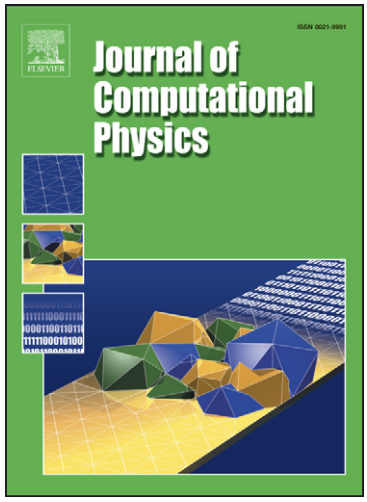


A spectrally refined interface approach for simulating multiphase flows

Olivier Desjardins, Heinz Pitsch

PII: S0021-9991(08)00588-3
DOI: 10.1016/j.jcp.2008.11.005
Reference: YJCPH 2335

To appear in: *Journal of Computational Physics*

Received Date: 20 May 2008
Revised Date: 29 September 2008
Accepted Date: 2 November 2008

Please cite this article as: O. Desjardins, H. Pitsch, A spectrally refined interface approach for simulating multiphase flows, *Journal of Computational Physics* (2008), doi: 10.1016/j.jcp.2008.11.005

A spectrally refined interface approach for simulating multiphase flows

Olivier Desjardins * and Heinz Pitsch

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

Abstract

This paper presents a novel approach to phase-interface transport based on pseudo-spectral sub-grid refinement of a level set function. In each flow solver grid cell, a set of quadrature points is introduced on which the value of the level set function is known. This methodology allows to define a polynomial reconstruction of the level set function in each cell. The transport is performed using a semi-Lagrangian technique, removing all constraints on the time step size. Such an approach provides sub-cell resolution of the phase-interface and leads to excellent accuracy in the transport, while a reasonable cost is obtained by pre-computing some of the metrics associated with the polynomials. To couple this approach with a flow solver, an converging curvature computation is introduced. First, a second order explicit distance to the sub-grid interface is reconstructed on the flow solver mesh. Then, a least squares approach is employed to extract the curvature from this distance function. This technique is found to combine the high accuracy and good conservation found in the particle level set method with the converging curvature usually obtained with classical high order PDE transport of the level set function. Tests are presented for both transport as well as two-phase flows, that suggest that this technique is capable

of retaining the thin liquid structures that are expected in turbulent atomization of liquids.

Key words: Multiphase flow, incompressible flow, DNS, spectral method, level set method, ghost fluid method, implicit scheme, primary atomization, sub-cell resolution

1 Motivation and objectives

Accurately simulating complex, turbulent, multiphase flows poses several major numerical challenges. The two phases can have different material properties, such as densities in 1 : 1000 ratio, which renders the discretization of the Navier-Stokes equations challenging. Moreover, the curvature of the phase-interface generates a surface tension force, which acts only at the interface itself. The singular nature of this force requires specific numerical treatments. Moreover, accurately computing this force can be problematic since it requires the knowledge of the interface curvature. The interface curvature is a high order term that is obtained by taking second derivatives, and it is hence prone to amplify numerical errors. Hence, ensuring the convergence of the curvature is a major hurdle of multiphase simulations. Finally, the accuracy of the interface transport itself can lead to difficulties, since the quality of the transport for most methods deteriorates greatly when considering small scale structures. While this may not be an issue for some applications, it can become critical for problems such as turbulent atomization, where the focus is precisely on the smallest liquid structures that are being generated by the flow. Because of all

* Corresponding author.
Email address: desjardi@stanford.edu (Olivier Desjardins).

these issues, multiphase flows remain difficult to simulate with good accuracy and robustness.

Several methods have been used in the past to handle the discontinuous material properties that can be found in two-phase flows. One of the most frequently used approaches is the Continuum Surface Force (CSF) model introduced by Brackbill [1]. The idea behind this method is to smear out the discontinuities over a few grid cells in order to resolve them. While this enables a standard discretization of the density jump and the surface tension force, it can be expected to deteriorate the accuracy of the small scale structures. The Ghost Fluid Method (GFM) [2] provides an interesting alternative to CSF by expressing all discontinuities explicitly. The discretization is performed on variables that are extended by continuity in order to remove the jumps. These jumps are then explicitly added, leading to a sharp description of the interfacial terms. Additionally, GFM naturally embeds the surface tension force in the pressure equation as a pressure jump. GFM has been used recently to simulate complex problems such as the atomization of liquid diesel jets [3,4,5], and possible improvements to this approach have been proposed [6].

However, all these methods rely on a numerical scheme to represent and transport the phase-interface, in order to localize the jumps and to provide the interfacial curvature. Many approaches have been developed to perform these tasks, the most common being probably the volume of fluid (VOF) [7] and the level set (LS) [8,9] methods. While the first tracks the liquid volume fraction, the second tracks the phase-interface itself in the form of an iso-contour of a level set function. Other techniques exist that rely on a two-dimensional unstructured mesh in combination with Lagrangian methods to represent and transport the interface [10]. All these approaches have some strengths and

shortcomings, and no method has clearly emerged as the ideal interface transport scheme.

Amongst the fundamental requirements of an interface transport scheme is the capability to compute a curvature that converges as the mesh is refined, and the capability to transport accurately small liquid structures without loosing mass. However, most of the schemes that try to improve the accuracy of the small scale transport, either by coupling LS with VOF information [11,12] or with Lagrangian particles [13] tend to degrade the accuracy of the curvature computation, because of the use of local corrections to the LS field. Coyajee *et al.* [14] showed that such an approach leads to inaccurate curvatures, and that a delocalization of the corrections should be devised to avoid this problem.

Other interesting strategies have been developed to obtain a converging curvature. Sussman *et al.* [6] uses a version of the coupled level set/volume of fluid (CLSVOF) method where the curvature is computed directly from the volume fraction scalar instead of the taking it from the level set field, leading to a second order converging curvature. However, this approach might seem unpractical in complex problems, since large seven points-stencils are associated with the curvature computation. Herrmann [15] proposed the refined level set grid (RLSG) method to locally refine the LS mesh, in order to control the errors associated with the interfacial transport, and to retain a converging curvature. However, this method can be demanding to implement, since it requires a separate mesh for the LS field. Moreover, it is unclear how much refinement can be afforded in realistic problems, considering that an explicit time integration is used, leading to smaller time steps as the LS mesh is refined. Lagrangian-based methods that extract a converging curvature from Lagrangian particles have been developed [16]. Classical Lagrangian

methods [13,10] naturally provide the strong solution rather than the weak solution to the interface transport equation, and therefore do not naturally perform curvature regularization. In the context of complex turbulent flows, Lagrangian-based methods can be challenging to employ. The complex nature of the turbulent velocity field can interfere with the ability of particle-based approaches to maintain sharp sub-grid interfacial structures, ultimately affecting the numerical accuracy and robustness of the transport.

In this work, the choice is made to improve the sub-cell representation of a level set function through a pseudo-spectral approach. In each cell, a polynomial reconstruction of the level set function is created, leading to highly improved accuracy of the transport at the smallest scales. By maintaining a Eulerian-type description of the interface, topology changes and characteristics crossings are handled automatically. Such a strategy is not new, and has been employed before [17,18,19]. However, all the previous work relied on a fully pseudo-spectral description of all the equations. Because of the cost associated with high order pseudo-spectral schemes, the order of the pseudo-spectral method presented by Marchandise *et al.* [18] remained limited. In Sussman and Hussaini [17], only level set transport tests were performed, without the coupling to the Navier-Stokes equations. Here, in a similar spirit as in the RLSG method [15], the pseudo-spectral description is used only for the LS, with the objective to introduce sub-cell interface resolution. Thanks to the potentially high order polynomial description, the frequent re-initialization step that is characteristic to level set methods becomes superfluous, since the increased accuracy handles both small and large gradients adequately. In order to allow for very fine resolution without affecting the time step size, the interface transport is performed using a semi-Lagrangian approach. Finally,

a method to extract the curvature is proposed that computes a converging curvature at the scale of the flow solver grid, therefore removing all possible coupling with sub-cell fluctuations.

This paper is organized as follows: the next section presents the spectrally refined interface approach, including the transport scheme and the approach used to extract the interfacial curvature. The third section presents the coupling with the Navier-Stokes solver, as well as the methods used to solve the momentum equations. Finally, the fourth section presents numerical tests used to validate the methodology, including the numerical simulation of a turbulent two-phase shear layer.

2 Spectrally Refined Interface (SRI) approach

This section describes in details the pseudo-spectral, collocation-based, polynomial reconstruction of a level set function that is used to improve the quality of the interface description by introducing sub-cell resolution.

2.1 Level set methodology

In the level set approach, the interface is defined implicitly as an iso-surface G_0 of a smooth function G. Formally, any function can be used as a level set function, however a signed distance to the interface is the most commonly used function, for its smooth behavior makes it easy to transport with good accuracy. Yet, the approach that is described in this work is within some constraints independent of the actual function used for G. Later in this section, a short discussion on the consequence of the choice of this function is given.

In the level set approach, the transport of the interface with a velocity $\boldsymbol{u}$ is described by solving a simple advection equation for the level set function,

$$\frac{\partial G}{\partial t} + \boldsymbol{u} \cdot \nabla G = 0. \tag{1}$$

The state-of-the-art for level set transport typically relies on WENO-type schemes [20,21,22] in order to combine accurate transport and numerical robustness. While these schemes will give good accuracy in general, they tend to be overly diffusive when transporting small-scale structures [23,24]. Moreover, Eq. 1 will ensure the transport of G, but not the preservation of its shape in the direction normal to the interface. As a result, the very nature of G is likely to vary with time, and very large or very small gradients are likely to be generated by Eq. 1. This issue requires a specific treatment, called re-initialization, during which the desired functional form of G is re-set, for example to a distance function, in order to avoid the development of excessive gradients that might lead to inaccuracy and ultimately to severe numerical instabilities. Such a re-initialization procedure can be based on a Hamilton-Jacobi partial differential equation (HJPDE) solved in pseudo-time for G, as in [22], or on geometrical algorithms, such as the fast marching method (FMM) [8]. All these approaches have in common the fact that they introduce errors in the transport of G, and typically lead to front displacement. These inaccuracies essentially manifest through the disappearance of small-scale interfacial features. These small-scale features, however, can be fundamental to the physics of multiphase flows. It is important to recognize that the improvement of interface representations in numerical simulations requires an accurate description of the smallest scales, hence the idea of introducing a sub-cell description of the interface.

2.2 Pseudo-spectral collocation-based sub-cell reconstruction

In order to enable sub-cell resolution, a polynomial reconstruction of the level set function is here generated in each cell by introducing a set of quadrature points. These points correspond to the locations where the nodal values of the level set function G are specified. Several considerations have to be taken into account while choosing the quadrature points, such as the accuracy of the resulting reconstruction. For example, a uniform distribution of quadrature points inside each cell is expected to lead to the Runge phenomenon, which will strongly limit the accuracy of the polynomial reconstruction [25,26]. To avoid this issue, the most logical approach is to employ Gauss quadrature. Here, the choice is made to use Gauss-Lobatto quadrature, such that some quadrature points are located on the cell faces. This property will be used to improve the continuity of the level set function across cells. The Gauss-Lobatto quadrature points can be based on any suitable family of orthogonal polynomials, such as Legendre or Chebyshev. Figure 1 shows an example of quadrature points for different numbers of unknowns p.

[Figure 1 about here.]

The multi-dimensional extension of this approach is straightforward for the structured cartesian mesh considered here. However, it is also possible for complex unstructured meshes by using linear transformations [27]. Sample two-dimensional sets of quadrature nodes in a cell are shown in Fig. 2.

[Figure 2 about here.]

In each flow solver cell, the quadrature points located on the right, top, and far face are considered as ghost nodes, whose G values are equal to those of the neighboring cells. For example, in the x-direction, the values of the quadrature nodes located at the right face of cell i are not stored, but considered to be equal to the G values of the quadrature nodes located at the left face of cell $i+1$. This is illustrated in Fig. 3 for a two dimensional configuration. This approach avoids any redundancy in computing face points, and naturally improves the continuity of the polynomials across cell faces.

[Figure 3 about here.]

Because of the structured environment which is considered here, one index will be used per direction to describe the computational space. Let the G value of the quadrature point (l, m, n) of the flow solver cell (i, j, k) be denoted $G_{i,j,k}^{l,m,n}$, and its position vector $\boldsymbol{x}_{i,j,k}^{l,m,n}$. Consider the number of quadrature points per direction to be p. For the sake of simplicity of notation, this number of points will be considered equal for each direction. Using the cardinal functions for algebraic interpolation $L^{\alpha}(\boldsymbol{x})$ for $\alpha \in [\![1, p]\!]$, the level set function reconstruction within cell (i, j, k) will be written

$$G_{i,j,k}(\boldsymbol{x}) = \sum_{l=1}^{p} \sum_{m=1}^{p} \sum_{n=1}^{p} L^{l}(\boldsymbol{x}) L^{m}(\boldsymbol{x}) L^{n}(\boldsymbol{x}) G_{i,j,k}^{l,m,n}. \tag{2}$$

Note that this reconstruction is of order $p-1$. This expression can be further simplified to account for the independence of the directions and the simple form of L^{α}. Consider a cell of unit size in one direction and let r_l with $l \in [\![1, p]\!]$ represent the position of the l^{th} quadrature point in that direction, for which then $r_1 = 0$ and $r_p = 1$. Similarly, let the flow solver mesh be defined by the location of the lower, left, proximal corner (x_i, y_j, z_k) for each cell (i, j, k). The

basis polynomials are then written

$$L^{\alpha}(r) = \frac{\prod_{\beta=1,\beta\neq\alpha}^{p}(r - r_{\beta})}{\prod_{\beta=1,\beta\neq\alpha}^{p}(r_{\alpha} - r_{\beta})}. \tag{3}$$

The polynomial reconstruction of the level set function for cell (i, j, k) will then be expressed directly as

$$G_{i,j,k}(x, y, z) = \sum_{l=1}^{p} L^{l}\left(\frac{x - x_i}{x_{i+1} - x_i}\right) \sum_{m=1}^{p} L^{m}\left(\frac{y - y_j}{y_{j+1} - y_j}\right) \sum_{n=1}^{p} L^{n}\left(\frac{z - z_k}{z_{k+1} - z_k}\right) G_{i,j,k}^{l,m,n}. \tag{4}$$

The computational cost of this approach remains limited, since some of the quantities related to the polynomial reconstruction can be pre-computed for a unit cell and stored. The one dimensional equivalent to Eq. 4 is

$$G_i(x) = \sum_{l=1}^{p} L^{l}\left(\frac{x - x_i}{x_{i+1} - x_i}\right) G_i^{l}. \tag{5}$$

This expression can be expanded into

$$G_i(x) = \sum_{l=1}^{p} \frac{\prod_{\beta=1,\beta\neq l}^{p}\left(\left(\frac{x-x_i}{x_{i+1}-x_i}\right) - r_{\beta}\right)}{\prod_{\beta=1,\beta\neq l}^{p}(r_l - r_{\beta})} G_i^{l}, \tag{6}$$

which can be re-written as

$$G_i(x) = P\left(\frac{x - x_i}{x_{i+1} - x_i}\right) \sum_{l=1}^{p} \frac{G_i^{l}}{\left(\frac{x-x_i}{x_{i+1}-x_i} - r_l\right) Q_l}, \tag{7}$$

where

$$P(r) = \prod_{\alpha=1}^{p}(r - r_{\alpha}) \quad \text{and} \quad Q_l = \prod_{\alpha=1,\alpha\neq l}^{p}(r_l - r_{\alpha}). \tag{8}$$

For $l \in [\![1, p]\!]$, the quantity Q_l can be pre-computed and stored, making the polynomial evaluation at any point in space efficient. In order to further reduce the computational cost associated with this sub-cell polynomial reconstruction, these polynomials are created only in a narrow band around the front, i.e. around the G_0 value of the level set function, as illustrated by Fig. 4.

[Figure 4 about here.]

2.3 Semi-Lagrangian transport

Having a sub-cell polynomial reconstruction of the level set function G in each cell is only the first step. An efficient and accurate transport scheme must now be devised. Classically, pseudo-spectral methods have been used to solve conservation laws by computing fluxes from the polynomials directly. This approach is spectrally accurate, but it leads to very strong time step size restrictions. Indeed, as can be seen in Fig. 1, the smallest distance between two quadrature nodes decreases faster for Gauss-type quadratures than for a uniform distribution as p increases, leading rapidly to very severe CFL restrictions. More precisely, it can be shown that the time step size should vary as p^{-2}. This makes this approach unsuited in the case where $p > 3$. As a consequence, a different approach has to be followed that allows to circumvent the time step restriction associated with sub-cell refinement.

Semi-Lagrangian (SL) transport naturally emerges as an attractive alternative. Instead of discretizing Eq. 1, SL transport consists of observing that G should be constant along the trajectory of material points evolving at velocity $\boldsymbol{u}$. Therefore, the trajectory that passes through $\boldsymbol{x}^{n+1}$ at time t^{n+1} can be followed backward in time to $t^n = t^{n+1} - \Delta t$ to obtain the old location $\boldsymbol{x}^n$. The value of the level set function G^{n+1} at $\boldsymbol{x}^{n+1}$ can simply be obtained by noting that $G^{n+1}\left(\boldsymbol{x}^{n+1}\right) = G^n\left(\boldsymbol{x}^n\right)$. Because of the Lagrangian nature of this method, larger time step sizes can be used. The only requirement is the computation of $\boldsymbol{x}^n$ from $\boldsymbol{x}^{n+1}$, which involves solving an ordinary differential equation. Moreover, this approach is efficient and easy to implement. These advantages make SL transport seemingly a beneficial method for the discretization of any advection term. But the method is typically avoided for conserved quantities, since

it does not have any conservation property. This limitation is not a problem in the case of level set, where the transport equation (Eq. 1) is already written in non-conservative form. Another commonly accepted limitation of SL transport is its tendency to be overly diffusive [28]. This is due to the interpolation step that has to be performed to compute $G^n(\boldsymbol{x}^n)$, for $\boldsymbol{x}^n$ is unlikely to coincide with the locations where G^n is known. However, in the framework of SRI, a polynomial reconstruction of order $p-1$ is readily available in each cell. Hence, this polynomial can simply be evaluated at the old location $\boldsymbol{x}^n$, and high accuracy can be expected from the SL transport. It is therefore expected that numerical diffusion will not be an issue. Figure 5 illustrates the transport procedure. Each quadrature point is advected backwards in time (Figs. 5(a) and 5(b)) using a Runge-Kutta scheme. The order of the time integration can be varied between first and fourth order. The effect of the temporal order of accuracy will be discussed at the end of this section. To construct the velocity vector in the RK algorithm, a tri-linear interpolation from the 8 closest points is used. At the old location, the polynomial reconstruction of G is evaluated (Fig. 5(c)), leading to the new G value at the new quadrature point location $\boldsymbol{x}^{n+1}$ (Fig. 5(d)). Note that this procedure is applicable only when the old location $\boldsymbol{x}^n$ lies within the narrow band where the polynomial reconstruction of G is available. In the case where $\boldsymbol{x}^n$ falls outside the refined region, the procedure is modified to use either a boundary condition value for the level set function or to revert back to unrefined level set values that can be transported using a classical scalar transport scheme. As an example, if the level set function is taken to be a signed distance function, i.e.

$$|\phi(\boldsymbol{x}, t)| = |\boldsymbol{x} - \boldsymbol{x}_\Gamma|, \tag{9}$$

where $\boldsymbol{x}_\Gamma$ corresponds to the point on the interface that is closest to $\boldsymbol{x}$, and

$\phi(\boldsymbol{x},t) > 0$ on one side of the interface, and $\phi(\boldsymbol{x},t) < 0$ on the other side, then the values of the level set function need to be provided outside of the narrow band. This can be done either by transporting the level set using a classical approach, preferably using fast, low-order accurate methods, or by extending the distance function from the narrow band to a larger band using a standard re-initialization technique. If the choice is made to use a sharp hyperbolic tangent function, i.e.

$$\psi(\boldsymbol{x},t) = \frac{1}{2}\left(\tanh\left(\frac{\phi(\boldsymbol{x},t)}{2\epsilon}\right)+1\right), \tag{10}$$

where ϵ is the thickness of the function, then, as long as ϵ is small enough, its values outside the refined narrow band can be approximated by 0 or 1, depending on which side of the interface is considered.

[Figure 5 about here.]

2.4 Stabilization technique

Ensuring the stability of the numerical method is fundamental. In multi-domain spectral methods, the stability is ensured by using a Riemann solver that introduces diffusion in the treatment of the fluxes at the cell faces [27,25]. Because SL transport is used here, this approach is not applicable. As a consequence, the stability of the proposed approach could be an issue. In order to alleviate this potential problem, two methods have been employed. The first method has already been described, and consists of reducing the cell to cell oscillations of the polynomials by enforcing that the face quadrature nodes share the same values. While not discretely ensuring the continuity of the polynomials across cells, this greatly reduces the oscillations between each

sub-cell reconstruction. The other method that is employed here to ensure the robustness of the SRI approach is to revert back to local tri-linear interpolations between quadrature nodes when the polynomial is found to oscillate, as shown in Fig. 6. Very simple and straightforward to implement, the idea behind this approach is to check that each polynomial evaluation lies between the level set values of the 8 closest quadrature points. If it is indeed the case, then the polynomial evaluation is considered valid, and therefore trusted. If it is not the case, then it means that locally the polynomial reconstruction is oscillating, and therefore it is replaced by the use of tri-linear interpolation. This was found to be sufficient to remove all oscillations from the computed solutions. Moreover, it was also found that it is only rarely necessary to revert to tri-linear interpolation, and therefore it is expected to have little impact on the accuracy of the method, especially since the second order error introduced in this tri-linear interpolation step is at the sub-cell level, therefore the error is of order $O\left((\Delta x/p)^2\right)$.

[Figure 6 about here.]

2.5 Curvature computation

A central element of multiphase models is the computation of the interface curvature. Indeed, this term governs the surface tension force, which itself is fundamental to capturing accurately two-phase flow phenomena. However, extracting a curvature that converges under mesh refinement from the sub-cell information that is available through SRI can be challenging, since it is likely that sub-cell structures will appear, that should not be seen by the momentum solver mesh. Two pathological cases are presented in Fig. 7. Figure 7(a)

illustrates the case when the interface is flat over each cell. This case would lead to a zero sub-cell curvature, while the curvature on the flow solver mesh is clearly non-zero. The opposite situation is shown in Fig. 7(b), where small sub-cell oscillations in front position are present. In this case, the sub-cell curvature is difficult to compute, and many different formulations are possible. Herrmann [15] chooses to compute local sub-cell curvatures, and then evaluates surface-averages. It is unclear whether such an approach will provide an adequate curvature, since the surface-averaged quantity can be polluted by sub-cell oscillations.

[Figure 7 about here.]

This suggests that the curvature should be computed from information resolvable by the flow solver mesh. In other words, since the length scales below $2\Delta x$ are not resolved by the flow solver, and therefore might not correspond to physical phenomena, these should be filtered out of the interface before the curvature is computed. In order to do this, two steps are introduced:

- *Reconstruction of a signed distance to the interface on the flow solver mesh.* This operation is done by combining a marching cubes (MC) algorithm [29] with a parallel fast marching method (FMM) [8,30], leading to a very fast and efficient algorithm. Only the narrow band of flow solver cells that are required in the computation of the curvature needs to possess the distance information. The initial distance to the interface is measured explicitly using a second order approach illustrated in Fig. 8. From the sub-cell interface information, an algorithm similar to marching cubes (MC) [29] is used to triangulate the interface. Each closest flow solver cell is then explicitly projected onto the triangulated interface, providing both the normal vector $\boldsymbol{n}$

and the distance to the interface. This information is then extended over a few cells using FMM. Because both MC and FMM are at best only second order accurate, the detection of the interface crossings shown in Fig 8(b) is performed using linear interpolation.

[Figure 8 about here.]

- *Least squares computation of curvature from the reconstructed distance function.* Following Marchandise *et al.* [18], a third order least squares algorithm is used to approximate the distance function resulting from the previous step. This approach is found to provide a mesh converging curvature, because of the tendency of the least squares method to smear out some of the numerical errors on the distance field. This is shown by evaluating the curvature of a circle of diameter $D = 1$ centered in a $[0, 2] \times [0, 2]$ domain discretized with various meshes, for which the errors are summarized in Table 1. At least first order convergence of the curvature is recovered, and the curvature is found to converge faster for poorly resolved structures.

[Table 1 about here.]

2.6 Re-initialization

Similar to the discontinuous Galerkin method of Marchandise *et al.* [18], the re-initialization of the level set function is found to be mostly superfluous when using the SRI method. Only when the gradient of the level set function becomes overly small or large, the need to re-initialize the G-field arises, since the triangulation of the sub-cell interface can then become inaccurate. As a result, a re-initialization step is necessary, however it is performed only rarely, typically for every 100 time steps. Two re-initialization strategies have

been employed and compared. The first consists simply of interpolating on the quadrature points the distance field that is reconstructed on the flow solver mesh. While being very inexpensive, this approach removes all the sub-cell information that was stored on the quadrature points, and therefore can introduce significant errors, i.e. of the same order as what is expected from a classical re-initialization step on the flow solver mesh. However, since this re-initialization needs to be performed only rarely, it is not expected to affect the quality of the simulations. The second approach tested here uses the triangulated interface to explicitly re-evaluate the distance of each quadrature point to the interface. This is much more accurate since the errors are second order, based on the sub-cell mesh. Obviously, the cost of this re-initialization step is much greater, since all quadrature points need to be explicitly projected onto the triangulated interface so that their distance can be evaluated. In realistic cases, this operation was found to have the cost of several flow solver time steps. Consequently, the first re-initialization strategy is preferred.

2.7 Solid body rotation of a notched disk

Having described the SRI approach in detail, numerical tests are now presented in order to assess the capability of the method to accurately represent small scale interface transport. The default SRI formulation employed in these test cases uses $p = 5$ Gauss-Lobatto quadrature nodes based on Legendre polynomials, and a second order Runge-Kutta time integration. The effect of varying these parameters will be evaluated throughout this section.

The solid body rotation of a notched circle has often been used to assess the quality of interface transport. In a $[-0.5, 0.5] \times [-0.5, 0.5]$ domain, a circle of

radius 0.15 with a notch of height 0.25 and width 0.05, initially centered at $(0, 0.25)$, undergoes a solid body rotation at angular velocity 2π. A 50^2 flow solver grid is used, and the time step size is set to $1/200$, meaning that 200 time steps are necessary to perform one full rotation of the circle. This leads to a CFL number close to 0.77. The level set function is taken to be a hyperbolic tangent function of thickness $\epsilon = \Delta x/p$. Figure 9 compares the exact solution with the computed solution after one rotation and after 50 rotations.

[Figure 9 about here.]

Even though the mesh used for this simulation is very coarse and resolves the notch on only two cells, the SRI solution appears excellent after one rotation, and it remains very satisfactory even after 50 rotations. This first result suggests that the SRI concept enables a highly accurate description of small interfacial features, even for long time transport.

2.7.1 Effect of temporal accuracy

In order to understand the effect of the order of accuracy of the Runge-Kutta scheme in the SL transport, the same notched circle is now transported using fourth order Runge-Kutta. Figure 10 presents the transported solution after one rotation and after 50 rotations. While the solution after one rotation shows very little difference compared to the interface location computed using second order Runge-Kutta, the solution after 50 rotations is greatly improved by using fourth order Runge-Kutta temporal integration, and compares very well with the exact solution.

[Figure 10 about here.]

2.7.2 *Effect of polynomial order*

The impact of the number of quadrature points per cell on the accuracy of the interfacial transport is now assessed. The previous test clearly showed that the temporal errors become dominant with 50 iterations, therefore the following tests will be performed with the fourth order Runge-Kutta. Figure 11 shows the performance of SRI with $p = 3$. While the solution after one rotation remains satisfactory, although slightly distorted, the interface rapidly deteriorates. Already after 10 rotations, the notch has disappeared. The poor accuracy of the polynomial reconstruction strongly limits the capability of transporting the small-scale notch, but also the capability of accurately representing the circle itself for a long time.

[Figure 11 about here.]

Figure 12 shows the same test case with $p = 9$. As expected, the accuracy is very satisfactory, and even after 50 iterations the computed interface location follows very accurately the exact solution. For the case of the solid body rotation of Zalesak's disk, these results suggest that $p = 5$ is enough to obtain a good solution, but that p could be increased in cases where spatial accuracy is more important. The temporal accuracy of the Runge-Kutta integration has an impact for long time transport, and fourth order seems desirable. However, it can be expected that for a less trivial problem, temporal accuracy may not play such an important role.

[Figure 12 about here.]

2.7.3 Effect of quadrature points

Finally, the impact of the distribution of quadrature points in the cells is discussed. Figure 13 presents results for the Zalesak's disk problem solved using Gauss-Lobatto quadrature based on Chebychev polynomials and Fig. 14 shows results obtained with a uniform distribution of quadrature nodes. The results obtained with the Chebychev-based quadrature are similar to those computed with the Legendre-based quadrature, although they seem slightly less accurate. Indeed, after 50 rotations, the notch does not appear to be as well preserved when using Chebychev-based Gauss-Lobatto quadrature. However, these differences are small and suggest that the accuracy of SRI is only weakly dependent on the choice of polynomials used in the Gauss-Lobatto quadrature.

[Figure 13 about here.]

The uniform distribution, on the other hand, gives very distorted solutions even for the first rotation, and most features of the notched circle are lost after 50 rotations. These poor results are expected, since the accuracy of the polynomial reconstruction is known to be much better when using Gaussian quadrature.

[Figure 14 about here.]

All these parametric tests suggest that:

- Gauss-Lobatto quadrature based on Legendre polynomials performs best,
- $p = 5$ is sufficient to accurately represent the notch, even on two grid cells,
- the accuracy of the Runge-Kutta integration becomes important for long-time transport.

The area conservation errors for different parameters are given in Table 2. Because the errors at the top and at the bottom of the notch tend to compensate, this measure does not represent the overall accuracy of the shape of the disk and might be misleading on its own. However, it provides some clues on the conservation property of SRI. The error values obtained are typically below one percent, even after 50 rotations, which is well below what was observed on the same mesh with the particle level set (PLS) method [13].

[Table 2 about here.]

2.8 Sphere in a deformation field

The velocity field for the previous case was linear, meaning that a tri-linear interpolation of the velocity to the quadrature points location will be exact. In general, this will not be the case, and therefore it is interesting to assess the accuracy of SRI for non-linear velocities. As an example of such a test case, the deformation of a three-dimensional sphere can be employed. This test case was first proposed by Enright *et al.* [13], using a velocity field discussed in LeVeque [31]. A sphere of radius 0.15 is placed at $(0.35, 0.35, 0.35)$ in a unit box, discretized by a 100^3 mesh. The velocity field is set to

$$\begin{aligned} u(x,y,z,t) &= 2\cos(\pi t/T)\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z), \\ v(x,y,z,t) &= -\cos(\pi t/T)\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z), \\ w(x,y,z,t) &= -\cos(\pi t/T)\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z), \end{aligned} \tag{11}$$

where $T = 3$. The time step size is set to 0.005, and the second order Runge-Kutta scheme is used for the temporal integration. Snapshots of the interface as resolved on the flow solver mesh at $t = 0, 0.3, 0.6, 1.0, 1.5, 2.0, 2.5, 3.0$ are

shown in Fig. 15. The geometrical features of the interface are very similar to the results of Enright *et al.* [13], where the thin sheet that is formed at $t = T/2$ is starting to disappear from the flow solver mesh, but the sphere at $t = T$ is still properly recovered.

[Figure 15 about here.]

The sub-cell interface reconstruction provided by the marching cubes algorithm is shown in Fig. 16, where it appears clearly that even at $t = T/2$, where the stretching is maximum, the sub-cell polynomial reconstruction is capable of retaining the thin sheet. Because this sheet is still properly resolved, it is fully recovered on the flow solver mesh when the flow is inverted.

[Figure 16 about here.]

The evolution of the volumetric error as a function of time is shown in Fig. 17.

[Figure 17 about here.]

Even at $t = T/2$, the error remains very small, and is comparable to the results of Enright *et al.* [13]. At the end of the simulation, the volume conservation error is back to less that 0.1%, which is more than an order of magnitude smaller than what was obtained by Enright *et al.* [13]. It is interesting to note that these good results are obtained with $p = 5$, where each cell of the refined band has to transport $(5-1)^3 = 64$ quadrature points, which is similar to the number of particles that were transported by Enright *et al.* [13].

3 Coupling with momentum solver

3.1 Incompressible Navier-Stokes equations

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

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla \cdot \left(\mu \left[\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^t\right]\right) + \boldsymbol{g}, \tag{12}$$

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

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = \frac{\partial \rho}{\partial t} + \boldsymbol{u} \cdot \nabla \rho = 0. \tag{13}$$

The interface Γ separates the liquid from the gaseous phase. In each phase, the material properties are constant, hence $\rho = \rho_l$ 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, $[\boldsymbol{u}]_\Gamma = 0$. However, the pressure is not continuous between the two phases, and we can write

$$[p]_\Gamma = \sigma\kappa + 2\,[\mu]_\Gamma\, \boldsymbol{n}^t \cdot \nabla \boldsymbol{u} \cdot \boldsymbol{n}, \tag{14}$$

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

3.2 Flow solver

3.2.1 Numerical methods

The flow solver used here is NGA, described extensively in Desjardins *et al.* [32]. This solver is structured, parallel, and has been designed for direct numerical simulations (DNS) and large eddy simulations (LES) of complex, reactive, turbulence flows. The numerical methods employed in this code are therefore tailored for the simulation of turbulence. The variables are staggered in space and time, and centered finite difference schemes are used to avoid all numerical dissipation. In the case of single phase computations, primary and secondary conservation are verified, meaning that mass, momentum, and kinetic energy are discretely conserved, which is known to be desirable when simulating turbulence [32]. At the phase-interface, these properties will be lost, because of the specific two-phase procedures described below. While the spatial order of accuracy of NGA can be arbitrarily high, only second order accuracy will be employed here, since the formal order of accuracy will be limited by the interfacial treatment.

Time integration is based on an iterative, second order, Crank-Nicolson formulation. In order to provide additional robustness, each sub-step of the Crank-Nicolson scheme is performed in a semi-implicit manner, using an approximate factorization approach similar to Choi and Moin [33] in order to decouple the spatial directions. Each resulting linear problems is then solved directly using a parallel polydiagonal solver.

3.2.2 Ghost Fluid approach

Because NGA is based on a fractional step approach, a Poisson equation is solved to enforce continuity (Eq. 13). However, the pressure exhibits a jump due to surface tension forces, as shown in Eq. 14, and the pressure gradient term in the Navier-Stokes equations contains the density, which is discontinuous. In order to discretize the pressure Poisson equation, as well as the pressure gradient term, the Ghost Fluid Method (GFM) [2] provides an attractive solution. By first extending the pressure across the interface using Taylor series, standard finite differencing can be used to compute derivatives of the pressure. The pressure jump is then added explicitly afterwards. In one dimension, if the interface Γ is located at x_Γ, between x_i and x_{i+1} and x_{i+1} is in the liquid, we introduce $\theta = (x_\Gamma - x_i)/\Delta x$, where $\Delta x = x_{i+1} - x_i$, as well as a modified density $\rho^\star = \rho_g\theta + \rho_l(1-\theta)$. The variable coefficient pressure Laplacian that is used at x_i in the pressure Poisson equation is then written

$$\left.\frac{\partial}{\partial x}\left(\frac{1}{\rho}\frac{\partial p}{\partial x}\right)\right|_{g,i} = \frac{\frac{1}{\rho^\star}\left(p_{l,i+1} - p_{g,i}\right) - \frac{1}{\rho_g}\left(p_{g,i} - p_{g,i-1}\right)}{\Delta x^2} - \frac{[p]_\Gamma}{\rho^\star \Delta x^2}. \tag{15}$$

More details on the derivation of GFM are provided in [34,35,2], and the derivation of Eq. 15 is given in Desjardins *et al.* [5]. This approach allows to use standard discretization techniques, and has the additional benefit of naturally embedding the surface tension force in a sharp manner in the pressure term, therefore completely avoiding spurious currents when the interface curvature is known exactly.

3.2.3 Viscous treatment

In order to facilitate the implicit formulation for the viscous term, the choice is made here to use CSF [1] to discretize the discontinuous density and viscosity

that appear in the viscous term of the Navier-Stokes equations. For this term only, both the viscosity and the density are modeled by

$$\begin{aligned} \rho(\boldsymbol{x},t) &= \rho_g + (\rho_l - \rho_g)\, H_\Gamma(\boldsymbol{x},t)\,, \\ \mu(\boldsymbol{x},t) &= \mu_g + (\mu_l - \mu_g)\, H_\Gamma(\boldsymbol{x},t)\,, \end{aligned} \tag{16}$$

where H_Γ is a smeared out Heaviside function as in [36]. The lack of a sharp model for the viscous term is not expected to influence the quality of the results significantly.

4 Validation and numerical results

Several two-phase flow test cases are now presented in order to assess the behavior of the proposed approach. The first case of a two-dimensional drop demonstrates that the spurious currents generated by curvature errors remain sufficiently small. Then, a standing wave is computed in order to verify the accurate description of surface tension and viscous forces. To assess the convergence of the method on a more complex two-phase flow problem, a Rayleigh-Taylor instability is computed. Finally, a turbulent two-phase shear layer simulation is presented, displaying the capability of the method to handle turbulent atomization problems.

4.1 Spurious currents

First, a two dimensional drop of diameter $D = 0.4$ is placed in the center of unit size box. Initially, the velocity field is zero, but because of inaccuracies in the computation of interfacial curvature, a spurious flow will be generated. The two fluids have the same density ρ and the same viscosity $\mu = 0.1$, the

surface tension coefficient σ is unity. In order to vary the relative importance of surface tension and viscous forces, the Laplace number $\mathrm{La} = 1/\mathrm{Oh}^2 = \sigma\rho D/\mu^2$ is changed by modifying the densities of both fluids, where Oh is the Ohnesorge number. To assess the intensity of the spurious currents, the Capillary number $\mathrm{Ca} = |u_{\max}|\mu/\sigma$ is computed at a non-dimensional time $t\sigma/(\mu D) = 250$. The simulations are performed on a 32×32 mesh, and the time step size is varied to verify the capillary CFL restriction. Detailed parameters and results are reported in Table 3.

[Table 3 about here.]

The resulting capillary numbers show little dependence on the Laplace number, and the values of Ca remain very small. Therefore, it is expected that the spurious currents should not affect two-phase flow simulations based on the SRI method.

4.2 Standing wave

Next, the interaction of surface tension forces with viscous effects is assessed by simulating the viscous decay of a two-dimensional standing wave with various density ratios. In a $[0, 2\pi] \times [0, 2\pi]$ domain, two fluids are initially separated by an interface defined by the zero iso-contour of

$$\phi(x, y) = \pi - y + A_0 \cos(2\pi x/\lambda), \tag{17}$$

where λ is set to 2π and A_0 is set to 0.01λ. In the x-direction, periodic boundary conditions are employed, while the y-direction assumes top and bottom symmetry. The surface tension coefficient is set to $\sigma = 2$, and the kinematic viscosity ν of both fluids is set to be equal. In the case of similar kinematic

viscosities, Prosperetti [37] provides a theoretical solution to the evolution of the wave amplitude, which we will use to compare our results. The time is non-dimensionalized using the inviscid oscillation frequency $\omega_0 = \sqrt{\frac{\sigma}{\rho_1+\rho_2}}$, where ρ_1 and ρ_2 are the densities in each fluid. Following the numerical study of Herrmann [15], two cases are considered. The first one assumes $\rho_1 = \rho_2 = 1$ and $\nu = 0.064720863$. The simulations are performed on various meshes, from 8×8 to 64×64, until $\omega_0 t = 20$ is reached. Figure 18 presents both the evolution of the wave amplitude with time for the different meshes in comparison to the theory, and the time evolution of the error in amplitude. The rms value of the error is then summarized in Table 4. Figure 19 shows that close to second order convergence is obtained for this problem. Moreover, while the 8×8 mesh predicts an incorrect frequency, leading to large errors in amplitude, the 16×16 mesh leads to very satisfactory results.

[Figure 18 about here.]

[Table 4 about here.]

[Figure 19 about here.]

The second case considers a density ratio of $\rho_2/\rho_1 = 1000$, and $\nu = 0.0064720863$. Figure 20 shows the results for this case, and Table 5 summarizes the rms of the amplitude error. Again the convergence shown in Fig. 21 is between first and second order, and the 16×16 solution is already very satisfactory. This confirms that the proposed approach is capable of accurately predicting this flow, even with a relatively small number of points per wavelength.

[Figure 20 about here.]

[Table 5 about here.]

[Figure 21 about here.]

4.3 Rayleigh-Taylor instability

The SRI approach is now employed to simulate the growth of a two-dimensional Rayleigh-Taylor instability. Numerous studies have used this problem to characterize the quality of interface transport methods, see e.g. [15]. However, many of these do not consider surface tension effects. The case studied here follows the simulation of Gomez *et al.* [38], which includes surface tension forces. In a $[1 \times 4]$ domain, two fluids about each other are initially separated by an interface defined by the zero iso-contour of

$$\phi\left(x, y\right) = y + A_0 \cos\left(2\pi x\right), \tag{18}$$

where A_0 is taken to be 0.05. The top fluid has a density $\rho_1 = 1.225$, while the density of the bottom fluid is set to $\rho_2 = 0.1694$. Both fluids have the same dynamic viscosity, $\mu_1 = \mu_2 = 0.00313$, and the surface tension coefficient is set to $\sigma = 0.1337$. Five different meshes are considered, ranging from 32×128 to 512×2048. Figure 22 presents the temporal evolution of the interface location for the finest mesh. These results are in good agreement with the simulations of Gomez *et al.* [38], and show the expected formation and growth of a spike of heavy fluid, while a bubble of light fluid rises. When comparing the solution between the various meshes for different times, as shown in Fig. 23, it can be seen that the mesh convergence is rather slow. A more quantitative evaluation of the rate of convergence of the solution is performed in Table 6, where the error in the maximum depth of the spike with the different meshes is reported at $t = 1.0$, 1.1 and 1.2, considering the finest solution as the reference solution. The error in spike penetration is then plotted as a function of the mesh size in

Fig. 24. The resulting error initially converges slowly, but reaches second order convergence rapidly. However, if one considers more complex features of the flow that involve larger curvatures, such as the extremities of the mushroom cap shape at the end of the spike, it is clear that slower convergence is achieved.

[Figure 22 about here.]

[Figure 23 about here.]

[Table 6 about here.]

[Figure 24 about here.]

4.4 Two-phase shear layer

Finally, the SRI approach is employed in a complex realistic two-phase flow problem, namely the simulation of a three-dimensional turbulent shear-layer. The computation is based on the experimental work of Ben Rayana [39]. The simulation is run on a $512 \times 128 \times 256$ mesh. Figure 25 illustrates the setup of the shear layer, with water flowing on a flat surface, while air is injected at higher velocity above the water surface. The two flows are separated at injection by a lip of thickness $e = 2.2$ mm, and their velocity profiles at injection are taken from the experimental measurements. The properties of both fluids, including the surface tension coefficient, are those of water and air, with the exception that the water density has been reduced to $\rho_l = 50$ kg/m^3 in order to ensure numerical stability. As in the experiment, the momentum flux ratio is set to $M = 16$, with a bulk air velocity of $U_g = 20$ m/s and a bulk water velocity of $U_l = 0.7746$ m/s, for a height of the water layer of 10 cm.

[Figure 25 about here.]

Figure 26 shows a top view of the interface in a region of about 9 cm $\times$ 7 cm right after the lip, both for the experiment and the simulation. From a qualitative point of view, the simulated interface corresponds to the experimental results. The first large Kelvin-Helmholtz–type structure is properly recovered, including some lateral wrinkling of the interface within the wave. Secondary lateral instabilities then follow, leading to fingering of the interface, and to the generation of droplets. While the experimental picture shown here does not display any ligament but only a few droplets, they were observed experimentally. Note that the difference in density ratio could explain the tendency of the simulation to generate more ligaments, potentially because of aerodynamic forces. The SRI approach appears robust even in the presence of a complex turbulent flow.

[Figure 26 about here.]

Finally, Fig. 27 presents the relative cost of the different components of the NGA code during the course of the shear layer simulation. Only the three main elements, namely the SRI solver, the pressure solver, and the velocity solver are included here. Note that the sum of these three components corresponds to almost 100% of the cost of the full NGA code. SRI is found to correspond to the third of the cost of the full code, well behind the pressure solver, which accounts for almost 50% of the simulation cost. This confirms that even for a realistic parallel simulation with complex topology, the SRI approach remains affordable, with a cost well below that of the pressure solver.

[Figure 27 about here.]

5 Conclusion

A spectral refinement approach for the description of interfacial flows has been proposed. This method introduces quadrature nodes in each cell to enable the construction of high order polynomials to represent a level set function with improved accuracy. By providing a sub-cell description of the interface structures, accurate transport becomes possible, even for the smallest resolved scales. The cost of the method remains reasonable even for large numbers of quadrature points thanks to a semi-Lagrangian transport scheme. On basic transport tests, this approach has been found to be more accurate than PLS [13], with the additional benefit that no frequent re-initialization was found to be necessary, as in [18]. The coupling of the SRI approach with the flow solver NGA, a structured code with numerical schemes tailored for turbulence simulation, has then been performed. Using GFM, a sharp description of the interfacial discontinuities is possible. Moreover, additional robustness is obtained by using a semi-implicit integration of the Navier-Stokes equations. The resulting solver is used to simulate several canonical two-phase flow problems, and satisfactory results are obtained. The full method has finally been employed for the simulation of a two-phase shear layer. Despite the complexity of the flow structures, the SRI approach was found to remain robust, and to predict an interface shape that agrees qualitatively with the experimental results.

Acknowledgements

The authors wish to express their gratitude to Dr. Guillaume Blanquart and Dr. Perrine Pepiot-Desjardins for many helpful discussions about this work, and to Dr. Guillaume Balarac for his work on the shear layer simulation. We also gratefully acknowledge funding by NASA and by the DOE through the ASC program.

References

[1] J. U. Brackbill, D. B. Kothe, C. Zemach, A continuum method for modeling surface tension, J. Comput. Phys. 100 (1992) 335–354.

[2] 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.

[3] O. Desjardins, V. Moureau, E. Knudsen, M. Herrmann, H. Pitsch, Conservative level set/ghost fluid method for simulating primary atomization, ILASS Americas 20th Annual Conference on Liquid Atomization and Spray Systems.

[4] 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.

[5] O. Desjardins, V. Moureau, H. Pitsch, An accurate conservative level set/ghost fluid method for simulating primary atomization, J. Comp. Phys. 227 (18) (2008) 8395–8416.

[6] 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.

[7] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Ann. Rev. Fluid Mech. 31 (1999) 567–603.

[8] J. A. Sethian, Level set methods and fast marching methods, 2nd Edition, Cambridge University Press, Cambridge, UK, 1999.

[9] S. Osher, R. Fedkiw, Level set methods and dynamic implicit interfaces, Springer, New York, 2003.

[10] S. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, J. Comput. Phys. 100 (1992) 25–37.

[11] 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.

[12] S. P. van der Pijl, A. Segal, C. Vuik, A mass-conserving level-set method for modelling of multi-phase flows, Int. J. Numer. Meth. Fluids 47 (2005) 339–361.

[13] 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.

[14] E. Coyajee, M. Herrmann, J. B. Boersma, Simulation of dispersed two-phase flow using a coupled volume-of-fluid/level-set method, in: Proceedings of the 2004 Summer Program, Center for Turbulence Research, Stanford, CA, 2004.

[15] 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.

[16] S. E. Hieber, P. Koumoutsakos, A lagrangian particle level set method, Journal of Computational Physics 210 (1) (2005) 342–367.

[17] M. Sussman, M. Y. Hussaini, A discontinuous spectral element method for the level set equation, Journal of Scientific Computing 19 (1) (2003) 479–500.

[18] 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.

[19] H. Touil, M. Hussaini, M. Sussman, Tracking discontinuities in hyperbolic conservation laws with spectral accuracy, J. Comput. Phys. 225 (2) (2007) 1810–1826.

[20] X. D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, J. Comput. Phys. 115 (1994) 200–212.

[21] G. S. Jiang, C. W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1996) 202–228.

[22] 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.

[23] J. Shi, Y.-T. Zhang, C.-W. Shu, Resolution of high order WENO schemes for complicated flow structures, Journal of Computational Physics 186 (2) (2003) 690–696.

[24] M. Herrmann, G. Blanquart, V. Raman, Flux corrected finite volume scheme for preserving scalar boundedness in reacting large-eddy simulations, AIAA Journal 44 (12) (2006) 2879–2886.

[25] P. G. Huang, Z. J. Wang, Y. Liu, An implicit space-time spectral difference method for discontinuity capturing using adaptive polynomials, AIAA Computational Fluid Dynamics Conference (2005).

[26] C. Canuto, M. Y. Hussaini, A. Quarteroni, T. A. Zang, Spectral methods in fluid dynamics, Springer-Verlag, 1988.

[27] Y. Liu, M. Vinokur, Z. J. Wang, Spectral difference method for unstructured grids I: Basic formulation, Journal of Computational Physics 216 (2) (2006) 780–801.

[28] R. Paoli, T. Poinsot, K. Shariff, Testing semi-Lagrangian schemes for two-phase flow applications, in: Proceedings of the 2006 Summer Program, Center for Turbulence Research, Stanford, CA, 2006.

[29] W. E. Lorensen, H. E. Cline, Marching cubes: A high resolution 3d surface construction algorithm, in: M. C. Stone (Ed.), Proceedings of the 14th Annual Conference on Computer Graphics and interactive Techniques, SIGGRAPH '87, ACM, New York, NY, 1987, pp. 163–169.

[30] M. Herrmann, A domain decomposition parallelization of the fast marching method, in Annual Research Briefs (Center for Turbulence Research, Stanford, CA) (2005).

[31] R. J. LeVeque, High-resolution conservative algorithms for advection in incompressible flow, SIAM Journal on Numerical Analysis 33 (2) (1996) 627–665.

[32] 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.

[33] H. Choi, P. Moin, Effects of the computational time step on numerical solutions of turbulent flow, J. Comput. Phys. 113 (1994) 1–4.

[34] M. Kang, R. Fedkiw, X. D. Liu, A boundary condition capturing method for multiphase incompressible flow, J. Sci. Comput. 15 (2000) 323–360.

[35] X. D. Liu, R. Fedkiw, M. Kang, Boundary condition capturing method for poisson equation on irregular domains, J. Sci. Comput. (2000) 151–178.

[36] 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.

[37] A. Prosperetti, Motion of two superposed viscous fluids, Phys. Fluids 24 (1981) 1217–1223.

[38] P. Gomez, J. Hernandez, J. Lopez, On the reinitialization procedure in a narrow-band locally refined level set method for interfacial flows, Int. J. Numer. Meth. Eng. 63 (10) (2005) 1478–1512.

[39] F. Ben Rayana, Contribution à l'étude des instabilités interfaciales liquide-gaz en atomization assistée et tailles de gouttes, Ph.D. thesis, Institut National Polytechnique de Grenoble, France (2007).

List of Figures

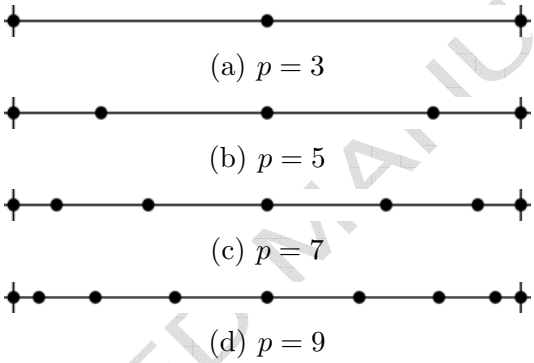
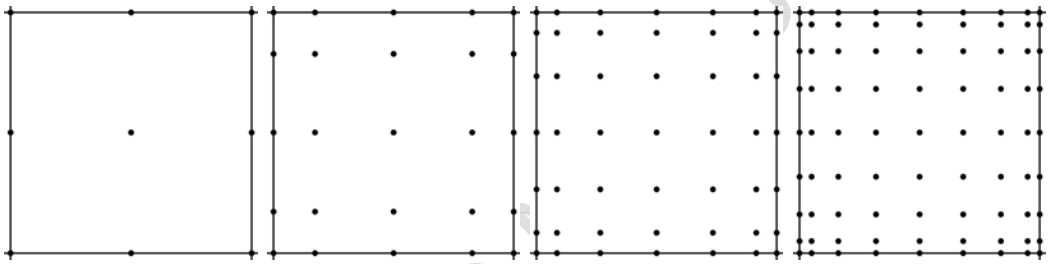


Figure 1. Location of the Gauss-Lobatto quadrature nodes based on the Legendre polynomials in one dimension for various numbers of unknowns p.



(a) $p = 3$ (b) $p = 5$ (c) $p = 7$ (d) $p = 9$

Figure 2. Location of the Gauss-Lobatto quadrature nodes based on the Legendre polynomials in two dimensions for various numbers of unknowns p^2.

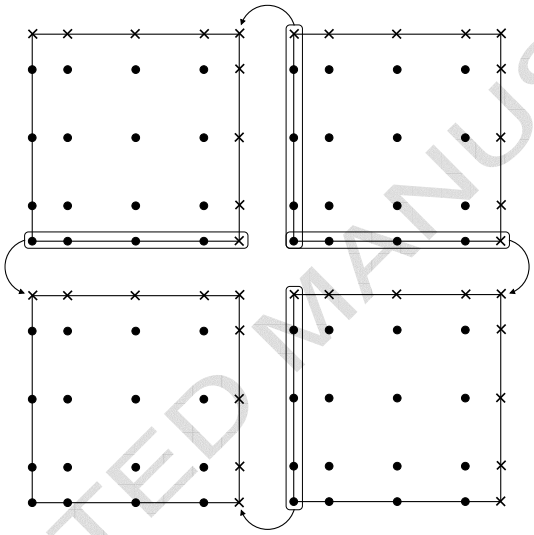


Figure 3. Ghost quadrature point locations (crosses), normal quadrature points (circles), and inter-cell communication patterns (arrows).

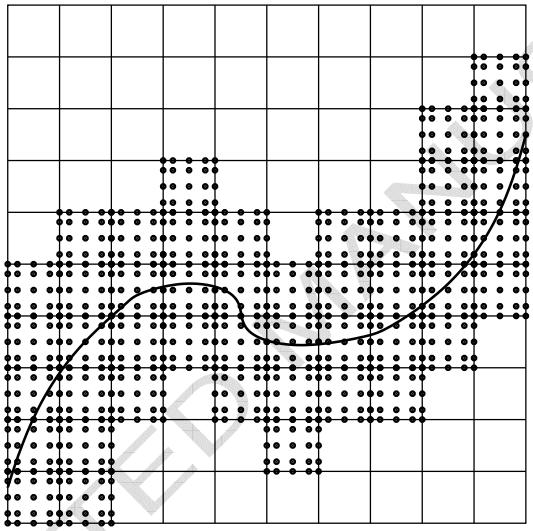


Figure 4. Narrow band spectral refinement around the G_0 iso-contour of the level set function G (thick line).

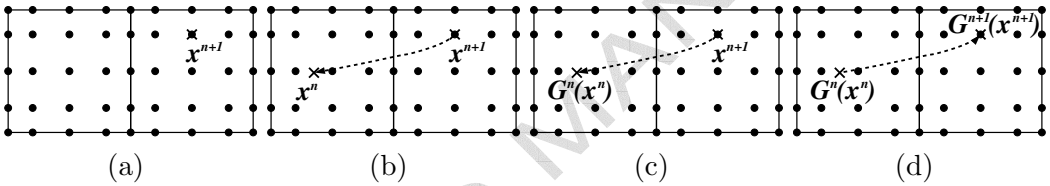


Figure 5. Semi-Lagrangian transport of the spectrally refined level set function G.

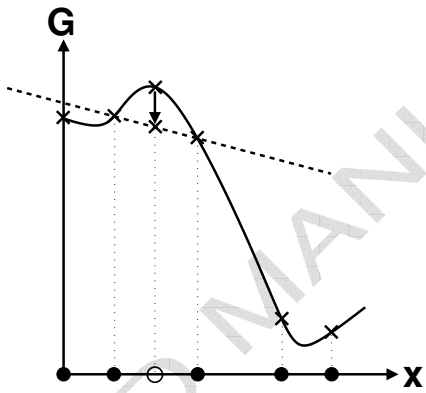
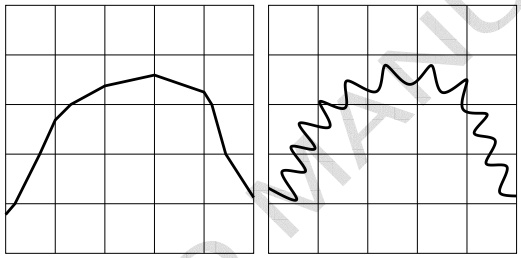
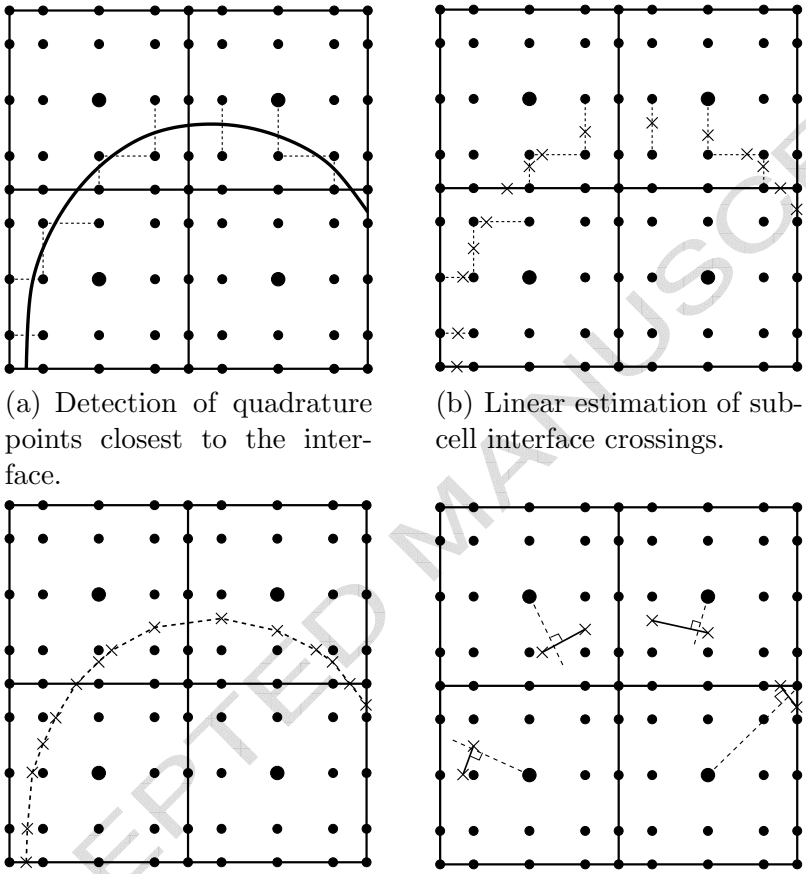


Figure 6. One dimensional illustration of the stabilization approach for SRI: when evaluating the polynomial function (solid line) at a new location (open circle) outside the quadrature points (filled circles), the resulting value is replaced by a linear interpolation between the closest neighbors (dashed line) if the polynomial evaluation does not lie between the values of the closest neighbors.



(a) (b)

Figure 7. Pathological cases for sub-cell curvature computation.



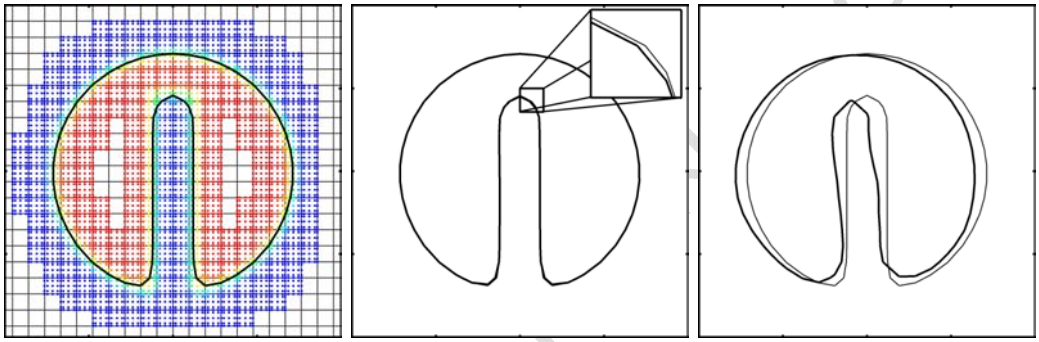
(a) Detection of quadrature points closest to the interface.

(b) Linear estimation of subcell interface crossings.

(c) Triangulation of the subcell interface.

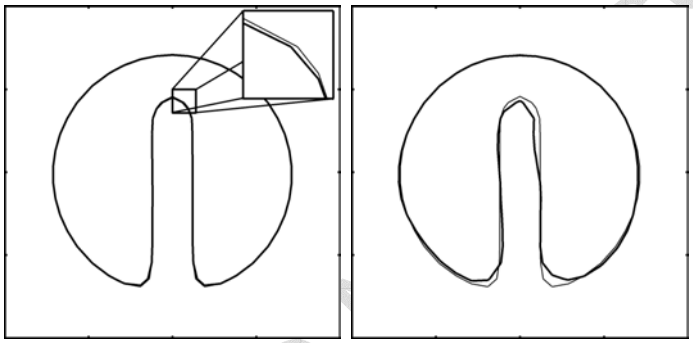
(d) Geometric projection of the flow solver centroids on the closest triangles.

Figure 8. Second order distance reconstruction on the flow solver mesh.



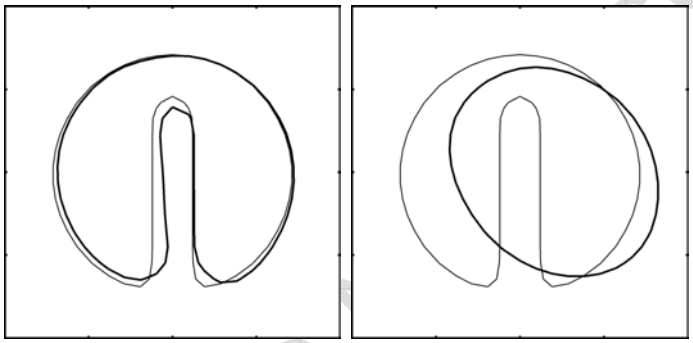
(a) (b) (c)

Figure 9. Solid body rotation of Zalesak's disk with second order Runge-Kutta for the SL transport, and $p = 5$. (a) Exact interface location (thick line), quadrature nodes colored by the level set function, and flow solver mesh. (b) Solution after one rotation: exact solution (thin line) and SRI solution (thick line). (c) Solution after 50 rotations: exact solution (thin line) and SRI solution (thick line).



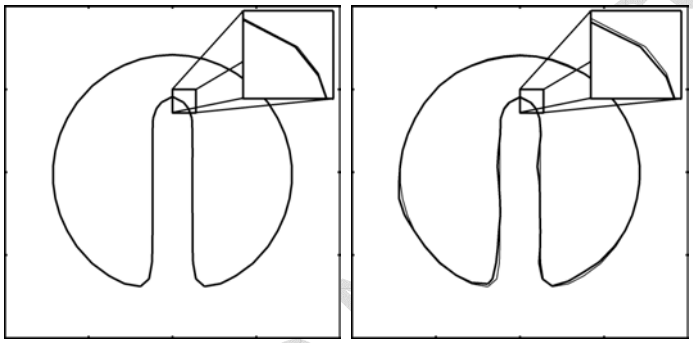
(a) (b)

Figure 10. Solid body rotation of Zalesak's disk with fourth order Runge-Kutta for the SL transport, and $p = 5$. (a) Solution after one rotation: exact solution (thin line) and SRI solution (thick line). (b) Solution after 50 rotations: exact solution (thin line) and SRI solution (thick line).



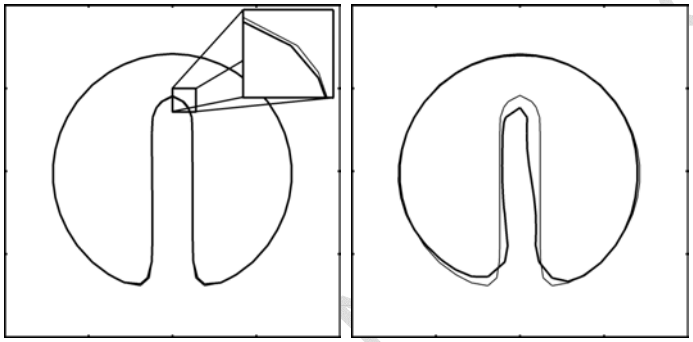
(a) (b)

Figure 11. Solid body rotation of Zalesak's disk with fourth order Runge-Kutta for the SL transport, and $p = 3$. (a) Solution after one rotation: exact solution (thin line) and SRI solution (thick line). (b) Solution after 10 rotations: exact solution (thin line) and SRI solution (thick line).



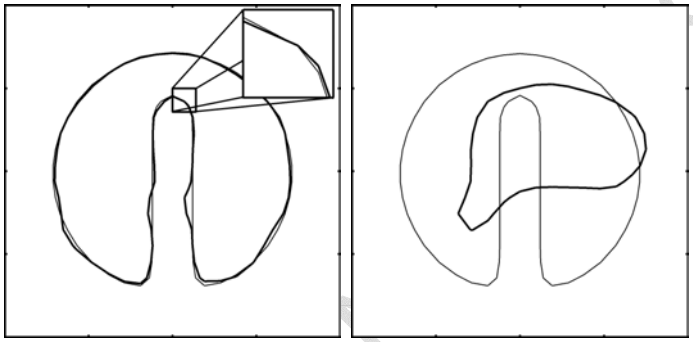
(a) (b)

Figure 12. Solid body rotation of Zalesak's disk with fourth order Runge-Kutta for the SL transport, and $p = 9$. (a) Solution after one rotation: exact solution (thin line) and SRI solution (thick line). (b) Solution after 50 rotations: exact solution (thin line) and SRI solution (thick line).



(a) (b)

Figure 13. Solid body rotation of Zalesak's disk with fourth order Runge-Kutta for the SL transport, and $p = 5$, using Gauss-Lobatto quadrature points based on the Chebychev polynomials. (a) Solution after one rotation: exact solution (thin line) and SRI solution (thick line). (b) Solution after 50 rotations: exact solution (thin line) and SRI solution (thick line).



(a) (b)

Figure 14. Solid body rotation of Zalesak's disk with fourth order Runge-Kutta for the SL transport, and $p = 5$, using uniformly distributed quadrature points. (a) Solution after one rotation: exact solution (thin line) and SRI solution (thick line). (b) Solution after 50 rotations: exact solution (thin line) and SRI solution (thick line).

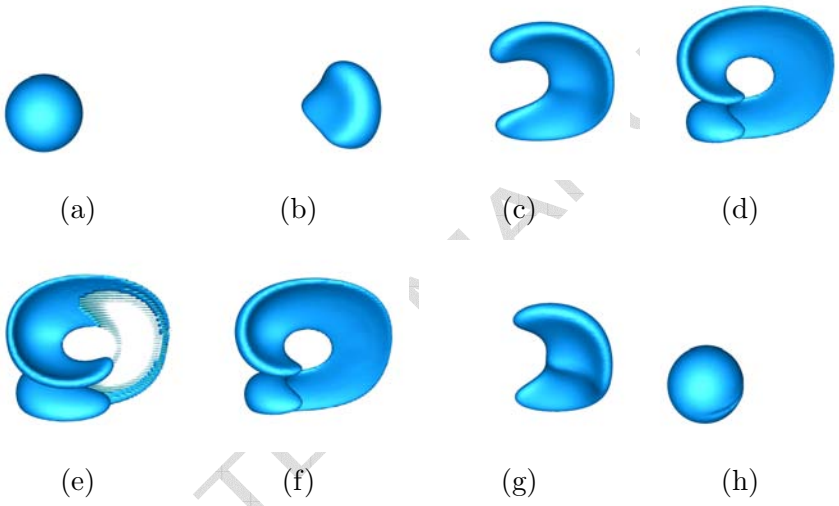


Figure 15. Sphere in a three-dimensional deformation velocity field. Evolution of the location of the interface at the flow solver level as a function of time.

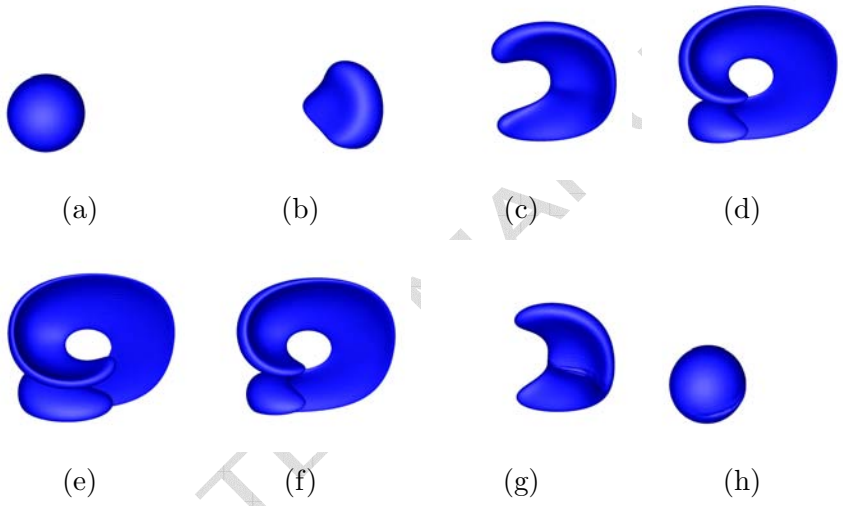


Figure 16. Sphere in a three-dimensional deformation velocity field. Evolution of the location of the interface at the sub-cell level as a function of time.

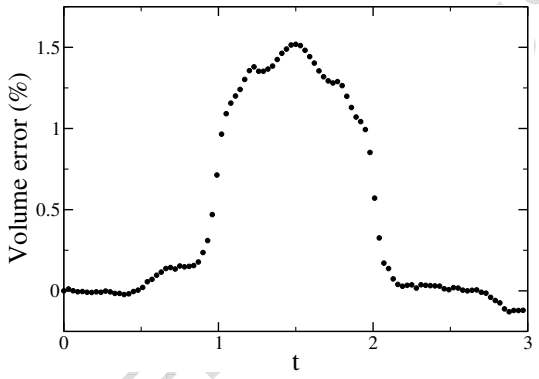
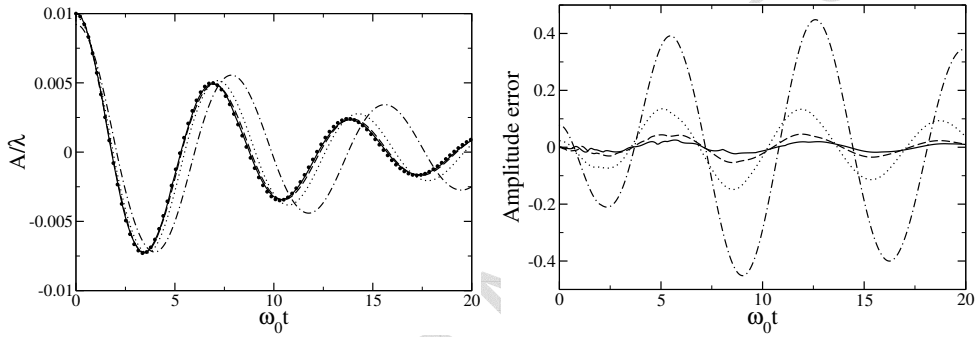


Figure 17. Volume error as a function of time for the sphere in a three-dimensional deformation velocity field.



(a) Time evolution of the wave amplitude

(b) Time evolution of the amplitude error

Figure 18. Damped surface wave problem with unity density ratio. 8×8 mesh (dash-dotted line), 16×16 mesh (dotted line), 32×32 mesh (dashed line), 64×64 mesh (solid line), and theory (symbols).

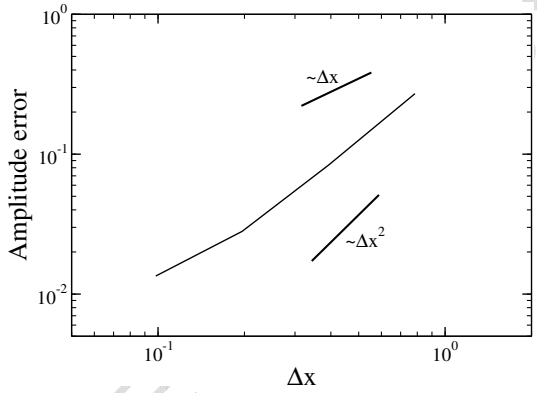
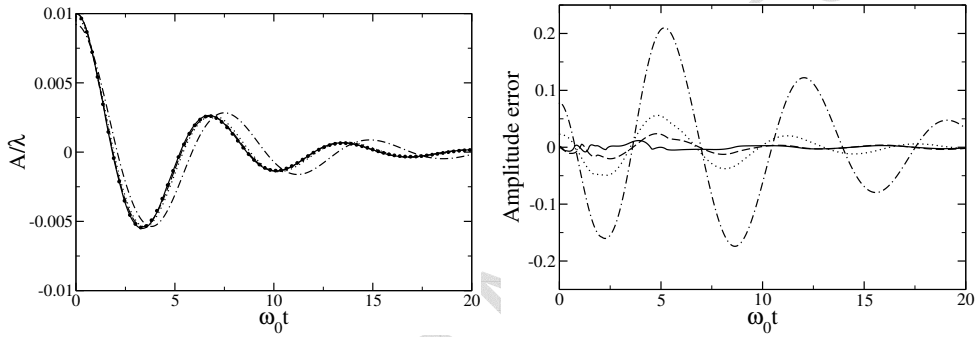


Figure 19. Convergence of the amplitude error for the standing wave problem with unity density ratio.



(a) Time evolution of the wave amplitude

(b) Time evolution of the amplitude error

Figure 20. Damped surface wave problem with density ratio 1 : 1000. 8×8 mesh (dash-dotted line), 16×16 mesh (dotted line), 32×32 mesh (dashed line), 64×64 mesh (solid line), and theory (symbols).

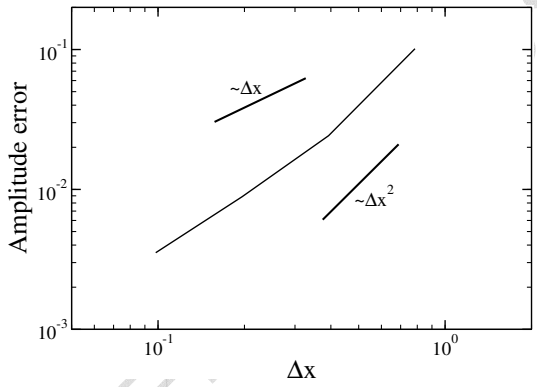
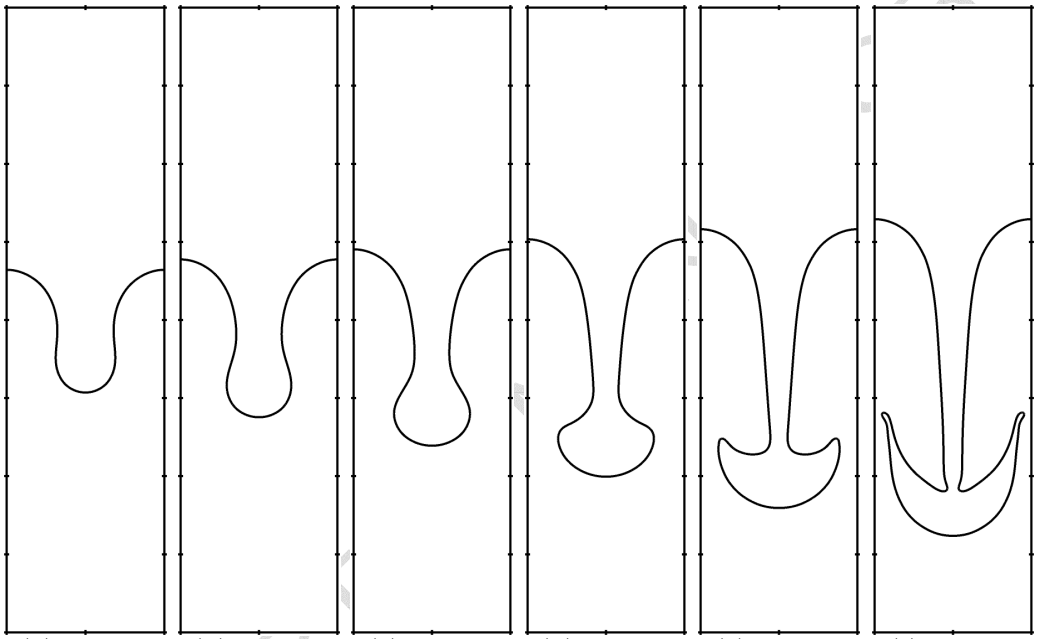
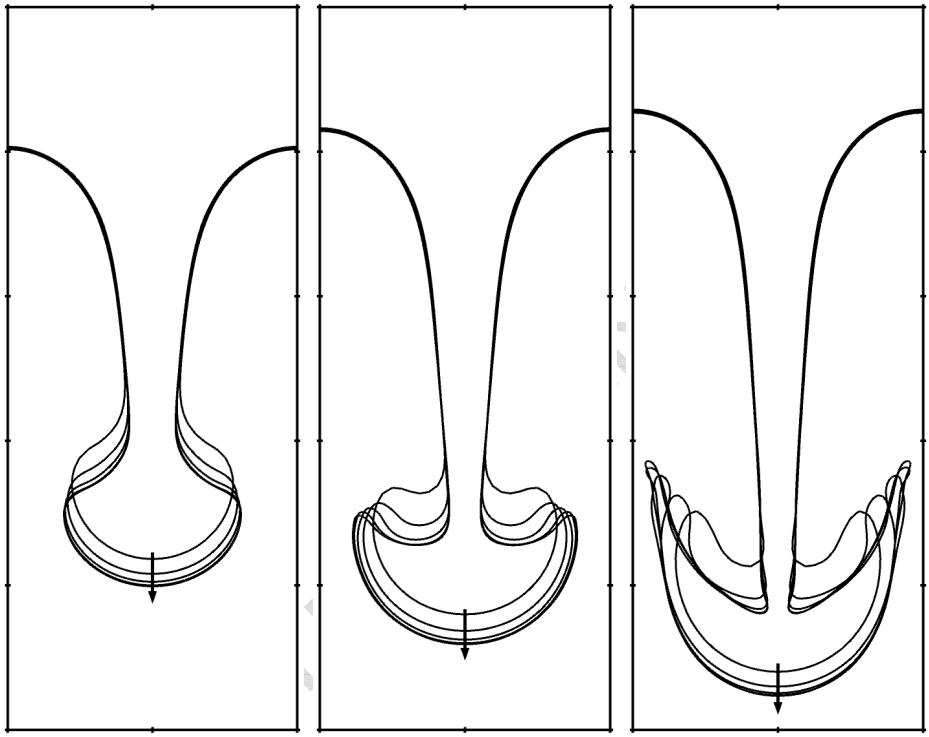


Figure 21. Convergence of the amplitude error for the standing wave problem with density ratio 1 : 1000.



(a) $t = 0.7$ (b) $t = 0.8$ (c) $t = 0.9$ (d) $t = 1.0$ (e) $t = 1.1$ (f) $t = 1.2$

Figure 22. Phase-interface shape as a function of time for the Rayleigh-Taylor instability problem on a 512×2048 mesh.



(a) $t = 1.0$ (b) $t = 1.1$ (c) $t = 1.2$

Figure 23. Phase-interface shapes as a function of time for the Rayleigh-Taylor instability problem. Arrow indicates increasing mesh sizes (32×128, 64×256, 128×512, 256×1024, and 512×2048).

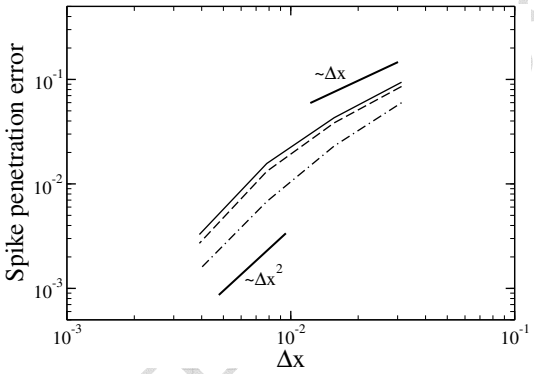


Figure 24. Convergence of the error in spike penetration for the Rayleigh-Taylor instability problem at different times: $t = 1.0$ (solid line), $t = 1.1$ (dashed line), and $t = 1.2$ (dot-dash line).

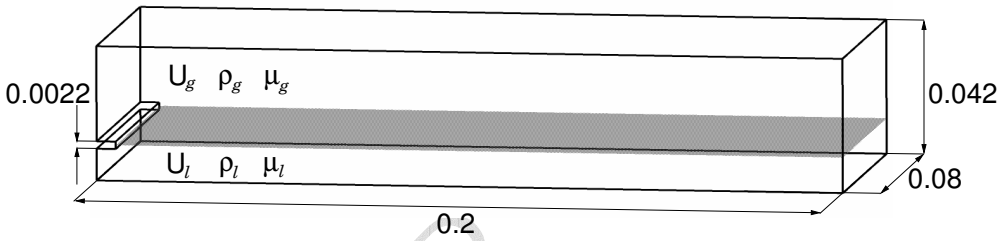
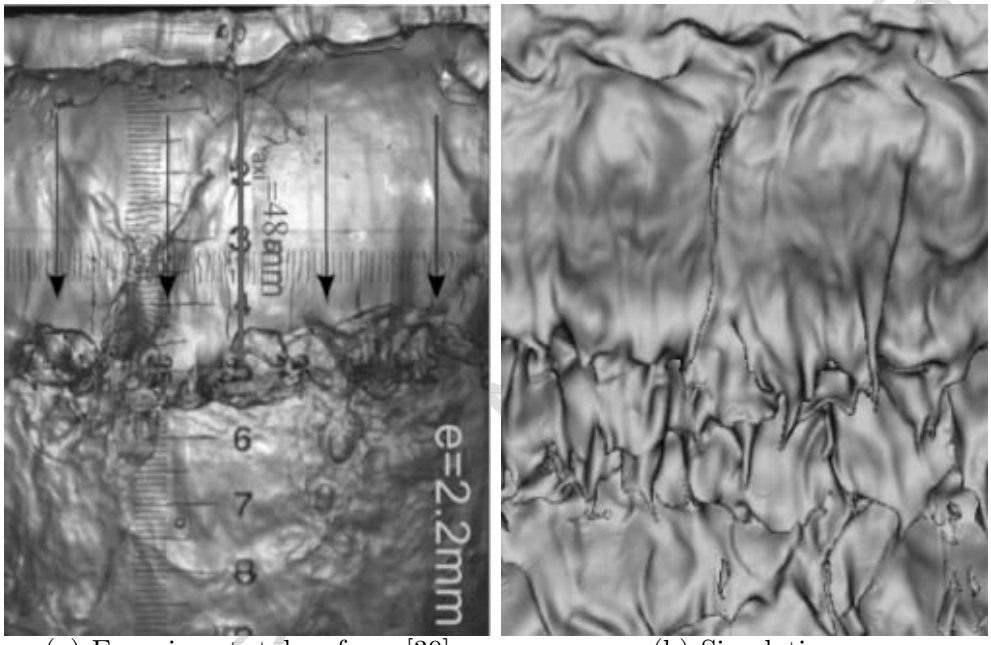


Figure 25. Schematics of the computational setup for the two-phase shear layer flow. Dimensions are given in meters.



(a) Experiment, taken from [39] (b) Simulation

Figure 26. Top view of the phase-interface in the shear layer flow. Flow direction is from top to bottom.

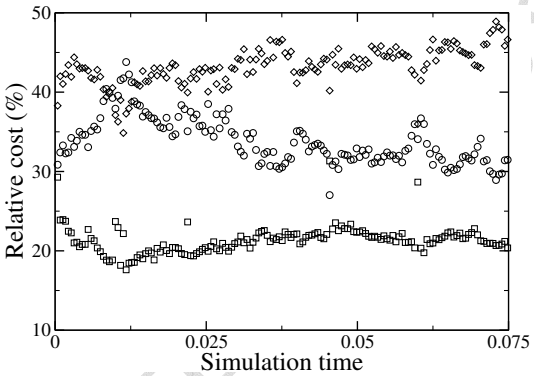


Figure 27. Relative cost of the different components of the NGA code during the course of the shear layer simulation: SRI solver (circles), velocity solver (squares), and pressure solver (diamonds).

List of Tables

Mesh	$D/\Delta x$	Error
8×8	4	0.62297
16×16	8	0.11251
32×32	16	0.06577
64×64	32	0.03378

Table 1
Convergence of the L_2 error of least squares curvature with mesh spacing.

p	Temporal order	% Area loss after one rotation	% Area loss after 50 rotations
5	2	0.366	1.100
5	4	0.366	0.595
9	4	0.488	0.899

Table 2
Area conservation errors for Zalesak's disk problem with different parameters.

ρ	0.3	3	30	300	3000	30000
La	12	120	1200	12000	120000	1200000
Δt	0.0006	0.002	0.006	0.02	0.06	0.2
Ca	1.10×10^{-5}	0.90×10^{-5}	2.93×10^{-5}	2.09×10^{-5}	7.46×10^{-5}	3.77×10^{-5}

Table 3
Dependence of the magnitude of parasitic currents with the Laplace number for a static droplet with surface tension on a 32×32 mesh.

Mesh	Error
8×8	0.27082
16×16	0.08356
32×32	0.02808
64×64	0.01346

Table 4
RMS value of the amplitude error for the standing wave problem with unity density ratio.

Mesh	Error
8×8	0.10127
16×16	0.02421
32×32	0.00887
64×64	0.00353

Table 5
RMS value of the amplitude error for the standing wave problem with density ratio $1 : 1000$.

Mesh	$t = 1.0$	$t = 1.1$	$t = 1.2$
32×128	0.09408	0.08566	0.05987
64×256	0.04300	0.03841	0.02306
128×512	0.01569	0.01318	0.00678
256×1024	0.00328	0.00270	0.00151

Table 6
Error in maximum penetration of the spike of heavy fluid compared to the finest simulation for the Rayleigh-Taylor instability problem, at different times using coarser meshes.