



An accurate conservative level set/ghost fluid method for simulating turbulent atomization

Olivier Desjardins^{*}, Vincent Moureau, Heinz Pitsch

Department of Mechanical Engineering, Stanford University, CA 94305, USA

ARTICLE INFO

Article history:

Received 15 January 2008

Received in revised form 23 April 2008

Accepted 31 May 2008

Available online 12 June 2008

Keywords:

Multiphase flow

Incompressible flow

DNS

Conservative level set

Ghost fluid method

Implicit scheme

Primary atomization

Mass conservation

ABSTRACT

This paper presents a novel methodology for simulating incompressible two-phase flows by combining an improved version of the conservative level set technique introduced in [E. Olsson, G. Kreiss, A conservative level set method for two phase flow, *J. Comput. Phys.* 210 (2005) 225–246] with a ghost fluid approach. By employing a hyperbolic tangent level set function that is transported and re-initialized using fully conservative numerical schemes, mass conservation issues that are known to affect level set methods are greatly reduced. In order to improve the accuracy of the conservative level set method, high order numerical schemes are used. The overall robustness of the numerical approach is increased by computing the interface normals from a signed distance function reconstructed from the hyperbolic tangent level set by a fast marching method. The convergence of the curvature calculation is ensured by using a least squares reconstruction. The ghost fluid technique provides a way of handling the interfacial forces and large density jumps associated with two-phase flows with good accuracy, while avoiding artificial spreading of the interface. Since the proposed approach relies on partial differential equations, its implementation is straightforward in all coordinate systems, and it benefits from high parallel efficiency. The robustness and efficiency of the approach is further improved by using implicit schemes for the interface transport and re-initialization equations, as well as for the momentum solver. The performance of the method is assessed through both classical level set transport tests and simple two-phase flow examples including topology changes. It is then applied to simulate turbulent atomization of a liquid Diesel jet at $Re = 3000$. The conservation errors associated with the accurate conservative level set technique are shown to remain small even for this complex case.

© 2008 Elsevier Inc. All rights reserved.

1. Motivation and objectives

In most propulsion devices, the fuel is introduced in liquid form in a combustion chamber, where it undergoes atomization, evaporation, mixing with air, and chemical reactions in the combustion process. Since the atomization process governs the liquid droplet diameter distribution, it strongly affects both the subsequent evaporation and combustion. Consequently, full predictive capabilities for numerical tools will only be achieved once the atomization is accurately modeled. However, no satisfying models exist to this date, mostly because of the high complexity of the physics involved. Surface instabilities, ligament formation, ligament stretching and fragmentation, and droplet coalescence, all interact with turbulence to transform large scale coherent liquid structures into small scale droplets. Such a problem has scarcely been studied numerically, because it poses several great challenges.

^{*} Corresponding author. Fax: +1 650 725 3525.

E-mail address: desjardi@stanford.edu (O. Desjardins).

The first challenge lies in the fact that the material properties, such as density and viscosity, are different in the two phases. Hence, a flow solver needs to be capable of handling large density ratios, of the order of 40 for Diesel engines, up to several hundreds for aircraft engines. Second, one of the characteristics of liquid–gas flows is the presence of a surface tension force, which exists only at the interface between the liquid and the gas. The singular nature of this force leads to a difficult discretization. High robustness is therefore required from the flow solver. A third challenge lies in the interface localization and transport. While many approaches have been developed, they all suffer from various limitations, so that no clear gold standard exists today. Prerequisites for such methods include high accuracy, robustness, and the capability of accurately extracting the interface normals and curvature. Moreover, in the case of incompressible flows, the interface transport and localization should ensure that the volume of each phase is exactly conserved. Another challenge comes from the small scales that the atomization process produces. The formation of always smaller liquid structures leads to a multi-scale problem that requires high resolution to tackle, and that will generally generate liquid structures at the limit of numerical resolution.

Among the available strategies to numerically transport an interface, the volume-of-fluid (VOF) method [1] is one of the most popular. Because it relies on a liquid volume fraction scalar to represent the interface, this method ensures discrete mass conservation. However, since the VOF scalar is discontinuous across the interface, a specific geometric advection scheme is required, which puts constraints on both the accuracy of the method and the time step size. Additionally, accessing quantities such as the interface normals or curvature can prove challenging.

The front-tracking approach was introduced by Unverdi and Tryggvason [2]. It consists of discretizing the interface using an unstructured moving mesh that is transported in a Lagrangian fashion. While enjoying the benefit of a purely Lagrangian transport, this method requires frequent mesh rearrangements that affect the conservation of the liquid volume. Moreover, the parallelization of such a method is very challenging. The main limitation of this approach is the lack of automatic topology modification. Any interface merging or break-up events have to be handled manually, which can be a complex procedure, especially for three-dimensional simulations. Since topology changes are extremely frequent in primary atomization, front-tracking methods seem unadapted.

The level set method [3,4] aims at representing the interface implicitly by an iso-level of a smooth function. This smooth function is preserved with a re-initialization process. Simple Eulerian scalar transport schemes can be used to transport this function, and therefore highly accurate methods are available. Moreover, parallelization is straightforward and highly efficient, and the smoothness of the level set function makes the interface normals and curvature readily available. However, level set methods are typically plagued by mass conservation issues, for no inherent conservation property of the level set function exists. This represents a severe drawback to level set methods, considering that inaccuracies in the liquid mass of fuel in a reactive simulation could lead to large errors in quantities such as temperature, or pollutant mass fractions.

In order to improve the mass conservation property of the level set method, several hybrid approaches have been proposed. Enright et al. [5] proposed a particle level set method (PLS), where Lagrangian markers are employed to correct the front location predicted by Eulerian transport. Sussman et al. [6] proposed to couple a level set method with the VOF technique (CLSVOF), hence benefiting from both the good mass conservation property of the VOF approach and the smooth interface description of the level set method. While both these methods have been quite successful, they suffer from additional problems. Their cost is typically much greater than the cost of a simple level set method, because many particles per cell are required for an accurate solution for the PLS approach, and because of the time step size restrictions for the geometric transport of the VOF scalar for the CLSVOF method. Moreover, the complexity of these techniques is significantly greater than that of a classical level set method.

Another attempt to alleviate the mass conservation issue of level set methods has been to refine the mesh locally in order to decrease the numerical errors associated with level set transport and re-initialization. This refinement can be used for the level set equation only, such as in the case of the refined level set grid (RLSG) method of Herrmann [7], or it can be a standard arbitrary mesh refinement (AMR) approach, where the Navier–Stokes equations are also solved on the refined mesh [8]. While this approach ensures a good resolution of all structures, it remains both challenging to implement on parallel systems and significantly more expensive than classical methods. Moreover, the time step size in the case of strong local refinement is likely to be extremely restrictive.

Recently, Olsson and Kreiss [9] and Olsson et al. [10] proposed a simple modification to the level set method in order to reduce mass conservation errors while retaining the simplicity of the original method. By replacing the usual signed distance function of the classical level set approach by a hyperbolic tangent profile that is transported and re-initialized using conservative equations, they showed in Olsson and Kreiss [9] that the mass conservation errors could be reduced by an order of magnitude in comparison with the results obtained with a signed distance function. In Olsson et al. [10], they improved their re-initialization equation, and further studied their approach in the context of finite elements. The work presented here is based on the conservative level set method with the improved re-initialization equation from Olsson et al. [10]. However, the choice was made to remain in the context of finite difference methods. Starting from the observation that the conservative level set approach is difficult to use in the context of complex turbulent flows, several key modifications to this approach are introduced, resulting in both improved accuracy and robustness.

Different strategies have been developed to handle the large density ratio and the surface tension force in a flow solver. The continuum surface force approach (CSF) [11] spreads out both the density jump and the surface tension force over a few cells surrounding the interface in order to facilitate the numerical discretization. Consequently, this approach tends to misrepresent the smallest front structures. In the context of finite differences, the ghost fluid method (GFM) [12] provides a very

attractive way of handling discontinuities by using generalized Taylor series expansions that directly include these discontinuities. Because GFM explicitly deals with the density jump, the resulting discretization is not affected by the density ratio. Similarly, the surface tension force can be included directly in the form of a pressure jump, providing an adequate sharp numerical treatment of this singular term.

In this work, we will use GFM for the surface tension term as well as for the density jump in the pressure term. However, the discretization of the viscous terms using GFM is rather complex, and is challenging to implement implicitly. In order to alleviate this issue, we will use the CSF approach to discretize the viscous terms. For a turbulent flow where the viscous contribution is small in comparison with the convective terms, we do not expect this to significantly affect the quality of the solution. However, it allows us to use implicit solvers to robustly handle the viscous terms.

In Section 2, the governing equations for the flow and for the level set approach are presented. Section 3 will present the accurate conservative level set technique (ACLS) as well as the results of interface transport tests. The GFM approach to handle the discretization of the Navier–Stokes equations will then be described in Section 4. Section 5 will concentrate on simple validation cases, such as the computation of a Rayleigh instability and of damped surface waves, to assess the accuracy of the overall method. We will finally present the simulation of the turbulent atomization of a liquid Diesel jet in Section 6.

2. Mathematical formulation

2.1. Incompressible Navier–Stokes equations

In order to describe the flow in two phases, the incompressible form of the Navier–Stokes equations is introduced,

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \frac{1}{\rho} \nabla \cdot (\mu [\nabla \mathbf{u} + \nabla \mathbf{u}^t]) + \mathbf{g}, \quad (1)$$

where \mathbf{u} is the velocity field, ρ is the density, p is the pressure, \mathbf{g} is the gravitational acceleration, and μ is the dynamic viscosity. The continuity equation can be written in terms of the incompressibility constraint

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = \frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho = 0. \quad (2)$$

The interface Γ separates the liquid from the gaseous phase. In each phase, the material properties are constant, allowing us to write $\rho = \rho_l$ in the liquid phase, while $\rho = \rho_g$ in the gas phase. Similarly, $\mu = \mu_l$ in the liquid and $\mu = \mu_g$ in the gas. At the interface, the material properties are subject to a jump that is written $[\rho]_\Gamma = \rho_l - \rho_g$ and $[\mu]_\Gamma = \mu_l - \mu_g$ for the density and the viscosity, respectively. The velocity field is continuous across the interface, $[\mathbf{u}]_\Gamma = 0$. However, the pressure is not continuous between the two phases, and we can write

$$[p]_\Gamma = \sigma \kappa + 2[\mu]_\Gamma \mathbf{n}^t \cdot \nabla \mathbf{u} \cdot \mathbf{n}, \quad (3)$$

where σ is the surface tension, κ is the interface curvature, and \mathbf{n} is the interface normal.

2.2. Level set equation

In the level set approach, the interface is defined implicitly as an iso-surface of a smooth function. This approach benefits from many advantages, including automatic handling of topology changes, efficient parallelization, as well as easy and accurate access to the interface normals and curvature. In this section, two different level set functions will be introduced: the popular distance function proposed by Chopp [13], and the hyperbolic tangent function that was used by Olsson and Kreiss [9] in the context of their conservative level set method.

2.2.1. Distance level set

The classical level set technique relies on representing the interface implicitly as the zero level set of a smooth function ϕ chosen to be the signed distance from the interface, i.e.,

$$|\phi(\mathbf{x}, t)| = |\mathbf{x} - \mathbf{x}_\Gamma|, \quad (4)$$

where \mathbf{x}_Γ corresponds to the closest point on the interface from \mathbf{x} , and $\phi(\mathbf{x}, t) > 0$ on one side of the interface, and $\phi(\mathbf{x}, t) < 0$ on the other side. With this definition of the level set function, the interface itself corresponds to the $\phi(\mathbf{x}, t) = 0$ iso-surface. This choice leads to a very smooth ϕ -field, which can be adequately transported and differentiated to compute the normal vector \mathbf{n} and the curvature κ of the interface defined as

$$\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|} \quad (5)$$

and

$$\kappa = -\nabla \cdot \mathbf{n}. \quad (6)$$

The transport of the interface can simply be described by

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

However, transporting the interface using Eq. (7) will distort the level set function, and the smoothness of ϕ will be lost, leading to numerical problems. In order to ensure that ϕ remains smooth, an additional treatment is introduced to reshape ϕ into a distance function. This re-initialization of the distance profile can be performed using different procedures. The most common method is to solve a Hamilton–Jacobi equation [14],

$$\frac{\partial \phi}{\partial \tau} + S(|\nabla \phi| - 1) = 0, \quad (8)$$

where S is a modified sign function as in [15], and τ represents a pseudo-time. This equation can be discretized with high accuracy, therefore leading to an accurate reconstruction of the distance profile. However, it suffers from CFL limitations, making it prohibitively expensive in complex situations such as highly stretched meshes or cylindrical coordinates. In order to circumvent these limitations, Sethian [4] proposed a fast marching approach for the distance re-initialization based on solving locally $|\nabla \phi| = 1$ while employing only points closer to the interface in the numerical stencil. Using a heap sort algorithm, this procedure can be made highly efficient, even on parallel systems [16]. However, the accuracy of this approach is limited, and the re-distancing of the points closest to the interface induces a displacement of the front.

One of the main limitations of the distance function level set approach in the context of multiphase flows is that neither the level set transport nor the re-initialization inherently conserve the volume of the region enclosed by the zero level set. For liquid/gas flows, for example, this can lead to gains or losses in the mass of the liquid, which can lead to substantial errors in many applications.

2.2.2. Hyperbolic tangent level set

Instead of a signed distance function, Olsson and Kreiss [9] and Olsson et al. [10] employed a hyperbolic tangent function ψ defined as

$$\psi(\mathbf{x}, t) = \frac{1}{2} \left(\tanh \left(\frac{\phi(\mathbf{x}, t)}{2\epsilon} \right) + 1 \right), \quad (9)$$

where ϵ is a parameter that sets the thickness of the profile. Rather than defining the interface location by the iso-surface $\phi = 0$, it is now defined by the location of the $\psi = 0.5$ iso-surface. The transport of the interface can still be performed by solving the same equation as Eq. (7) for ψ . However, it can also be written in conservative form provided the velocity field \mathbf{u} is solenoidal, i.e. $\nabla \cdot \mathbf{u} = 0$, namely,

$$\frac{\partial \psi}{\partial t} + \nabla \cdot (\mathbf{u}\psi) = 0. \quad (10)$$

With the level set transport equation written in conservative form, and the given definition of ψ , it is clear that the scalar ψ should be a conserved quantity. As in the case of the level set function ϕ , nothing insures that solving Eq. (10) will preserve the form of the hyperbolic tangent profile ψ . As a result, an additional re-initialization equation needs to be introduced to re-establish the shape of the profile. As in [10], this equation is written

$$\frac{\partial \psi}{\partial \tau} + \nabla \cdot (\psi(1 - \psi)\mathbf{n}) = \nabla \cdot (\epsilon(\nabla \psi \cdot \mathbf{n})\mathbf{n}). \quad (11)$$

This equation is advanced in pseudo-time τ , it consists of a compression term on the left hand side that aims at sharpening the profile, and of a diffusion term on the right hand side that ensure the profile remains of characteristic thickness ϵ , and is therefore resolvable on a given mesh. It should be noted that this equation is also written in conservative form. As a result, solving successively for Eqs. (10) and (11) should accomplish the transport of the $\psi = 0.5$ iso-surface, preserve the shape of the hyperbolic tangent profile, and ensure the conservation of ψ .

For the sake of simplicity, the following symbolic definitions can be introduced:

$$\begin{aligned} (\text{scal}) &= \nabla \cdot (\mathbf{u}\psi), \\ (\text{comp}) &= \nabla \cdot (\psi(1 - \psi)\mathbf{n}), \text{ and} \\ (\text{diff}) &= \nabla \cdot (\epsilon(\nabla \psi \cdot \mathbf{n})\mathbf{n}). \end{aligned} \quad (12)$$

3. Accurate conservative level set (ACLS) method

3.1. Original conservative level set method

The conservative level set method of Olsson and Kreiss [9] and Olsson et al. [10] aims at reducing the mass conservation errors by exploiting the discrete conservation of the ψ -scalar. In the limit where the thickness ϵ of the hyperbolic tangent

profile ψ goes to zero, the volume integral of the ψ -function approaches the volume enclosed in the $\psi = 0.5$ iso-surface, namely,

$$\lim_{\epsilon \rightarrow 0} \int_V \psi(\mathbf{x}; t) d\mathbf{x} = \int_V H(\psi(\mathbf{x}; t) - 0.5) d\mathbf{x}, \quad (13)$$

where H is the Heaviside function and V is a volume. Since all the equations that need to be solved for ψ are conservative, discrete conservation of the volume enclosed in the interface becomes possible. Clearly, for a given numerical mesh with a spacing Δx , taking $\epsilon \ll \Delta x$ would lead to strong under-resolution of the hyperbolic tangent profile, and hence the numerical transport and re-initialization of ψ would suffer from severe numerical problems. In order to sufficiently resolve ψ , Olsson and Kreiss [9] proposed to use $\epsilon = \Delta x/2$, which leads to a hyperbolic tangent profile represented on two to three mesh points. With such a discretization of the profile, discretely solving Eqs. (10) and (11) becomes possible. However, the volumetric integral of ψ does not exactly correspond to the volume enclosed in the $\psi = 0.5$ iso-surface. Consequently, the volume enclosed in the $\psi = 0.5$ iso-surface will not be discretely conserved. Olsson and Kreiss [9] observed however that such an approach greatly reduced the conservation errors. The underlying conservation of ψ provides an anchor to the $\psi = 0.5$ iso-surface, preventing the accumulation of transport and re-initialization inaccuracies leading to large mass conservation errors. In their numerical tests, Olsson and Kreiss [9] obtained very encouraging results where discrete conservation errors were reduced by an order of magnitude in comparison to classical level set approaches.

3.2. Computation of the interface normals

In Olsson and Kreiss [9], it was mentioned that the choice of numerical method for the transport of the ψ -quantity was based on three considerations.

First, the discrete conservation of the transport should be ensured. Second, no spurious oscillations should be introduced, and third, the thickness of the hyperbolic tangent profile should be kept constant. The first point is straightforward, and satisfied by most numerical methods. Similarly, the third point can be ensured by solving the re-initialization equation for the hyperbolic tangent profile, Eq. (11). However, the second point is difficult to enforce in general, and it requires non-oscillatory transport schemes, which are typically expensive. Indeed, the cost of total variation diminishing (TVD) or total variation bounded (TVB) schemes is typically more than the cost of simple non-TVD(B) transport schemes. Note that this cost could be significantly reduced by using low order TVD schemes away from the interface, such as a first order upwind scheme. In addition, achieving the TVD(B) property constitutes a challenge, especially for high order schemes. For example, WENO-type schemes [17,18] are not TVD: most likely they are at best only TVB when combined with a TVD Runge–Kutta time integration [3]. Moreover, the effectiveness of non-oscillatory schemes is strongly conditioned on the divergence-free quality of the velocity field. In complex, realistic turbulence simulations, ensuring that the velocity field is discretely divergence free to machine accuracy is a challenge. As a result, we can say that avoiding spurious oscillations, although desirable, is simply impossible to achieve in general. Consequently, the robustness of the method should not be based on this property.

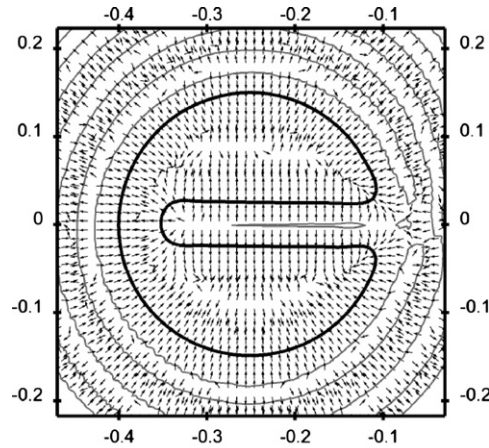
A convenient method to compute the normal vectors is to write

$$\mathbf{n} = \frac{\nabla \psi}{|\nabla \psi|}, \quad (14)$$

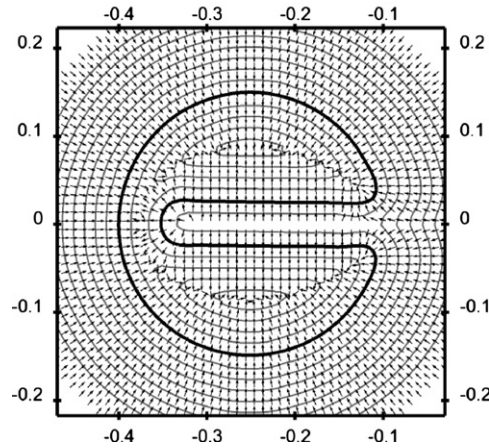
as in Olsson and Kreiss [9] and Olsson et al. [10]. However, this approach is strongly sensitive to spurious oscillations in the ψ -field.

Indeed, an oscillation in ψ will appear as a large change in direction of the normal vector. As a result, the normals obtained by Eq. (14) are not appropriate to use in the re-initialization equation, Eq. (11). As this equation contains a compression term that moves the level set scalar ψ along the directions defined by the normal vectors in order to re-form a hyperbolic tangent function, we can expect that having normals that point in the wrong direction will lead to severe numerical difficulties. More precisely, (comp) will create an accumulation of ψ where the normal vectors are facing each other. In the presence of parasitic oscillations of the normal vectors, this means that spots of the scalar ψ will form spuriously in the domain, leading to an unphysical displacement of the liquid mass, similar to jetsam/flotsam problems of some VOF methods [1]. To illustrate this point, a few steps of Zalesak's disk [19] problem are computed. This test case, for which the detailed parameters will be given in Section 3.7.2, is often employed to assess the accuracy of level set methods. No re-initialization is used, the level set function is simply transported using a non-TVD(B) scheme. It can be observed in Fig. 1a that the resulting normal vectors are alternating direction, as can be expected from taking the gradient of a field with spurious oscillations. The consequences of this, including the formation of jetsam/flotsam, are severe and will be shown in Section 3.7.2.

In order to remedy this problem, we propose to first recompute ϕ from the ψ -function using a standard re-distancing algorithm, then to use Eq. (5) to compute the normal from the smooth, reconstructed distance function ϕ . This distance reconstruction can be performed efficiently using a fast marching method (FMM), therefore, it does not affect the overall cost of the method significantly. The specific cost increase due to the FMM will be discussed in comparison with the cost of a non-oscillatory scheme in the following section. Moreover, the distance ϕ for the points closest to the interface can simply be obtained by inverting the hyperbolic tangent function, meaning that no spurious displacement of the interface will be induced by this operation. It is then straightforward to access the interface normals by Eq. (5). It can be seen in Fig. 1b that the resulting normals are smooth, and that they are perfectly useable to perform the re-initialization.



(a) Normals from ψ . Interface location (black line), and iso-contours of ψ (grey lines).



(b) Normals from ϕ . Interface location (black line), and iso-contours of ϕ (grey lines).

Fig. 1. Computation of interface normals in the presence of spurious oscillations in ψ .

3.3. High order level set transport

Having modified the way the normal vectors are computed, the non-oscillatory property of the level set transport becomes unnecessary. Therefore, we can take advantage of fast, high order, non-TVD(B) scalar transport schemes. A commonly used approach is the High Order Upstream Central (HOUC- n , where n is the order of the scheme) class of schemes, employed for example in [8] for level set transport. These schemes are implemented in the context of a numerical code developed for accurate simulations of turbulent reactive flows, NGA [20]. Following the notations used in [20], the n th order level set transport scheme can be written as

$$(\text{scal-}n) = \sum_{i=1}^3 \left(\frac{1}{J} \frac{\delta_{2\text{nd}}}{\delta_{2\text{nd}} \zeta_i} \left[\frac{J}{h_i} u_i \bar{\psi}^{\zeta_i} \right] \right), \quad (15)$$

where u_i is the i th component of the velocity vector, ζ_i represents the computational space, which is related to physical space x_i through $h_i = dx_i/d\zeta_i$ and $J = \prod_{i=1}^3 h_i$. The second order interpolation operator is defined for a variable α by

$$\bar{\alpha}^{2\text{nd} \zeta_1}(\zeta_1, \zeta_2, \zeta_3) = \frac{\alpha(\zeta_1 + 1/2, \zeta_2, \zeta_3) + \alpha(\zeta_1 - 1/2, \zeta_2, \zeta_3)}{2}, \quad (16)$$

and the second order differentiation operator is defined by

$$\frac{\delta_{2\text{nd}} \alpha}{\delta_{2\text{nd}} \zeta_1}(\zeta_1, \zeta_2, \zeta_3) = \frac{\alpha(\zeta_1 + 1/2, \zeta_2, \zeta_3) - \alpha(\zeta_1 - 1/2, \zeta_2, \zeta_3)}{2}. \quad (17)$$

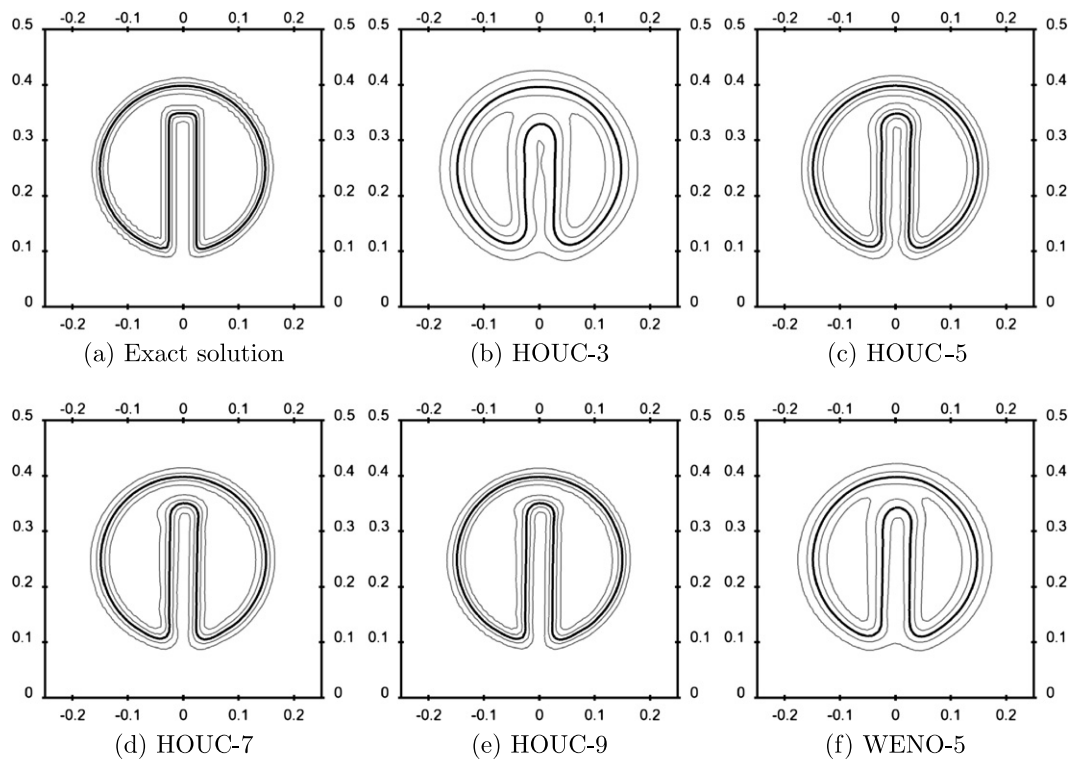


Fig. 2. Contours of ψ for the transport without re-initialization of Zalesak's disk.

$\bar{\psi}_{\zeta_i}$ is an n th order interpolation of the ψ variable to the i -direction cell face, which can be based on a HOUC approach or a WENO-type scheme [17,18]. In Fig. 2, the importance of using accurate transport schemes can be assessed by comparing the quality of the rotated Zalesak's disk on a 100×100 mesh. As expected, the accuracy of the transported ψ increases as the order of accuracy of the chosen scheme increases. Despite the fact that none of the HOUC schemes used are non-oscillatory, the solution after one rotation remains acceptable, without being noticeably affected by spurious oscillations. Moreover, the sharpness of the profile is much better preserved when the order of accuracy of the transport scheme is higher, and when non-TVD(B) schemes are preferred to a TVB scheme such as WENO-5. Having a more accurate transport scheme will allow to rely much less on the re-initialization equation. Instead of spreading out the profile due to transport inaccuracies, and then re-sharpen it using the re-initialization equation, it is highly beneficial to try to keep the profile as close to the original hyperbolic tangent as possible. In order to find the right compromise, it is instructive to compare the computational cost associated with the individual transport schemes. The cost of the different schemes used in Fig. 2 is compared in Table 1. It can be seen that the non-oscillatory scheme is by far the most expensive scheme, even though its actual accuracy should theoretically be between third and fifth order. On the other hand, if one accepts to use other non-TVD(B) schemes, a ninth order HOUC can be used for notably less cost than the fifth order WENO scheme.

It is also interesting to compare the actual cost of the entire procedure, including the cost of the FMM that is used prior to computing the normal vectors. Table 2 shows that the additional cost of the FMM remains low, as long as a banded approach is used, in the case where the fifth order HOUC is used for scalar transport. As a conclusion, even though an additional step has to be performed compared to the classical conservative level set method, the cost of the full proposed approach is still lower than that of using a lower order non-oscillatory scheme. It is also more accurate, and much more robust, since it can be argued that even a non-oscillatory scheme will lead to spurious oscillations in the presence of a complex, turbulent field that is not discretely divergence free.

Table 1
Cost of the different scalar schemes tested for the transport of Zalesak's disk

Scalar scheme	Time (seconds per time step)
HOUC-3	0.0440
HOUC-5	0.0607
HOUC-7	0.0798
HOUC-9	0.0997
WENO-5	0.1177

Table 2Additional cost of the fast marching method performed on n bands (FMM(n)) for Zalesak's disk

Scalar scheme	Time (seconds per time step)	Increase in cost (%)
Houc-5	0.0607	–
Houc-5 + FMM(5)	0.0645	6.3
Houc-5 + FMM(10)	0.0712	17.3
Houc-5 + FMM(20)	0.0839	38.2

3.4. Conservative re-initialization

Now that we have established how to transport the level set function ψ , and how to extract the normal vectors \mathbf{n} from it, the actual discretization of the re-initialization step can be discussed. The re-initialization equation for the hyperbolic tangent level set given in Eq. (11) is not the exact same form that was employed by Olsson and Kreiss [9] and Olsson et al. [10] performed this discretization in the context of finite elements. For finite differences, we observed that this step requires specific attention in order to avoid using overly large stencils, that would be detrimental to the robustness and accuracy of the solution. The values of the level set functions ϕ and ψ are stored at the cell centers, but the compressive flux $\mathbf{F}_C = \psi(1 - \psi)\mathbf{n}$ and the diffusive flux $\mathbf{F}_D = \epsilon(\nabla\psi \cdot \mathbf{n})\mathbf{n}$ need to be computed at the cell faces, in order to obtain cell centered residuals when taking the divergence of the fluxes. Consequently, if a standard centered differencing approach is used to compute \mathbf{n} from ϕ , as shown in Fig. 3a, computing \mathbf{F}_C and \mathbf{F}_D will require one additional interpolation, leading to an effective five points stencil in one dimension. This was found to be a poor discretization strategy subject to spurious oscillations. In order to ensure the compactness of the discretization stencils, the concept of face normals was introduced instead. These face normal vectors can be computed using more compact stencils, as shown in Fig. 3. Using the notations employed in [20], the x_i -component of the x_i -face gradient of a scalar quantity a can be written as

$$(\nabla^{x_i} a)_{x_j} = \begin{cases} \frac{\delta_{2nd} a}{\delta_{2nd} x_i} & \text{for } i = j, \\ \frac{\delta_{2nd} a}{\delta_{2nd} x_j} \bar{a}^{2nd x_i} & \text{for } i \neq j, \end{cases} \quad (18)$$

where $\delta_{2nd} a / \delta_{2nd} x_i$ 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. These face gradients are used to compute the normals from ϕ as well as the $\nabla\psi$ term in the (diff) term of Eq. (11). The x_i -face normal vector can then be defined by

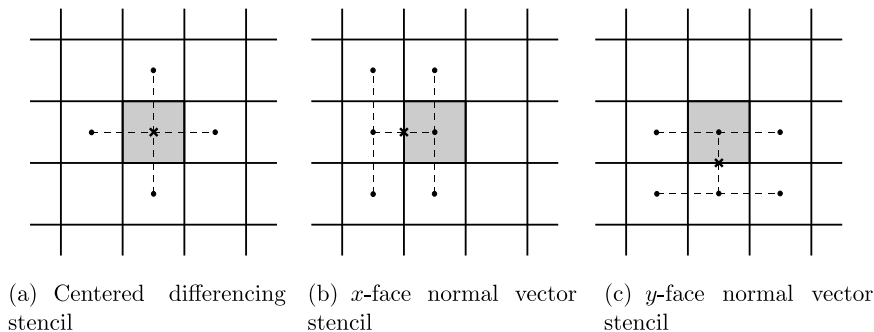
$$\mathbf{n}^{x_i} = \frac{\nabla^{x_i} a}{|\nabla^{x_i} a|}. \quad (19)$$

The compression term of Eq. (11) is discretized with second order accuracy in computational space, so as to ensure the discrete equivalence of the divergence operator used for the continuity equation and the one used for the re-initialization equation. The discrete version of (comp) reads

$$(\text{comp} - 2) = \sum_{i=1}^3 \left(\frac{1}{J} \frac{\delta_{2nd}}{\delta_{2nd} \zeta_i} \left[\frac{J}{h_i} \mathbf{n}^{x_i} \overline{\psi(1 - \psi)^{2nd \zeta_i}} \right] \right). \quad (20)$$

The diffusive term of Eq. (11) is discretized similarly by writing

$$(\text{diff} - 2) = \sum_{i=1}^3 \left(\frac{1}{J} \frac{\delta_{2nd}}{\delta_{2nd} \zeta_i} \left[\frac{J}{h_i} \epsilon \mathbf{n}^{x_i} (\mathbf{n}^{x_i} \cdot \nabla^{x_i} \psi) \right] \right). \quad (21)$$

**Fig. 3.** Discretization of the normals.

With this approach, a compact scheme can be used to discretize the full conservative re-initialization equation: 3 points are used in one dimension, 9 points in two dimensions and 27 points in three dimensions. This scheme is discretely conservative, second order accurate, and has been found to be robust in our test cases.

It should be noted that this conservative re-initialization converges very fast. Indeed, as noticed by Olsson et al. [10], with a choice of parameters following $\Delta x \sim \Delta t \sim \epsilon$, one or two time steps are enough to reach steady state. The spreading of the hyperbolic tangent induced by one time step of transport (Eq. 10) occurs on a length scale which can be expressed as $l_{\text{conv}} \approx \alpha_{\text{conv}} \Delta x$, where α_{conv} represents the convective CFL number. Similarly, the compression term (comp) of the conservative re-initialization (Eq. 11) for p time steps $\Delta \tau$ will displace the ψ -scalar over a length scale $l_{\text{comp}} \approx \alpha_{\text{comp}} p \Delta x$, where α_{comp} represents the compressive CFL number for the conservative re-initialization. To ensure that the displaced ψ -scalar is properly reshaped into a hyperbolic tangent function, it is desirable to have $l_{\text{comp}} \approx l_{\text{conv}}$, which leads to $p \alpha_{\text{comp}} \approx \alpha_{\text{conv}}$. For the sake of simplicity, the choice is made to always use $p = 2$, and $\alpha_{\text{comp}} \approx \alpha_{\text{conv}}/2$, instead of checking the convergence of the re-initialization equation at each step. This approach has been found to be sufficient to ensure the proper behavior of the re-initialization equation, as demonstrated by the examples discussed below.

3.5. Curvature computation

In order to accurately predict surface tension forces, it is of fundamental importance to have a converging curvature computation. The curvature could be directly computed from the hyperbolic tangent level set function ψ , but following the previous discussions about the presence of spurious oscillations in ψ , it is clear that differentiating the hyperbolic tangent level set should be avoided, and that the curvature should be computed from ϕ . However, since the distance level set field ϕ is recomputed at every step from an FMM, inaccuracies in the distance field are also expected. Since two successive levels of differentiation are applied to the ϕ -field to obtain the curvature, it is to be expected that a ϕ -field computed through the FMM with at best second order accuracy will provide first order normals, and a curvature that will not converge under mesh refinement. In order to verify this property, the curvature has been computed by taking a second order divergence of the face normals. Such a scheme gives a compact curvature computation, using 27 points in three dimensions, which only differs from the standard, second order, compact curvature computation [4] by the way the normalization is done. This scheme was tested by computing the curvature from an FMM-reconstructed distance, for a circle of radius 0.5 centered in a squared domain of size $[0, 2] \times [0, 2]$. The L_2 error is shown in Table 3. Clearly, no convergence is obtained under mesh refinement.

In order to remedy this problem, a least squares approach was introduced. While the finite difference approach proposed above uses a 27-point stencil in three dimensions, a second order polynomial reconstruction of ϕ only requires 10 points to be obtained. If the same 27-point stencil is used to perform the least squares reconstruction instead of only the 10 necessary points, the size of the statistical sample will be rather large, and it can be expected that the numerical errors due to the inaccuracy of the FMM will be smeared out. This approach was proposed recently by Marchandise et al. [21], and successfully employed to obtain converging curvatures from a discontinuous Galerkin scalar field. The results for the test case used above, but with the least squares curvature computation are shown in Table 4. It can indeed be observed that first order convergence is recovered.

3.6. ACLS solution procedure

A brief summary of the ACLS solution procedure is given here:

- Using a semi-implicit Crank–Nicolson time integration, advance the ψ field by solving Eq. (10). Unless specified otherwise, HOUC-5 will be used for this step.

Table 3

Evolution of the finite difference curvature error with mesh spacing

Mesh	Error
8×8	0.17772
16×16	0.09200
32×32	0.15270
64×64	0.12575

Table 4

Evolution of the least squares curvature error with mesh spacing

Mesh	Error
8×8	0.28207
16×16	0.17276
32×32	0.08279
64×64	0.04737

- Use FMM to efficiently reconstruct ϕ from ψ .
- Compute the face normals from ϕ .
- Compute the least squares curvature from ϕ .
- Perform the conservative re-initialization step: using a semi-implicit Crank–Nicolson time integration, Eq. (11) is advanced.

3.7. Level set transport tests

3.7.1. Long time conservation of an under-resolved sphere

In order to evaluate the capability of the proposed ACLS method to conserve the mass at low resolution, a poorly resolved sphere of diameter D is transported in a uniform velocity field with a CFL number of 0.5. For comparison, the same test is conducted with a standard distance function level set using WENO-5 for the transport and FMM for the distance re-initialization, which is performed every 10 time steps. With only five mesh points in the diameter, we observe in Fig. 4 that the standard distance level set method loses all the mass in the sphere in less than $50D$, even with the low re-initialization frequency. However, we can see that the ACLS approach maintains the mass indefinitely. After being transported for a distance larger than $500D$, the mass is still adequately maintained. Only very limited oscillations in the mass can be observed, which can be attributed to the very poor discretization of the sphere.

3.7.2. Zalesak's disk

In order to assess the capability of the ACLS method to adequately transport thin structures and sharp corners, the solid body rotation of a notched circle is simulated. The circle is 0.15 in radius, with a notch of width 0.05 and height 0.25. It is centered initially at $(0.5, 0.75)$ in a $[-0.5, 0.5] \times [-0.5, 0.5]$ domain, discretized on a 100×100 mesh. The solid body rotation is applied to this shape until one revolution is completed. The time integration of the full revolution is performed using a semi-implicit Crank–Nicolson in 500 time steps, leading to a CFL number of 0.6.

First, the full ACLS procedure is tested and compared to the approach where the normal vectors are directly obtained from ψ . For both methods, HOUC-5 is used for the transport equation, Eq. (10). Fig. 5 compares the transported ψ -field with the

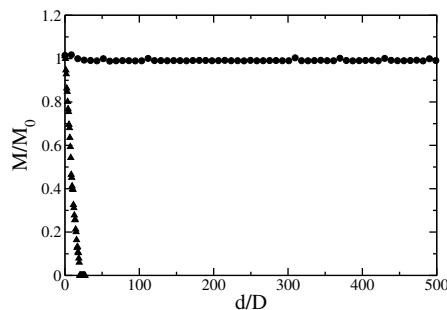


Fig. 4. Evolution of the mass in an under-resolved droplet as a function of the distance traveled; standard distance level set approach (triangles) and ACLS (circles).

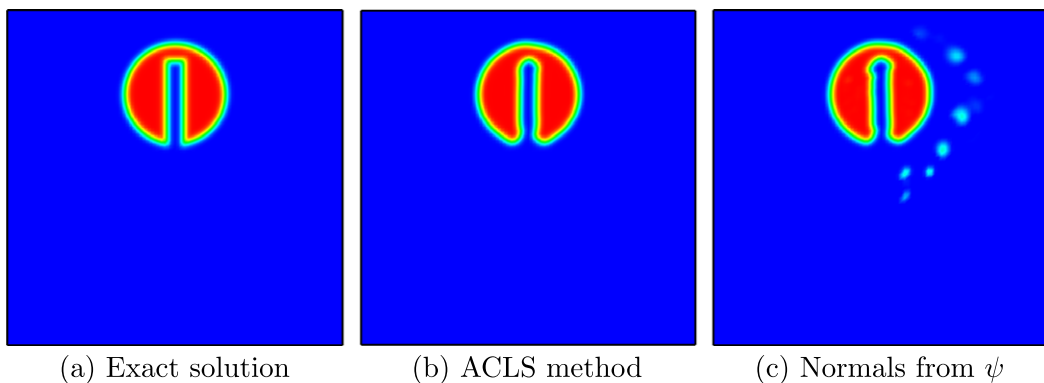


Fig. 5. ψ -Scalar for Zalesak's disk problem with ACLS or by computing the normals from ψ , after a full rotation. Color map is from $\psi = 0$ to $\psi = 1$. (For interpretation of the references in color in this figure legend, the reader is referred to the web version of this article.)

exact solution for both cases. Clearly, the ACLS method leads to very satisfactory results, with a properly maintained hyperbolic tangent thickness, while the method where the normals are obtained from the ψ -field shows many regions of accumulated ψ -scalar away from the notched disk. While this is not a major problem for this case, it is clear that it will become critical to the stability of the method for realistic problems.

In Fig. 6, the notched circle shape is compared with the exact solution using different level set transport schemes to discretize Eq. (10), namely HOU-5 and WENO-5. While both results are very satisfactory, it is clear that HOU-5 leads to a more accurate solution due to both the increased accuracy of HOU-5 in comparison to WENO and because the HOU-5 scheme will rely much less on the conservative re-initialization. These results confirm that it is beneficial to avoid spreading of the ψ profile, and also that the HOU-5 solution behaves adequately despite the absence of any non-oscillatory property.

As a third test case, the influence of the conservative re-initialization frequency is analyzed. Fig. 7 compares the results obtained by re-initializing at each time step and re-initializing every 10 time steps. Clearly, very little differences can be observed in the results, suggesting that the accuracy of the ACLS approach is not affected by the re-initialization frequency. This appears as a strong advantage compared to classical level set approaches, for which more re-initialization usually means less accurate results.

Finally, a more detailed mesh convergence analysis is performed using both a 50×50 mesh and a 200×200 mesh in addition to the 100×100 mesh introduced before. The CFL number is kept constant for the different cases. Fig. 8 compares the results after one rotation for the three meshes employed. The computed interface converges toward the exact solution in a satisfactory manner. Fig. 9 shows the time evolution of the normalized area enclosed in the interface for the three different meshes. Even with the coarsest mesh, the conservation errors remain very limited, and these errors are significantly reduced by going to finer meshes. Table 5 summarizes the maximum area conservation errors for the three meshes. Between the

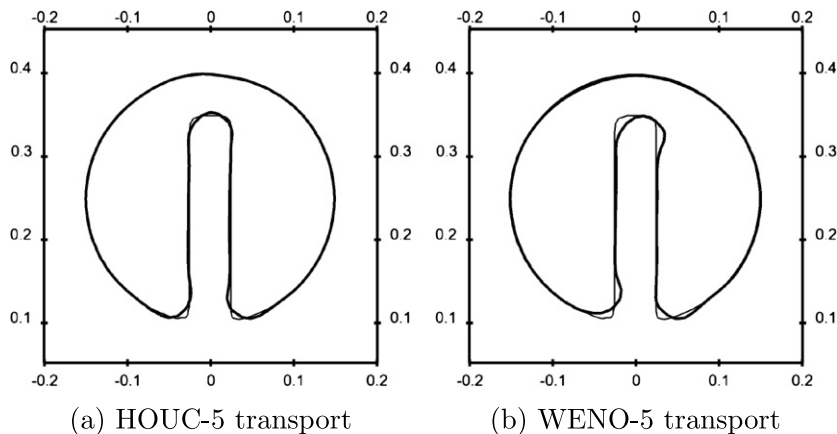


Fig. 6. Effect of the transport scheme on Zalesak's disk after a full rotation with ACLS: computed interface location (thick line) and exact interface location (thin line).

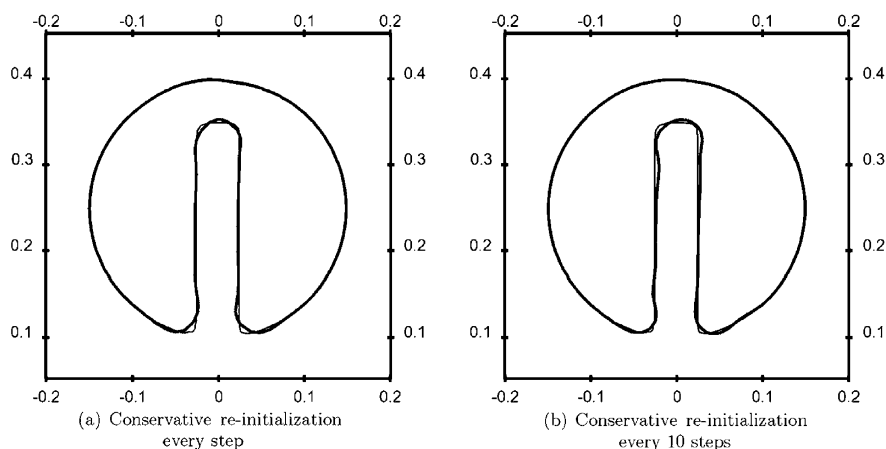


Fig. 7. Effect of the re-initialization frequency on Zalesak's disk after a full rotation with ACLS: computed interface location (thick line) and exact interface location (thin line).

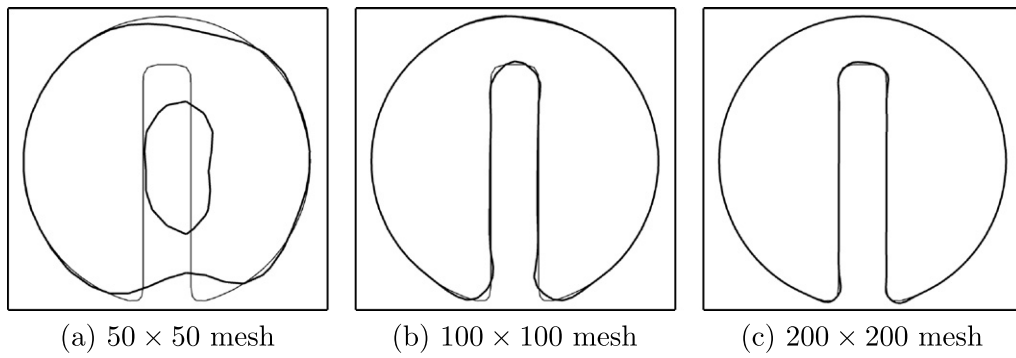


Fig. 8. Effect of the mesh spacing on Zalesak's disk after a full rotation with ACLS: computed interface location (thick line) and exact interface location (thin line).

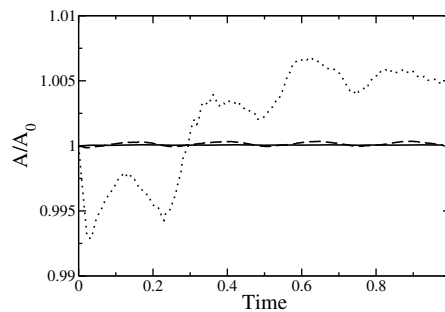


Fig. 9. Temporal evolution of the normalized area enclosed in the interface for Zalesak's disk using the ACLS method on different meshes: 50×50 mesh (dotted line), 100×100 mesh (dash line), and 200×200 mesh (solid line).

Table 5

Evolution of the maximum area conservation error with mesh spacing for Zalesak's disk

Mesh	Error (%)
50×50	0.7167
100×100	0.0352
200×200	0.0085

50×50 mesh and the 100×100 mesh, the error is strongly reduced. This can be attributed to the fact that numerical errors lead to the closure of the notch for the coarsest mesh simulation. Between the medium and the fine simulation, second order convergence of area conservation errors is observed.

3.7.3. Circle in a deformation field

The deformation of a circle by a single vortex has been considered to assess the ability of numerical methods to resolve thin filaments, e.g. see [5]. In a $[0, 1] \times [0, 1]$ domain, a circle of radius 0.15 is initially centered at (0.5, 0.75). The velocity field is obtained from the stream function

$$\Psi(\mathbf{x}, t) = \frac{1}{\pi} \sin^2(\pi x) \cos^2(\pi y) \cos(\pi t/T), \quad (22)$$

where T is set to 8. At $t = T/2$, the deformation will be maximum, then the process is inverted until $t = T$, at which point the circle should be back to its initial shape and location. The simulation is conducted on two meshes, namely a 128×128 and a 256×256 mesh. For the coarse run, the time step size is set to $\Delta t = 0.01$, leading to a maximum CFL number of 1.28. For the finer mesh, the time step size is divided by two. Fig. 10 shows the result of both runs at $t = T/2$. Clearly, at this time, the width of the trailing ligament should fall below the resolution of the 128×128 mesh, meaning that the numerically correct solution should be to lose the corresponding area. However, because of the conservative re-initialization step, the ACLS method attempts to maintain this area on the mesh, which leads to the formation of drops of the size of one or two grid cells. When the velocity field is reversed, this area is recovered. In Fig. 11, it can be observed that the circle at $t = T$ displays some deformation because of the droplets that have been created. However, going to a finer mesh greatly improves these results.

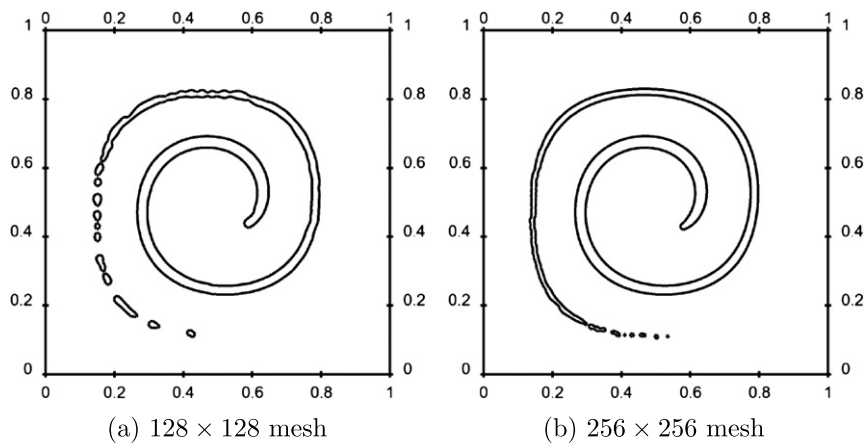


Fig. 10. Interface location for the circle in a deformation field test case at $t = T/2$ for different mesh sizes.

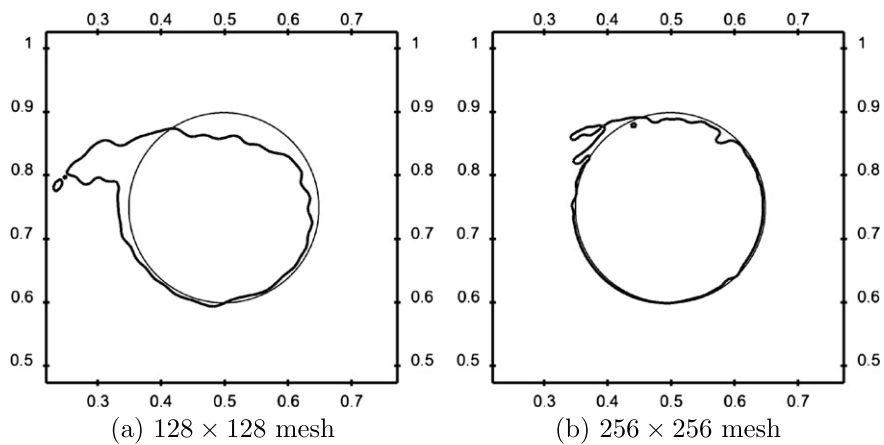


Fig. 11. Interface location for the circle in a deformation field test case at $t = T$ for different mesh sizes: computed interface (thick line) and exact solution (thin line).

Finally, Fig. 12 shows the evolution of the normalized area with time for the two meshes. Note that even on the coarse mesh, the error at maximum stretching remains below 4%, while the fine mesh shows a maximum error below 0.5%. At $t = T$, most of the area is recovered, leading to an error of less than 0.1% on the coarse mesh, and less than 0.01% for the fine mesh. In comparison, Herrmann [7] reported for a standard level set technique an area conservation error at $t = T$ of more than 30% on a 128×128 mesh, and more than 4% on a 256×256 mesh.

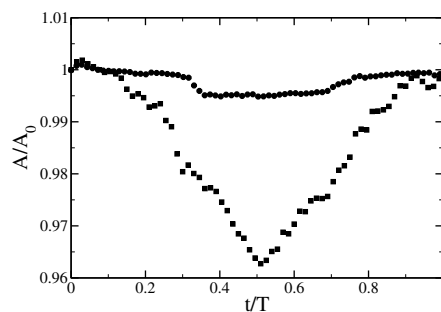


Fig. 12. Temporal evolution of the normalized area for the circle in a deformation field test case for different mesh sizes: 128×128 mesh (squares) and 256×256 mesh (circles).

4. Solution of the Navier–Stokes equation

This section describes the coupling with the flow solver NGA [20]. NGA solves the variable density, low Mach number Navier–Stokes equations using high order conservative finite difference methods that are staggered in time and space. Such methods have been shown to be highly suited for turbulence simulations [20], and therefore are expected to be highly beneficial for ensuring the accuracy of turbulent multiphase simulations. The Navier–Stokes equations are solved using the fractional step approach of Pierce and Moin [22]. In order to handle the large density variation across the phases, this code was modified to solve the incompressible Navier–Stokes equations by assuming a constant density. The ghost fluid technique [12] is then used to account explicitly for the density jump and the surface tension through the pressure term, while the continuum surface force approach [11] is used to model the jump in the viscous stresses.

4.1. Ghost fluid methodology

One of the central issues with two-phase flow simulations lies in the numerical discretization of the pressure gradient term. Indeed, this term contains the density, which exhibits a jump at the interface, and the pressure itself is discontinuous at the interface because of both the surface tension force and the viscous jump, as seen in Eq. (3). An efficient discretization of this term requires a specific treatment. The ghost fluid method is used here to account for the pressure jump caused by surface tension, while the jump caused by the discontinuous viscous stress is handled by the continuum surface force technique. GFM relies on the assumption that all the jump conditions for a given variable $[\zeta]_r$ and its spatial derivatives $[\partial\zeta/\partial x]_r, [\partial^2\zeta/\partial x^2]_r, \dots$ are known at the interface Γ . Then, the GFM is based on the extension by continuity of ζ_i in the gas and of ζ_g in the liquid. This extension allows to define the jump of ζ , $[\zeta]$, not only at the interface, but also in the neighborhood of the interface. Since the jump $[\zeta]$ is defined everywhere, it may be expanded by using a Taylor series expansion around the interface:

$$[\zeta] = [\zeta]_r + (x - x_r) \left[\frac{\partial\zeta}{\partial x} \right]_r + \frac{1}{2} (x - x_r)^2 \left[\frac{\partial^2\zeta}{\partial x^2} \right]_r + \mathcal{O}((x - x_r)^3). \quad (23)$$

Then, spatial derivatives may be expressed using only values in the same phase. If the interface Γ is located somewhere between x_{i-1} and x_{i+1} , the first derivative may be written:

$$\left. \frac{\partial\zeta}{\partial x} \right|_{g,i} = \frac{\zeta_{g,i+1} - \zeta_{g,i-1}}{2\Delta x} + \mathcal{O}(\Delta x^2) = \frac{\zeta_{l,i+1} - [\zeta]_{i+1} - \zeta_{g,i-1}}{2\Delta x} + \mathcal{O}(\Delta x^2), \quad (24)$$

where

$$[\zeta]_{i+1} = [\zeta]_r + (x_{i+1} - x_r) \left[\frac{\partial\zeta}{\partial x} \right]_r + \mathcal{O}((x_{i+1} - x_r)^2). \quad (25)$$

This methodology may be used to discretize the pressure Laplacian for an interface located between x_i and x_{i+1} with $x - x_r > 0$ corresponding to the liquid phase:

$$\left. \frac{\partial}{\partial x} \left(\frac{1}{\rho} \frac{\partial P}{\partial x} \right) \right|_{g,i} = \frac{1}{\rho_g} \frac{P_{g,i+1} - 2P_{g,i} + P_{g,i-1}}{\Delta x^2} + \mathcal{O}(\Delta x^2), \quad (26)$$

$$\left. \frac{\partial}{\partial x} \left(\frac{1}{\rho} \frac{\partial P}{\partial x} \right) \right|_{g,i} = \frac{1}{\rho_g} \frac{P_{l,i+1} - [P]_{i+1} - 2P_{g,i} + P_{g,i-1}}{\Delta x^2} + \mathcal{O}(\Delta x^2), \quad (27)$$

where

$$[P]_{i+1} = [P]_r + (x_{i+1} - x_r) \left[\frac{\partial P}{\partial x} \right]_r + \mathcal{O}((x_{i+1} - x_r)^2). \quad (28)$$

These equations cannot be used in this form, because unlike the pressure jump at the interface, which is a known quantity that depends on local surface tension, the pressure gradient jump at the interface $[\partial P/\partial x]_r$ is not known *a priori* and has to be derived. With the assumption that the velocity is continuous across the interface, the pressure gradient divided by the density that appears on the RHS of the momentum equation also has to be continuous:

$$\left[\frac{1}{\rho} \frac{\partial P}{\partial x} \right]_r = 0. \quad (29)$$

This equation allows to evaluate the pressure gradient jump across the interface from the pressure gradient in the gas or in the liquid:

$$\left[\frac{\partial P}{\partial x} \right]_r = -\rho_l \left[\frac{1}{\rho} \frac{\partial P}{\partial x} \right]_{g,r} = -\rho_g \left[\frac{1}{\rho} \frac{\partial P}{\partial x} \right]_{l,r}. \quad (30)$$

By using the Taylor series expansion of the pressure gradient in the gas or in the liquid from the interface, the pressure jump can be taken equal to the pressure gradient at $i + 1/2$:

$$\left. \frac{\partial P}{\partial x} \right|_{g,r} = \left. \frac{\partial P}{\partial x} \right|_{g,i+1/2} + \mathcal{O}(x_{i+1/2} - x_r), \quad (31)$$

and

$$[P]_{i+1} \simeq [P]_r - (x_{i+1} - x_r) \rho_l \left[\frac{1}{\rho} \right] \left. \frac{\partial P}{\partial x} \right|_{g,i+1/2}, \quad (32)$$

$$[P]_{i+1} \simeq [P]_r - (x_{i+1} - x_r) \rho_l \left[\frac{1}{\rho} \right] \frac{P_{g,i+1} - P_{g,i}}{\Delta x}, \quad (33)$$

$$[P]_{i+1} \simeq [P]_r - (x_{i+1} - x_r) \rho_l \left[\frac{1}{\rho} \right] \frac{P_{l,i+1} - [P]_{i+1} - P_{g,i}}{\Delta x}. \quad (34)$$

The last equation has to be inverted to obtain $[P]_{i+1}$. Then, introducing the index $\theta = (x_r - x_i)/\Delta x$ and a modified density $\rho^\star = \rho_g \theta + \rho_l(1 - \theta)$, the pressure jump at x_{i+1} reads:

$$[P]_{i+1} \simeq \frac{\rho_g}{\rho^\star} [P]_r + \left(1 - \frac{\rho_g}{\rho^\star} \right) (P_{l,i+1} - P_{g,i}). \quad (35)$$

Finally, the pressure jump at x_{i+1} may be replaced in Eq. (27), leading to the following discretization of the Laplacian:

$$\left. \frac{\partial}{\partial x} \left(\frac{1}{\rho} \frac{\partial P}{\partial x} \right) \right|_{g,i} = \frac{\frac{1}{\rho^\star} (P_{l,i+1} - P_{g,i}) - \frac{1}{\rho_g} (P_{g,i} - P_{g,i-1})}{\Delta x^2} - \frac{[P]_r}{\rho^\star \Delta x^2}. \quad (36)$$

The extension of this expression to two or three dimensions is straightforward. This result for the pressure equation was obtained by Kang et al. [23] and Liu et al. [24]. Not using discrete operators to differentiate or interpolate quantities that vary by several orders of magnitude across the interface is likely to reduce the numerical discretization errors, and therefore the GFM is expected to provide improved accuracy. One additional benefit of using GFM is that because the surface tension force is embedded in the pressure gradient, the discrete balance of pressure forces and surface tension forces is guaranteed, whereas several other methods require an explicit treatment to obtain this property [7,25].

4.2. Viscous formulation

Although advantageous, the GFM is challenging to apply in the presence of the viscous term. While formulations have been proposed [23], they are not easy to implement, and an implicit formulation has proven challenging to develop. Consequently, in the present method the continuum surface force (CSF) approach by Brackbill et al. [11] is used for the viscous terms. Considering we are interested in turbulent problems, the viscous terms are expected to be significant only at the smallest scales, which do not contain much energy, and therefore this choice is not likely to influence significantly the quality of the results. In place of using a smeared-out Heaviside function to compute the density and viscosity, as in Olsson and Kreiss [9], we directly use ψ :

$$\begin{aligned} \rho(\mathbf{x}, t) &= \rho_g + (\rho_l - \rho_g) \psi(\mathbf{x}, t), \\ \mu(\mathbf{x}, t) &= \mu_g + (\mu_l - \mu_g) \psi(\mathbf{x}, t). \end{aligned} \quad (37)$$

Using this approach, an implicit time integration is straightforward, and therefore the time step size is not limited by the viscous CFL condition.

4.3. Time integration

As already mentioned before, ensuring an implicit time integration for each step of the method is one of our objectives. While this will permit to handle cylindrical coordinates as well as non-uniform meshes, this also provides increased robustness, which is a highly desirable property for any multiphase model. Moreover, using implicit methods frees the time step size from most CFL constraints, and therefore leads to a significant speed-up of the simulations. The Navier–Stokes equations are solved using the second order semi-implicit Crank–Nicolson scheme of Pierce and Moin [22]. This approach, inspired by the classical fractional step method [26], makes use of an iterative temporal advancement with staggering in time between the velocity field and the scalar and density fields. In the context of multiphase flow simulations, the level set field is advanced first from $t^{n-1/2}$ to $t^{n+1/2}$ using the velocity at t^n . The velocity field is then advanced from t^n to t^{n+1} , and the level set information is used at $t^{n+1/2}$ to solve the Poisson equation for pressure. This variable-coefficient, elliptic equation is solved using a combination of Krylov-based methods [27] preconditioned using a multi-grid solver [28]. For all time advancement steps, including the conservative re-initialization, an implicit correction is computed using an approximate factorization technique similar to the one proposed by Choi and Moin [29], where spatial directions are decoupled. Performing this

implicit correction requires the inversion of a linear system with multiple diagonals, for each scalar and velocity component, as well as for each spatial direction, which is done using a parallel poly-diagonal linear solver.

4.4. Full solution procedure

The full solution procedure is here summarized:

- Using the ACLS methodology, advance the interface implicitly from $t^{n-1/2}$ to $t^{n+1/2}$ using the velocity at t^n .
- Advance the velocity field implicitly from t^n to t^{n+1} by solving Eq. (1) without pressure gradient.
- Project the velocity field by solving the Poisson equation making use of GFM. The solution of the pressure equation is computed using a Krylov-based method, preconditioned by a multi-grid solver.
- Correct the velocity at t^{n+1} using the pressure gradient, again using GFM.

5. Applications

In order to assess the capability of the proposed method to tackle a wide range of multiphase problems, several test cases involving surface tension, topology changes, and high density ratios, are investigated. For all these problems, the minimum number of points required to obtain a good description of the physics is discussed. The proposed methodology is then used to compute the turbulent atomization of a liquid Diesel jet.

5.1. Parasitic currents

The errors in curvature computation will lead to discrete errors in the surface tension force, ultimately generating spurious velocities at the interface between the two phases. To assess the importance of these parasitic currents, a static two-dimensional drop with a diameter of $D = 0.4$, placed at the center of a unit box, is computed. The viscosity of both fluids is set to $\mu = 0.1$, the surface tension coefficient to $\sigma = 1$, and the density ratio to unity. By changing the density ρ of both fluids, the Laplace number $La = 1/Oh^2 = \sigma\rho D/\mu^2$ can be varied. The capillary number $Ca = |u_{\max}|\mu/\sigma$ is computed at a non-dimensional time of $t\sigma/(\mu D) = 250$. The results for a 32×32 mesh, presented in Table 6, show that the capillary number remains small independently of the Laplace number.

To assess mesh convergence of the parasitic currents, the Laplace number is fixed to 12,000, while mesh spacing is varied. Table 7 shows the resulting capillary numbers, that display first order convergence.

While these values are slightly larger than those obtained by previous studies that benefited from higher order curvature computations [7,30,31], they remain very small, and have not been found to affect the accuracy of the computed solutions in an unreasonable way.

5.2. Standing wave

Next, the viscous damping of a surface wave is investigated. This test case will help assess the capability of the proposed method to accurately simulate problems where viscosity and surface tension forces interact. In a two-dimensional domain of size $[0, 2\pi] \times [0, 2\pi]$, two superimposed fluids with density ρ_1 and ρ_2 are initially separated by a flat interface, slightly perturbed by a sine wave profile, namely,

$$\phi(x, y) = \pi - y + A_0 \cos(2\pi x/\lambda), \quad (38)$$

where the wavelength of the perturbation λ is set to 2π , and the initial amplitude of the wave is $A_0 = 0.01\lambda$. Periodic boundary conditions are used for the x -direction, and slip walls are used in the y -direction. In the case where both fluids have the same kinematic viscosity ν , Prosperetti [32] employed initial value theory to derive an analytical expression to the evolution

Table 6

Dependence of the magnitude of parasitic currents with the Laplace number for a static droplet with surface tension on a 32×32 mesh

La	12	120	1200	12,000	120,000	1,200,000
Ca	4.54×10^{-5}	3.67×10^{-5}	3.62×10^{-5}	4.15×10^{-5}	3.75×10^{-5}	8.19×10^{-6}

Table 7

Dependence of the magnitude of parasitic currents with mesh spacing for a static droplet with surface tension with $La = 12,000$

Mesh	8×8	16×16	32×32	64×64	128×128
Ca	1.61×10^{-4}	8.95×10^{-5}	4.15×10^{-5}	2.24×10^{-5}	1.16×10^{-5}

of the wave amplitude with time. The reader is referred to Prosperetti [32] for the details of these analytical results. We simply recall here the inviscid oscillation frequency $\omega_0 = \sqrt{\frac{\sigma}{\rho_1 + \rho_2}}$ that is used to non-dimensionalize the computational time.

Following the case study presented by Herrmann [7], we investigate a first case with unity density ratio, and a second with a density ratio of 1000. For both cases, three different meshes are tested, namely an 8×8 mesh, a 16×16 mesh, and a 32×32 mesh. The simulations are performed up to a time of $\omega_0 t = 20$, leading to approximately three full oscillation periods. During these simulations, we ensure that the conservative re-initialization is called at least once. Moreover, the FMM is employed at each iteration in order to reconstruct ϕ from ψ , as described in Section 3.2.

For the first case, the non-dimensional surface tension coefficient is set to $\sigma = 2$, and the non-dimensional kinematic viscosity in both fluids is set to $\nu = 0.064720863$. Both fluid densities are set to unity. The time step size is taken to be $\Delta t = 0.01$, regardless of the mesh size. The time evolution of the wave amplitude for the different meshes as well as for the theoretical results are shown for this first case in Fig. 13a. The error between the exact solution and the computed solution, normalized by A_0 , is shown in Fig. 13b. It can be observed that even with a mesh as coarse as 8×8 , adequate results are obtained. However, a slight discrepancy in the oscillation period leads to a significant error on this mesh. As expected, increasing the mesh size to 16×16 or 32×32 gives much more accurate results. The RMS values of the error in amplitude for the three meshes employed are summarized in Table 8. Second order convergence is observed. These results are comparable to previous work [7,31], and suggest that about 10 mesh points per waves are necessary to capture the physics accurately.

The next case involves a high density ratio, obtained by setting $\rho_1 = 1$ and $\rho_2 = 1000$. The kinematic viscosity of both fluids is set to $\nu = 0.0064720863$. Again the results are compared to the work of Prosperetti [32], and we can see in Fig. 14 that good agreement is obtained, even with the coarsest mesh. Table 9 summarizes the RMS of the errors in amplitude, and we can see that at least first order convergence is obtained. This slow convergence rate has also been re-

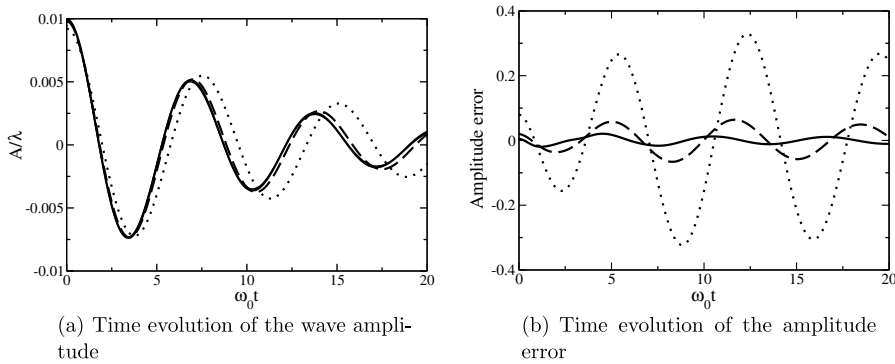


Fig. 13. Damped surface wave problem with unity density ratio. 8×8 mesh (dotted line), 16×16 mesh (dashed line), 32×32 mesh (solid line), and theory (thin line).

Table 8

RMS value of the amplitude error for the standing wave with unity density ratio

Mesh	Error
8×8	0.1996610
16×16	0.0395346
32×32	0.0103786

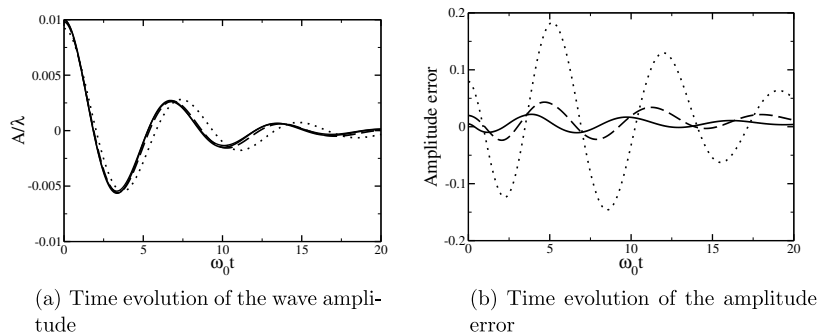


Fig. 14. Damped surface wave problem with density ratio 1:1000. 8×8 mesh (dotted line), 16×16 mesh (dashed line), 32×32 mesh (solid line), and theory (thin line).

Table 9
RMS value of the amplitude error for the standing wave with density ratio 1:1000

Mesh	Error
8 × 8	0.0892302
16 × 16	0.02043
32 × 32	0.00971

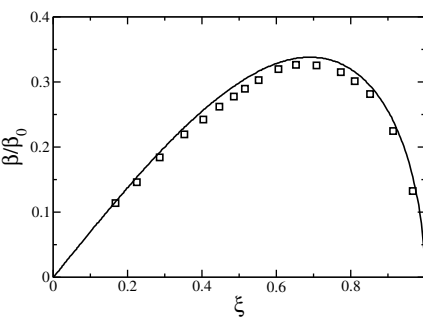


Fig. 15. Growth rate of the disturbance as a function of its wavelength for the capillary instability. Simulation with 24 points in the radial direction (symbols), and linear theory by Weber [34] (line).

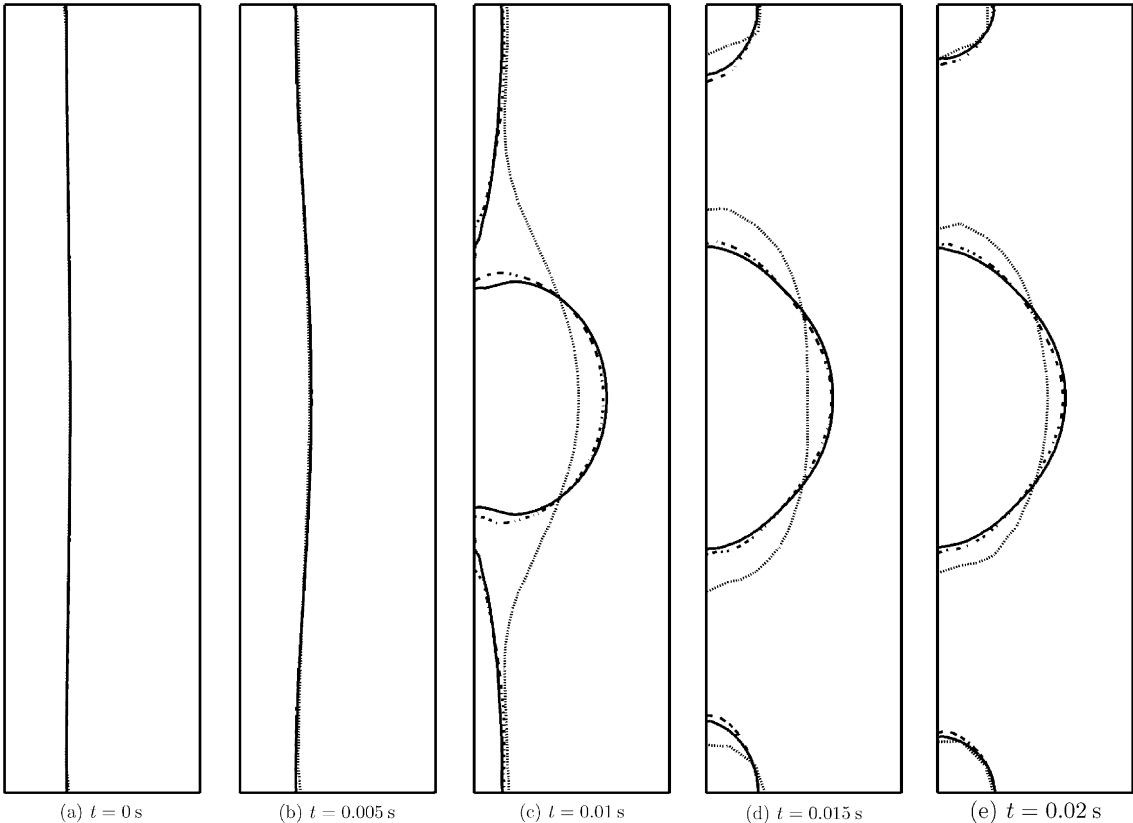


Fig. 16. Interface shape for the capillary instability where $\lambda = 12r_0$, computed with three different meshes: a 32×8 mesh (dotted line), a 64×16 mesh (dashed line), and a 128×32 mesh (solid line).

ported by Herrmann [7]. These results suggest again that about 10 points per wavelength are sufficient to accurately describe the physics of this problem, even with large density ratios.

5.3. Capillary instability

The computation of a Rayleigh instability is performed in order to assess in details the capability of the proposed approach to simulate surface tension-driven instabilities. This test case follows the work of Ménard et al. [33], where the dispersion relation was computed, and completes it by investigating mesh convergence. Following Ménard et al. [33], the capillary instability of a stationary water column in air is computed on a two-dimensional axisymmetric domain of size $[0, \lambda] \times [0, 3r_0]$, where r_0 is the mean column radius, which is set to $r_0 = 1/3 \times 10^{-3}$ m, and the initial perturbation wavelength is λ . The initial level set field therefore reads

$$\phi(x, y) = r_0 - y + A_0 \cos(2\pi x/\lambda), \quad (39)$$

where A_0 is the initial disturbance amplitude set to $A_0 = 10^{-4}\Delta y$, where Δy is the mesh spacing in the y -direction. The computational domain has periodic conditions in the x -direction, and a slip wall in the y -direction. Whereas Ménard et al. [33] used 61 grid points in the y -direction, we choose to decrease this number to only 24. Keeping $\Delta x = \Delta y$, the number of points

Table 10

Convergence rates of the L_1 -errors of different quantities, at different times, for the capillary instability test case

Quantity	$t = 5 \times 10^{-3}$ s	$t = 1 \times 10^{-2}$ s	$t = 1.5 \times 10^{-2}$ s
u	1.06	1.27	0.79
v	1.05	1.52	1.09
ψ	1.42	2.24	2.04

Table 11

Physics parameters for the liquid jet atomization test case

ρ_l/ρ_g	μ_l/μ_g	Re_l	We_l
40	40	3000	10,000

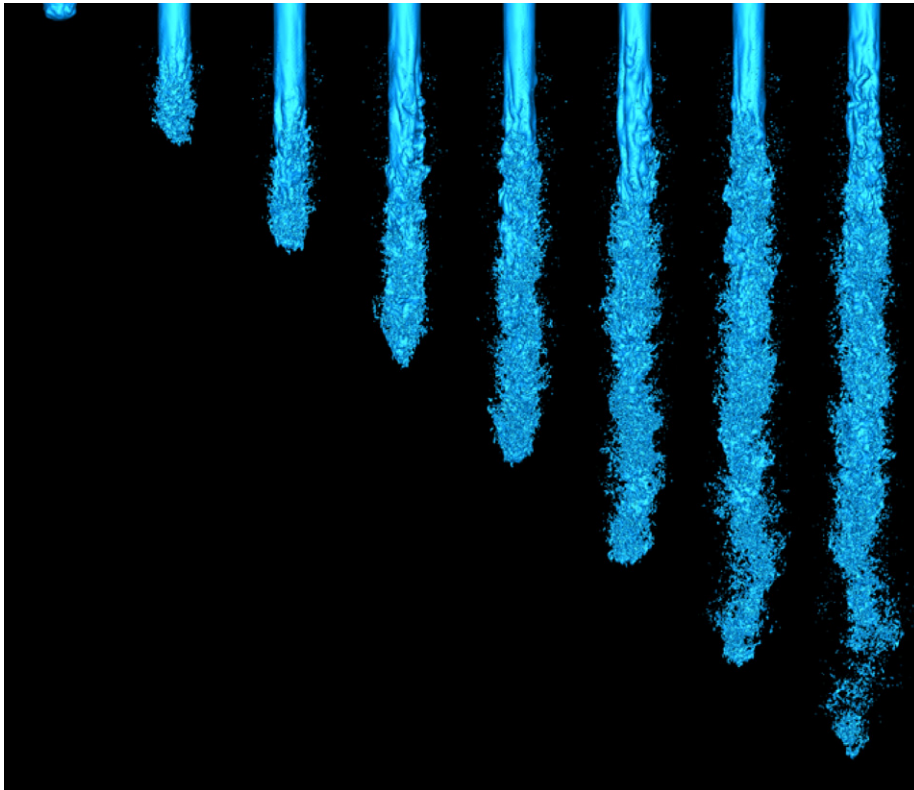


Fig. 17. Turbulent atomization of a liquid Diesel jet. $\Delta t = 2.5$ between each image.

in the x -direction is varied in order to change the disturbance wavelength λ . We then compute the non-dimensional growth rate β/β_0 of the disturbance, where $\beta_0 = \sqrt{\sigma/(\rho_l r_0^3)}$, as a function of its non-dimensional wavenumber $\xi = 2\pi r_0/\lambda$, and compare it to the linear theory due to Weber [34]. The results, shown in Fig. 15, are very satisfactory, and suggest that the proposed methodology succeeds at capturing this capillary instability with as little as eight points per column radius.

As a second step, we propose to compare the results for $\lambda = 12r_0$ obtained with three different meshes, namely a 32×8 mesh, a 64×16 mesh, and a 128×32 mesh. Fig. 16 shows the time evolution of the interface location for the different meshes. It can be observed that the medium and the fine mesh solution are very close to each other, while the coarse mesh solution is significantly different, and underpredicts the growth rate of the instability. Table 10 compares the errors for both the velocity and the level set, assuming the fine mesh solution is the exact solution, both before break-up and after break-up. The order of convergence that is obtained for the L_1 -norm of the errors for u , v , and ψ is between first and second order, even during and after the break-up of the column. Again, these results suggest that the medium mesh is sufficient to obtain an accurate description of the physics, and therefore it can be concluded that about 10 grid points in the column are necessary to appropriately predict capillary break-up.

5.4. Turbulent atomization of a liquid Diesel jet

All cases performed until now are very limited in terms of complexity, because they cover only low Reynolds number flows, or flows governed by surface tension and viscous effects. In order to assess the performance of the proposed approach in the presence of fully developed turbulence, the simulation of a turbulent liquid jet in quiescent air is conducted. The properties for the simulation are inspired by liquid Diesel injection systems, although both the Reynolds number and Weber number have been reduced to make the simulation possible. The parameters employed are summarized in Table 11.

No sub-grid scale model is employed for this simulation, even though it seems likely that the smallest structures are not fully resolved. This simulation can provide some much needed insights, both on the resolution requirements to simulate

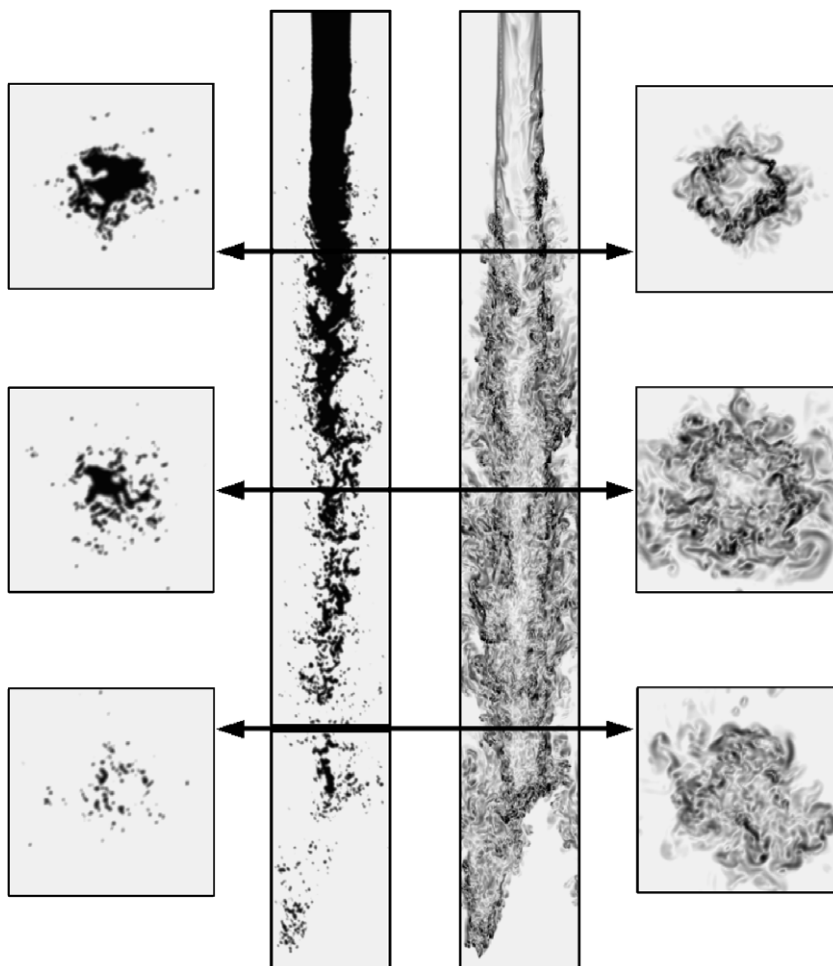


Fig. 18. ψ -Field (left) and magnitude of the vorticity (right) on two-dimensional axial and lateral cuts at $t = 22.8$ for the turbulent liquid jet case.

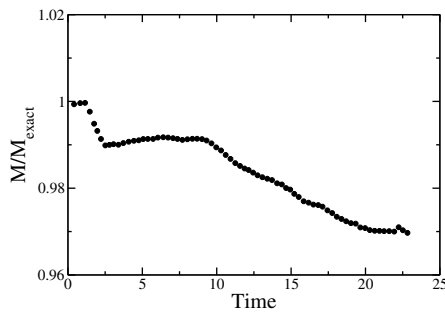


Fig. 19. Mass conservation errors for the turbulent liquid jet case.

turbulent atomization, and on the performance of the present method in the context of turbulent flows. The computation is performed on a domain of size $24D \times 3D \times 3D$, where D is the jet diameter, discretized on a $1024 \times 128 \times 128$ mesh. The inflow conditions are obtained by first simulating a turbulent pipe using the liquid properties, and storing the time-dependent velocity information. This information is then re-injected in the computational domain.

Instantaneous snapshots of the interface at different times are presented in Fig. 17. The interface displays a complex, turbulent behavior, as the liquid jet undergoes turbulent atomization. Many complex phenomena interact, leading to a fast break-up of the liquid core into ligaments and sheets, then droplets. It is interesting to note that by the end of the computational domain, the liquid core has fully disintegrated. The ψ -field as well as the magnitude of the vorticity field are presented in several two-dimensional cuts in Fig. 18. The fully developed nature of the turbulence appears clearly, along with the chaotic nature of the interface. Even for such a complex, turbulent, three-dimensional flow, the proposed multi-phase method appears robust.

Finally, we compare the mass enclosed in the $\psi = 0.5$ iso-contour with the expected liquid mass as a function of time in Fig. 19. With a maximum of 3% mass loss, the methods appears very satisfactory in terms of mass conservation, even in a complex turbulent case.

6. Conclusion

An accurate conservative level set method that provides accurate and robust interface transport with good mass conservation properties has been developed, and coupled to a Navier–Stokes solver using a ghost fluid approach. The conservative level set approach provides a simple way of reducing the mass conservation errors that are known to plague level set methods. In order to improve accuracy and robustness of the method, the normals are computed by first reconstructing a distance level set field, which is done efficiently using an FMM. A converging curvature is obtained by using a second order, compact, least squares reconstruction. The accuracy of the level set transport is ensured by using HOUK schemes. The GFM allows for a sharp treatment of the interface, and for a robust description of the surface tension forces. Additional robustness is obtained by solving all equations using implicit methods, which provides the additional benefit of allowing larger time steps. This approach is employed in a range of test cases, showing the good behavior of the method, and is then used in the simulation of turbulent atomization of a liquid jet. Even for such a complex problem, the method is robust and mass conservation errors are shown to remain small.

Acknowledgments

The authors wish to express their gratitude to Mr. Ed Knudsen for providing the parallel fast marching algorithm used in this work. We are also thankful to Dr. Madhusudan Pai for his fruitful comments on a draft of this manuscript, and to Prof. Marcus Herrmann and Dr. Guillaume Blanquart for many helpful discussions about this work. We also gratefully acknowledge funding by NASA and by the DOE through the ASC program.

References

- [1] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, *Ann. Rev. Fluid Mech.* 31 (1999) 567–603.
- [2] S. Unverdi, G. Tryggvason, A front-tracking method for viscous incompressible multi-fluid flows, *J. Comput. Phys.* 100 (1992) 25–37.
- [3] S. Osher, R. Fedkiw, *Level Set Methods and Dynamic Implicit Interfaces*, Springer, New York, 2003.
- [4] J.A. Sethian, *Level Set Methods and Fast Marching Methods*, second ed., Cambridge University Press, Cambridge, UK, 1999.
- [5] D. Enright, R. Fedkiw, J. Ferziger, I. Mitchell, A hybrid particle level set method for improved interface capturing, *J. Comput. Phys.* 183 (2002) 83–116.
- [6] M. Sussman, E.G. Puckett, A coupled level set and volume of fluid method for computing 3D and axisymmetric incompressible two-phase flows, *J. Comput. Phys.* 162 (2000) 301–337.
- [7] 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.
- [8] R.R. Nourgaliev, T.G. Theofanous, High-fidelity interface tracking in compressible flows: unlimited anchored adaptive level set, *J. Comput. Phys.* 224 (2007) 836–866.
- [9] E. Olsson, G. Kreiss, A conservative level set method for two phase flow, *J. Comput. Phys.* 210 (2005) 225–246.

- [10] E. Olsson, G. Kreiss, S. Zahedi, A conservative level set method for two phase flow II, *J. Comput. Phys.* 225 (2007) 785–807.
- [11] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, *J. Comput. Phys.* 100 (1992) 335–354.
- [12] 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 (1999) 457–492.
- [13] D.L. Chopp, Computing minimal surfaces via level set curvature flow, *J. Comput. Phys.* 106 (1993) 77–91.
- [14] M. Sussman, P. Smereka, S. Osher, A level set method for computing solutions to incompressible two-phase flow, *J. Comput. Phys.* 114 (1994) 146–159.
- [15] D. Peng, B. Merriman, S. Osher, H. Zhao, M. Kang, A PDE-based fast local level set method, *J. Comput. Phys.* 155 (1999) 410–438.
- [16] M. Herrmann, A domain decomposition parallelization of the fast marching method, in: *Annual Research Briefs, Center for Turbulence Research, Stanford, CA*, 2003.
- [17] X.D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, *J. Comput. Phys.* 115 (1994) 200–212.
- [18] G.S. Jiang, C.W. Shu, Efficient implementation of weighted ENO schemes, *J. Comput. Phys.* 126 (1996) 202–228.
- [19] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, *J. Comput. Phys.* 31 (1979) 335–362.
- [20] 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.
- [21] 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.
- [22] C.D. Pierce, P. Moin, Progress-variable approach for large eddy simulation of turbulent combustion, Technical Report TF80, Flow Physics and Computation Division, Department of Mechanical Engineering, Stanford University, 2001.
- [23] M. Kang, R. Fedkiw, X.D. Liu, A boundary condition capturing method for multiphase incompressible flow, *J. Sci. Comput.* 15 (2000) 323–360.
- [24] X.D. Liu, R. Fedkiw, M. Kang, Boundary condition capturing method for Poisson equation on irregular domains, *J. Sci. Comput.* (2000) 151–178.
- [25] M.M. Francois, S.J. Cummins, E.D. Dendy, D.B. Kothe, J.M. Sicilian, M.W. Williams, A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework, *J. Comput. Phys.* 213 (2006) 141–173.
- [26] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier–Stokes equations, *J. Comput. Phys.* 59 (1985) 308–323.
- [27] H.A. van der Vorst, *Iterative Krylov Methods for Large Linear Systems*, first ed., Cambridge University Press, Cambridge, 2003.
- [28] R.D. Falgout, U.M. Yang, HYPRE: a library of high performance preconditioners, in: P.M.A. Sloot, C.J.K. Tan, J.J. Dongara, A.G. Hoekstra (Eds.), *Computational Science – ICCS 2002 Part III*, vol. 2331, Springer-Verlag, Berlin, 2002, pp. 632–641.
- [29] H. Choi, P. Moin, Effects of the computational time step on numerical solutions of turbulent flow, *J. Comput. Phys.* 113 (1994) 1–4.
- [30] M. Sussman, K.M. Smith, M.Y. Hussaini, M. Ohta, R. Zhi-Wei, A sharp interface method for incompressible two-phase flows, *J. Comput. Phys.* 221 (2007) 469–505.
- [31] S. Popinet, S. Zaleski, A front-tracking algorithm for accurate representation of surface tension, *Int. J. Numer. Method Fluid* 30 (1999) 775–793.
- [32] A. Prosperetti, Motion of two superposed viscous fluids, *Phys. Fluid* 24 (1981) 1217–1223.
- [33] T. Ménard, S. Tanguy, A. Berlemont, Coupling level set/VOF/ghost fluid methods: Validation and application to 3D simulation of the primary break-up of a liquid jet, *Int. J. Multiphase Flow* 33 (2007) 510–524.
- [34] C. Weber, Disintegration of liquid jets, *Z. Angew. Math. Mech.* 11 (1931) 136.