V European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2010
J. C. F. Pereira and A. Sequeira (Eds)
Lisbon, Portugal,14-17 June 2010

AN OPTIMAL APPROACH FOR VELOCITY INTERPOLATION IN MULTILEVEL VOF METHOD

A. Cervone*, S. Manservisi* and R. Scardovelli*

*DIENCA - University of Bologna,
via dei Colli, 16 - Bologna, Italy
e-mail: {a.cervone,sandro.manservisi,ruben.scardovelli}@unibo.it

Key words: Fluid Dynamics, Two Phase Flow, Volume Of Fluid Method, Multilevel Approach

Abstract. *The use of mesh refinement techniques is becoming more and more popular in computational fluid dynamics, from multilevel approaches to adaptive mesh refinement. In particular, direct numerical simulations of two-phase flow have seen recently the introduction of a finer grid for interface representation. In general, the mesh used for surface tension calculations requires higher resolution than the grid where the conservation equations are discretized and the introduction of these auxiliary grids overcomes some well-known limitations of Volume-of-Fluid (VOF) and Level Set methods. In particular for VOF methods, the interface discontinuity at the cell boundary can cause the deformation of the interface shape at low resolution even for simple translations and solid-body rotations. In fact different interface lines interact with each other in the reconstruction step at a distance of a couple of grid spacings, causing distortion and eventually numerical break-up of the interface. The methodology we present provides a smoother description of the interface and its geometrical properties which are necessary for an accurate capillary force evaluation. The use of different grids for Navier-Stokes equations and interface representation requires the projection of the velocity field from the coarse to the higher resolution grid. This procedure is not trivial, especially for incompressible flows where the divergence-free constraint must be satisfied on both grids. In this paper we present a new method to estimate the velocity field on the fine mesh based on an optimal approach. This algorithm allows us to satisfy the divergence-free constraint in every cell of the fine grid in two- and three-dimensional geometry. This is achieved by a constrained minimization of an objective functional. The proposed functional is the integral of the square of the error between the computed and the linear interpolation velocity field. The minimization is subject to the divergence-free constraint which therefore must be always satisfied. In this way, the projected velocity field is divergence-free and the interface advection does not generate unrealistic situations that could lead to an unphysical behavior. In the last part of the paper we present some results obtained with the above method in two-phase flow simulations in order to assess its reliability and robustness.*

1 INTRODUCTION

In this work we present a multilevel VOF method and a projection velocity approach that improve two phase flow simulations with front tracking techniques. Adaptive Mesh Refinement (AMR)[6,9,14] is a common technique nowadays and many authors have introduced approaches in which the computational grid for interface resolution is finer than the surrounding grid.[4] The use of a finer grid allows to solve some difficulties of front capturing algorithms, especially Volume-of-Fluid (VOF) methods, and to reproduce correctly a drop or a filament with a characteristic length comparable to the grid spacing. These numerical issues may change the fluid topology and lead to an unphysical evolution of the interface.

VOF method uses the color function which represents the fraction of volume in each cell occupied by the reference phase.[2,15] In each cell the interface is approximated with a segment in two dimensions or a plane in three dimensions and its orientation is obtained with discrete numerical schemes applied to the color function. In this work we present a refinement technique for VOF simulations called multilevel VOF that introduces a fine grid for the interface to capture structure of small dimensions. In order to have minimal memory requirements and low CPU load we implement some techniques that exploit the sparsity of the VOF data.

In multilevel methods particular care must be taken in the projection of the velocity from the coarse to the fine grid.[11] The divergence-free constraint is not satisfied when the projection is computed by simple linear or quadratic interpolation. To this purpose we introduce a novel approach that produces a projected velocity field close to standard linear interpolation and able to preserve the divergence-free constraint.

In the simulation sections the algorithms are tested in two- and three-dimensional domains. The features of the technique are stressed in some kynematic simulations that show mass conservation properties and very detailed interface structures. In the final section we deals with typical three-dimensional advection methods, like Lagrange-Lagrange-Lagrange advection algortihms, which do not have divergence-free velocity field and therefore cannot, in principle, conserve the fluid mass.

2 VOLUME-OF-FLUID METHOD

The Volume-of-Fluid (VOF) method is one of the most popular techniques to study two-phase flows on fixed grids.[15] We consider a domain Ω with boundary Γ. In the single–fluid formulation of the Navier-Stokes equations we introduce the characteristic function

$$\chi(\boldsymbol{x},t) = \int_{\Omega_1(t)} \delta(\boldsymbol{x}' - \boldsymbol{x})\, d\boldsymbol{x}' \quad \forall \boldsymbol{x} \in \Omega\,, \tag{1}$$

where Ω_1 is the portion of Ω occupied by the main phase. From this definition it follows that χ is equal to 1 in the reference phase, zero in the other phase and discontinuous across the interface. Under the hypotheses of immiscible fluids with no phase change, the

function χ behaves like a passive scalar and is simply advected by

$$\frac{\partial \chi}{\partial t} + (\boldsymbol{u} \cdot \nabla)\chi = 0\,, \qquad \text{in } \Omega \times [0,T]\,. \tag{2}$$

The color function C is a discrete function defined as

$$C_i(t) = \frac{1}{V_i}\int_{\Omega_i} \chi(\boldsymbol{x},t)\, d\boldsymbol{x}\,, \tag{3}$$

where V_i is the volume of the i-th computational cell Ω_i. The color function takes a value between zero and one in the cells cut by the interface. In the Piecewise–Linear Interface Calculation (VOF/PLIC) algorithm the interface in each mixed cell is reconstructed by a segment or a portion of a plane, under the volume conservation constraint. The reconstruction provides an approximate characteristic function $\widetilde{\chi}$ which is used to compute the reference phase fluxes across the cell boundary to update the C data.

2.1 Interface reconstruction

We consider widely used reconstruction and advection algorithms in order to focus on multilevel approach and velocity field interpolation. In two-dimensional geometries the ELVIRA reconstruction algorithm[13] calculates local height function values by adding the C data along the columns and rows of a 3×3 block around the central cell. It then derives six normal vector candidates $\widetilde{m}$ by evaluating backward, central and forward differences of the height function. With each candidate the central cell segment is reconstructed and then extended to the whole block of cells defining an approximate $\widetilde{C}$ distribution. The selected candidate minimizes the discrete error E in L_2 between the real data C and the approximate values $\widetilde{C}$

$$E(\widetilde{m}) = \left(\sum_k (\widetilde{C}_k(\widetilde{m}) - C_k)^2\right)^{\frac{1}{2}}\,. \tag{4}$$

This technique reproduces any linear interface exactly.

The ELVIRA algorithm has been extended to three-dimensional Cartesian grids[12] and it requires a $5\times5\times5$ block of cells and 144 normal vector candidates to reconstruct exactly any linear interface. This approach is very expensive when the number of cut cells is large. For this reason we consider a *reduced* version of this reconstruction algorithm, that relies only on a $3\times3\times3$ cell stencil. In this algorithm we compute the local height function and consider only the forward, central and backward finite differences along the three spatial directions. In addition, we test the normal vector with the Parker–Youngs method which is particularly efficient at low resolutions. The normal is selected by evaluating the same error (4) of the two-dimensional case. This approach does not reproduce all linear interfaces exactly, but is a good compromise between the number of candidates and the accuracy of the reconstruction.

2.2 Interface advection

In this work we use some split techniques to propagate separately the interface along each coordinate direction. Unsplit algorithms are less popular because they are generally much more complex and rather cumbersome to implement in an efficient way,[5] in particular in three dimensions. Most split techniques rely on the conservative form of (2) written as

$$\frac{\partial \chi}{\partial t} + \nabla \cdot (\chi \boldsymbol{u}) = \chi \nabla \cdot \boldsymbol{u} \,. \tag{5}$$

Let us integrate over the cell Ω_i the one-dimensional equation along the x direction to get

$$\frac{\partial C_i(t)}{\partial t} + \frac{1}{V_i} \int_{\Gamma_i} \chi(\boldsymbol{x}, t)\, \boldsymbol{u} \cdot \boldsymbol{n}\, dS = C_i \frac{\partial u}{\partial x} \,, \tag{6}$$

where the term $\partial u/\partial x$ can be computed as the cell mean value. We note that it may be different from zero even for divergence-free velocity fields. If we consider two consecutive discrete times t^k and $t^{k+1} = t^k + \Delta t$ and approximate the spatial derivative with central finite differences we get

$$C_i^{k+1} = C_i^k - \Delta \widetilde{\Phi}_i + \widetilde{C}_i \Delta u \,, \tag{7}$$

where $\widetilde{\Phi}_i$ is the normalized flux. The symbol $\Delta\alpha$ denotes the difference between the right and left boundary value of the scalar quantity α.

In the *Eulerian implicit* (E) scheme $\widetilde{C}_i = C_i^{k+1}$ so that (7) becomes

$$C_i^{k+1} = a \left(C_i^k - \Delta \widetilde{\Phi}_i \right) \,, \tag{8}$$

where $a = 1/(1 - \Delta u)$ is the expansion/contraction coefficient of the Eulerian step.

In a similar way, in the *Lagrangian explicit* (L) scheme $\widetilde{C}_i = C_i^k$ so that (7) becomes

$$C_i^{k+1} = b\, C_i^k - \Delta \widetilde{\Phi}_i \,, \tag{9}$$

where $b = (1 + \Delta u)$ is the corresponding expansion/contraction coefficient of the Lagrangian step.

In two dimensions the two previous steps can be combined in an area-preserving algorithm. This method is called *geometrical unsplit* since it can be seen as a linear mapping between two different tessellations of the plane. The method combines an Eulerian step in the x direction followed by a Lagrangian step in the y direction to form the linear transformation[2]

$$\Pi_{xy} = \begin{cases} x' = a(x + u_i) \\ y' = by + v_i \end{cases} \,, \tag{10}$$

where $a = 1/(1 - \Delta u)$ and $b = 1 + \Delta v$. The discrete version of the area conservation constraint can be written as $\Delta u + \Delta v = 0$, then $a\, b = 1$. This corresponds to the area

preserving property that does not generate undershoots or overshoots. By alternating the implicit and explicit steps no coordinate direction is preferred.

For three-dimensional simulations we need to look for an area–preserving algorithm by combining three steps of the form (7). The constraint can now be formally written as $abc = 1$, where a, b and c are the expansion/contraction coefficients of each single split step. We use the Eulerian implicit step along the x direction and the Lagrangian explicit step along the z direction as the first and last step of the three-dimensional algorithm. Then, we define a new intermediate step along the y direction, that we call the *Modified Eulerian* (M) step, given by

$$b = \frac{1}{ac} = \frac{1-\Delta u}{1+\Delta w} = 1 + \frac{\Delta v}{1+\Delta w}\,. \tag{11}$$

This is an algebraic definition that satisfies the three-dimensional discrete divergence-free constraint $\Delta u+\Delta v+\Delta w = 0$, but the Modified Eulerian scheme cannot ensure $0 \le C \le 1$. As a matter of fact, this step may introduce small inconsistencies that are seen as local overshoots or undershoots. The magnitude of these errors decreases as the time step is reduced.

2.3 Multilevel approach

The multilevel technique has been proposed to reduce the weaknesses of front-capturing algorithms, such as VOF and Level Set.[7,8,10] If the interface structure characteristic length is comparable to the grid spacing, artificial break-up or coalescence can occur. To this purpose we introduce a separate mesh for interface advection that is derived from the velocity and pressure grid with a mid-point refinement algorithm. The coarse grid quantities and the fine ones are identified with the superscript c and f, respectively.

In this work we present some kinematic simulations and focus on the advection equation of the color function. The divergence-free constraint

$$\int_\Omega q^c \nabla\cdot\boldsymbol{u}^c dV = 0\,, \qquad\qquad \int_\Omega q^f \nabla\cdot\boldsymbol{u}^f dV = 0\,, \tag{12}$$

is satisfied on both grids. If the coarse velocity field $\boldsymbol{u}^c$ is substituted at the fine level we obtain

$$\int_\Omega q^f \nabla\cdot\boldsymbol{u}^c dV = \int_\Omega q^f\, R^{fc}(\boldsymbol{u}^c,\boldsymbol{u}^f)\, dV\,, \tag{13}$$

where R^{fc} is the fine-to-coarse mass transfer operator defined by

$$R^{fc}(\boldsymbol{u}^f,\boldsymbol{u}^c) = \nabla\cdot(\boldsymbol{u}^f - \boldsymbol{u}^c)\,. \tag{14}$$

This operator quantifies the mass conservation error when the divergence-free constraint is not satisfied on the fine grid. Since we do not solve the Navier-Stokes equations on the fine grid, we must rely on a projection operator to get the fine velocity field from the coarse one. We remark that the operator R^{fc} is equal to zero only when the projection operator preserves the divergence-free constraint.

2.3.1 Numerical implementation

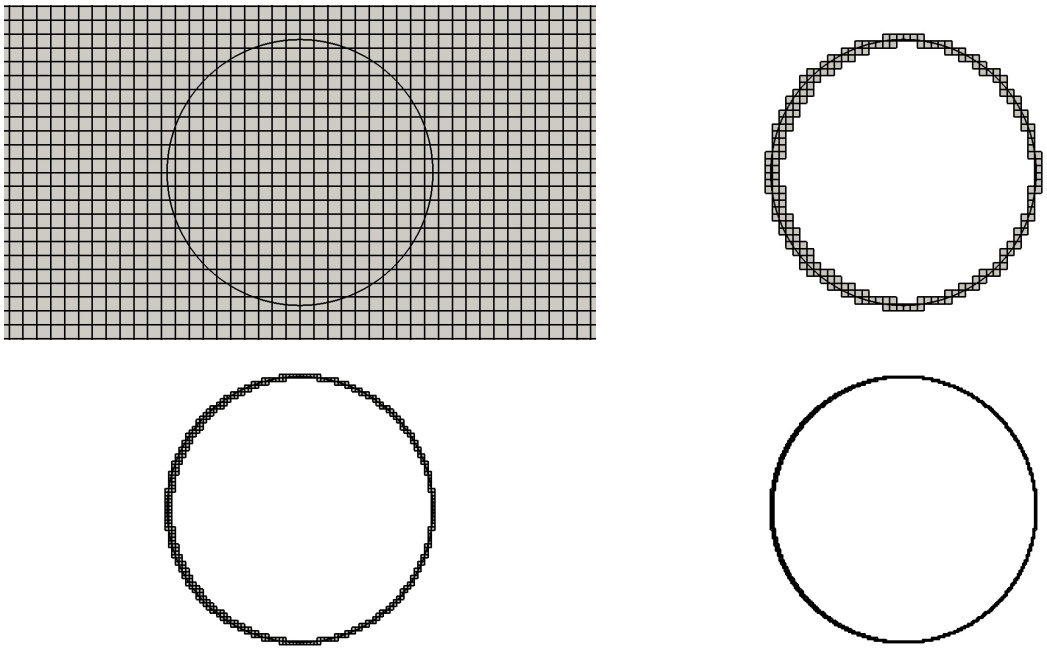
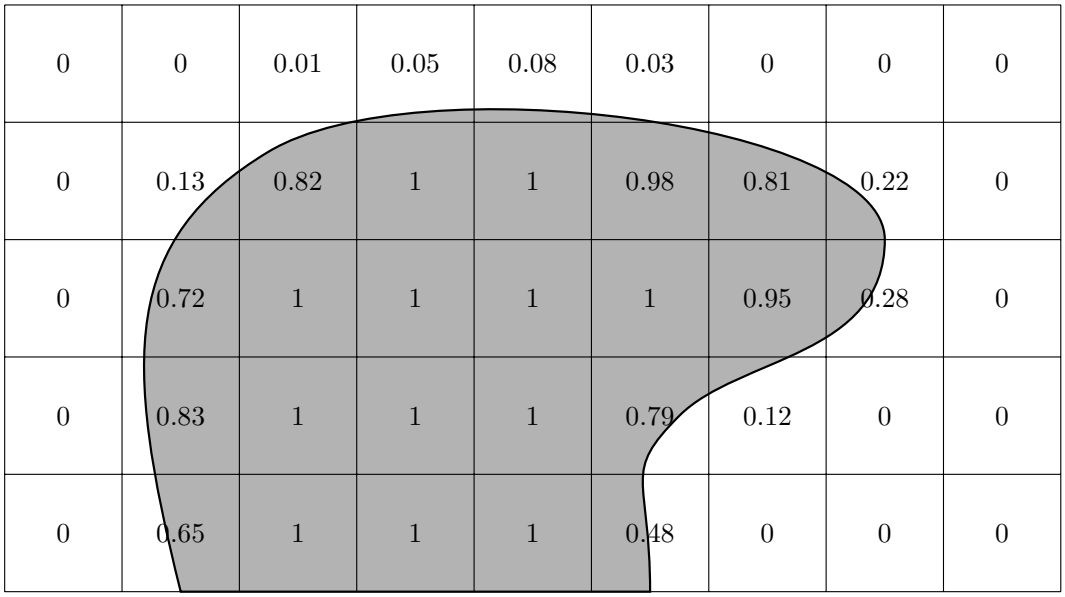


Figure 1: Full memorization at the coarse level on a grid with 24×42 square cells, sparse memorization with one refinement level, $f = c + 1$ and 48×84 cells, and then with two and three refinement levels (left to right and top to bottom).

The introduction of a finer grid can lead to large memory requirements and CPU time due to the huge number of fine cells. To avoid performance decreases of the overall numerical model we compress the C data and perform the reconstruction/advection algorithm simultaneously over grid subregions. An example of the sparse memorization is given in Fig. 1 where we show on the top left the stored cells for a circular interface on a coarse grid and the stored cells at different refinement levels. The total number of cells increases by a factor of four in 2D (eight in 3D) while the number of cells in the sparse approach increases approximately only by a factor of two (four) since only the cut cells are stored.

The C data storage is similar to the Compressed Row Storage algorithm (CRS).[3] We show a two-dimensional example in Fig. 2 where for each row we store only the number of entries n_c, the C data and their column numbers. The empty cells are discarded and a sequence of n consecutive full cells is stored as a single value equal to n. For example in the third row of Fig.2, we memorize in the second column the number 4 to represent the sequence of four consecutive full cells. With this technique we can use many refinement levels and keep the storage requirements proportional to the length of the interface divided by the fine grid spacing. In order to use the sparse matrix storage technique we need to

implement an efficient numerical algorithm to extract and compress the data.



row	n_c	C						column					
1	4	0.01	0.05	0.08	0.03			3	4	5	6		
2	6	0.13	0.82	2	0.98	0.81	0.22	2	3	4	6	7	8
3	4	0.72	4	0.95	0.28			2	3	7	8		
4	4	0.83	3	0.79	0.12			2	3	6	7		
5	3	0.65	3	0.48				2	3	6			

Figure 2: A color function distribution on a 9×5 Cartesian mesh (top) and the corresponding stored data (bottom), with row number, number of cells n_c, color function in the mixed and consecutive full cells, and column position.

In the reconstruction and advection algorithms we do not perform the VOF calculations on a single block of 3×3 cells but use three complete rows of n cells along the horizontal coordinate. Then, we compute all the normal vectors and the fluxes and store them again in the compact form. The normal vectors can be stored in the same way as C data. The three-dimensional extension of this approach is straightforward and shows similar features. The $3 \times n$ cell block is replaced by a $3 \times 3 \times n$ cell block.

3 OPTIMAL APPROACH FOR VELOCITY REFINEMENT

We consider a velocity field that satisfies the divergence-free condition on a given coarse grid. If we introduce a refined grid and compute the velocity field on the additional nodes as a weighted average of the values on the coarse grid, then the refined velocity field does not satisfy the divergence-free constraint. To satisfy this we introduce an optimal

constrained approach where the objective functional is the sum of the squared difference between the velocity and the linear interpolated values. The minimization is solved by using the Lagrange multiplier method.

3.1 Two-dimensional projection

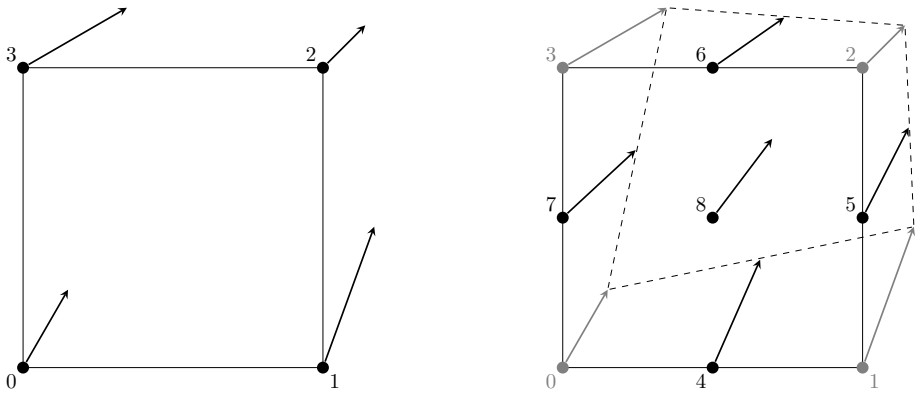


Figure 3: The coarse element with four nodes (left) and the refined one with nine nodes (right).

We consider the case shown in Fig. 3, where we calculate the velocity vector at the nodes 4, 5, 6, 7 and 8 starting from the known values at 0, 1, 2 and 3. The refined velocity field must satisfy the following discrete divergence-free condition in each sub-cell

$$D_0 = u_4 + u_8 - u_7 - u_0 + v_8 + v_7 - v_4 - v_0 = 0 \tag{15a}$$
$$D_1 = u_1 + u_5 - u_8 - u_4 + v_5 + v_8 - v_1 - v_4 = 0 \tag{15b}$$
$$D_2 = u_8 + u_6 - u_3 - u_7 + v_6 + v_3 - v_8 - v_7 = 0 \tag{15c}$$
$$D_3 = u_5 + u_2 - u_8 - u_6 + v_2 + v_6 - v_5 - v_8 = 0\,. \tag{15d}$$

In the linear system (15) there are four equations and ten unknowns, which are the velocity components of the additional nodes. This system is clearly underdetermined so we can set the value of some unknowns to their weighted average. In particular we set

$$\begin{aligned} v_4 &= (v_0 + v_1)/2\,, & u_5 &= (u_1 + u_2)/2\,, \\ v_6 &= (v_2 + v_3)/2\,, & u_7 &= (u_3 + u_0)/2\,, \\ u_8 &= (u_0 + u_1 + u_2 + u_3)/4\,, & v_8 &= (v_0 + v_1 + v_2 + v_3)/4\,, \end{aligned} \tag{16}$$

and solve for u_4, v_5, u_6 and v_7.

We introduce the functional

$$J = \frac{1}{2}(u_4 - \tilde{u}_4)^2 + \frac{1}{2}(v_5 - \tilde{v}_5)^2 + \frac{1}{2}(u_6 - \tilde{u}_6)^2 + \frac{1}{2}(v_7 - \tilde{v}_7)^2\,, \tag{17}$$

where the target values are

$$\tilde{u}_4 = (u_0 + u_1)/2\,, \qquad \tilde{v}_5 = (v_1 + v_2)/2\,, \tag{18a}$$

$$\tilde{u}_6 = (u_2 + u_3)/2\,, \qquad \tilde{v}_7 = (v_3 + v_0)/2\,. \tag{18b}$$

The augmented Lagrangian functional P, given by the sum of J and (15), becomes

$$P = J + \sum_{i=0}^{3} \gamma_i D_i\,, \tag{19}$$

where γ_i are the Lagrange multipliers associated to the discrete divergence D_i. We can now determine the minimum of P by setting

$$\begin{aligned}\delta P = (u_4 - \tilde{u}_4)\delta u_4 + (v_5 - \tilde{v}_5)\delta v_5 + (u_6 - \tilde{u}_6)\delta u_6 + (v_7 - \tilde{v}_7)\delta v_7 + \gamma_0(\delta u_4 + \delta v_7) + \\ + \gamma_1(-\delta u_4 + \delta v_5) + \gamma_2(\delta u_6 - \delta v_7) + \gamma_3(-\delta u_6 - \delta v_5) + \sum_{i=0}^{3} \delta\gamma_i D_i = 0\,. \end{aligned} \tag{20}$$

Since all variations in (20) are independent, we can set each of them to zero and get a linear system of eight equations in u_4, v_5, u_6, v_7 and the four Lagrangian multipliers. The system does not have full rank, since the (15) are not linearly independent. Therefore, we must enforce an additional condition and impose that each sub-cell divergence equals to one quarter of the coarse cell value. In all kinematic tests of the next section this value is zero. The analytical solution for the velocity components is

$$u_4 = \frac{2u_0 + 2u_1 + v_0 - v_1 + v_2 - v_3}{4}\,, \tag{21a}$$

$$v_5 = \frac{u_0 - u_1 + u_2 - u_3 + 2v_1 + 2v_2}{4}\,, \tag{21b}$$

$$u_6 = \frac{2u_2 + 2u_3 + v_0 - v_1 + v_2 - v_3}{4}\,, \tag{21c}$$

$$v_7 = \frac{u_0 - u_1 + u_2 - u_3 + 2v_3 + 2v_0}{4}\,. \tag{21d}$$

We remark that these optimized values depend on both components of the coarse velocity field.

3.2 Three-dimensional projection

The three-dimensional projection algorithm is a straightforward extension of the two-dimensional case. We consider the case of a hexahedron with eight subcells and eight discrete divergence-free conditions. The number of unknowns is now 57. In this case we can set the velocity in the center of each edge, in the center of the hexahedron and also the velocity component in the center of each face that is perpendicular to the face itself.

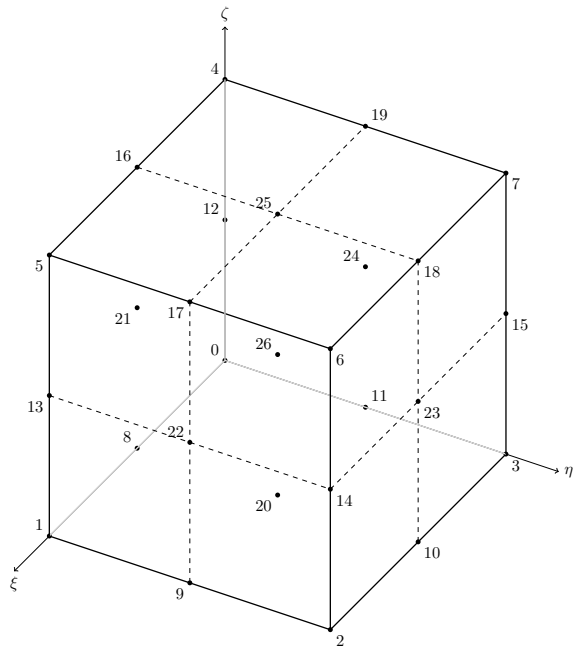


Figure 4: The refined right hexahedron in three dimensions.

There are 12 unknowns left that are u_{20}, v_{20}, u_{21}, w_{21}, v_{22}, w_{22}, u_{23}, w_{23}, v_{24}, w_{24}, u_{25}, v_{25} of Fig. 4. At this point, we define the functional J and the augmented Lagrangian functional P as in the two-dimensional case. The linear system has now 20 unknowns and must be solved with the additional condition that each subcell has one eighth of the coarse divergence value. The analytical solution for the first two unknowns is

$$\begin{aligned} u_{20} = (&4u_0 + 4u_1 + 4u_2 + 4u_3 + 3v_0 - 3v_1 + 3v_2 - 3v_3 + v_4 - v_5 + v_6 - v_7 + \\ &+ 2w_0 - 2w_1 - 2w_2 + 2w_3 - 2w_4 + 2w_5 + 2w_6 - 2w_7)/16\,, \\ v_{20} = (&3u_0 - 3u_1 + 3u_2 - 3u_3 + u_4 - u_5 + u_6 - u_7 + 4v_0 + 4v_1 + 4v_2 + 4v_3 + \\ &+ 2w_0 + 2w_1 - 2w_2 - 2w_3 - 2w_4 - 2w_5 + 2w_6 + 2w_7)/16\,. \end{aligned}$$

The other can be obtained with simple index permutations. We remark that also in this case each optimized velocity depends on all three velocity components.

4 NUMERICAL TESTS

In the next subsections we compare the performance of the optimized velocity projection algorithm and the simple linear interpolation. The velocity field is modulated in time with a cosinusoidal function so that at the end of the simulation period T the interface

should be exactly on top of the initial configuration $\widehat{C}$. We define the *mass error*

$$E_m = \frac{|\sum_j C_j - \sum_j \widehat{C}_j|}{\sum_j \widehat{C}_j}\,, \tag{23}$$

where the sum spans over all the cells of the computational domain. We also define the *geometrical error*

$$E_g = \sum_j |C_j - \widehat{C}_j|\,, \tag{24}$$

and the *relative geometrical error*

$$E_r = \frac{\sum_j |C_j - \widehat{C}_j|}{\sum_j \widehat{C}_j}\,. \tag{25}$$

The mass conservation is affected only by the velocity interpolation method and the advection algorithm. The other errors may depend on the reconstruction algorithm as well.

4.1 Two-dimensional tests

In these tests we use the ELVIRA reconstruction method and advect the interface with the geometrical unsplit algorithm with different refinement levels. The *single–vortex*

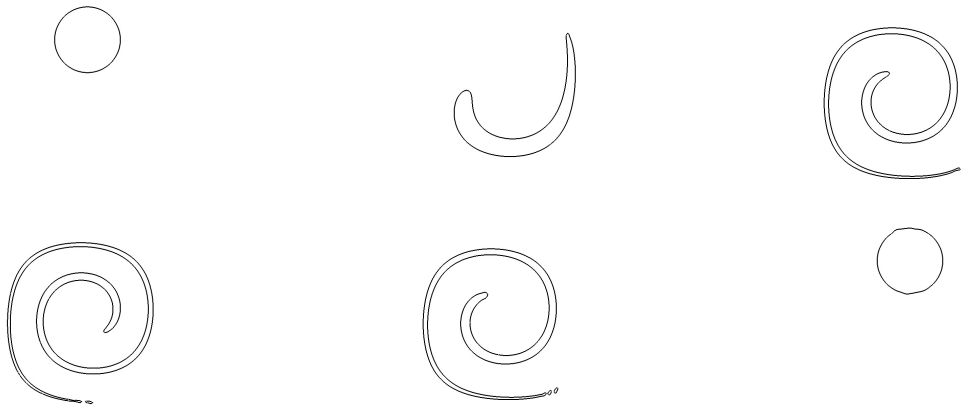


Figure 5: Single–vortex test with period $T = 8$, $\max(CFL) = 1$, coarse grid with 32×32 cells and three refinement levels. The interface is shown at times $t = 0$, $T/8$, $T/4$, $3T/8$, $T/2$, T (left to right, top to bottom).

test is a popular kinematic test that stretches and deforms the interface to a large extent and shows the excellent mass conservation property of the multilevel VOF method when

Figure 6: Single–vortex test of Fig. 5, but with two refinement levels. The mixed cells that are stored in the compact memorization matrix are shown at times $t = T/8$, $T/4$ (left to right).

	$f = c + 2$		$f = c + 3$		$f = c + 4$	
	opt	lin	opt	lin	opt	lin
E_m	3.926e-16	1.547e-03	5.889e-16	3.180e-06	1.177e-15	5.040e-07
E_r	2.033e-02	7.558e-03	4.234e-03	4.216e-03	1.090e-03	1.051e-03
E_g	1.437e-03	1.044e-03	2.969e-04	2.957e-04	7.705e-05	7.436e-05

Table 1: Mass E_m, relative geometrical E_r and geometrical E_g errors for the optimal (opt) and linear (lin) interpolations, for the single–vortex test of Fig. 5 and three different refinement levels, $f = c + 2$, $f = c + 3$ and $f = c + 4$.

coupled to the optimized velocity projection. The velocity field is derived from the stream function

$$\psi(x, y, t) \;=\; \frac{\cos(\pi t/T)}{\pi} \sin^2(\pi\, x)\, \sin^2(\pi\, y)\,, \tag{26}$$

with $u = -\partial\psi/\partial y$ and $v = \partial\psi/\partial x$. The partial derivatives are approximated by finite differences, so that the discrete divergence-free condition is exactly satisfied. The circle at $t = 0$ is centered at $(0.5, 0.75)$ with radius $R = 0.15$, and the computational grid is the unit square $[0, 1] \times [0, 1]$.

In Figs. 5 and 6 we can see the interface evolution of the vortex test with $T = 8$ over a coarse grid with 32×32 with different fine resolutions. Fig. 5 shows the interface reconstruction at different times with a fine grid of 256×256 cells while Fig. 6 shows the storage pattern with a 128×128 cell domain. Tab. 1 summarizes the errors obtained with the optimal and linear approaches for different refinement levels. The optimal control approach for the velocity refinement satisfies accurately the mass conservation while the linear interpolation does not. The values shown in the table are computed at the end of the simulation but mass error remains of the same order at each time step. The geometrical error is largely dependent on the reconstruction method. Even if the geometrical errors are close, the results with linear interpolation are lower. In fact, in the optimal interpolation

each refined velocity depends on both components on the coarse grid inducing a small distortion of the refined velocity field.

	$CFL = 1$		$CFL = 0.1$	
	opt	lin	opt	lin
E_m	5.889e-16	3.180e-06	9.620e-15	3.159e-06
E_r	4.234e-03	4.216e-03	6.277e-03	6.339e-03
E_g	2.969e-04	2.957e-04	4.437e-04	4.480e-04

Table 2: Mass E_m, relative geometrical E_r and geometrical E_g errors for the optimal (opt) and linear (lin) interpolations, for the single–vortex test of Fig. 5 with three refinement levels, $f = c + 3$, and different values of the CFL number.

In Tab. 2 we compare the result for the case $f = c + 3$ and a variable CFL number. Again, the mass conservation is accurately satisfied only by the optimized refinement projection. The geometrical errors increase since when $CFL = 0.1$ we compute 10 times the reconstructions of the case $CFL = 1$. The geometrical error in both cases is similar, however, when we increase the number of time steps by decreasing the CFL number, we observe that the optimal approach is more performing.

Finally we consider a different test and show the performance of the multilevel VOF approach with several grid refinements. The velocity field is again derived from the stream function

$$\psi(x, y, t) \;=\; \frac{\cos(\pi t/T)}{4\pi} \sin(4\pi\,(x + 0.5))\,\cos(4\pi\,(y + 0.5))\,. \tag{27}$$

This test is usually called *four vortex test*. The circle is now centered at $(0.5, 0.5)$ with radius $R = 0.15$ and the computational domain is again the unit square. The simulation is shown in Fig. 7 with six levels of grid refinement, corresponding to a mesh of 2048×2048 cells. With this resolution our multilevel approach is very close to a front tracking algorithm. The mass is also conserved very accurately throughout the simulation, i.e. $E_M \simeq 10^{-16}$.

4.2 Three-dimensional tests

We have selected a few three-dimensional tests to show how the optimal projection approach performs when it is coupled to different split advection schemes. In the simulations we always consider the 3D reduced ELVIRA reconstruction algorithm. In the first test we consider a two-dimensional velocity field in a three-dimensional domain. We introduce a vector stream function $\boldsymbol{\psi}(x, y, z, t)$ which is defined from the scalar stream function $\psi(x, y, t)$ of (26) as

$$\boldsymbol{\psi} = (0, 0, \psi(x, y, t))\,, \quad \boldsymbol{u} = \nabla \times \boldsymbol{\psi}\,. \tag{28}$$

In the simulation we use two different split advection techniques. The first one, denoted by EML, is a sequence of an Eulerian implicit, a Modified Eulerian and a Lagrangian explicit

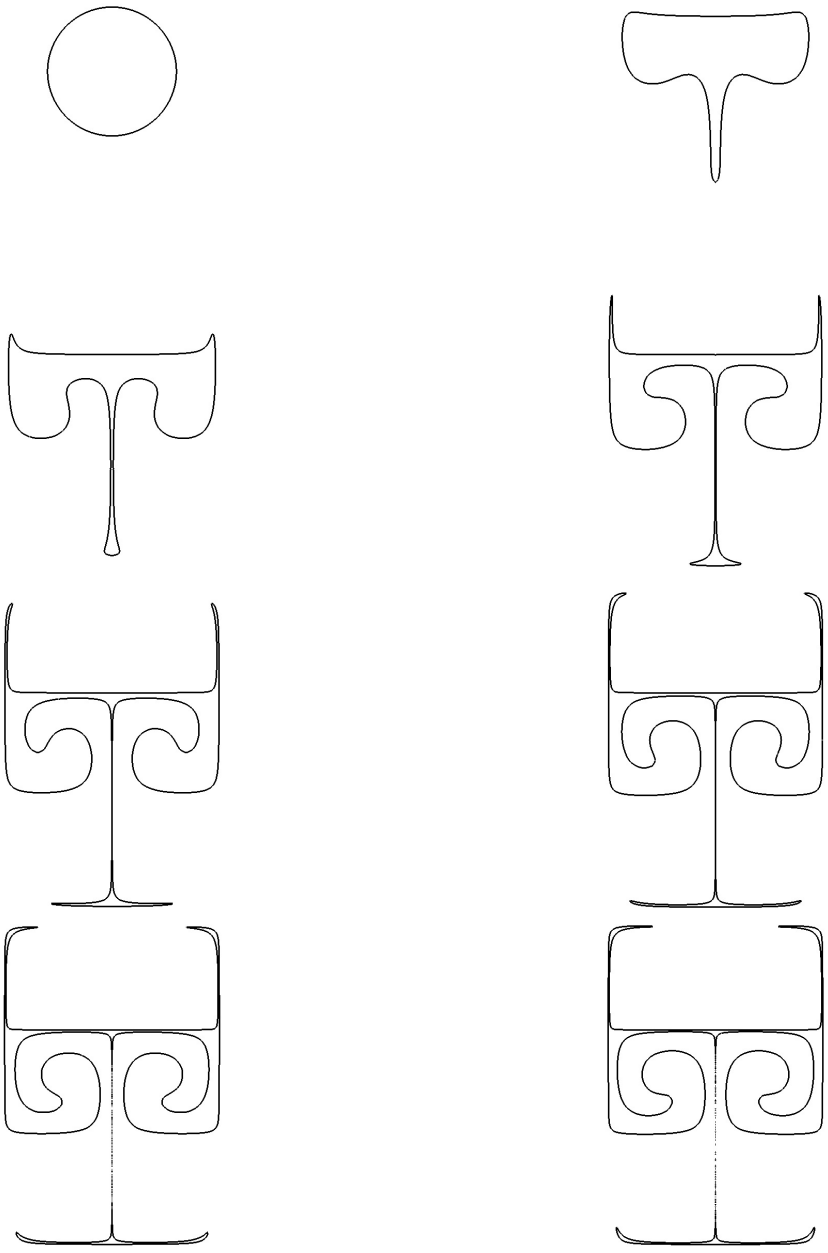


Figure 7: Four vortex test with period $T = 2$. The interface is shown at times $t = 0$, 1/8, 1/4, 3/8, 1/2, 5/8, 3/4, 7/8, 1.

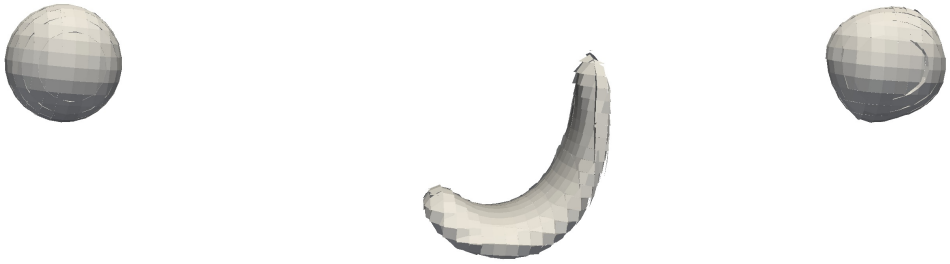


Figure 8: 3D single–vortex test with period $T = 2$, coarse grid with $16 \times 16 \times 16$ cells and a single refinement level. The interface is shown at times $t = 0$, $T/2$, T.

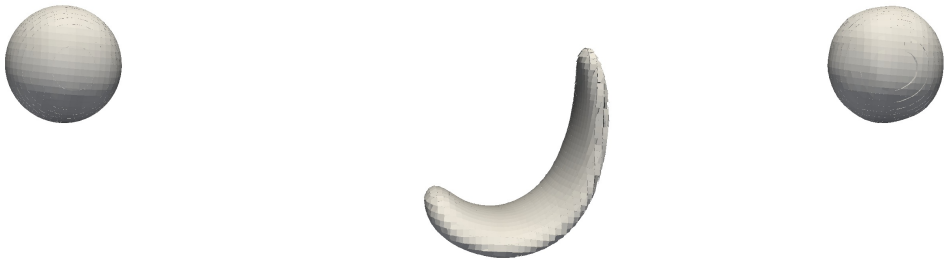


Figure 9: 3D single–vortex test with period $T = 2$, coarse grid with $16 \times 16 \times 16$ cells and two refinement levels. The interface is shown at times $t = 0$, $T/2$, T.

step. The second one is given by three consecutive Lagrangian explicit steps (LLL). Since the coarse velocity field is two-dimensional, we put the Modified Eulerian step along the third direction. This velocity component is exactly zero for the linear interpolation and very small for the optimal projection.

The interface at initial time, half period and final time is shown in Figs. 8, 9 and 10 for a different number of refinement levels. The corresponding errors are given in Tab. 3 for the two different interpolation methods and with the EML split advection. The results are in agreement with those of the two-dimensional case. The mass conservation is very accurate only with the optimal velocity refinement, while the geometrical errors do not change significantly. In Tab. 4 we consider two levels of grid refinement and change the CFL number. Again, the mass conservation is very accurate only with the optimal algorithm combined with the EML split advection. The sequence of three Lagrangian steps does not conserve the mass in spite of the chosen interpolation method. The geometrical errors do not change significantly with the interpolation method and the advection scheme.

For the fully three-dimensional velocity field we define a new scalar and a new vector

	$f = c + 2$		$f = c + 3$		$f = c + 4$	
	opt	lin	opt	lin	opt	lin
E_m	2.233e-14	4.029e-07	3.681e-15	2.824359e-06	4.540e-15	6.540e-06
E_r	8.257e-02	8.207e-02	2.210e-02	2.186416e-02	7.289e-03	7.187e-03
E_g	1.167e-03	1.160e-03	3.121e-04	3.090973e-04	1.030e-04	1.016e-04

Table 3: Mass E_m, relative geometrical E_r and geometrical E_g errors for the optimal (opt) and linear (lin) interpolations, for the 3D single–vortex test with period $T = 2$, $\max(CFL) = 0.1$, EML advection, coarse grid with 16 cells and three different refinement levels, $f = c + 1$, $f = c + 2$ and $f = c + 3$.

$CFL = 0.1$

	EML		LLL	
	opt	lin	opt	lin
E_m	2.233e-14	4.029e-07	7.597e-03	7.581e-03
E_r	8.257e-02	8.207e-02	8.942e-02	8.888e-02
E_g	1.167e-03	1.160e-03	1.264e-03	1.256e-03

$CFL = 0.01$

	EML		LLL	
	opt	lin	opt	lin
E_m	1.043e-13	1.649e-05	7.796e-04	7.581e-03
E_r	8.507e-02	8.456e-02	8.489e-02	8.888e-02
E_g	1.202e-03	1.195e-03	1.200e-03	1.256e-03

$CFL = 0.001$

	EML		LLL	
	opt	lin	opt	lin
E_m	1.645e-14	1.831e-05	7.815e-05	8.067e-05
E_r	8.559e-02	8.506e-02	8.474e-02	8.409e-02
E_g	1.210e-03	1.202e-03	1.198e-03	1.188e-03

Table 4: Mass E_m, relative geometrical E_r and geometrical E_g errors for the optimal (opt) and linear (lin) interpolations, for the 3D single–vortex test with period $T = 2$, EML and LLL advections, coarse grid with $16 \times 16 \times 16$ cells, two refinement levels, $f = c + 2$, and different values of the CFL number.

Figure 10: 3D single–vortex test with period $T = 2$, coarse grid with $16 \times 16 \times 16$ cells and three refinement levels. The interface is shown at times $t = 0$, $T/2$, T.

stream function

$$\psi_1(x, y, z, t) = \frac{\cos(\pi t/T)}{\pi} \sin^2(\pi\, x)\, \sin^2(\pi\, y)\, \sin^2(\pi\, z)\,, \tag{29}$$

$$\boldsymbol{\psi}(x, y, z, t) = (\psi_1, \psi_1, \psi_1)\ ; \quad \boldsymbol{u} = \nabla \times \boldsymbol{\psi}\,. \tag{30}$$

The discrete velocity field is divergence-free when the curl is calculated with central finite differences.

In Fig. 11 we show the evolution of the interface with the optimal velocity interpolation. In this case we observe that also the optimal approach does not preserve exactly the mass because the modified Eulerian step can generate small overshoots and undershoots, as we have pointed out previously. The mass error is $3.451e-6$ and remains several orders of magnitude below the geometrical error. The mass error for the linear interpolation is only a couple of orders of magnitude larger and does not lead to significant differences. The situation changes for dynamical simulations where the divergence-free constraint is very important to compute a consistent velocity field around the interface.[1]

5 CONCLUSIONS

The focus of this paper is a novel approach to project the velocity field from a coarse field, where the velocity and pressure fields are computed, to a fine grid where the interface is advected and its geometrical properties are calculated. There are several reasons to investigate the feasibility of a multilevel approach where the interface is evolved on a grid with a higher resolution than the one used for the dynamical variables. We recall that interface details smaller than a few grid spacings, such as high curvature regions and thin filaments, are poorly resolved and usually lead to numerical brakup or coalescence of the interface. With this technique these problems are not removed but they develop on a smaller spatial scale. Furthermore, as we increase the number of refinement levels the interface representation becomes smoother and smoother. We have shown that the optimal projection is feasible in both two and three dimensions and that the divergence-free constraint is satisfied very accurately. When the optimal velocity interpolation is coupled to an advection scheme that does generate numerical undershoots or overshoots

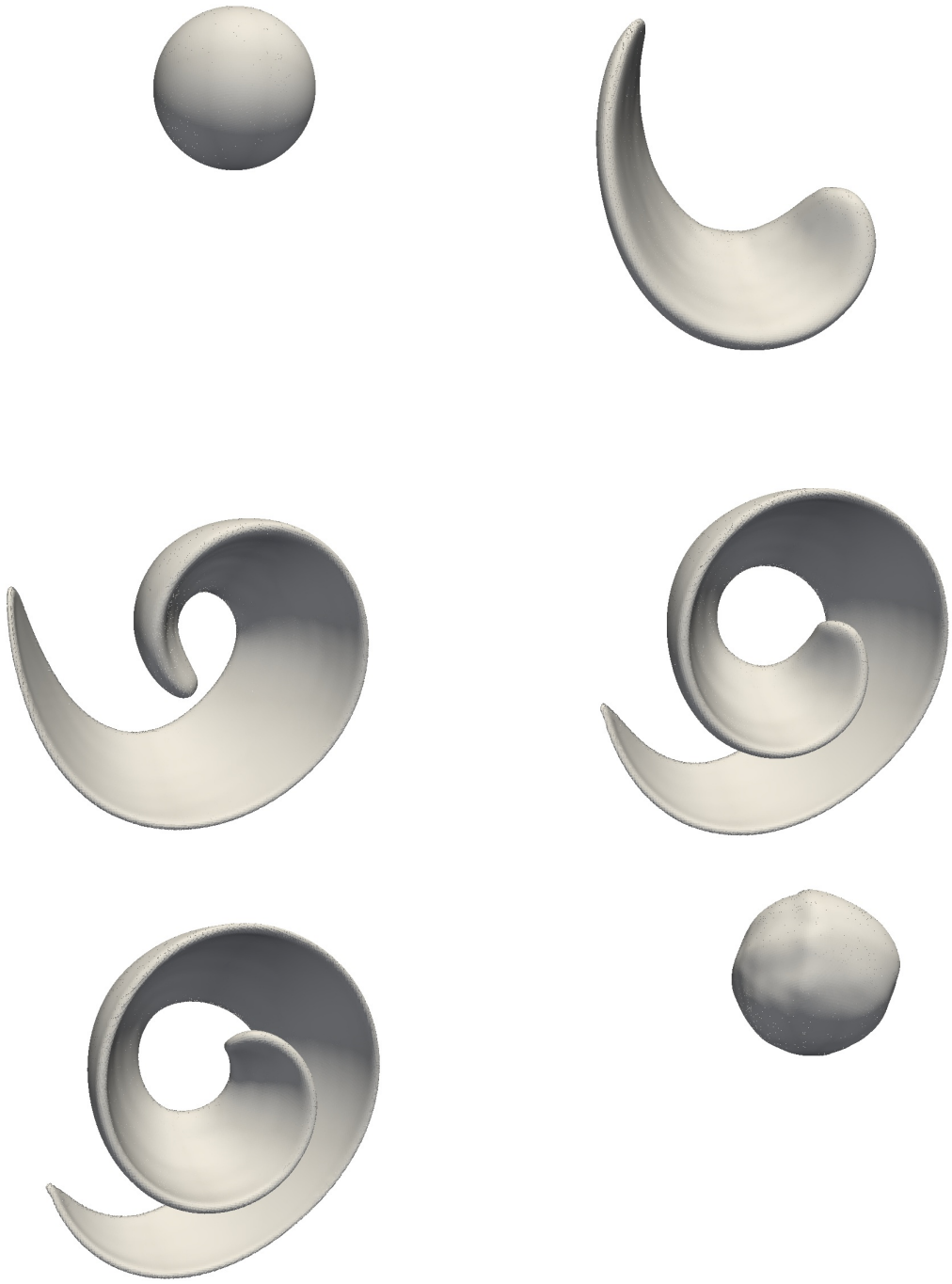


Figure 11: Fully-3D single–vortex test with period $T = 4$, coarse grid with $16 \times 16 \times 16$ cells and five refinement levels. The interface is shown at times $t = 0$, $T/8$, $T/4$, $3T/8$, $T/2$, T (left to right, top to bottom).

the divergence value is around machine precision. Furthermore, we have found that the geometrical error is comparable to the linear interpolation, even if the optimal approach slightly deforms the refined velocity field. The optimal velocity interpolation can also be applied to other front-capturing algorithms, such as the Level Set method.

REFERENCES

[1] E. Aulisa, S. Manservisi, and R. Scardovelli. A novel representation of the surface tension force for two-phase flows with reduced spurious currents. *Comput. Method. Appl. M.*, 195:6239–6257, 2006.

[2] E. Aulisa, S. Manservisi, R. Scardovelli, and S. Zaleski. A geometrical area-preserving volume-of-fluid advection method. *J. Comput. Phys.*, 192:355–364, 2003.

[3] R. Barrett and C. Romine. *Templates for the solution of linear systems: building blocks for iterative methods*. Society for Industrial Mathematics, 1994.

[4] A. Cervone, S. Manservisi, and R. Scardovelli. Simulation of jets with a finite element Navier–Stokes solver and a multilevel VOF approach. In *Proceedings of 11th Triennial International Annual Conference on Liquid Atomization and Spray Systems*, Vail, CO, USA, 2009.

[5] A. Cervone, S. Manservisi, R. Scardovelli, and S. Zaleski. A geometrical predictor–corrector advection scheme and its application to the volume fraction function. *J. of Comput. Phys.*, 228(2):406–419, 2009.

[6] D. Fuster, A. Bague, T. Boeck, L. Le Moyne, A. Leboissetier, S. Popinet, P. Ray, S. Scardovelli, and S. Zaleski. Simulation of primary atomization with an octree adaptive mesh refinement and VOF method. *Int. J. Multiphas. Flow*, 35(6):550–565, 2009.

[7] M. Gorokhovski and M. Herrmann. Modeling primary atomization. *Annual Reviews of Fluid Mechanics*, 40:343–366, 2008.

[8] M. Herrmann. A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids. *J. Comput. Phys.*, 227(4):2674–2706, 2008.

[9] L.H. Howell and J.B. Bell. An adaptive mesh projection method for viscous incompressible flow. *SIAM J. Sci. Comput.*, 18(4):996–1013, 1997.

[10] H. Kohno and T. Tanahashi. Numerical analysis of moving interfaces using a level set method coupled with adaptive mesh refinement. *Int. J. Numer. Meth. Fl.*, 45(9):921–944, 2004.

[11] S. Li and H. Li. A novel approach of divergence-free reconstruction for adaptive mesh refinement. *J. Comput. Phys.*, 199(1):1–15, 2004.

[12] G. H. Miller and P. Colella. A conservative three-dimensional Eulerian method for coupled solid-fluid shock capturing. *J. Comput. Phys.*, 183(1):26–82, 2002.

[13] J.E. Pilliod and E.G. Puckett. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *J. Comput. Phys.*, 199(2):465–502, 2004.

[14] S. Popinet. An accurate adaptive solver for surface-tension-driven interfacial flows. *J. Comput. Phys.*, 228(16):5838–5866, 2009.

[15] R. Scardovelli and S. Zaleski. Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics*, 31(1):567–603, 1999.