

SIMPLIFIED VOLUME OF FLUID METHOD (SVOF) FOR TWO-PHASE FLOWS

MACIEJ MAREK, WOJCIECH ANISZEWSKI
AND ANDRZEJ BOGUSŁAWSKI

*Institute of Thermal Machinery,
Czestochowa University of Technology,
Armii Krajowej 21, 42-200 Czestochowa, Poland
aniszewski@imc.pcz.czyst.pl*

(Received 7 May 2008)

Abstract: The PLIC approach has been usually used in recent implementations of the VOF method *i.e.* the interphasal surface is approximated by a plane with an arbitrary orientation with respect to the computational cell. Although this method is accurate, it is rather difficult to implement, as a large number of orientations need to be taken into account and the calculation of volume fraction fluxes is not straightforward. A simpler approach to VOF – SLIC – requires much less effort from the programmer but the interface approximation by a plane parallel to the cell surfaces is too crude and the results are not satisfactory. The method presented in the article may be considered as an intermediate approach between PLIC and SLIC – fluxes are computed directly only for the interface’s special orientations and linear interpolation is used for calculation of the fluxes for the remaining cases. Some classical tests of the proposed method are performed and an example of a broken dam problem simulation is presented.

Keywords: two-phase flow modeling, volume of fluid, eulerian simulation

1. Introduction

The modeling of two-phase flows is not an easy task, both from a physical and numerical point of view. The complexity of the phenomenon arises from the presence of an interphase surface (front, interface) on which physical properties change discontinuously (*e.g.* density, viscosity, pressure). This surface may be considered as a moving boundary, where appropriate boundary conditions must be imposed and an evolution of which needs to be found as a part of the solution. In the case of immiscible non-reacting fluids, the interface is simply advected with the velocity of the flow.

Many methods for tracking the interphase surface can be found in the literature. The most popular of those are: the front tracking method [1] (interface modeled as a set of connected markers), the Level Set method [2] (interface captured implicitly as the zero level set of a signed distance function) and the Volume of Fluid method (VOF). We will deal with the latter in more detail in this paper.

The Volume of Fluid (VOF) method is one of the best established interface tracking methods currently in use [3]. A large number of applications have been reported for incompressible (see for example [4]), as well as for compressible flows [5]. Originating in the works of Hirth and Nichols [6], it is based on the idea of a volume fraction function C , which is an integral of a characteristic function $\chi(x,y)$ of the tracked phase over a computational cell. On a regular mesh with rectangular cells Δx and Δy , in size, this becomes (a two-dimensional case is assumed for the convenience of presentation):

$$C_{ij} = \frac{1}{\Delta x \Delta y} \int_0^{\Delta x} \int_0^{\Delta y} \chi(x,y) dx dy. \quad (1)$$

This function, as described in [6, 7] gives the volume fraction of the tracked phase in the computational cell, *i.e.* it is equal to 1 when the cell is full, vanishes if the cell is empty and has an intermediate value when the cell contains the interface. Since χ is passively advected with the flow, we can expect its material derivative to vanish:

$$\frac{\partial \chi}{\partial t} + \mathbf{V} \cdot \nabla \chi = 0, \quad (2)$$

where \mathbf{V} is a velocity vector with components $\mathbf{V} = [u, v]$ in \mathbb{R}^2 and $[u, v, w]$ in \mathbb{R}^3 . This, along with the assumption of a zero divergence for the velocity field (in an incompressible flow):

$$\nabla \cdot \mathbf{V} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \quad (3)$$

enables us to write

$$\frac{\partial \chi}{\partial t} + \frac{\partial(\chi u)}{\partial x} + \frac{\partial(\chi v)}{\partial y} = 0. \quad (4)$$

Equation (2) is the same for all other passively advected scalars – $\phi(\mathbf{x})$ can serve as an example, a distance function used in Level-Set methods [2]. However, the conservative form (4) is characteristic for the VOF method, and so are the means of dealing with this equation, taking also into account a discontinuous nature of function χ .

Let us denote the value of C in n -th timestep for the (i,j) -th cell by $C_{i,j}^n$. Equation (4) may be discretised in the following way. Let the $F_{i,j}^n$, $G_{i,j}^n$ denote the flux of volume fraction C leaving the cell (i,j) in direction x and y , respectively. We thus have:

$$\frac{C_{i,j}^{n+1} - C_{i,j}^n}{\Delta t} = \frac{F_{i-1/2,j}^n - F_{i+1/2,j}^n}{\Delta x} + \frac{G_{i,j-1/2}^n - G_{i,j+1/2}^n}{\Delta y} \quad (5)$$

which can be solved for $C_{i,j}^{n+1}$, if we know the way to calculate the F and G fluxes. An accurate calculation of fluxes is crucial for the quality of advection. Advection can be split into two substeps [8], so that we find all the fluxes in the x direction in one substep. Then, advection in the y direction is performed in the second substep.

The way of calculating the fluxes depends on the chosen fluid interface representation. One approach, now of historical significance only, is SLIC (*Simple Line Interface Calculation*) [6], in which the fluid interface is represented as a line parallel to the cell faces.

In Figure 1, an interface is shown with its normal vector \mathbf{m} inside a cell $\Delta x = \Delta y = h$ in size. Let $u_{i+1/2,j}$ be the value of velocity component u on the right

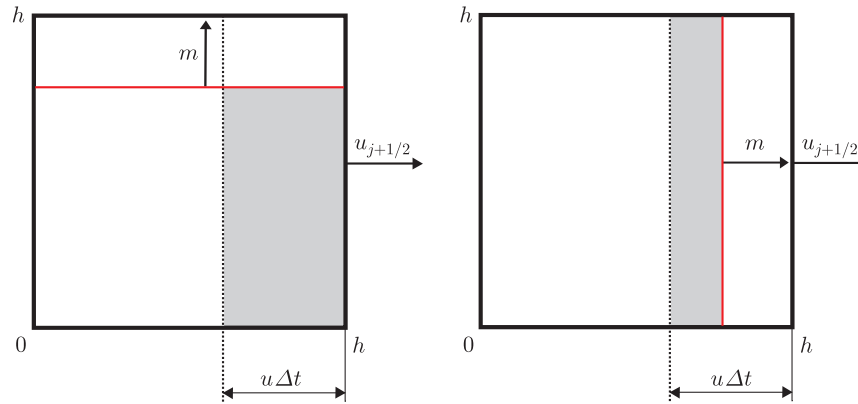


Figure 1. SLIC representation of the interface. An example of flux calculation for two orientations of the interface (2D) – the shaded area corresponds to the flux value

edge of cell (i, j) . The vertical dotted line cuts off a part of the cell (advection control volume) – a rectangle $u_{i+1/2, j} \Delta t$ in width (assuming that $u_{i+1/2, j}$ is positive.) All the fluid on the right side of this line will be transferred to cell $(i+1, j)$.

In more recent VOF applications, the interface is represented as a line that, unlike in SLIC, can have an arbitrary position with respect to the cell faces. This method is termed PLIC (*Piecewise Linear Interface Calculation*). If the vector normal to the interface $\mathbf{m} = [m_x, m_y]$ is known, we can write its equation in local (cell) coordinates:

$$m_x x + m_y y = \alpha. \quad (6)$$

An individual cell with an interface is shown in Figure 2. The CB line is the interface. Like in the SLIC example, when considering advection in the horizontal direction with a positive velocity $u_{i+1/2, j}$ all fluid positioned right to the dotted line EF will be transferred to the neighboring cell. To calculate the flux $F_{i+1/2, j}$ on the cell face, the area $FG BCE$ has to be found which can be a rectangle, trapezium or triangle. In \mathbb{R}^3 , these areas become intersections of tetrahedra and cuboids.

The interface reconstruction itself is a nontrivial task, especially in \mathbb{R}^3 . In the interface Equation (6), normal components have to be found, using for example Young's scheme [8], along with the α term of (6). Finding α is far more difficult and can be done analytically or by a fitting procedure. A brief description can be found in [7].

As the PLIC method is capable of dealing with interfaces of any orientation with respect to the computational cell (control volume), it is much more accurate than the SLIC method and many of the shortcomings of SLIC may be avoided (like “flotsam and jetsam” [8]). The main disadvantage of the PLIC approach is that the method implementation is rather difficult, especially in 3D. To calculate volume fraction fluxes a large number of orientations of the interface in the control volume along with various positions of the advection control volume has to be considered. Supposing the flux on the control volume's right face is needed and the velocity u is directed as shown in Figure 3, the advection control volume in this case is a rectangular prism $ABCDEFGH$ (the AB length is equal to $u \Delta t$) and the volume fraction amount that should be transferred to the adjacent control volume is the solid $ABCDIJKL$

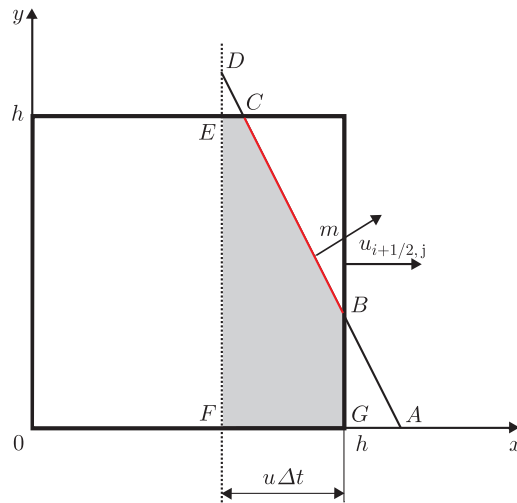


Figure 2. PLIC representation of interface (2D)

volume, bounded by the advection control volume faces and the interface. An analysis of the remaining orientations is similar and it is quite simple for a specific case, but Figure 3 gives an idea of how many cases should be considered in 3D.

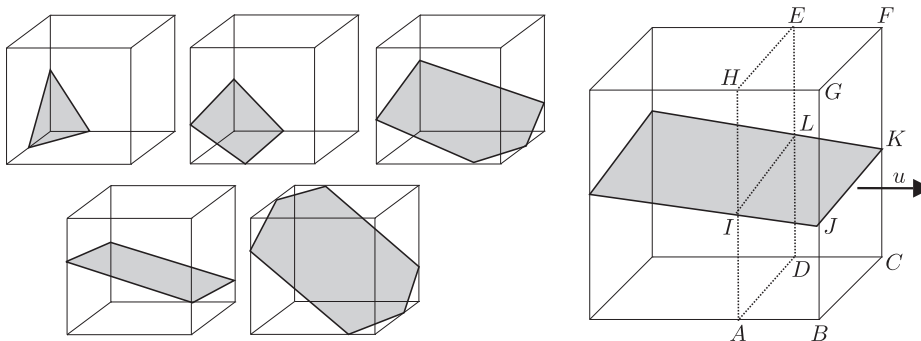


Figure 3. A few different orientations of the interface and example case for the calculation of volume fraction flux. The tracked phase is assumed to be below the interface

The fact that PLIC treats various orientations of the interface exactly and SLIC oversimplifies the problem taking only trivial orientations into account, suggests the possibility of a compromise between these two approaches. The main concept is as follows: calculate exact fluxes for trivial orientations just like in the SLIC method, then use the vector normal to the interface to estimate the actual orientation and interpolate smoothly between exact values. This method turns out to be much more accurate than SLIC and comparable with PLIC in standard tests performed in Section 4. In the next section details of the 2D and 3D algorithms will be discussed. To make the presentation easier, the proposed method will be called SVOF (Simplified Volume of Fluid method), as it seems to be a simpler, more attractive alternative compared to PLIC VOF.

SVOF has been recently developed by the authors and used with success also for modeling of solidification (unpublished results). However, the authors are bound

to admit that when preparing this article they came across a publication [9] released a few months ago in which a very similar method was described, although derived in a somewhat different context. Bearing that in mind, the authors cannot claim that the method presented in this paper is original.

2. SVOF – 2D algorithm

First, the 2D version of the method will be described for the advection of volume fraction function C in a given velocity field $\vec{V} = [u(x,y), v(x,y)]$. The computational grid is assumed to be a staggered, regular array of square cells h in size. The volume fraction is assigned to the center of a given cell and the velocity components – to its faces. The component u is stored on the vertical faces, the component v – on the horizontal ones. In the case of collocated grids, the velocity on a specific cell face may be found by interpolation.

The advection is performed using directional splitting, *i.e.* each time step is divided into two stages: it is only the u component of the velocity field that is considered (advection along x direction) in the first stage, and it is the v component that is chosen (advection along y direction) in the second stage. The first stage will be discussed in detail below. The second stage will be dealt with in a similar way.

1. For each vertical cell face the donor cell must be identified. A donor cell is defined as a cell from which the flux will flow to the adjacent cell (acceptor) (see [6]). Donor cells may be assigned solely by checking the velocity sign $u_{i+1/2,j}$ on the given cell face (Figure 4):

$$(i^{(d)}, j^{(d)}) = \begin{cases} (i, j) & \text{for } u_{i+1/2,j} > 0, \\ (i+1, j) & \text{for } u_{i+1/2,j} < 0, \end{cases} \quad (7)$$

where $(i^{(d)}, j^{(d)})$ denotes the donor cell indices.

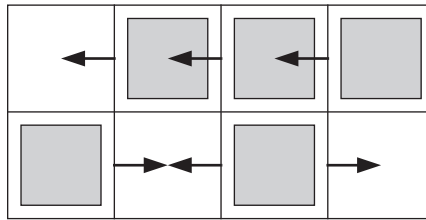


Figure 4. Donor cells identification. For the given directions of velocities, donor cells are depicted as gray boxes

2. Calculation of vector $\mathbf{m} = [m_x, m_y]$ normal to the interface for each donor cell. Youngs method is employed (see [8]), that is:

$$\begin{aligned} m_x^{i,j} &= -[C_{i+1,j+1} + C_{i+1,j-1} - C_{i-1,j+1} - C_{i-1,j-1} + 2(C_{i+1,j} - C_{i-1,j})], \\ m_y^{i,j} &= -[C_{i-1,j+1} + C_{i+1,j+1} - C_{i-1,j-1} - C_{i+1,j-1} + 2(C_{i,j+1} - C_{i,j-1})]. \end{aligned} \quad (8)$$

3. Transformation of a donor cell into a standard form. Calculation of the fluxes is more convenient when only one cell and outgoing flux configuration is to be considered. The standard form of a donor cell is shown in Figure 5 (on the right) and the flux is, by definition, directed outwards the bottom face in this

form. This configuration may be obtained by right angle rotations of the donor cell and the normal vector assigned to the cell. The components m_x^L , m_y^L of the normal vector in the local coordinate system are:

$$[m_x^L, m_y^L] = \begin{cases} [m_y, -m_x] & \text{for } u_{i+1/2,j} > 0, \\ [-m_y, m_x] & \text{for } u_{i+1/2,j} < 0. \end{cases} \quad (9)$$

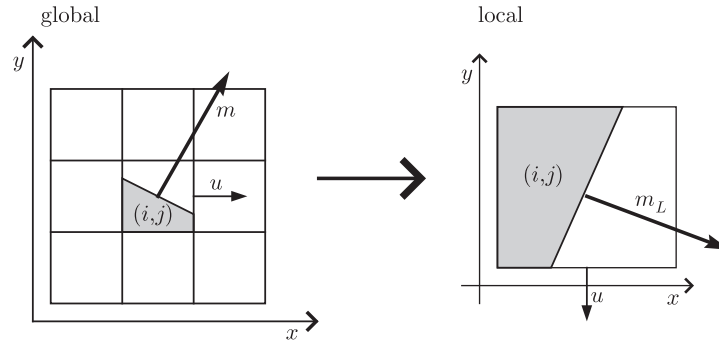


Figure 5. Donor cell transformation (left picture) into the standard form (right picture)

4. Interpolated fluxes calculation. This is an essential part of the SVOF algorithm. It is only fluxes for simple SLIC-type orientations that are calculated exactly. In Figure 6 these orientations are shown on the left (trivial cases) – (u), (m), (b). There is another orientation which is a mirror reflection of case (m), but it is not necessary to consider it as a separate case. The f flux may be expressed as:

$$f = \begin{cases} \max\{0, u\Delta t/h - (1-C)\} & \text{for case (u),} \\ C \cdot u\Delta t/h & \text{for case (m),} \\ \min\{C, u\Delta t/h\} & \text{for case (b),} \end{cases} \quad (10)$$

where C is the volume fraction in a given cell and the Δt time step is chosen to satisfy the $u\Delta t/h < 1$ Courant-Friedrichs-Lévy (CFL) condition. The graph of the $f(u\Delta t/h)$ function is shown on the right in Figure 6 (solid lines).

The f flux for the remaining orientations is calculated by means of an interpolation between exact values. Examples of such intermediate cases are denoted as (i). The trivial cases can be selected only on the basis of the m_y^L normal vector component sign. Using a simple, linear interpolation, we can define the $f^{(i)}$ flux as:

$$f^{(i)} = \begin{cases} w_i f^{(m)} + (1-w_i) f^{(u)} & \text{for } m_y^L < 0, \\ w_i f^{(m)} + (1-w_i) f^{(b)} & \text{for } m_y^L > 0, \end{cases} \quad (11)$$

where the upper indices denote fluxes corresponding to the trivial cases and w_i is the interpolation weight. There are many possible choices for w_i . The authors have found that a value based on the angle between a normal vector and a local x -axis give satisfactory results:

$$w_i = \frac{2}{\pi} \arccos\left(\frac{|m_x^L|}{|m|}\right). \quad (12)$$

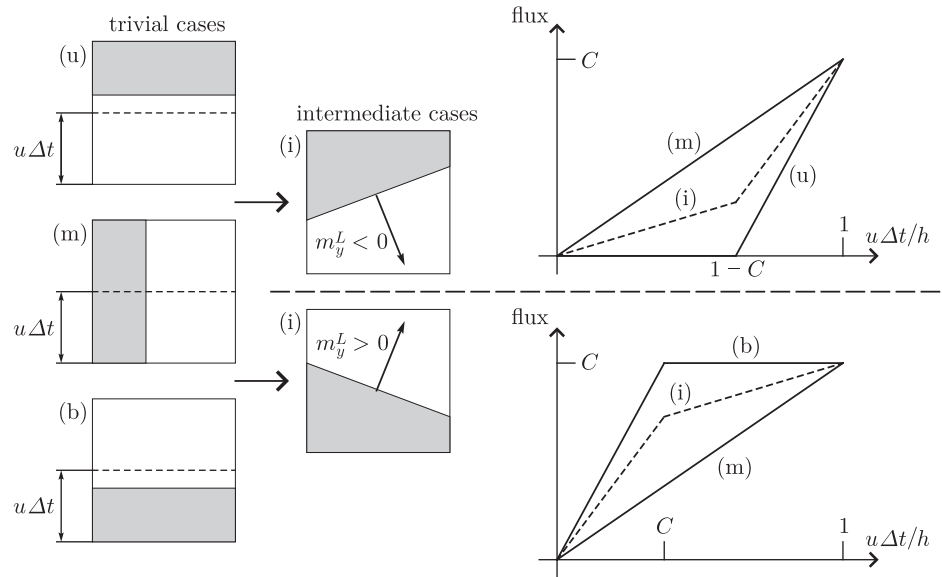


Figure 6. Calculation of interpolated fluxes

5. Volume fraction update. For every donor cell and the acceptor cell assigned to it, the volume fractions must be changed depending on the $f^{(i)}$ interpolated flux value:

$$C_d \rightarrow C_d - f^{(i)}, \quad C_a \rightarrow C_a + f^{(i)}. \quad (13)$$

This scheme ensures volume fraction conservation in the domain.

The second stage, advection in the y -direction, is performed in exactly the same way. The only difference is that the v -component is taken instead of u , donor-acceptor pairs are oriented in the y -direction (Figure 4 should be rotated by 90°) and the donor cell transformation into the standard form is:

$$[m_x^L, m_y^L] = \begin{cases} [-m_x, -m_y] & \text{for } v_{i,j+1/2} > 0, \\ [m_x, m_y] & \text{for } v_{i,j+1/2} < 0. \end{cases} \quad (14)$$

The method order in time may be increased by exchanging the directions of advection in every time step (Strang splitting, see [8]).

3. SVOF – 3D algorithm

The algorithm presented in the previous section may be surprisingly easily generalized to three-dimensional domains (this is not the case for PLIC). The $\mathbf{V} = [u, v, w]$ velocity field has three components now, hence, three advection steps must be performed every time step, along the splitting scheme lines. The algorithm details will not be described here but the main ideas are as follows.

The donor cells are identified in an analogous way to the 2D case based on the velocity sign on a given cell face. Donor cell transformation into the standard form is slightly more complicated, as there are three components to transform and there are six faces of a control volume (*i.e.* six cases instead of four in 2D). The standard form is shown in Figure 7) – the outgoing flux is located at the bottom face and it is directed outwards the cell.

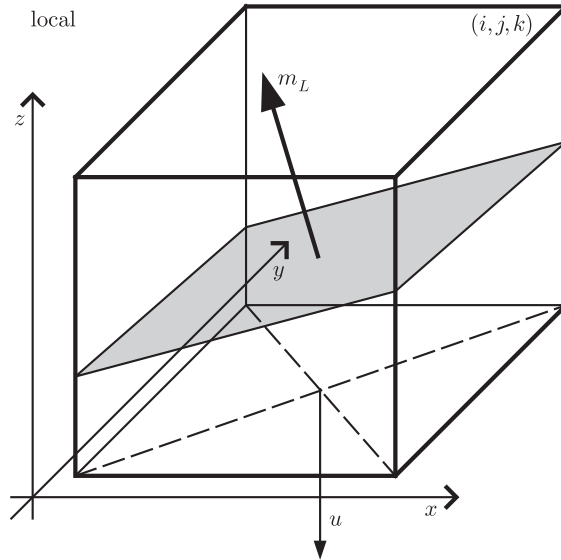


Figure 7. Donor cell standard form in 3D

Another difference is the normal vector estimation which may also be found using Youngs' method. The formulas for 3D are more complex but they can be derived in a similar way. A great advantage of the SVOF approach is that interpolated fluxes are calculated in the same manner as in the 2D scheme. The trivial cases for interpolation are chosen on the basis of the m_z^L sign and the w_i weight is:

$$w_i = 1 - \frac{2}{\pi} \arccos \left(\frac{|m_{xy}^L|}{|m|} \right), \quad (15)$$

where m_{xy}^L – normal vector projection on the xy plane of the local coordinate system.

4. Sample test results

A typical test of any interface tracking method is to assess its performance in a passive velocity field. In order to compare the three methods – PLIC, SVOF and SLIC – a circle in a square domain of a unit size has been subjected to the velocity field:

$$\mathbf{V} = [\sin(\pi x) \cdot \cos(\pi y), -\cos(\pi x) \cdot \sin(\pi y)]. \quad (16)$$

The results are shown in Figure 8. It can be noticed that the interface shape obtained with the use of SVOF (the middle column) is nearly the same as that calculated with PLIC. However, a small deformation has appeared in the thick part of the tracked phase and the filament is slightly shorter (PLIC results are taken as a reference). Nevertheless, it seems that some considerable improvement of the SLIC method has been achieved – SLIC calculations (the right column) reveal a deficiency of the method, the interface is highly distorted and the filament is much shorter.

In Figure 9 it is possible to compare the initial shape of the interface and the final shape obtained by reversing the velocity field at the end of the previous simulation and performing the same number of additional time steps. For a perfect advection scheme, the initial and final shapes should be identical. In this test, the

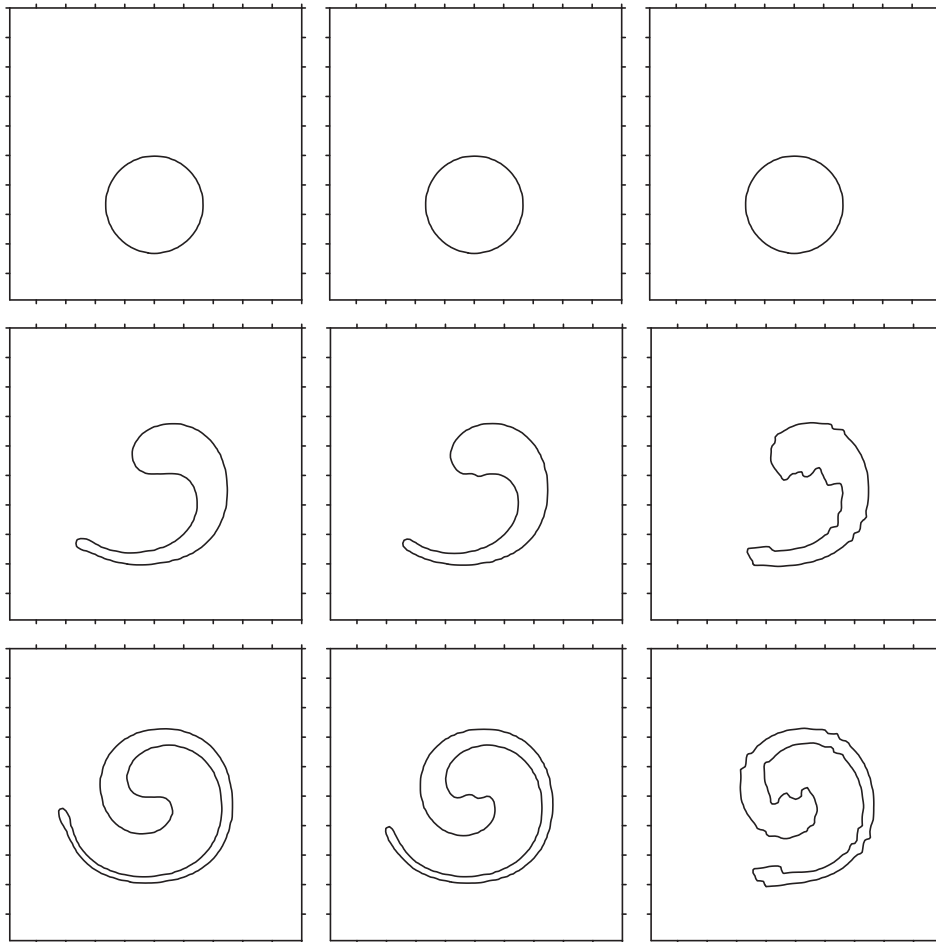


Figure 8. Test of passive advection in shear flow. The three columns correspond to: PLIC (left), SVOF (middle) and SLIC (right). Grid 100×100 , $\Delta t = 10^{-3}$, 10^4 time steps

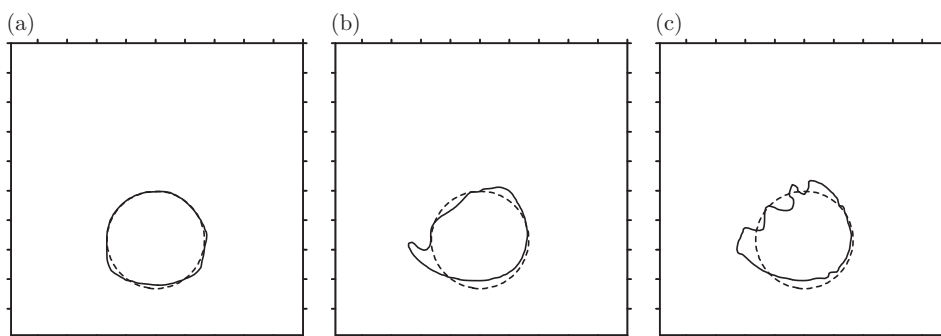


Figure 9. Test of passive advection in shear flow. Initial position of the interface (dashed line) and final state (solid line) for: (a) PLIC, (b) SVOF, (c) SLIC. Grid 100×100 , $\Delta t = 10^{-3}$, $2 \cdot 10^4$ time steps

superiority of PLIC is most visible. Both SVOF and SLIC calculations lead to irregular shapes (SVOF results seem to be significantly better, though).

Another test has been performed for the 3D version of the algorithm. A sphere has been placed in a cube of a unit size and subjected to the velocity field:

$$\begin{aligned} u &= 2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z)\cos(\pi t/3), \\ v &= -\sin(\pi x)\sin^2(2\pi y)\sin(2\pi z)\cos(\pi t/3), \\ w &= -\sin(\pi x)\sin(2\pi y)\sin^2(2\pi z)\cos(\pi t/3). \end{aligned} \quad (17)$$

In Figure 10 the tracked phase evolution is presented. The stages from (a) to (d) correspond to an evolution from the initial state to the middle of the simulation, then the flow direction is reversed and stage (e) should be identical to stage (c). Stage (f) is the final state. The initial shape of the interface (wireframe) has been superimposed on the final shape (smooth surface). Just like in the 2D algorithm's case, small distortions of the interfaces can be noticed. Although, bearing in mind the simplicity of the method and the fact that this test is considered as hard for an advection scheme, the results are quite satisfactory.

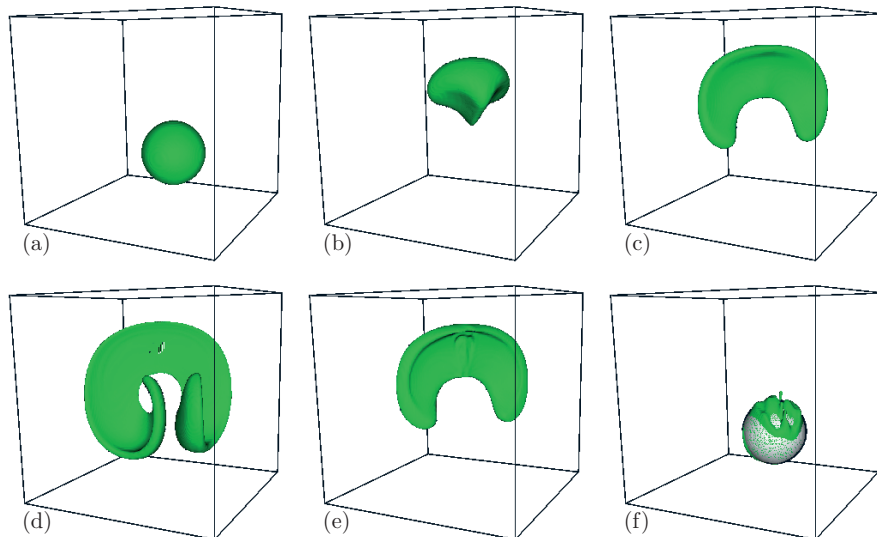


Figure 10. Test of passive advection in 3D shear flow.
Grid $150 \times 150 \times 150$, $\Delta t = 10^{-3}$, 10^3 time steps

In the end, a hypothetical application of the SVOF method to a simulation of a two-phase flow will be presented. Calculations of the well-known “broken dam” problem have been performed using an in-house 2nd order flow solver and SVOF for tracking the interface. Sample results are shown in Figure 11. In the beginning, the tracked phase (dark area) is placed at one side of the domain, as if bounded by a wall and a thin dam. The simulation starts when the dam breaks and the tracked phase collapses due to gravity, then it hits the opposite wall. The density ratio has been equal to 20:1 and the viscosity has been relatively large.

5. Conclusions

The method proposed in this paper – SVOF – may be considered as an improved SLIC algorithm version which is significantly more accurate but nearly as easy to

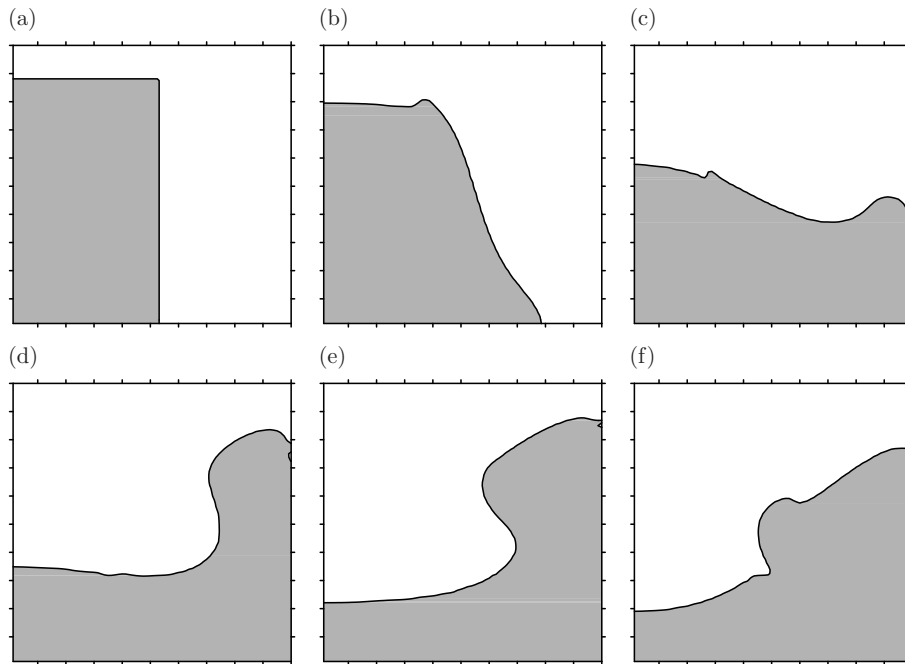


Figure 11. Broken dam problem (SVOF). Grid 80×80 , full-slip boundary conditions, density ratio 20:1

implement. SVOF avoids an intricate geometrical analysis of different orientations of the interface in a control volume which is characteristic of the PLIC method. However, in many classical passive advection tests, SVOF provides results comparable to that obtained with the use of an exact PLIC approach.

Acknowledgements

The investigations presented in this paper have been obtained under statutory research Project No. BS-1-103-301/2004/P.

References

- [1] Tryggvason G, Bunner B, Juric D, Tauber W, Nas S, Han J, Al-Rawahi N and Jan Y-J 2002 *J. Comput. Phys.* **169** 708
- [2] Osher S and Sethian J A 1988 *J. Comput. Phys.* **79** 12
- [3] Sussman M and Puckett E G 2000 *J. Comput. Phys.* **162** 331
- [4] Annaland M, Deen N and Kuipers J 2005 *Chem. Engng Sci.* **60** 2999
- [5] Puckett E G and Saltzman J S 1992 *Physica D* **60** 84
- [6] Hirt C W and Nichols B D 1981 *J. Comput. Phys.* **39** 201
- [7] Gueyffier D, Li J, Nadim A, Scardovelli R and Zaleski S 1999 *J. Comput. Phys.* **152** 423
- [8] Pilliod J E Jr. and Puckett E G 2004 *J. Comput. Phys.* **199** 265
- [9] Yokoi K 2007 *J. Comput. Phys.* **226** 1985

