

\* Corresponding author.

E-mail address: jom48@cornell.edu (J.O. McCaslin).

by Sethian [31] and bypasses Courant–Friedrichs–Lewy (CFL) limitations by solving the re-initialization equation directly in its Eikonal form.

The implicit definition of the interface as an iso-surface of a smooth function is attractive for multiphase flow calculations, as it lends itself to convenient computation of interfacial quantities such as curvature and automatic handling of interface topology changes. This is in contrast to the geometric representation of the interface in volume-of-fluid (VOF) methods [13], which require interface reconstruction from the cell volume fractions by way of simple-line interface calculation (SLIC) [16] or piecewise linear interface calculation (PLIC) [21], for example.

Despite the benefits of a smooth representation of the interface, the primary downfall of the level set application to multiphase flows is its lack of any inherent conservation property. Olsson and Kreiss [17] and Olsson et al. [18] alleviated this issue by introducing a hyperbolic tangent level set that resembles the liquid volume fraction. Re-initialization is used to maintain a hyperbolic tangent profile after the level set is transported. However, oscillatory normals obtained from differentiation of the conservative level set function were shown to be inadequate for use in the re-initialization equation by Desjardins et al. [6], who proposed to combine the conservative level set with a re-distancing algorithm performed efficiently using a fast marching method [29,1]. This was done in order to obtain accurate normals from the smooth distance field. Different methods based on the conservative level set idea have been shown to be successful for a variety of applications, including large-eddy simulations of multiphase reacting flows [23], compressible flows [33], turbulent atomization [6], two-phase electrohydrodynamic flows [40,14], and modeling of contact angle [44,28].

While re-initialization is needed to preserve the hyperbolic tangent profile and improve conservation, excessive use of it may deteriorate the solution by compounding errors. In particular, it is clear that an unperturbed level set should ideally not be re-initialized. Efforts have been made to limit the amount of re-initialization to prevent such accumulation of errors. In particular, Owkes and Desjardins [20] proposed a metric to calibrate re-initialization based on the global maximum amount of level set deformation that comes from numerical diffusion through level set transport. However, regions with little to no level set transport still suffer from errors due to the global nature of the re-initialization equation. Another study proposed a local form of re-initialization by tying the amount of level set deformation to local flow field kinematics [27]. While accounting for relevant physical sources of level set deformation, the approach did not account for numerical diffusion contributions to level set deformation and was not formulated conservatively. The essence of the present work is to combine both kinematic and numerical sources of level set deformation into a local coefficient that determines the necessary amount of re-initialization, formulated in a conservative manner. The methodology presented herein may be useful in general to compressive or counter-gradient transport schemes [42,41,2].

Details of the classical and conservative level set methods are given in Section 2. Motivation and conservative formulation of the method are presented in Section 3, followed by a detailed description of how to determine the appropriate amount of local re-initialization in Section 4. In Section 5 Zalesak's disk [45] and a drop deforming in a vortex are presented. The latter exhibits flow kinematics that benefit from local re-initialization, while the former does not. This is intended to ensure that the local formulation of the re-initialization equation does not degrade previous results. Congruently in Section 6, we apply the method to two different flows, one which does not have need for local treatment and one that does. The former is the viscous damping of a free surface and the latter a drop impacting a shallow pool.

## 2. Mathematical formulation

### 2.1. Classical level set

The classical level set method introduced by Osher and Sethian [19] relies on a signed distance function  $\phi(\mathbf{x}, t) = \pm\|\mathbf{x} - \mathbf{x}_I\|$ , where  $\mathbf{x}_I$  is the location on the interface  $\Gamma$  that provides the minimum Euclidean distance from location  $\mathbf{x}$ . The sign of  $\phi$  is designated as positive on one side of the interface and negative on the other. The motion of the level set is captured through the evolution equation

$$\frac{\partial\phi}{\partial t} = F\|\nabla\phi\|, \quad (1)$$

where the speed function  $F$  gives the speed of the interface in its normal direction. After being transported,  $\phi$  should be re-initialized to a true signed distance function, as Eq. (1) may not preserve the property

$$\|\nabla\phi\| = 1. \quad (2)$$

This may be done by solving a Hamilton–Jacobi equation [22], but Sethian introduced a FMM approach that bypasses CFL limitations by solving  $\|\nabla\phi\| = 1$  directly in its Eikonal form [31]. Velocity extension is an alternative approach to  $\phi$  preservation that was introduced by Adalsteinsson and Sethian [1]. The idea is to extend the speed function  $F$ , which only has physical meaning at the interface, to the entire domain by solving

$$\nabla\phi \cdot \nabla F = 0. \quad (3)$$

If this is done, then theoretically the evolution of  $\phi$  will not lead to deformation and re-initialization becomes unnecessary. Geometrically, Eq. (3) enforces that  $F$  is constant along normals from the interface. Further details regarding velocity ex-

tensions are provided elsewhere [32,3,4]. An attractive attribute of the classical level set is that interfacial normals  $\mathbf{n}$  and curvature  $\kappa$  are easily obtained from differentiation of  $\phi$ , defined as

$$\mathbf{n} = \frac{\nabla\phi}{\|\nabla\phi\|} \quad (4)$$

and

$$\kappa = -\nabla \cdot \mathbf{n}. \quad (5)$$

## 2.2. Conservative level set

The conservative level set of Olsson and Kreiss [17] and Olsson et al. [18] replaces the distance function by a hyperbolic tangent profile, defined as

$$\psi(\mathbf{x}, t) = \frac{1}{2} \left( \tanh\left(\frac{\phi(\mathbf{x}, t)}{2\varepsilon}\right) + 1 \right), \quad (6)$$

where the thickness of the profile is specified by  $\varepsilon$ . In the conservative level set notation, the interface now corresponds to the  $\psi = 0.5$  iso-surface. The level set is transported via the evolution equation

$$\frac{\partial\psi}{\partial t} + \mathbf{u} \cdot \nabla\psi = 0, \quad (7)$$

where  $\mathbf{u}$  is the velocity field and  $t$  is time. Note that in the presence of a solenoidal velocity field, Eq. (7) can be rewritten as

$$\frac{\partial\psi}{\partial t} + \nabla \cdot (\mathbf{u}\psi) = 0, \quad (8)$$

allowing for discretely conservative transport of  $\psi$  and providing conservation properties to the algorithm. After transporting  $\psi$  by Eq. (8),  $\psi$  should be restored to the hyperbolic tangent profile to improve numerical robustness and improve conservation of the mass under the  $\psi = 0.5$  iso-surface. Considering a one-dimensional level set  $\psi(\phi)$ , where  $\phi$  plays the role of the spatial coordinate, Eq. (6) satisfies

$$\frac{\partial\psi}{\partial\phi} = \frac{\psi(1-\psi)}{\varepsilon}. \quad (9)$$

The solution to this equation in higher dimensions is compatible with the steady state solution of

$$\frac{\partial\psi}{\partial\tau} = \nabla \cdot ((\varepsilon\nabla\psi \cdot \mathbf{n}) - \psi(1-\psi)\mathbf{n}), \quad (10)$$

where  $\mathbf{n}$  is the unit vector normal to the interface and  $\tau$  is a pseudo-time coordinate. The successive solution of Eqs. (8) and (10) provides that the phase interface is advected by the fluid velocity and maintains discrete conservation of  $\psi$ . When transporting and re-initializing discretely, the time step  $\Delta t_{\text{reinit}}$  and pseudo-time step  $\Delta\tau_{\text{reinit}}$  are governed by the CFL conditions associated with the transport and re-initialization equations, Eqs. (8) and (10), respectively. Assuming a uniform mesh size  $\Delta x$ , these conditions are

$$\text{CFL}_{\text{trans}} = \frac{\max(|\mathbf{u}|)\Delta t_{\text{trans}}}{\Delta x} \quad (11)$$

and

$$\text{CFL}_{\text{reinit}} = \max\left(\frac{\Delta\tau_{\text{reinit}}}{\Delta x}, \frac{4\varepsilon\Delta\tau_{\text{reinit}}}{\Delta x^2}\right). \quad (12)$$

The two terms in Eq. (12) come from the diffusive and compressive fluxes in the re-initialization equation, respectively. Note that sub-steps may be required for the solution of the transport equation if the maximum allowable  $\Delta t_{\text{trans}}$  from Eq. (11) is smaller than the flow solver time step  $\Delta t$ . Similarly, the solution of the re-initialization equation will require sub-steps if the prescribed amount of total re-initialization time  $\Delta\tau$  is larger than the maximum allowable  $\Delta\tau_{\text{reinit}}$  from Eq. (12).

In the framework of the accurate conservative level set method (ACLS) of Desjardins et al. [6], which is the framework for the current work,  $\mathbf{n}$  in the re-initialization equation is computed via Eq. (4) from a re-distanced  $\phi$  that is obtained from a FMM performed prior to re-initialization. Once computed,  $\mathbf{n}$  is kept constant during the solution of Eq. (10). Note that other three-dimensional reformulations of Eq. (9) are possible, and one alternative was proposed by Shukla et al. [33] that avoids differentiating  $\mathbf{n}$ , but at the cost of the conservative property of Eq. (10).

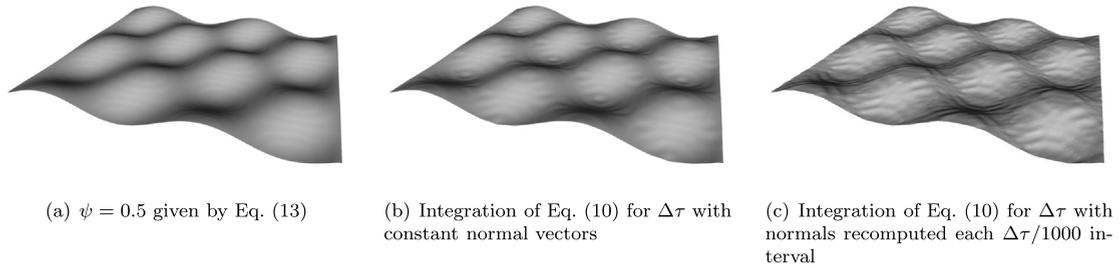


Fig. 1. The feedback mechanism between erroneous normals and the solution of Eq. (10).

Table 1

$L_2$  and  $L_\infty$  norms of the curvature error on a  $100^3$  mesh for re-initialization of a wave.

	$L_2$	$L_\infty$
Test 1	0.00128	24.5695
Test 2	0.01137	109.9825

### 3. Localized conservative re-initialization

#### 3.1. Motivation

Re-initialization is necessary for hyperbolic tangent profile preservation and numerical stability, as there is no diffusive term in the transport. Since Eq. (10) has to be discretized, its numerical solution will be associated with errors. In particular, the action of re-initialization is to move the  $\psi = 0.5$  iso-surface in the direction of the normal vector  $\mathbf{n}$ , which is also approximated discretely. If the level set is stationary, any error dissipation that could occur by means of numerical diffusion from the transport equation is not possible. For stationary level sets, a feedback mechanism establishes as follows: re-initialization will move  $\psi = 0.5$  in the direction of  $\mathbf{n}$ ,  $\phi$  will be recomputed at the next iteration of the simulation and differ from the previous iteration due to numerical errors despite the fact that the interface has not moved,  $\mathbf{n}$  will be recomputed from the new  $\phi$  value and also differ slightly, and the interface will be moved in a different direction due to another application of Eq. (10). This process will be repeated throughout the simulation, leading to an overall degradation of the computational results if excessive re-initialization is used. This feedback mechanism is identical to if Eq. (10) were being solved while updating  $\mathbf{n}$ . This effect will manifest itself in computations as spurious displacement of the interface. We illustrate this by constructing a stationary interface in a  $100 \times 100 \times 100$  computational domain defined by  $x \in [-0.5, +0.5]$ ,  $y \in [0, 1]$ , and  $z \in [-0.5, +0.5]$ . The interface is prescribed as

$$\psi(x, y, z) = \frac{1}{2} \left( \tanh \left( \frac{0.5 - y + A_0 \sin(4\pi x) \cos(4\pi z)}{2\varepsilon} \right) + 1 \right) \tag{13}$$

and is shown in Fig. 1(a). The amplitude  $A_0$  of the initial interface displacement is set to  $3\Delta x$ . To illustrate the feedback mechanism that leads to spurious  $\psi$  displacement, we perform two tests. The first test, shown in Fig. 1(b), computes  $\mathbf{n}$  and keeps  $\mathbf{n}$  constant while subsequently iterating Eq. (10) for a full integration time of  $\Delta\tau = 2.5$ . It is clear from Fig. 1(b) that this does not alter the  $\psi = 0.5$  iso-surface very much, as we have not updated the normals in between successive solutions of Eq. (10). The second test, shown in Fig. 1(c), also integrates Eq. (10) for a full  $\Delta\tau = 2.5$ , but  $\phi$  is recomputed and  $\mathbf{n}$  updated at every  $\Delta\tau/1000$  interval. It is clear from Fig. 1(c) that this moves the interface significantly relative to the first test. We measure the spurious displacement of  $\psi$  through the interfacial curvature  $\kappa = -\nabla \cdot \mathbf{n}$ . Table 1 shows the  $L_2$  and  $L_\infty$  curvature errors for the two cases, demonstrating that interfacial topology deteriorates by almost an order of magnitude between the two tests.

Owkes and Desjardins [20] correlated the integration pseudo-time step for re-initialization with the maximum amount of normal transport in the domain that  $\psi$  has undergone using

$$\Delta\tau = f_{\text{reinit}} \Delta t \max(|\mathbf{u} \cdot \mathbf{n}|), \tag{14}$$

where  $\Delta t$  is the transport time step and  $f_{\text{reinit}}$  is a parameter set by the user based on the nature of the flow considered. Choosing  $f_{\text{reinit}} = 0$  equates to no re-initialization, and  $f_{\text{reinit}} = 1$  allows re-initialization to move the interface as much as it was moved during the transport step. This has been shown by Owkes and Desjardins [20] to improve the topology of the interface for Zalesak’s disk [45] and droplet stretching test cases. However, this does not adequately address issues that arise in multiphase flows due to the existence of both stationary and moving regions of the interface. Such regions of the interface will be referred to as “inactive” and “active”. Obvious examples of such flows could include a drop impacting a free surface, as in Fig. 2(a), or the wavy interfacial flow of a free surface, as in Fig. 2(b). If the amount of re-initialization is calibrated according to an inactive region of the interface, i.e., regions where  $|\mathbf{u} \cdot \mathbf{n}|$  on  $\Gamma$  is small, then the computation

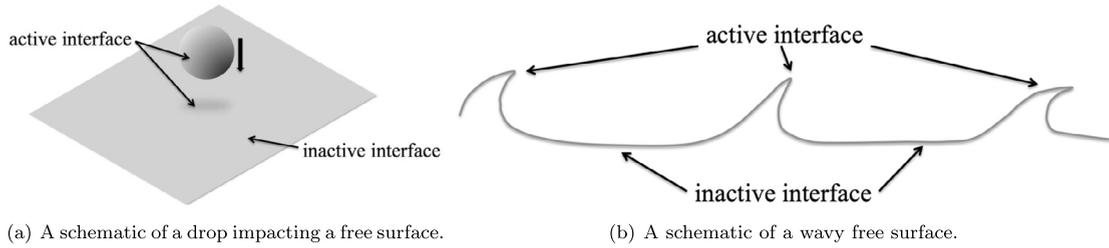


Fig. 2. Examples of multiphase flows that require spatially varying re-initialization.

may suffer from mass conservation errors in active regions due to  $\psi$  not being sufficiently re-initialized. On the other hand, catering the amount of re-initialization to an active region of the interface, i.e., where  $|\mathbf{u} \cdot \mathbf{n}|$  on  $\Gamma$  is large, can lead to the spurious motion of  $\psi$  in inactive regions. This is precisely the behavior observed in the test case of Fig. 1, for which the level set undergoes re-initialization despite the fact that the interface has not moved. We conclude that an appropriate solution to avoid both spurious  $\psi$  displacement in inactive regions and conservation errors in active regions is to allow for the amount of re-initialization to vary spatially in the computational domain. A strategy for enabling such localized re-initialization is discussed in the following section.

### 3.2. Conservative formulation

To allow for local re-initialization that maintains discrete conservation of  $\psi$ , we introduce a spatially and temporally varying factor  $\alpha(\mathbf{x}, t)$  into Eq. (10) to yield

$$\frac{\partial \psi}{\partial \tau} = \nabla \cdot (\alpha (\varepsilon (\nabla \psi \cdot \mathbf{n}) - \psi(1 - \psi)) \mathbf{n}). \quad (15)$$

Physical reasoning for the choice of  $\alpha$  is discussed in detail in Section 4. Expanding Eq. (15) leads to

$$\frac{\partial \psi}{\partial \tau} = \alpha \nabla \cdot ((\varepsilon (\nabla \psi \cdot \mathbf{n}) - \psi(1 - \psi)) \mathbf{n}) + ((\varepsilon (\nabla \psi \cdot \mathbf{n}) - \psi(1 - \psi)) \mathbf{n} \cdot \nabla \alpha, \quad (16)$$

hence  $\psi$  relaxes to the hyperbolic tangent solution of Eq. (6) if  $\mathbf{n} \cdot \nabla \alpha = 0$ ; that is if  $\alpha$  does not vary in the direction specified by the interfacial normal vector  $\mathbf{n}$ . If  $\alpha$  has this property, then Eq. (16) reduces to

$$\frac{\partial \psi}{\partial \tilde{\tau}} = \nabla \cdot ((\varepsilon (\nabla \psi \cdot \mathbf{n}) - \psi(1 - \psi)) \mathbf{n}), \quad (17)$$

where  $\tilde{\tau} = \alpha \tau$  is the localized pseudo-time. In computations, the total integration time is now specified by  $\Delta \tilde{\tau} = \alpha \Delta \tau$ . Setting  $\Delta \tau = 1$  simply allows the coefficient  $\alpha$  to determine the amount of re-initialization in the form of the local  $\Delta \tilde{\tau}$ . The Eikonal equation

$$\mathbf{n} \cdot \nabla \alpha = 0 \quad (18)$$

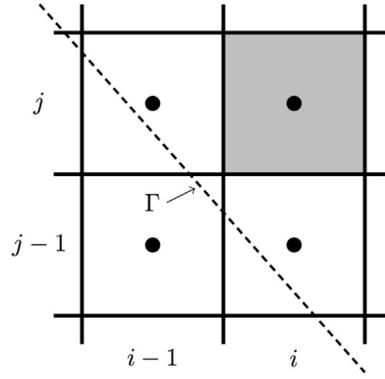
is identical in form to Eq. (3) and is solved most efficiently using a fast marching method in conjunction with the  $\|\nabla \phi\| = 1$  condition. The ACLS framework [6] already relies on a fast marching solution of  $\|\nabla \phi\| = 1$  in order to obtain accurate interfacial normals, so the majority of the computations involved in the solution of Eq. (18) are already being performed. The procedure for solving Eq. (18) is given next.

#### 3.2.1. Fast marching extension of $\alpha$

Here we present a basic description of the fast marching method such that it informs the solution of the Eikonal equation, but more detailed descriptions are provided by Sethian [29,31,30]. In the ACLS method of Desjardins et al. [6], a parallel FMM is used to re-distance the signed distance function  $\phi$  such that

$$\|\nabla \phi\| = 1, \quad (19)$$

in order to compute normals and curvature from Eqs. (4) and (5). The FMM solves Eq. (19) by using ordered upwind finite differences to express  $\nabla \phi$ . The upwind direction is always determined by the location of the interface, such that stencil points used in the calculation of  $\phi$  at a given computational zone always have a  $\|\phi\|$  of lesser value than the zone being computed. This causality assumption of the FMM is maintained through a heap sort that provides the order in which each zone should be computed. For clarity, consider the 2D example shown in Fig. 3. The location of the  $\phi = 0$  iso-contour is given by the dashed line, and the zone  $(i, j)$  is the zone at which we are calculating  $\phi_{i,j}$ . Based on the topology of  $\phi = 0$ , the values  $\phi_{i-1,j}$  and  $\phi_{i,j-1}$  have already been computed and will be used in the calculation of  $(\nabla \phi)_{i,j}$ . Using upwind differences,  $\|\nabla \phi\|$  can be approximated as



**Fig. 3.** 2D example of the FMM procedure. The dashed line shows the  $\phi = 0$  iso-contour, and the grey zone  $(i, j)$  is the zone at which  $\phi$  is being calculated.

$$\|\nabla\phi\|_{i,j} = \left[ \left( \frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta x} \right)^2 + \left( \frac{\phi_{i,j} - \phi_{i,j-1}}{\Delta y} \right)^2 \right]^{1/2}. \tag{20}$$

Setting Eq. (20) equal to 1 and solving the quadratic equation

$$\left( \frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta x} \right)^2 + \left( \frac{\phi_{i,j} - \phi_{i,j-1}}{\Delta y} \right)^2 - 1 = 0 \tag{21}$$

for  $\phi_{i,j}$  provides two roots. In most cases, one root will be smaller than the neighboring values  $\phi_{i-1,j}$  and  $\phi_{i,j-1}$  while the other will be larger. Taking the larger of the roots maintains the causal nature of the FMM and gives the proper solution for  $\phi_{i,j}$ . In the rare event that both roots are smaller than the neighboring values, each coordinate direction is considered separately, and the minimum  $\phi_{i,j}$  between the directions is taken. This guarantees a solution that maintains the causal nature of the FMM. This process is repeated until  $\phi$  has been computed everywhere in the domain. If the upwind stencil that corresponds to each zone  $(i, j)$  in the FMM calculation of  $\phi_{i,j}$  is stored, then virtually no additional cost is required in solving Eq. (18). The relation between  $\phi$  and  $\mathbf{n}$  is given by Eq. (4), a calculation that is performed right after  $\phi$  has been re-distanced to satisfy Eq. (19). This implies that the solution of the Eikonal equation for  $\alpha$  is equivalent to the solution of

$$\nabla\phi \cdot \nabla\alpha = 0, \tag{22}$$

where gradients are computed based on the stencils stored during the FMM procedure. Returning to our 2D example, Eq. (22) can be written at zone  $(i, j)$  as

$$\alpha_{i,j} \left( \frac{\delta\phi_x}{\Delta x^2} + \frac{\delta\phi_y}{\Delta y^2} \right) - \left( \alpha_{i-1,j} \frac{\delta\phi_x}{\Delta x^2} + \alpha_{i,j-1} \frac{\delta\phi_y}{\Delta y^2} \right) = 0, \tag{23}$$

where  $\delta\phi_x = \phi_{i,j} - \phi_{i-1,j}$  and  $\delta\phi_y = \phi_{i,j} - \phi_{i,j-1}$ . All values of  $\phi$  are known and  $\alpha_{i-1,j}$  and  $\alpha_{i,j-1}$  have already been computed, so  $\alpha_{i,j}$  is readily available from Eq. (23). The procedure outlined here works the same way in 3D, the differences being that more complex configurations are possible, and there can be as many as three neighbors used in the calculation of  $\phi_{i,j,k}$  and  $\alpha_{i,j,k}$ .

### 3.3. Summary of the proposed algorithm

For clarity, a summary of the proposed solution procedure is provided here.

- Advance  $\psi$  by solving Eq. (8). Details regarding its discretization are given by Desjardins et al. [6].
- Reconstruct  $\phi$  from  $\psi$  efficiently using a FMM. The stencils used in the FMM are stored.
- Compute  $\mathbf{n}$  at zone faces from  $\phi$  via Eq. (4) [6].
- Compute the curvature from  $\phi$  using least squares [6].
- Compute  $\alpha$  as described in Section 4.
- Perform a fast marching extension of  $\alpha$  to satisfy Eq. (18) in a band of cells containing the interface, using the stencils stored above.
- Re-initialize  $\psi$  by solving Eq. (17). Details regarding its discretization are given by Desjardins et al. [6].

## 4. Determination of $\alpha$

The coefficient  $\alpha$  should be defined in a way that reflects how much re-initialization is locally required. The amount of necessary re-initialization must depend on how much the level set has been locally deformed. Two sources of level set

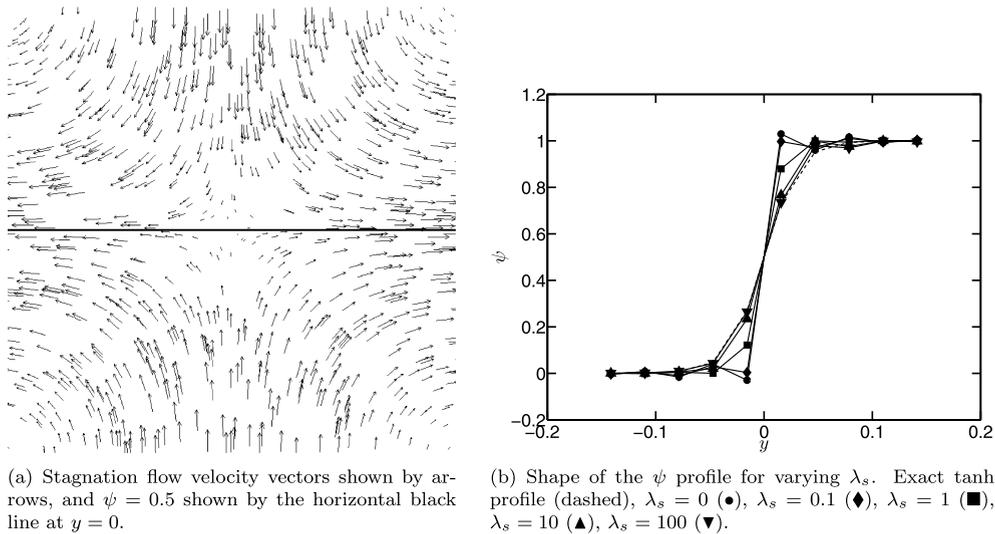


Fig. 4. Stagnation flow used for the calibration of  $\lambda_s$ .

deformation are identified: (1) the kinematics of the velocity field and (2) numerical diffusion associated with the evolution of  $\psi$  through the discrete solution of Eq. (8). Measures for both types of level set deformation are proposed in the following section.

#### 4.1. Deformation of $\psi$ by flow kinematics

The hyperbolic tangent level set function is distorted due to transport by the velocity field. This deformation occurs when velocity gradients cause relative motion of  $\psi$  iso-surfaces, as measured by the normal component of the strain rate tensor  $\mathbf{S}$ , where

$$\mathbf{S}(\mathbf{x}, t) = \frac{1}{2}(\nabla \mathbf{u}(\mathbf{x}, t) + \nabla \mathbf{u}(\mathbf{x}, t)^T) \quad (24)$$

is the symmetric portion of the velocity gradient tensor  $\nabla \mathbf{u}$ . We therefore define  $\alpha$  as

$$\alpha(\mathbf{x}_\Gamma, t) = \lambda_s |\mathbf{n}^T \cdot \mathbf{S}_\Gamma \cdot \mathbf{n}| \varepsilon, \quad (25)$$

where  $\mathbf{S}_\Gamma$  is the strain rate tensor evaluated at the interface, and  $\lambda_s$  is a free parameter that must be calibrated by gauging the amount of re-initialization required to maintain a hyperbolic tangent  $\psi$  field. In Eq. (25), the factor  $\varepsilon$  was added for dimensional consistency. In order to calibrate  $\lambda_s$ , a hyperbolic tangent level set is positioned in a stagnation flow, as shown in Fig. 4(a). The computational domain is composed of  $100 \times 100$  cells in a 2D region defined by  $x \in [-\pi/2, \pi/2]$  and  $y \in [-\pi/2, \pi/2]$ , with a velocity field prescribed as

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} +\cos(x + \pi/4) \sin(y - \pi/4) \\ -\sin(x + \pi/4) \cos(y - \pi/4) \end{pmatrix}. \quad (26)$$

Fig. 4(b) shows profiles of  $\psi(y)$  for different values of  $\lambda_s$  compared to an exact hyperbolic tangent profile. It is clear from Fig. 4(b) that no re-initialization ( $\lambda_s = 0$ ) leads to a heavily distorted  $\psi$  profile. Larger values of  $\lambda_s$  preserve the  $\psi$  profile, but very large values lead to significant increases in computational cost. It appears in Fig. 4(b) that for  $\lambda_s \geq 10$ , the profile shape is fairly insensitive to  $\lambda_s$ . Table 2 reports the  $L_2$  error in the profile compared to the exact hyperbolic tangent profile and the number of pseudo-time steps of Eq. (17) required for a range of values, defined as  $N_\tau = \max(\Delta \tilde{\tau}) / \Delta \tau_{\text{reinit}}$ . A value of  $\lambda_s = 10$  is chosen as a good compromise between preservation of the  $\psi$  profile and computational cost.

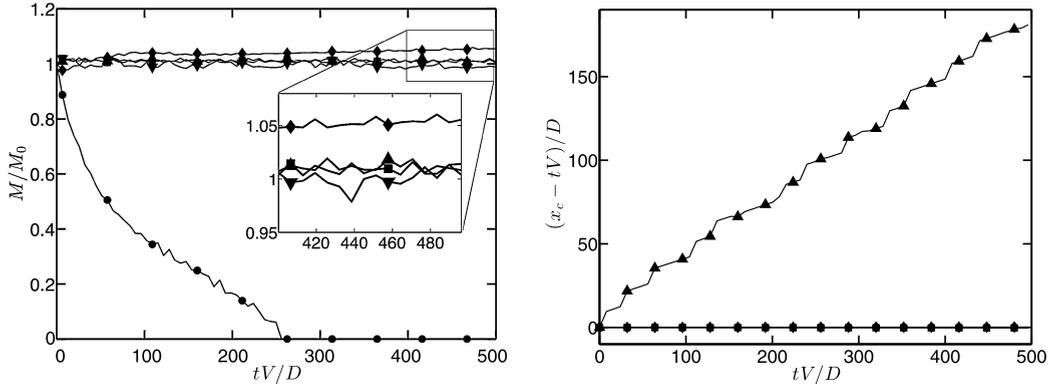
#### 4.2. Numerical diffusion of $\psi$

Even in the absence of normal strain,  $\psi$  may undergo numerical diffusion associated with the non-oscillatory schemes used to solve the transport equation [6]. Consequently, some re-initialization is required to keep the hyperbolic tangent function from becoming overly diffused in the direction of transport. The minimum value of re-initialization that is required in the absence of normal strain must depend on how much the level set has been transported in the normal direction. This can be measured by defining  $\alpha$  as

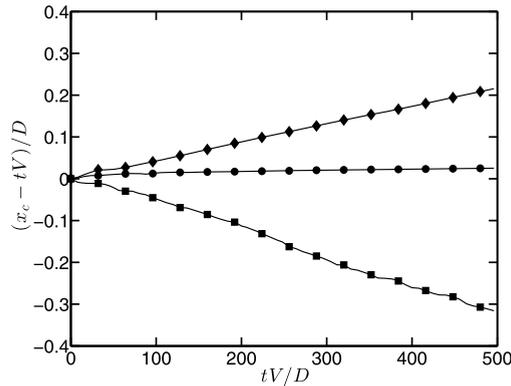
$$\alpha(\mathbf{x}_\Gamma, t) = \lambda_n |\mathbf{u}_\Gamma \cdot \mathbf{n}|, \quad (27)$$

**Table 2**  
 $L_2$  error of the  $\psi$  profile in the stagnation flow and number of pseudo-time steps  $N_\tau$  of Eq. (17) for different values of  $\lambda_s$ .

$\lambda_s$	$L_2$ error	$N_\tau$
0	$7.1676 \times 10^{-2}$	0
0.1	$5.7002 \times 10^{-2}$	1
1	$1.9271 \times 10^{-2}$	1
10	$2.2718 \times 10^{-3}$	1
100	$5.3449 \times 10^{-4}$	5



(a) Conservation of a coarsely represented drop for varying values of  $\lambda_n$ .  $\lambda_n = 0$  (●),  $\lambda_n = 0.1$  (◆),  $\lambda_n = 1$  (■),  $\lambda_n = 10$  (▲),  $\lambda_n = 100$  (▼).  
 (b) Normalized location of drop centroid  $x_c/D$  as a function of  $tV/D$ .  $\lambda_n = 0.1$  (●),  $\lambda_n = 1$  (◆),  $\lambda_n = 10$  (■),  $\lambda_n = 100$  (▲).

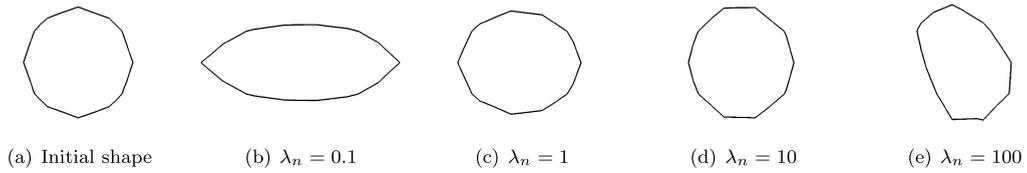


(c) The same as Fig. 5(b), without showing  $\lambda_n = 100$ .

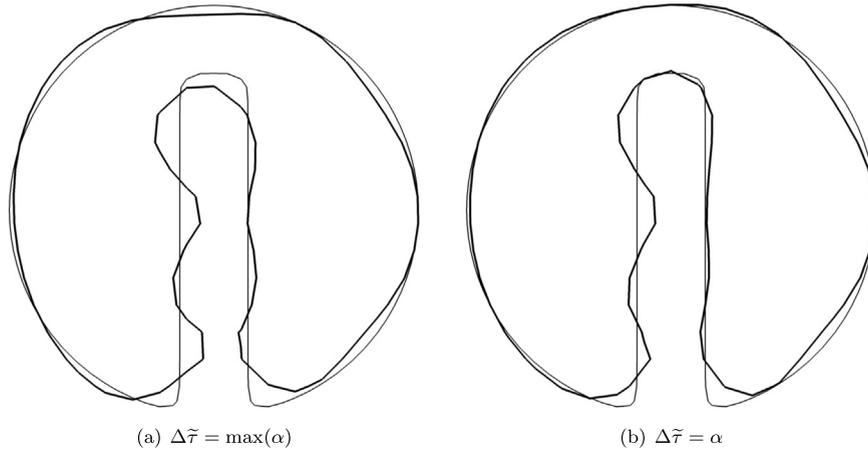
**Fig. 5.** Drop advection test case used to calibrate  $\lambda_n$ .

where  $\mathbf{u}_F$  is the velocity at the  $\psi = 0.5$  iso-surface. Similar to  $\lambda_s$  in Eq. (25),  $\lambda_n$  in Eq. (27) is a parameter that determines how much re-initialization is applied. The value of  $\lambda_n$  is calibrated by ensuring that liquid mass is conserved for a coarsely represented drop that is advected in a uniform flow field for long time scales. The drop is composed of 4 computational cells across its diameter. The drop is advected through a periodic unit box on a  $32 \times 32 \times 32$  grid for the equivalent of 500 drop diameters. Fig. 5(a) shows the liquid mass  $M$  normalized by the initial value  $M_0$  plotted against dimensionless time  $tV/D$ , where  $V$  is the velocity of the drop and  $D$  is its diameter. It is clear from Fig. 5(a) that if  $\lambda_n = 0$ , the entire drop disappears after traveling roughly 250 diameters, due to numerical diffusion. Increasing the value of  $\lambda_n$  preserves the drop, and  $\lambda_n \geq 1$  provides nearly perfect mass conservation. Refining the values of  $\lambda_n$  shows that  $\lambda_n = 0.5$  is a good minimum value for countering numerical diffusion in terms of mass conservation for this case.

The value of  $\lambda_n$  also affects the shape and location of the drop, as shown in Figs. 5(b) and 6. Fig. 6 shows the center plane of the final shape of the drop at  $tV/D = 500$  for different values of  $\lambda_n$ , along with the initial drop shape for comparison. Fig. 5(b) shows the deviation of the drop centroid  $x_c$  from perfect translation given by  $tV$  as a function of  $tV/D$ , normalized by drop diameter. The value  $(x_c - tV)/D$  remaining zero indicates no interference of re-initialization with the translation of the drop centroid by the prescribed velocity field. Fig. 5(c) shows  $(x_c - tV)/D$  for the first three values of  $\lambda_n$ . The slight drift in  $x_c$  for  $\lambda_n = 0.1$  is likely caused by the motion of the centroid due to the reshaping of the drop. Choosing  $\lambda_n = 1$  or  $\lambda_n = 10$  causes  $x_c$  to deviate on the order of a quarter of a diameter,  $\lambda_n = 10$  deviating more than  $\lambda_n = 1$ . Choosing  $\lambda_n = 100$



**Fig. 6.** Effect of  $\lambda_n$  on the shape of a coarsely defined drop advected in a uniform flow field at  $tV/D = 500$ .



**Fig. 7.** Effect of local re-initialization on the transport of Zalesak's disk on a  $50 \times 50$  mesh. The exact solution is given by the underlying thin line.

leads to significant deviation of  $x_c$ . Considering both displacement of  $x_c$  and the final shapes shown in Fig. 6, results indicate that  $\lambda_n$  on the order of 1 is a good choice.

Having isolated two types of level set deformation, the local interfacial value of  $\alpha$  is defined as

$$\alpha(\mathbf{x}_\Gamma, t) = \max(\lambda_n |\mathbf{u}_\Gamma \cdot \mathbf{n}|, \lambda_s |\mathbf{n}^\top \cdot \mathbf{S}_\Gamma \cdot \mathbf{n}| \varepsilon), \quad (28)$$

with the calibrated values of  $\lambda_s$  and  $\lambda_n$  set to 10 and 0.5, respectively. Eq. (28) provides the value of  $\alpha$  at the interface only, hence  $\alpha$  needs to be extended to the computational domain in which Eq. (17) is solved. In practice, the interfacial  $\alpha$  field is filtered with its immediate neighbors in each direction according to the equation

$$\alpha_{\text{filtered}} = \mathcal{F}_z \mathcal{F}_y \mathcal{F}_x \alpha, \quad (29)$$

where the operators  $\mathcal{F}_x$ ,  $\mathcal{F}_y$ , and  $\mathcal{F}_z$  represent three-point stencils in the  $x$ ,  $y$ , and  $z$  directions with weighting coefficients according to Simpson's Rule, i.e., the left and right weights are  $1/6$ , and the centered weight is  $2/3$ . Then, the fast marching solution of the Eikonal equation described in Section 3.2.1 is performed in a band of five cells on either side of the interface. This is sufficient, as ten cells across the interface is significantly wider than the region in which the compressive and diffusive fluxes in Eq. (17) are non-zero. Setting  $\Delta\tilde{\tau} = \alpha$  in this band allows flow kinematics to determine how much  $\psi$  has deformed and therefore how much it should be re-initialized, and it also prevents numerical diffusion from causing mass loss errors due to transport of the level set.

## 5. Numerical tests

### 5.1. Zalesak's disk

We examine the solid body rotation of Zalesak's disk [45] to see how local re-initialization affects the transport of a complex geometry with sharp corners. In this test, we transport and re-initialize a rotating notched circle with radius 0.15, notch width 0.05, notch height 0.25, and center initially located at  $(x, y) = (0, 0.25)$  in a unit box domain centered on  $(0, 0)$ . The disk is subject to the velocity field

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -2\pi y \\ +2\pi x \end{pmatrix}. \quad (30)$$

Tests are performed on  $50 \times 50$  and  $100 \times 100$  meshes to compare local re-initialization with  $\Delta\tilde{\tau} = \alpha$  to its global counterpart,  $\Delta\tilde{\tau} = \max(\alpha)$ . Figs. 7 and 8 indicate a slight improvement in the interface topology near the top of the disk when local re-initialization is used. Small differences are expected, as the solid-body rotation of Zalesak's disk does not lead to

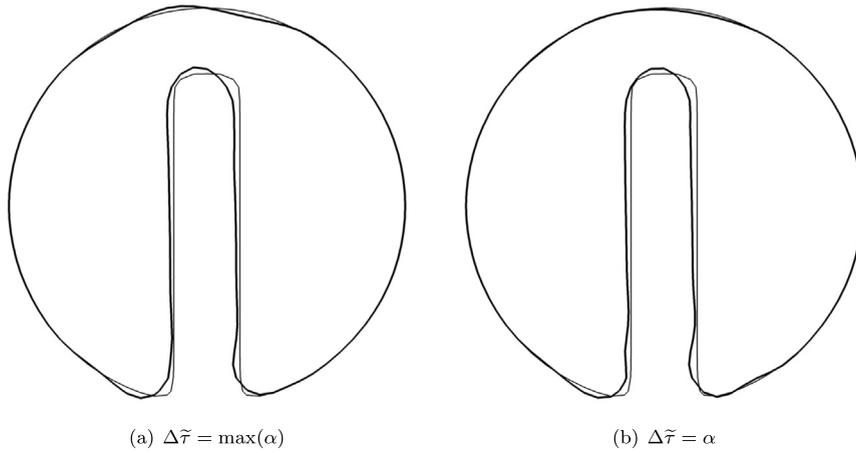


Fig. 8. Effect of local re-initialization on the transport of Zalesak’s disk on a 100 × 100 mesh. The exact solution is given by the underlying thin line.

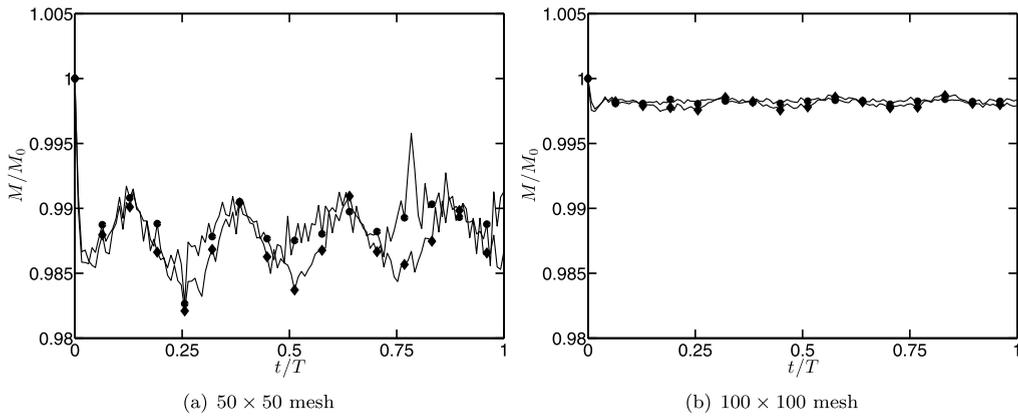


Fig. 9. Effect of local re-initialization on the mass beneath  $\psi = 0.5$  for Zalesak’s disk.  $\Delta\tilde{\tau} = \max(\alpha)$  ( $\bullet$ ),  $\Delta\tilde{\tau} = \alpha$  ( $\blacklozenge$ ).

deformation that varies locally to a high degree. The mass beneath  $\psi = 0.5$  was computed after one revolution for both meshes, and the differences between local and global treatments shown in Fig. 9 are negligible. The noted decrease in calculated mass from the initial condition is due to the inability of the mesh to resolve the sharp corners of the disk. This effect decreases when resolution is increased.

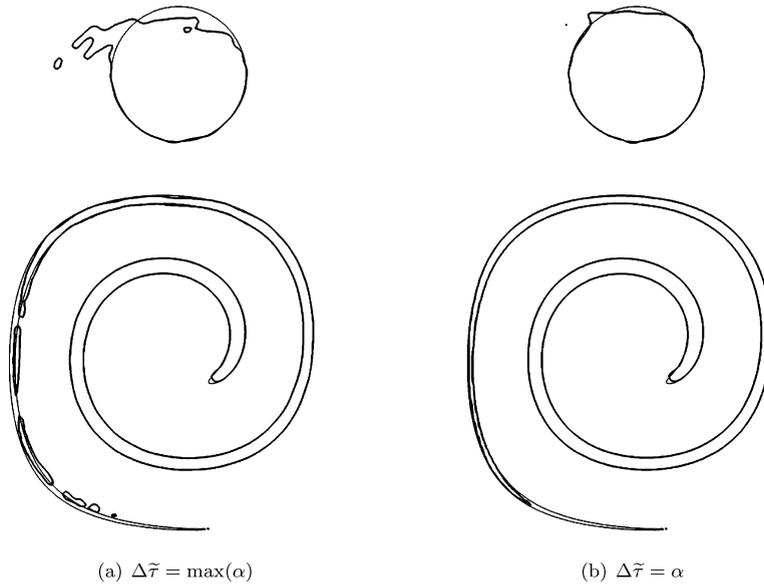
A uniform  $\alpha$  field is suitable for this type of flow, and results confirm that a varying  $\alpha$  does not degrade the solution. In fact  $\alpha$  is nearly uniform for this case, so the fact that the results are not worsened by possible spurious errors associated with a locally varying  $\alpha$  indicates that the FMM approach is enforcing Eq. (18) with sufficient accuracy.

### 5.2. Drop deformation

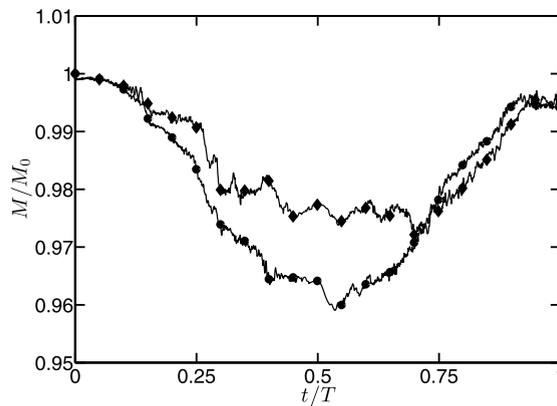
Here we examine the effect of local re-initialization in the context of a two-dimensional drop deforming in an unsteady vortex given by the velocity field

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -2 \sin^2(\pi x) \sin(\pi y) \cos(\pi y) \cos(\pi t/T) \\ +2 \sin^2(\pi y) \sin(\pi x) \cos(\pi x) \cos(\pi t/T) \end{pmatrix}, \tag{31}$$

where  $t$  is time and  $T$  is the full period of the deformation process. Thus, the drop experiences maximum deformation at  $t/T = 1/2$ , at which point the flow reverses. At  $t/T = 1$ , the drop should return to the initial condition. The tests are performed on a  $128 \times 128$  mesh. Fig. 10 compares local treatment to the case with  $\Delta\tilde{\tau} = \max(\alpha)$ . For each case, the initial shape is shown at  $t/T = 0$  overlaid by the final shape at  $t/T = 1$ . Maximum deformation at  $t/T = 1/2$  is shown beneath, underlaid by a  $1024 \times 1024$  solution for reference. In the absence of local treatment, re-initialization tends to form drops as the two fronts in the tail approach one another and the distance between them falls below the resolution capability of the method. It is clear that reducing the amount of re-initialization allows the fronts to get very close to one another with minimal topology changes, which reduces the deformation of the drop in the final state  $t/T = 1$ . This is also seen in the plots of mass beneath the  $\psi = 0.5$  iso-surface as a function of  $t/T$ , shown in Fig. 11. Local re-initialization has a reduced tendency to reform the interface into multiple structures, and thus mass errors associated with the tails of the hyperbolic



**Fig. 10.** Effect of local re-initialization on a drop deforming in a vortex. The initial condition is shown by the thin line, overlaid by the final shape. The shape at maximum deformation is shown beneath. Maximum deformation on a  $1024 \times 1024$  grid is shown for reference by the thin line.



**Fig. 11.** Liquid mass beneath  $\psi = 0.5$  for a drop deforming in a vortex.  $\Delta\tilde{\tau} = \max(\alpha)$  (●),  $\Delta\tilde{\tau} = \alpha$  (◆).

tangent function are not as significant. This is evident in the roughly 1.5% improvement in conservation of liquid mass at maximum deformation in Fig. 11.

Although a 1.5% conservation improvement is modest, it is stressed that the new formulation obtains this improvement at no additional computational cost. It requires the calculation of the rate of strain tensor and a matrix–vector multiplication, but the cost is minimal because this is only done in the cells containing the interface. The FMM extrapolation step requires a three-dimensional loop through the cells in the narrow band when updating  $\alpha$  within the band. Expressing the cost per time step with  $\Delta\tilde{\tau} = \alpha$  as  $t_{\text{local}}$  and the cost per time step with  $\Delta\tilde{\tau} = \max(\alpha)$  as  $t_{\text{global}}$ , we assess the relative performance by defining the total relative cost ratio  $T_{\text{rel}}$  as

$$T_{\text{rel}} = \frac{\sum_1^{N_t} t_{\text{local}}}{\sum_1^{N_t} t_{\text{global}}}, \quad (32)$$

where  $N_t$  is the number of time steps. A variety of timing tests were performed on Intel Xeon dual 6-core X5670 CPUs (a total of 12 cores per CPU), and it was observed that  $T_{\text{rel}}$  varied in a range of approximately  $\pm 1\%$ . For example,  $T_{\text{rel}} = 1.0120$  when run on 4 cores, but  $T_{\text{rel}} = 0.9902$  when run on 12 cores. In any case, such minor differences in timing are within our estimated timing error and vary slightly from one instance to another.

Even though the FMM extrapolation of  $\alpha$  is essentially free within the context of ACLS, it is of interest to assess the performance of the approach without the filtering step that enhances tangential smoothness of the  $\alpha$  field and without the extrapolation of  $\alpha$  that enforces the condition described by Eq. (18). There are errors associated with the extrapolation of  $\alpha$ ,

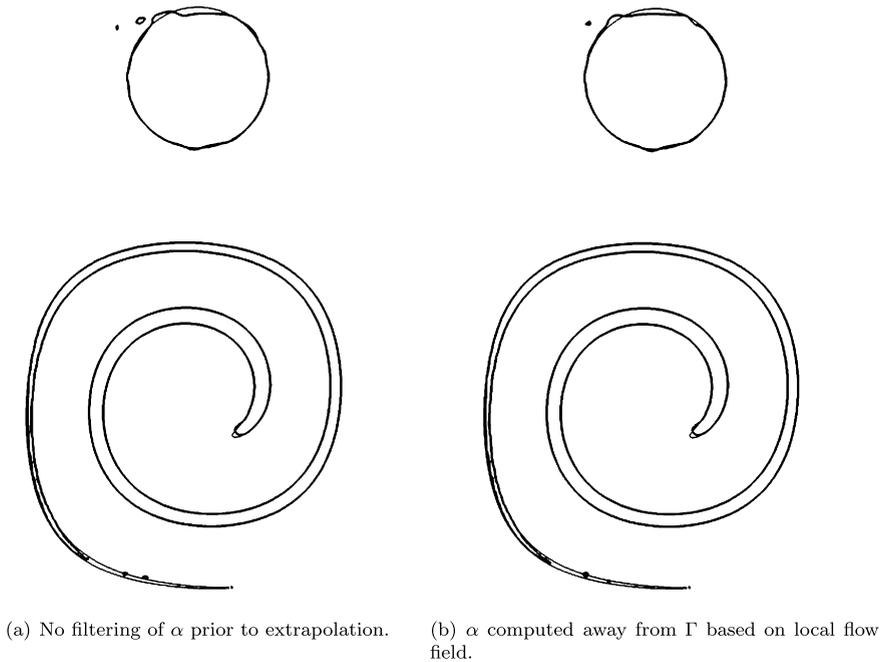


Fig. 12. Effect of filtering  $\alpha$  and enforcing Eq. (18) on the interface topology for the deformation test case.

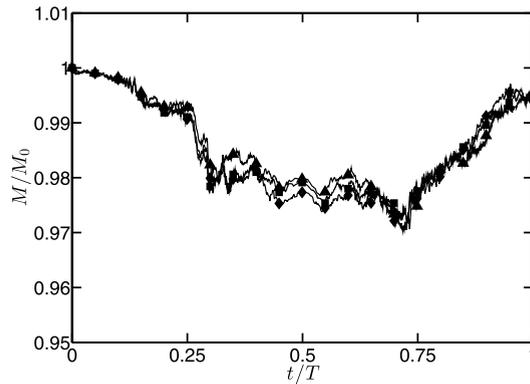
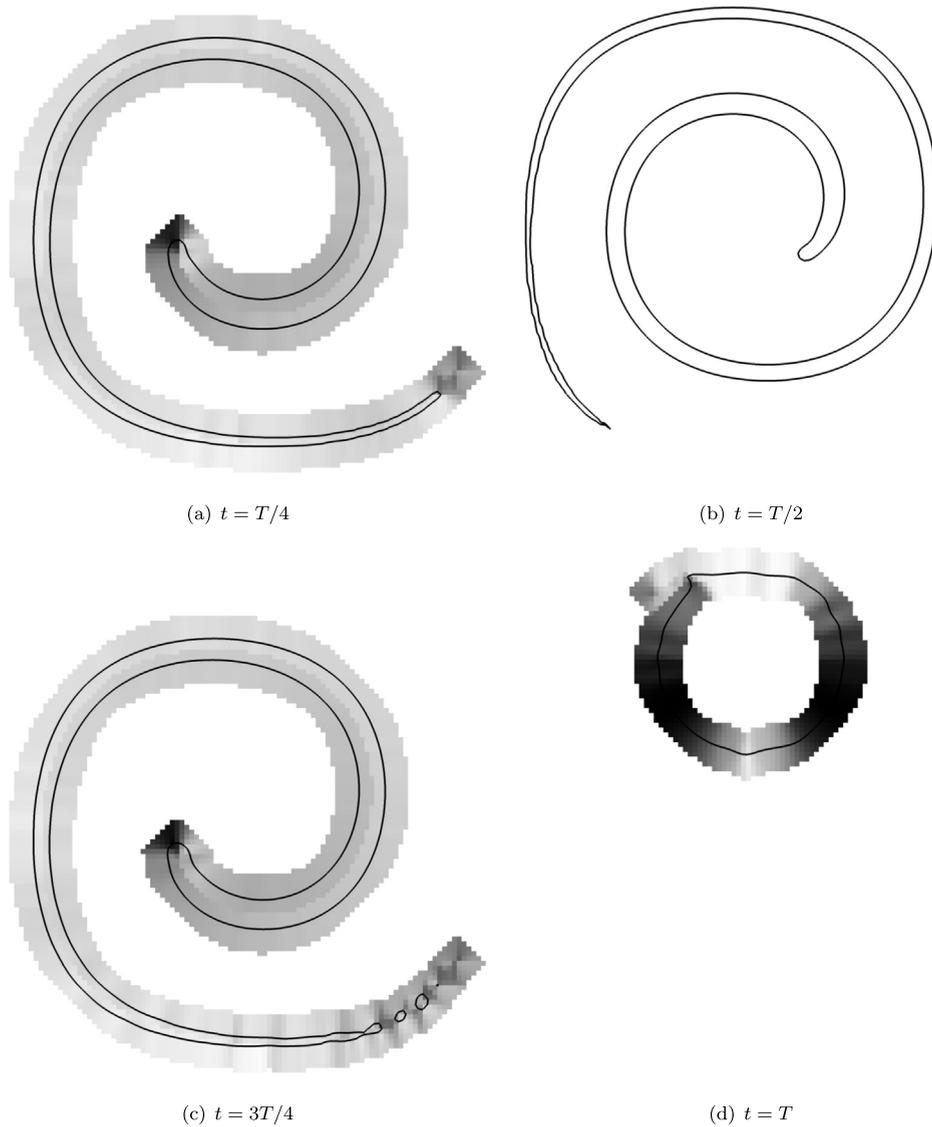


Fig. 13. Effect of filtering and extrapolating  $\alpha$  on liquid mass beneath  $\psi = 0.5$  for a drop deforming in a vortex. Filtered and extrapolated from  $\Gamma$  ( $\blacklozenge$ ), extrapolated from  $\Gamma$  without filtering ( $\blacksquare$ ), computed away from  $\Gamma$  based on local flow field ( $\blacktriangle$ ).

and it is possible that these errors may be comparable to the errors that would arise if Eq. (18) is not enforced, and the local velocity field is instead used to compute  $\alpha$  away from the interface. Fig. 12(a) shows the resulting interface when  $\alpha$  is not filtered, and Fig. 12(b) shows the resulting interface when the local velocity field is used to compute  $\alpha$  away from  $\Gamma$  rather than through extrapolation. Very minor differences in interface topology are observed, which is also the case for mass conservation shown in Fig. 13. From this we conclude that for this test case, local re-initialization is beneficial even in the absence of a mechanism for solving the Eikonal equation to enforce Eq. (18). Note that the velocity field given by Eq. (31) has very small gradients in the normal direction, and the difference between  $\alpha$  extrapolation and local calculation may be more significant for a different flow.

The deformation test case differs significantly from Zalesak’s disk in that there is very distinct coexistence of active and inactive regions of the interface. The ends of the ligament experience significant deformation, while the sides are very inactive. This is directly captured in the  $\alpha$  field, the progression of which is visible in Fig. 14. At maximum deformation the vortex reverses direction. When this happens the ligament stops moving, resulting in a temporarily uniform  $\alpha = 0$  field that does not artificially move the level set. The  $\mathbf{n} \cdot \nabla \alpha = 0$  property of the field is also visible in Fig. 14.



**Fig. 14.** Progression of the  $\alpha$  field for the drop deformation case on a  $128^2$  mesh. Darker tones indicate increasing values of  $\alpha$ .

## 6. Applications

### 6.1. Standing wave

Following the computational setup of Herrmann [11] and Desjardins et al. [6], we simulate the viscous damping of a surface wave. The simulation is performed on a two-dimensional domain defined by  $x \in [-\pi, \pi]$  and  $y \in [-\pi, \pi]$ , with the initial level set prescribed to be

$$\psi(x, y, t = 0) = \frac{1}{2} \left( \tanh \left( \frac{A_0 \cos(2\pi x/\eta) - y}{2\varepsilon} \right) + 1 \right), \quad (33)$$

where  $\eta = 2\pi$  is the perturbation wavelength and  $A_0 = 0.01\eta$  is the initial amplitude of the perturbation. Slip wall boundary conditions are used in the  $y$ -direction, and periodic conditions are used in  $x$ . Fluid densities are prescribed as  $\rho_1 = \rho_2 = 1$  and kinematic viscosities  $\nu_1 = \nu_2 = 0.064720863$ , providing unity ratios for both quantities. The multiphase Navier–Stokes solver NGA [5] is coupled to the proposed interface capture scheme. Results are shown for  $8 \times 8$ ,  $16 \times 16$ ,  $32 \times 32$ , and  $64 \times 64$  meshes in Fig. 15, compared to the analytical solution derived by Prosperetti [24]. Results are non-dimensionalized by  $\eta$  and the inviscid oscillation frequency  $\omega_0 = \sqrt{\sigma/(\rho_1 + \rho_2)}$ , where  $\sigma = 2$  is the surface tension coefficient. The amplitude error is defined as the amplitude difference between the computed and exact solutions, normalized by the initial amplitude  $A_0$ . The effect of local re-initialization on the amplitude error is shown in Fig. 16, demonstrating slight differences in interface

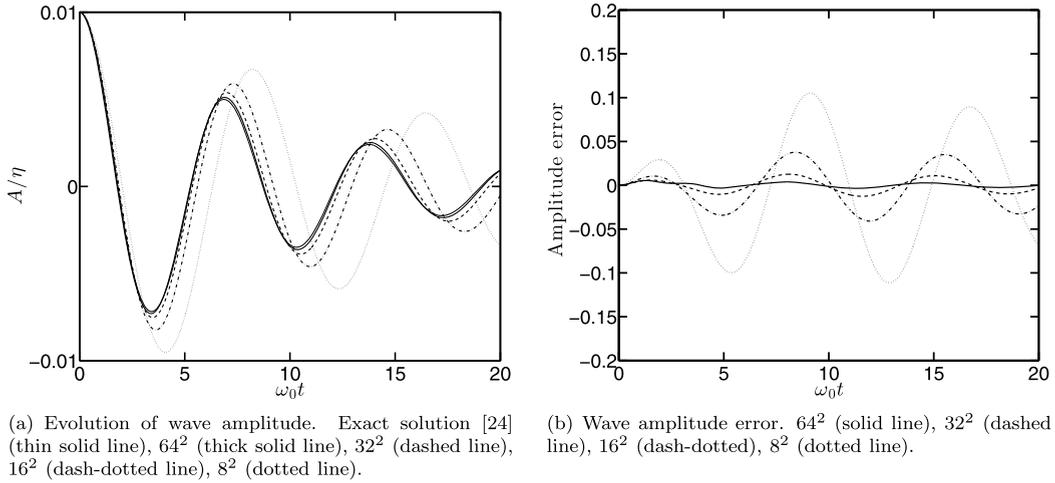


Fig. 15. Viscous damping of a free surface for different levels of mesh resolution.

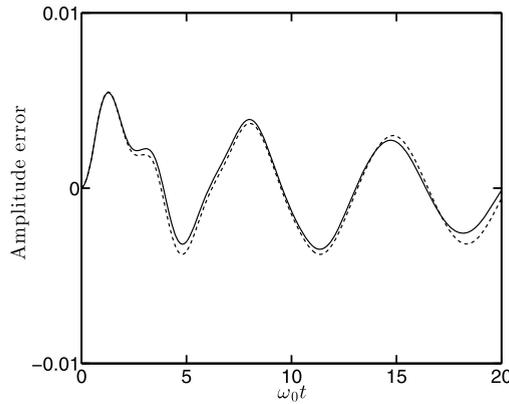


Fig. 16. Effect of local re-initialization on amplitude error on a  $64^2$  mesh.  $\Delta\tilde{\tau} = \alpha$  (solid line),  $\Delta\tilde{\tau} = \max(\alpha)$  (dashed line).

Table 3  
 $L_2$  norm of amplitude error on a  $64 \times 64$  mesh shown in Fig. 16.

	$L_2$ error
Local	$6.8130 \times 10^{-3}$
Global	$7.1797 \times 10^{-3}$

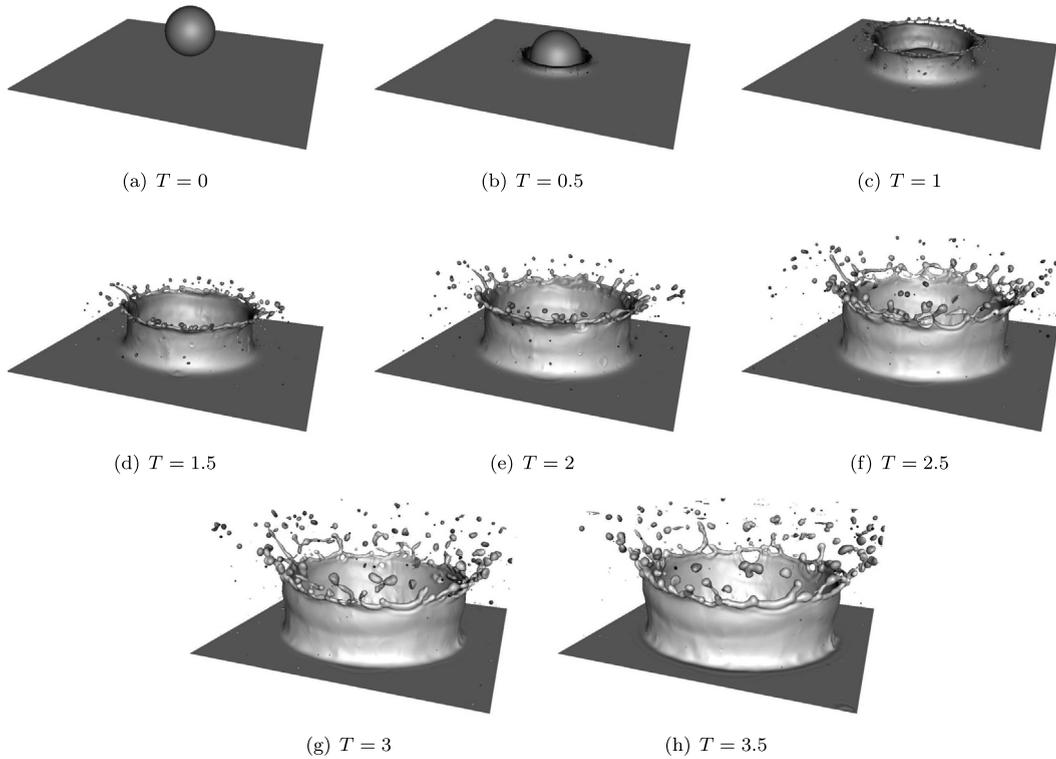
topology. Table 3 gives the  $L_2$  norm of the amplitude error shown in Fig. 16. Similar to Zalesak’s disk, we do not expect local treatment to improve the computation of the standing wave, as the flow kinematics deform the level set in a uniform manner. However, this test confirms that the varying  $\alpha$  field does not introduce spurious errors that degrade the results. Note that again, negligible differences in computational cost between the local and global formulations are observed for the standing wave simulation.

6.2. Drop impact

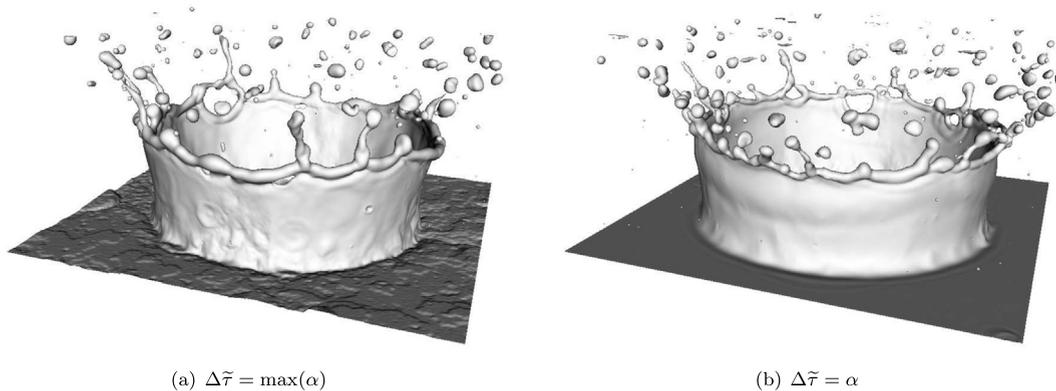
To further demonstrate the improvement obtained with the modified equation, we simulate a drop impacting a shallow liquid pool. Similar to the deformation test case of Section 5.2, we expect interface topology to benefit from spatially varying re-initialization due to the coexistence of active and inactive regions. The simulation performed uses the same parameters as the simulation of Rieber and Frohn [26], and these parameters are summarized in Table 4. The Weber number defined by the liquid density  $\rho_l$ , initial drop velocity  $V$ , drop diameter  $D$ , and surface tension  $\sigma$  is  $We = \rho_l V^2 D / \sigma$ . Similarly, the Ohnesorge number is defined as  $Oh = \mu_l / \sqrt{\sigma \rho_l D}$ , where  $\mu_l$  is the liquid viscosity. The pool depth  $h$  normalized by drop diameter is  $H = h/D$ , and  $T = tV/D$  is the dimensionless simulation time. The simulation is run for a time of  $\Delta T = 3.5$ . The multiphase Navier–Stokes solver NGA [5] is coupled to the proposed level set method and used on a  $512 \times 1024 \times 1024$

**Table 4**  
Parameters for the drop impact simulation.

We	Oh	H	$\Delta T$
598	0.0014	0.116	3.5



**Fig. 17.** Growth of the splashing lamella resulting from drop impact.

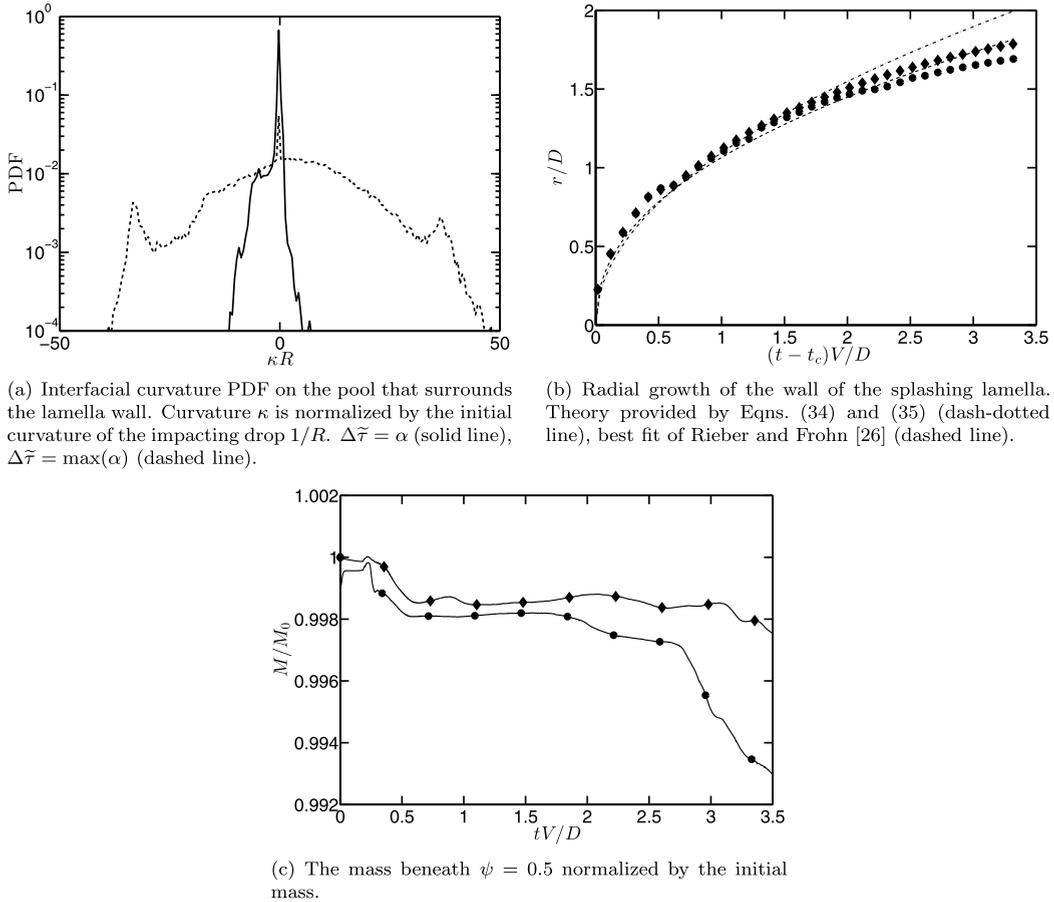


**Fig. 18.** The effect of local re-initialization on the structure of the splashing lamella and surrounding pool at  $T = 3.5$ .

mesh defined on the region  $x \in [0, 2.5D]$ ,  $y \in [-2.5D, 2.5D]$ , and  $z \in [-2.5D, 2.5D]$ . The  $x$  coordinate is aligned with the direction of impact. The splashing lamella that forms, often referred to as a “milk crown”, grows radially outward from the initial region of impact, as is evident in Fig. 17, where the liquid–gas interface is shown at different  $T$ . High-speed liquid is ejected from the impact region at early times. As the walls of the splashing lamella rise, the rim of the crown becomes unstable and ligaments form that eventually pinch off into droplets.

It has been shown theoretically by Yarin and Weiss [43] that the radial growth of the splashing lamella is determined by

$$\frac{r}{D} = k\sqrt{T_c} \tag{34}$$



**Fig. 19.** Curvature PDF of the pool surrounding the lamella wall, radial growth of the lamella wall, and conservation errors for the drop impact simulation.  $\Delta\tilde{\tau} = \alpha$  ( $\blacklozenge$ ),  $\Delta\tilde{\tau} = \max(\alpha)$  ( $\bullet$ ).

for  $T_c \gg 1$ , where  $T_c = (t - t_c)V/D$  is the dimensionless time measured from the time  $t_c$  when the drop makes first contact with the pool, and  $r$  is the radius of the lamella wall. Yarin and Weiss [43] provide the proportionality constant  $k$  depending only on the velocity distribution in the lamella wall. The observations of Rieber and Frohn [26] offer an improvement, given as

$$k = (6H)^{-1/4}. \tag{35}$$

Spurious interfacial corrugations due to excessive re-initialization in the inactive liquid pool affect the radial propagation of the lamella wall, as visualized in Fig. 18. Side-by-side results of the simulation with and without local re-initialization at  $T = 3.5$  are shown. The mechanism for error accumulation described in Section 3.1 leads to artificial motion of the level set that is clearly visible on the surrounding stationary interface. These spurious corrugations on the surrounding pool are quantified by computing the probability density function (PDF) of interfacial curvature outside of the lamella wall at  $T = 3.5$ , shown in Fig. 19(a). The curvature  $\kappa$  is normalized by  $1/R$ , which is the curvature of the initial liquid drop. Without local re-initialization, the PDF illustrates the tendency of the interface to acquire regions with curvature magnitude 30–40 times larger than the initial drop. The PDF that corresponds to local re-initialization much more closely resembles the PDF of the initially flat interface, which is a delta function at  $\kappa R = 0$ . The slight spread of the PDF is due to curvature in the surrounding pool generated by impacting drops, which are visible in Fig. 18(b).

Fig. 19(b) shows the radial growth at the base of the lamella wall for the simulation compared to the growth provided by Eqs. (34) and (35) and the best fit of Rieber and Frohn [26]. The simulation data agrees very well with the simulation data of Rieber and Frohn [26]. It is likely that the deviation from theory for high values of  $T_c$  is due to the finite domain size of the simulation. Local re-initialization also provides improvements in mass conservation for the same reasons as discussed in Section 5.2, as shown in Fig. 19(c).

Fig. 20 shows the relative wall clock time per time step of the simulation, including the relative time spent in the multiphase, velocity, and pressure solvers within the code. The total relative time  $T_{\text{rel}}$  defined by Eq. (32) is provided for each of the different solvers in Table 5, as well as the overall total relative time. This demonstrates that for this particular simulation, local re-initialization actually leads to a reduction in overall computational cost. This is explained by the tendency

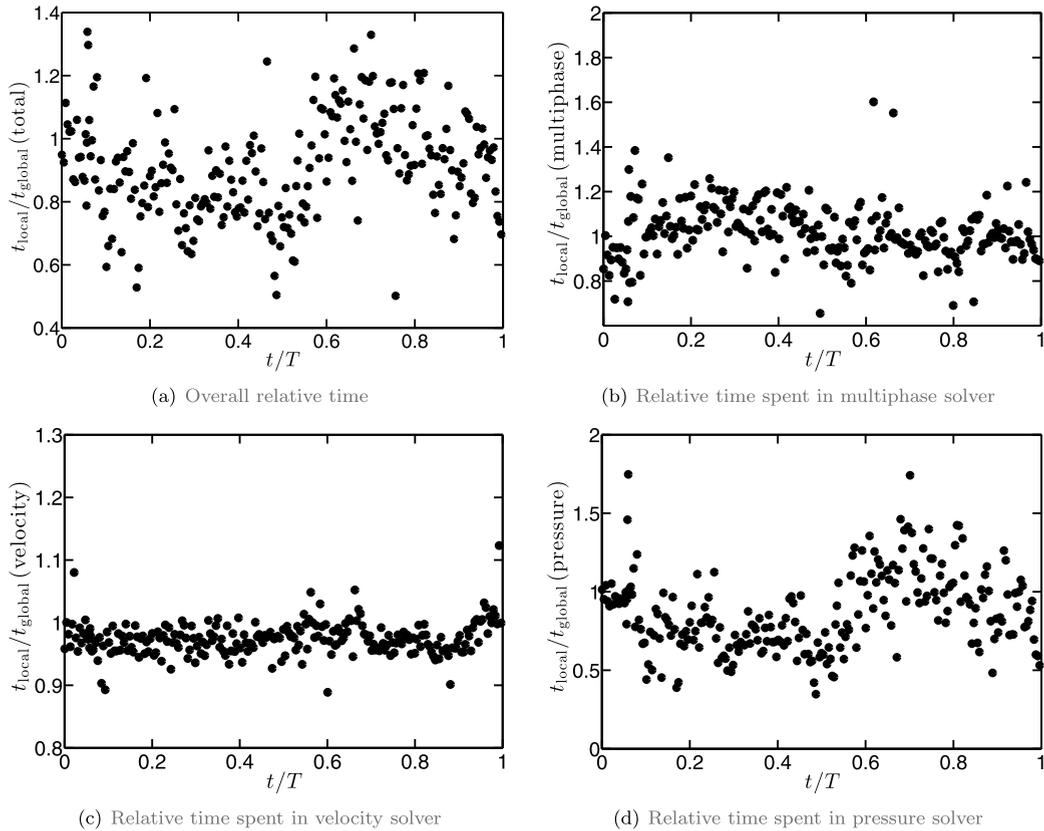


Fig. 20. Relative cost per time step for the different solvers in the code.

Table 5

Relative cost  $T_{\text{rel}}$  (defined by Eq. (32)) for the different solvers in the drop impact simulation.

	Total	Multiphase	Velocity	Pressure
$T_{\text{rel}}$	0.8821	1.0056	0.9770	0.8035

of the pressure Poisson equation, which is necessary to enforce the solenoidal condition for incompressible simulations, to become ill-conditioned in the presence of high density ratios. With a density ratio of  $10^3$  for this simulation, the smoother interface that results from local re-initialization leads to faster convergence of the pressure solver used within NGA. The relative timing for the multiphase and velocity solvers remains the same within our timing errors, and these minor changes in timing are dominated by the roughly 20% difference in Poisson solver convergence time, as convergence of the pressure system comprises roughly 60% of a given time step. Simulations to assess timing were run on 2112 computational cores on Titan, the Cray XK7 supercomputer at Oak Ridge National Laboratory's Leadership Computing Facility.

The formation of the initial sheet during drop impact is difficult to resolve. As the width of the sheet approaches the level of resolution, the tendency of re-initialization is to break the sheet into structures that are resolvable on the mesh. As is evident from the deformation test case in Section 5.2, the proposed method alleviates this tendency to some degree. Fig. 21 shows a top view of the interface at  $T = 0.7$ , illustrating that although the sheet is likely under-resolved, the new method has less of a tendency to break the sheet into multiple structures, leading to an increase in effective resolution.

## 7. Conclusion

A straightforward modification to the conservative re-initialization equation has been derived that allows for spatial variation in the amount of re-initialization of the conservative level set. A variety of methods and applications are based on the conservative re-initialization equation [6,14,17,18,23,28,33,40,44], and all are potentially subject to spurious interfacial motion when re-initialization fluxes move the level set in the direction of imperfect normals. Whether or not this becomes severe depends on the extent to which the flow at hand contains both active and inactive regions of interface. While this effect may not become severe for many multiphase flow applications, the milk crown presented here demonstrates a case for which it does. Schemes with more diffusive properties could be used in implementing the conservative level set method to help alleviate this issue, but this would just further distort the interface due to numerical errors rather than actual

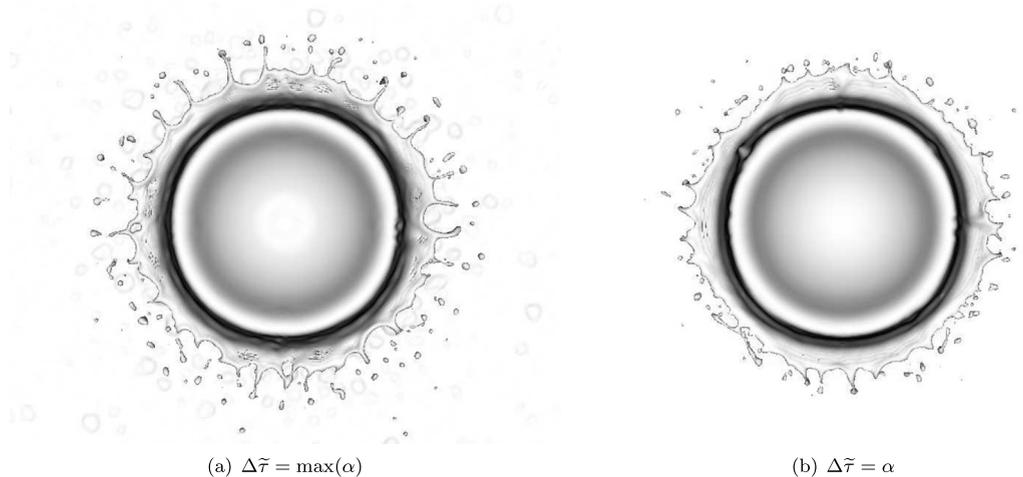


Fig. 21. Effect of local re-initialization on the formation of the liquid sheet at  $T = 0.7$ .

physical processes. This work offers a superior solution, which simply removes re-initialization altogether from regions in which it is not necessary. Upon solving the additional Eikonal equation that arises through an efficient fast marching method, the proposed equation becomes equivalent to the original equation, scaled by a factor that sets the local amount of re-initialization. Within the context of the accurate conservative level set method [6], this modification has minimal if any additional computational cost relative to the original method, and even shows a reduction in cost for one of the simulations contained herein.

The amount of local re-initialization is calibrated by both flow field kinematics and numerical diffusion that lead to level set deformation and diffusion. It is shown with Zalesak's disk and a viscously damped standing wave computation that the method does not degrade results for cases in which a uniform amount of re-initialization is suitable. The spatially variable nature of the method yields noted improvements for flows that contain both active and inactive regions of interface, as shown by a drop deforming in a vortex and a drop impacting a shallow pool. Results from the drop impact simulation indicate that the elimination of spurious interfacial corrugations in the pool surrounding the impact region improves the radial growth of the lamella wall, reduces mass conservation errors, and reduces computational cost by increasing the rate of convergence of the pressure solver due to better conditioning of the pressure Poisson equation. Given that these improvements are computationally free within the local framework, the proposed approach should be viewed as a useful extension of the conservative level set method.

## Acknowledgements

Computational resources at Oak Ridge National Laboratory's Leadership Computing Facility that contributed to this work are gratefully acknowledged.

## References

- [1] D. Adalsteinsson, J.A. Sethian, The fast construction of extension velocities in level set methods, *J. Comput. Phys.* 148 (1) (Jan. 1999) 2–22.
- [2] A. Albadawi, D. Donoghue, A. Robinson, D. Murray, Y. Delauré, On the analysis of bubble growth and detachment at low capillary and bond numbers using volume of fluid and level set methods, *Chem. Eng. Sci.* 90 (Mar. 2013) 77–91.
- [3] D. Chopp, Recent advances in the level set method, in: *Handbook of Biomedical Image Analysis*, Kluwer Academic, 2005, pp. 201–256 (Chapter 4).
- [4] D.L. Chopp, Another look at velocity extensions in the level set method, *SIAM J. Sci. Comput.* 31 (5) (2009) 3255–3273.
- [5] 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) (Jul. 2008) 7125–7159.
- [6] O. Desjardins, V. Moureau, H. Pitsch, An accurate conservative level set/ghost fluid method for simulating turbulent atomization, *J. Comput. Phys.* 227 (18) (Sep. 2008) 8395–8416.
- [7] D. Enright, R. Fedkiw, J. Ferziger, I. Mitchell, A hybrid particle level set method for improved interface capturing, *J. Comput. Phys.* 183 (1) (Nov. 2002) 83–116.
- [8] D. Enright, F. Losasso, R. Fedkiw, A fast and accurate semi-Lagrangian particle level set method, *Comput. Struct.* 83 (6–7) (Feb. 2005) 479–490.
- [9] D. Enright, S. Marschner, R. Fedkiw, Animation and rendering of complex water surfaces, in: *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, ACM Trans. Graph. 21 (Jul. 2002) 736.
- [10] A. Gravouil, N. Moës, T. Belytschko, Non-planar 3D crack growth by the extended finite element and level sets, Part II: Level set update, *Int. J. Numer. Methods Eng.* 53 (11) (2002) 2569–2586.
- [11] M. Herrmann, A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids, *J. Comput. Phys.* 227 (4) (Feb. 2008) 2674–2706.
- [12] S.E. Hieber, P. Koumoutsakos, A Lagrangian particle level set method, *J. Comput. Phys.* 210 (1) (Nov. 2005) 342–367.
- [13] C. Hirt, B. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, *J. Comput. Phys.* 39 (1) (Jan. 1981) 201–225.
- [14] Y. Lin, Two-phase electro-hydrodynamic flow modeling by a conservative level set model, *Electrophoresis* 34 (5) (Nov. 2013) 736–744.

- [15] N. Moës, A. Gravouil, T. Belytschko, Non-planar 3D crack growth by the extended finite element and level sets, Part I: Mechanical model, *Int. J. Numer. Methods Eng.* 53 (11) (Apr. 2002) 2549–2568.
- [16] W. Noh, P. Woodward, SLIC (simple line interface calculation), in: *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics*, Enschede, 1976, pp. 330–340.
- [17] E. Olsson, G. Kreiss, A conservative level set method for two phase flow, *J. Comput. Phys.* 210 (1) (Nov. 2005) 225–246.
- [18] E. Olsson, G. Kreiss, S. Zahedi, A conservative level set method for two phase flow II, *J. Comput. Phys.* 225 (1) (Jul. 2007) 785–807.
- [19] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations, *J. Comput. Phys.* 79 (1) (Nov. 1988) 12–49.
- [20] M. Owkes, O. Desjardins, A discontinuous Galerkin conservative level set scheme for interface capturing in multiphase flows, *J. Comput. Phys.* 249 (1) (Sep. 2013) 275–302.
- [21] B. Parker, D. Youngs, Two and three dimensional Eulerian simulation of fluid flow with material interfaces, UK Atomic Weapons Establishment, 1992.
- [22] D. Peng, B. Merriman, S. Osher, H. Zhao, M. Kang, A PDE-based fast local level set method, *J. Comput. Phys.* 155 (2) (Nov. 1999) 410–438.
- [23] H. Pitsch, O. Desjardins, G. Balarac, M. Ihme, Large-eddy simulation of turbulent reacting flows, *Prog. Aerosp. Sci.* 44 (6) (Aug. 2008) 466–478.
- [24] A. Prosperetti, Motion of two superposed viscous fluids, *Phys. Fluids* 24 (7) (1981) 1217–1223.
- [25] C.W. Rhee, L. Talbot, J.A. Sethian, Dynamical behaviour of a premixed turbulent open V-flame, *J. Fluid Mech.* 300 (1995) 87–115.
- [26] M. Rieber, A. Frohn, A numerical study on the mechanism of splashing, *Int. J. Heat Fluid Flow* 20 (5) (Oct. 1999) 455–461.
- [27] Y. Sato, B. Ničeno, A conservative local interface sharpening scheme for the constrained interpolation profile method, *Int. J. Numer. Methods Fluids* 70 (4) (Oct. 2012) 441–467.
- [28] Y. Sato, B. Ničeno, A new contact line treatment for a conservative level set method, *J. Comput. Phys.* 231 (10) (May 2012) 3887–3895.
- [29] J.A. Sethian, A fast marching level set method for monotonically advancing fronts, *Proc. Natl. Acad. Sci. USA* 93 (4) (Feb. 1996) 1591–1595.
- [30] J.A. Sethian, Fast marching methods, *SIAM Rev.* 41 (1999) 199–235.
- [31] J.A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, 1999.
- [32] J.A. Sethian, P. Smereka, Level set methods for fluid interfaces, *Annu. Rev. Fluid Mech.* 35 (1) (Jan. 2003) 341–372.
- [33] R.K. Shukla, C. Pantano, J.B. Freund, An interface capturing method for the simulation of multi-phase compressible flows, *J. Comput. Phys.* 229 (19) (Sep. 2010) 7411–7439.
- [34] J. Strain, Fast tree-based re-distancing for level set computations, *J. Comput. Phys.* 152 (2) (Jul. 1999) 664–686.
- [35] J. Strain, Tree methods for moving interfaces, *J. Comput. Phys.* 151 (2) (May 1999) 616–648.
- [36] J. Strain, A fast modular semi-Lagrangian method for moving interfaces, *J. Comput. Phys.* 161 (2) (Jul. 2000) 512–536.
- [37] J. Strain, A fast semi-Lagrangian contouring method for moving interfaces, *J. Comput. Phys.* 170 (1) (Jun. 2001) 373–394.
- [38] M. Sussman, P. Smereka, S. Osher, A level set method for computing solutions to incompressible two-phase flow, *J. Comput. Phys.* 114 (Sep. 1994) 146–159.
- [39] Y.-h.R. Tsai, Rapid and accurate computation of the distance function using grids, *J. Comput. Phys.* 178 (1) (May 2002) 175–195.
- [40] B. Van Poppel, O. Desjardins, J. Daily, A ghost fluid, level set methodology for simulating multiphase electrohydrodynamic flows with application to liquid fuel injection, *J. Comput. Phys.* 229 (20) (Oct. 2010) 7977–7996.
- [41] E.D. Villiers, A. Gosman, H.G. Weller, Large eddy simulation of primary diesel spray atomization, *SAE Transact.* 113 (3) (2004) 193–206.
- [42] H.G. Weller, The development of a new flame area combustion model using conditional averaging, Thermo-Fluids Section Report TF 9307, Tech. rep., Imperial College of Science, Technology and Medicine, 1993.
- [43] A. Yarin, D. Weiss, Impact of drops on solid surfaces: self-similar capillary waves, and splashing as a new type of kinematic discontinuity, *J. Fluid Mech.* 283 (1995) 141–173.
- [44] S. Zahedi, K. Gustavsson, G. Kreiss, A conservative level set method for contact line dynamics, *J. Comput. Phys.* 228 (17) (Sep. 2009) 6361–6375.
- [45] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, *J. Comput. Phys.* 31 (3) (Jun. 1979) 335–362.
- [46] J. Zhu, J.A. Sethian, Projection methods coupled to level set interface techniques, *J. Comput. Phys.* 102 (1) (Sep. 1992) 128–138.