A comparison of interface tracking methods in a low-Mach formulation

Victor Boniou, Thomas Schmitt, Aymeric Vié

A comparison of interface tracking methods in a low-Mach formulation

Victor Boniou [*1], Thomas Schmitt[1], Aymeric Vié[1,2]

[1]Laboratoire EM2C UPR 288, CNRS, CentraleSupélec, Université Paris-Saclay, 3, rue Joliot-Curie 91192 Gif-sur-Yvette cedex France
[2]Fédération de Mathématiques de CentraleSupélec, CNRS, 3, rue Joliot-Curie 91192 Gif-sur-Yvette cedex France
*Corresponding author email: victor.boniou@centralesupelec.fr

Abstract

The objective of the present work is to compare the main Eulerian sharp-interface tracking methods using up-to-date numerical strategies from the literature. The study is focused on the specific case of low-Mach number formulation for the phase dynamics, and on a finite-volume cartesian-grid discretization. Volume of fluid and Level Set methods both rely on the resolution of an Eulerian transport equation that describes the interface and an additional step to prevent huge deviation from the interface representation due to numerical diffusion and dispersion in the transport step. In the VOF framework, this results in the transport of the volume fraction with a piecewise linear construction of the interface (PLIC). In the LS framework, this takes the form of a distance function transport which can be either a signed distance (here called Standard Level Set (SLS)) or an hyperbolic tangent (called the Conservative Level Set (CLS)) followed by a reinitialization step which ensures that the transported variable remains a signed distance or an hyperbolic tangent respectively. For each method, the numerical scheme used for advection and additional step are selected because of their proven accuracy and effectiveness in the literature for our specific framework.
Our comparison is based on 2D and 3D canonical cases of the literature (Zalesak's Disk rotation, vortex-in-a-box, sphere deformation). Our attention is drawn on a detailed analysis of mass conservation and transport accuracy, with the use of shared metrics for all methods.

Keywords
Incompressible flow, Volume of fluid, Level set, Accurate conservative level set

Introduction
Direct Numerical Simulation (DNS) of two-phase problems implies that at least two non-miscible fluids are present. This subject has been widely explored in the last decades, and several approaches were developed in order to capture the interface. This work restrains the scope to the simulation of incompressible two-phase flows with a sharp interface representation. To this aim, the most used techniques in the literature are Volume of fluid (VOF) [1], Level set (LS) [2] and Front tracking (FT) [3]. While VOF and LS methods rely on an Eulerian representation of the interface, FT methods use Lagrangian markers all belonging to the interface. This last method will not be investigated here, it is a totally different way of handling the interface transport.
Making a choice between those methods is not trivial as they all seem to have advantages and drawbacks regarding the application they are used for. Comparison studies are already available in the literature. However, they are often limited to 2D [4, 5] or to a single class of method [6, 7, 8]. Moreover, recent improvements have to be included for an updated study. Here, we investigate the capability of VOF, Standard LS and Conservative LS for under-resolved structures handling, mass conservation, accuracy and computational cost on adapted 2D and 3D test cases.
Sec. 2 gives a brief description of the three methods considered in this work. A comparison of these methods on test cases is presented in Sec. 3 and finally, a conclusion on the interface tracking method choice is detailed in Sec. 4.

Interface description

The Eulerian representation of the interface relies on an indicator function χ defined as follows

$$\chi(\mathbf{x}) = \left\{ \begin{array}{ll} 0 & \text{if } \mathbf{x} \in \Omega_g \\ 1 & \text{if } \mathbf{x} \in \Omega_l \end{array} \right. \tag{1}$$

with Ω_g the gas part, Ω_l the liquid part and $\mathbf{x}$ any point of the domain.
Its evolution is given by

$$\frac{D\chi}{Dt} = \frac{\partial \chi}{\partial t} + \mathbf{u} \cdot \nabla \chi = 0 \tag{2}$$

with $\mathbf{u}$ the velocity vector.

To solve this last equation numerically, χ needs a discrete representation. The VOF method intents to solve equation (2) on the computational mesh by introducing the volume fraction of liquid f [9] which is a discrete version of χ. For a given computational cell $\mathcal{C}$ of volume $\mathcal{V}_\mathcal{C}$

$$f = \frac{\int_\mathcal{C} \chi(\mathbf{x})d\mathbf{x}}{\mathcal{V}_\mathcal{C}} \tag{3}$$

f is still a very sharp and challenging quantity to solve numerically. As a circumvent, the standard LS method solves the minimal signed distance to the interface ϕ instead [2]

$$\phi(\mathbf{x}) = \left\{ \begin{array}{ll} -\min\limits_{\forall \mathbf{x}_\Gamma \in \Gamma} |\mathbf{x}_\Gamma - \mathbf{x}| & \text{if } \mathbf{x} \in \Omega_g \\ \min\limits_{\forall \mathbf{x}_\Gamma \in \Gamma} |\mathbf{x}_\Gamma - \mathbf{x}| & \text{if } \mathbf{x} \in \Omega_l \\ 0 & \text{if } \mathbf{x} \in \Gamma \end{array} \right. \tag{4}$$

where $\mathbf{x}_\Gamma$ is a point belonging to the interface Γ. However, this method does not solve a quantity related to χ any more and suffers from mass conservations issues. To improve mass conservation, the LS method can also use another definition of ϕ which is a smooth Heaviside ψ computed from ϕ, this is the conservative LS [15]

$$\psi = \frac{1}{2} \left(\tanh \left(\frac{\phi}{2\epsilon} \right) \right) \tag{5}$$

with 2ϵ the interface thickness.

The evolution of the color function $c = f, \phi$ or ψ is deduced from the following advection equation which, in the context of low Mach two-phase flows, with divergence-free velocity $\nabla \cdot \mathbf{u} = 0$ is

$$\frac{\partial c}{\partial t} + \nabla \cdot (\mathbf{u}c) = 0 \tag{6}$$

VOF method

In this work, the VOF method is based on a PLIC [10] representation of the interface .

The interface is reconstructed as a line (plane in 3D) in a mixed cell $(0 < f < 1) : \mathbf{x} \cdot \mathbf{n} = d$. The interface normal $\mathbf{n}$ is evaluated at second order using the ELVIRA [10] procedure while the plane parameter d is deduced from the normal and the volume fraction using the analytic relations of a chopped cube [11].

In order to alleviate possible numerical diffusion or dispersion, fluxes are computed from the geometric reconstruction using the Weymouth and Yue dimensional-splitting (WY) scheme [12]. This allows to control the thickness of the interface and maintain it to one cell.

A dimensional-splitting scheme results in solving successive 1D advection problems :

$$\frac{\partial f}{\partial t} + \frac{\partial (u_s f)}{\partial x_s} = f \frac{\partial u_s}{\partial x_s} \tag{7}$$

with s the sweep direction. $\nabla \cdot \mathbf{u} = 0$ does not imply $\dfrac{\partial u_s}{\partial x_s} = 0$ and a direct consequence of this is the RHS of equation (7). Hence, it is non trivial to have an exactly conservative scheme solving those equations successively. In the WY scheme, the RHS of equation (7) $f\dfrac{\partial u_s}{\partial x_s}$ is replaced by $f_c\dfrac{\partial u_s}{\partial x_s}$ with $f_c = \chi(\mathbf{x}_{\mathcal{C}})$ the indicator function evaluated at the center of a given cell. After each sweep, a PLIC reconstruction is performed for flux computation of the next step. This gives the following algorithm for a timestep

1. Compute the compression/dilatation factor f_c

2. Perform a PLIC reconstruction by computing $\mathbf{n}$ and d in all mixed cells

3. Solve (7) in a given direction using Euler Implicit scheme

4. Repeat 2 and 3 for all directions to obtain f^{n+1}

It can be shown that this algorithm leads to exact mass conservation given a divergence-free velocity field $\mathbf{u}$. WY is the only method able to keep exact mass conservation in a directional-splitting fashion. Unsplit methods also achieve such conservation [13], but the geometrical flux construction is way more demanding in term of implementation and computation effort. Both approaches are limited to second order accuracy because of the geometrical nature of the fluxes based on linear reconstruction.

SLS method
The standard LS advection method relies on the transport of ϕ. The discretization has a huge impact on the mass conservation as numerical diffusion leads to artificial mass loss. This subject has been widely explored, and it has been shown that level set performs well with high-order schemes. A complete comparison of ENO, WENO and HOUC schemes of different orders has been done in [7]. Here, a WENO5 scheme is used for advection.
One important aspect of the SLS method is to keep the property $|\nabla\phi| = 1$. If this is no longer true, then the computation of topological properties from ϕ (such as normal and curvature) would suffer from huge errors and spurious behaviour. The diffusion and dispersions of numerical schemes used to solve advection will lead to the loss of this property, hence an additional step is required. The redistancing step can be written as an evolution equation of ϕ in pseudo time τ [2]

$$\frac{\partial \phi}{\partial \tau} + \text{sign}(\phi_0)\left(|\nabla\phi| - 1\right) = 0 \tag{8}$$

This results in solving a Hamilton-Jacobi equation with the corresponding Hamiltonian $\mathcal{H}\left(\phi, \nabla\phi\right) = \text{sign}(\phi_0)\left(1 - |\nabla\phi|\right)$. It can be again solved with high accuracy with a HJ-WENO5 scheme [14]. This leads to the following algorithm for a time step

1. Advance the interface by solving equation (6) to obtain ϕ^*

2. Compute the regularised distance sign $\text{sign}(\phi_0) = \dfrac{\phi^*}{\sqrt{(\phi^*)^2 + \Delta x^2}}$

3. Perform 2 iterations of equation ((8) to obtain ϕ^{n+1} with the pseudo time step $\Delta\tau = 0.5\Delta x$

The overall method does not conserve mass as the reinitialization step is not conservative and conservation of ϕ doesn't imply conservation of mass. However, the scheme accuracy can be very good if high order schemes are applied, the implementation is straightforward even in multidimensional cases and the computational cost is fairly low.

CLS method

Conservative LS method relies on a sharper representation of the interface by introducing ψ, a smooth version of the indicator function χ. ψ is a volume fraction with a controlled interface width ϵ, usually chosen as $\epsilon = \Delta x / 2$.

For the same reasons that ϕ cannot maintain the property $|\nabla \phi| = 1$ during transport, there is no guarantee that the hyperbolic tangent profile ψ will remain unchanged. This takes the form of local modifications of the interface thickness which would lead to a bad representation of the interface and topology computation from it.

An additional equation has to be solved in pseudo time to overcome this problem [15]

$$\frac{\partial \psi}{\partial \tau} + \nabla \cdot (\psi(1 - \psi)\mathbf{n} - \epsilon(\nabla \psi)) = 0 \tag{9}$$

The stable and accurate method considered here is the ACLS of Chiodi et al. [16] where the reinitialization is reformulated to

$$\frac{\partial \psi}{\partial \tau} = \nabla \cdot \left(\frac{1}{4 \cosh^2 \left(\frac{\phi_{map}}{2\epsilon} \right)} \left(|\nabla \phi_{map} \cdot \mathbf{n}| - 1 \right) \mathbf{n} \right) \tag{10}$$

with $\phi_{map} = \epsilon \log \left(\frac{\psi}{1 - \psi} \right)$.

In order to ensure ψ boundness, a BHOUC5 [17] discretization is used for (6) which is a HOUC5 scheme with a switch to first order upwind when undershoots or overshoots occur in the transport process. The terms in equation (10) are discretized using second order finite differences while the normal $\mathbf{n}$ is computed based on $\mathbf{n} = \dfrac{\nabla \phi_{FMM}}{|\nabla \phi_{FMM}|}$ with ϕ_{FMM} a distance function computed from a Fast Marching Method (FMM) algorithm [18]. The construction of ϕ_{FMM} is fundamental in the method as it removes all oscillatory behaviour of ψ in the computation of normals. This leads to the following algorithm for a time step

1. Advance the interface by solving equation (6) to obtain ψ^*

2. Compute the signed distance ϕ_{FMM} from the isocontour $\psi^* = 0.5$ and ϕ_{map} from ψ^*

3. Perform 1 iteration of equation (10) to obtain ψ^{n+1} with the pseudo time step $\Delta \tau = 0.25 \Delta x$

The overall method leads to a better mass conservation than the SLS because both transport and reinitialization steps are conservative and the transported color function ψ represents the liquid volume in the limit $\Delta x \to 0$. Moreover, it allows to use high order schemes too. However, the FMM algorithm implies more implementation and computation effort. To improve efficiency, the FMM reconstruction and reinitialization are only performed in a narrow band of 10 cells near the interface.

Results

Error metrics

The evaluation of the different methods is based on accuracy and mass conservation. To provide an equal base of comparison for accuracy, the shape error E_{shape} is based on the regularized Heaviside function $H_\epsilon(c)$ which is defined as f for VOF, $\frac{1}{2} \tanh \left(\frac{\phi}{2\epsilon} \right)$ for SLS or ψ for CLS. They all are smooth versions of χ with an interface thickness of $2\epsilon = \Delta x$ such that $H_\epsilon(c) \xrightarrow[\Delta x \to 0]{} \chi$. The accuracy error is then defined as

$$E_{shape} = \sum_{i=1}^{N_C} |H_\epsilon(c_{i,T}) - H_\epsilon(c_{i,0})| \mathcal{V}_i \tag{11}$$

with 0 the initial time of the simulation, T the final time, $N_{\mathcal{C}}$ the number of cells in the computational domain and $\mathcal{V}_i$ the volume of the cell $\mathcal{C}_i$.

Regarding mass conservation, VOF achieves it at machine precision and will not be displayed for obvious reasons. The CLS method conserves ψ up to machine precision, however it does not corresponds exactly to the volume enclosed in the 0.5 isocontour. This is why a simplex decomposition is performed in each cell in order to find the intersections between the cell and the interface and compute the related volume. This method leads to a second order approximation of a volume enclosed in a given isocontour [19]. This same approach is performed for enclosed volume in the 0 isocontour of SLS. The mass error is then defined as

$$E_{mass} = \int_{t=0}^{T} |\Delta V| dt \tag{12}$$

with $\Delta V = V(t + dt) - V(t)$ the variation of liquid volume computed from the simplex decomposition method evaluated at time t and $t + dt$. Compared to a more classical measure of mass loss based on the difference between initial and final volume V_0 and V_T, this metric gives a better overview of the mass variation of a method with respect to time.

Zalesak's disk rotation

The Zalesak's disk [20] test case consists in a notched circle of radius 0.15 initially centered at $(0.5, 0.75)$ in a $[1 \times 1]$ domain. The notched width is 0.05 and notched length is 0.25. The velocity field is a solid rotation defined as $\mathbf{u} = \begin{pmatrix} 2\pi(0.5 - y) \\ 2\pi(x - 0.5) \end{pmatrix}$. The results are given for a full rotation of the disk corresponding to a simulation time $T = 1$ at a CFL number of 0.5.
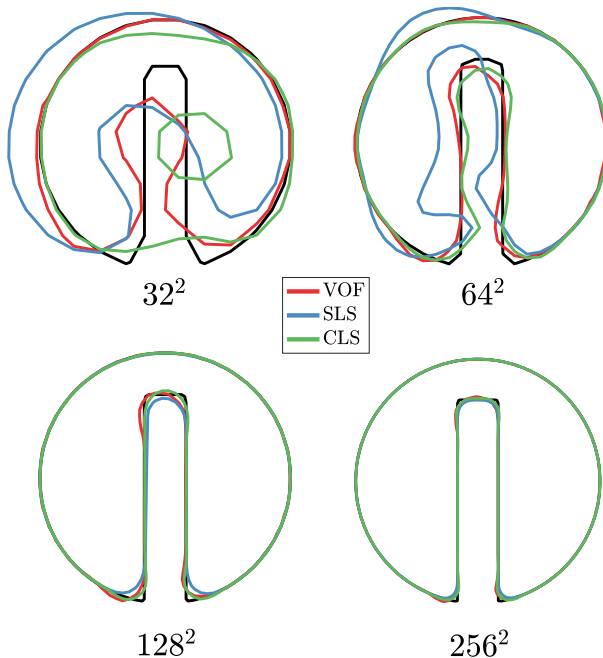

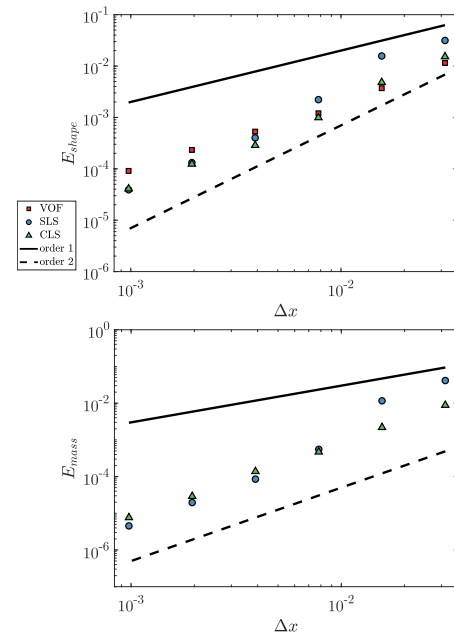**Figure 1.** Initial and final shape for the Zalesak's disk rotation

Figure 2. Mesh convergence of E_{shape} and E_{mass} for the Zalesak's disk rotation

The final shape is compared with the initial shape for all methods in Figure 1. At the lowest resolution 32^2, VOF keep the notch while SLS is shifted and CLS merge the two sides of the notch. This shows how VOF is able to capture poorly resolved structure. From 64^2 resolution all methods maintain the notch during the whole computation. In Figure 2, the error convergence is displayed for all three methods. One can notice that VOF performs well even at very low resolution while SLS and CLS perform better for high resolution with an asymptotic second order behaviour. Regarding mass conservation, CLS is better than SLS for the low resolution meshes while they both have the same conservation properties for the highest resolution.

Vortex in a box

Another classical test case is the vortex-in-a-box first used by Leveque to evaluate high order advection schemes in incompressibles flows [21]. A circle of radius 0.15 is initially centered at $(0.5, 0.75)$ in a $[1 \times 1]$ domain. The velocity field is deduced from the stream function $\Psi = \dfrac{1}{\pi} \sin^2(\pi x) \sin^2(\pi y) \cos\left(\pi \dfrac{t}{T}\right)$ such that it is reversed at $t = T/2$. The results are given for the final time $T = 8$ at an initial CFL number of 0.32 (Δt is kept constant for the whole simulation).
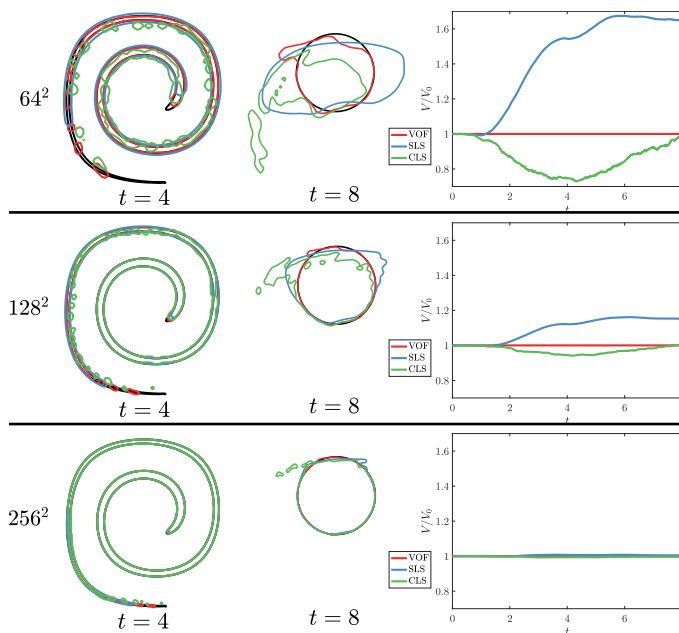
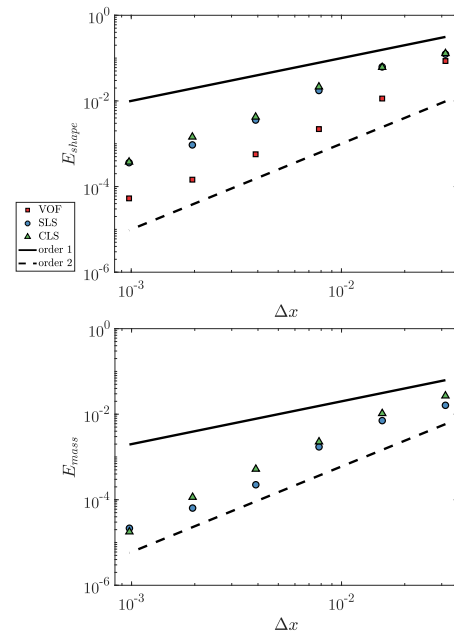**Figure 3.** Vortex in a box shape at $t = 4$ and $t = 8$ with the temporal mass evolution

Figure 4. Mesh convergence of E_{shape} and E_{mass} for the Vortex in a box

In Figure 3, VOF and CLS tend to produce numerical atomization in the thinner structures of the serpentine while the SLS shows a more robust behaviour at the cost of mass conservation. In Figure 4 VOF is performing better for all resolutions, this is expected as the 1024^2 case still implies a thin tail which is not well resolved. Surprisingly, SLS is better at conserving mass than CLS based on our total volume variation metric. However, the CLS method is able to retrieve a final mass close to the initial one as demonstrated in Figure 3 which is not the case at all for SLS.

Sphere deformation

A 3D test case is the sphere deformation, also presented in [21]. A sphere of radius 0.15 is initially centered at $(0.35, 0.35, 0.35)$ in a $[1 \times 1 \times 1]$ domain. It is then advected by a velocity field which induces a combination of stretching in the x-y plane and the x-z plane with an inversion at $t = T/2$. The results are given for the final time $T = 3$ at an initial CFL number of 0.32 (Δt is kept constant for the whole simulation).

The same conclusions can be drawn from this 3D test case : VOF and CLS produce some numerical atomization when the interface is under-resolved as it can be seen in Figure 5 for $t = 1.5$ at a mesh resolution of 128^3. From Figure 6, VOF is still the most accurate method while CLS and SLS show similar mass conservation. Apparently, the transition from 2D to 3D doesn't affect the overall behaviour of the methods.
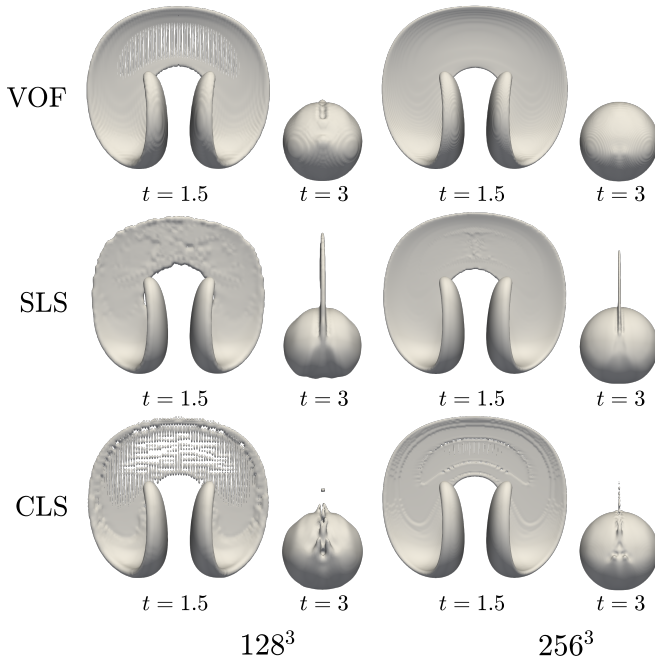
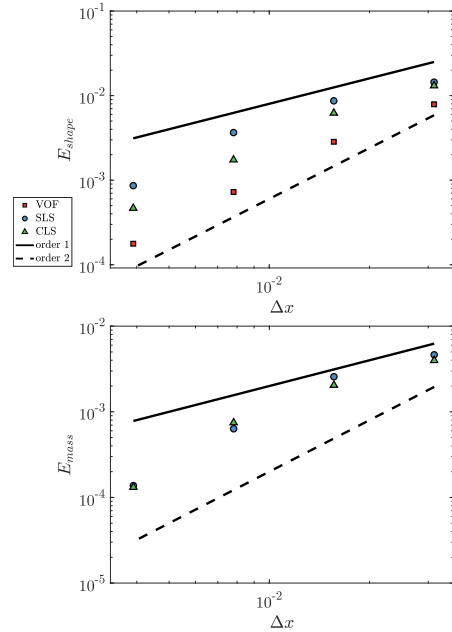**Figure 5.** Shape at $t = 1.5$ and $t = 3$ for the sphere deformation

Figure 6. Mesh convergence of E_{shape} and E_{mass} for the sphere deformation

A word on computational time

To complete the comparison, the computational cost is compared between the methods. In Figure 7, the mean walltime per iteration is given for all cases. 2D cases were run on 16 CPUs while the 3D case was run on 64 CPUs. In 2D configurations, the VOF method is more effi-
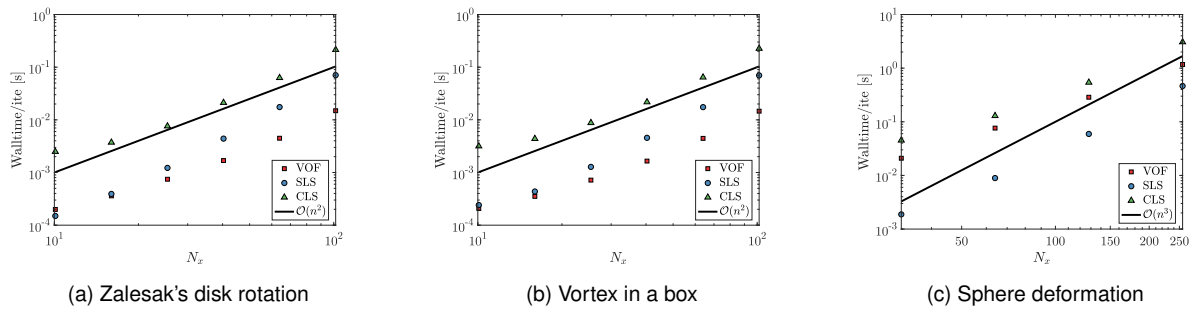

(a) Zalesak's disk rotation

(b) Vortex in a box

(c) Sphere deformation

Figure 7. Walltime per iteration for the different test cases

cient than SLS and CLS and seems to scale better with number of elements. This is because VOF method only requires to compute fluxes and reconstruction on the interface cells and their neighbours, hence the computational time does not scale in N_{elem} but in N_Γ. Also, 2D computation of geometric flux and reconstruction is really fast in a split fashion. As expected, SLS is more efficient than CLS because of the reinitialization which is more demanding in the case of CLS.

In 3D, geometry operations are more expensive, so VOF falls behind SLS in efficiency. However, the scaling remains better with the same explanations as 2D. Finally, CLS is still the most expensive method in 3D.

Conclusion

The 3 canonical test cases considered in this work enlighten the capabilities and limits of VOF, SLS and CLS methods.

When the interface is well resolved, the SLS method is very interesting for its simplicity, efficiency and high accuracy. Surprisingly, the CLS doesn't seem to improve mass conservation

with respect to the E_{mass} metric. Moreover, the method implies an increase in computational time which is more important than SLS and VOF. Finally the VOF method is capable of handling under-resolved structures and conserves mass exactly at the cost of more implementation efforts and computational time in 3D.

Please, note that other metrics can be used for mass conservation and would lead to different conclusions. For example, if one chose to compute the mass error as $V_T - V_0$, then the CLS method is better than SLS for the reversed test cases as it is able to retrieve the mass that was lost during the first half of the simulation. However a gain or a loss of mass is still a mass variation due to numerical errors, and in real applications, the flow field won't reverse itself to allow compensation of mass errors. This is why E_{mass} has been introduced here.

Finally, this work is limited to uniform cartesian grids where VOF remains fairly simple and effective. This is not the case anymore when dealing with unstructured meshes because of the geometric nature of the method.

Acknowledgements

References

[1] Noh, W. F. and Woodward, P.,June 28-July 2. 1976, 15th international conference on numerical methods in fluid dynamics.

[2] Sussman, M., Smereka, P., Osher, S., 1994, *Journal of Computational physics*, 114(1), pp. 146-159.

[3] Unverdi, S. O., Tryggvason, G., 1979, *Journal of computational physics*, 100 (1), pp. 25-37.

[4] Gerlach, D., Tomar, G., Biswas, G., Durst, F., 2006, *International Journal of Heat and Mass Transfer*, 49(3-4), 740-754.

[5] Abadie, T., Aubin, J., Legendre, D., 2015, *Journal of Computational Physics*, 297, 611-636.

[6] Fuster, D., Agbaglah, G., Josserand, C., Popinet, S., Zaleski, S., 2009, *Fluid dynamics research*, 41(6), 065001.

[7] Nourgaliev, R. R., Theofanous, T. G., 2007, *Journal of Computational Physics*, 224(2), pp. 836-866.

[8] Solomenko, Z., Spelt, P. D., Naraigh, L. O., Alix, P., 2017, *International Journal of Multiphase Flow*, 95, 235-256.

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

[10] Pilliod, J. E., Puckett, E. G., 2004, *Journal of Computational Physics*, 199(2), pp. 465-502.

[11] Scardovelli, R., Zaleski, S., 2000, *Journal of Computational Physics*, 164(1), pp. 228-237.

[12] Weymouth, G., Yue, D., 2010, *Journal of Computational Physics*, 229(8), pp. 2853-2865.

[13] Owkes, M., Desjardins, O., 2014, *Journal of Computational Physics*, 270, pp. 587-612.

[14] Jiang, G. S., Peng, D., 2000, *SIAM Journal on Scientific computing*, 21(6), pp. 2126-2143.

[15] Olsson, E., Kreiss, G., 2005, *Journal of computational physics*, 210(1), pp. 225-246.

[16] Chiodi, R., Desjardins, O., 2017, *Journal of Computational Physics*, 343, pp. 186-200.

[17] Herrmann, M., Blanquart, G., Raman, V., 2006, *AIAA journal*, 44(12), pp. 2879-2886.

[18] McCaslin, J. O., Courtine, É., Desjardins, O., 2014 *Journal of Computational Physics*, 274, pp. 393-412.

[19] Min, C., Gibou, F., 2008, *Journal of Computational Physics*, 227(22), pp. 9686-9695.

[20] Zalesak, S. T., 1979, *Journal of computational physics*, 31 (3), pp. 335-362.

[21] Leveque, R. J., 1996, *SIAM Journal on Numerical Analysis*, 33(2), pp. 627-665.