Journal of Computational Physics 229 (2010) 2853–2865

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Conservative Volume-of-Fluid method for free-surface simulations on Cartesian-grids

G.D. Weymouth, Dick K.-P. Yue *

Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA

ARTICLE INFO

Article history:
Received 9 April 2009
Received in revised form 3 November 2009
Accepted 14 December 2009
Available online 4 January 2010

Keywords:
Cartesian-grid
Volume-of-Fluid
Breaking waves

ABSTRACT

This paper contributes to the state of the art in Cartesian-grid methods through development of new advection and reconstruction Volume-of-Fluid (VOF) algorithms which are applicable to two and three-dimensional flows. A computationally efficient and second-order VOF reconstruction method is presented which uses no inversions to determine the interface normal direction. Next, the lack of conservation of fluid volume in previous VOF advection methods are shown to be due to improper treatment of one-dimensional stretching in the velocity field. This paper uses simple explicit time stepping and a cell-center estimate of the volume fraction in the dilatation term to achieve a completely conservative advection method. The new methods are simple, robust and shown to out perform existing approaches for canonical test problems relevant to breaking wave flows.

1. Introduction

The free-boundary in multi-phase flows (systems with two or more fluid materials, i.e., those with a liquid/gas interface) presents a severe computational modeling challenge. By definition, the position and forcing on a free-boundary is unknown *a priori* which greatly complicates the solution of the boundary value problem of the fluid system. While methods which rely on linearization or surface tracking (such as [1]) have been successfully used to model low-steepness free-surface flows, they have a limited ability to model energetic and topologically complex free-boundaries such as breaking waves.

Methods which immerse physical boundaries on Cartesian background grids can model complex multi-phase flows and are well suited to study energetic and breaking waves. These so-called 'surface capturing' methods use a characteristic function to define the location of the free surface in time. Because the grid does not follow the material interface the numerical problems of grid skew and connectivity are avoided.

The function describing the fluid type, the so-called fluid 'color function', $c(\vec{x})$ is defined as

$$c(\vec{x}) = \begin{cases} 1 & \text{if } \vec{x} \in \text{ 'dark' fluid} \\ 0 & \text{if } \vec{x} \in \text{ 'light' fluid} \end{cases} \tag{1}$$

In the physical simulations to follow, the dark fluid is water and the light fluid is air but any appropriate two-phase flow (oil and water, etc.) can be simulated. If the color-function is known it is simple to determine the spatially variable fluid property fields, for example

$$\rho(\vec{x}) = \rho_w c(\vec{x}) + \rho_a[1 - c(\vec{x})] \tag{2}$$

* Corresponding author.
E-mail addresses: weymouth@mit.edu (G.D. Weymouth), yue@mit.edu (D.K.-P. Yue).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.12.018

where the (assumed constant) densities of water ρ_w and air ρ_a have been used with the color-function to define the global field $\rho(\vec{x})$. Using this field in the momentum equation allows one velocity field to describe the complete flow over both the air and water, and explicit definition of the interface location is not required. This is what enables flows with arbitrarily complex topology to be simulated.

There are different classes of free-boundary capturing methods, notably the level-set method and the Volume-of-Fluid (VOF) method. Level-set methods are a well established and vibrant research topic. Recent papers have applied level-set methods to the direct numerical simulations of turbulent breaking waves [2] and recent extensions of level-set methods have achieved sharp interface characteristics [3]. However, in this paper we focus on improvements to the direct VOF method. Variations of VOF have been used in CFD simulations for more than 25 years [4], and the use and improvement of VOF for complex free-surface flows is a subject of active current research, e.g. [5].

In the VOF method, the characteristic function f is obtained by integrating the color-function over each of the computational volumes Ω:

$$f = \frac{\int_\Omega c(\vec{x})\,\mathrm{d}v}{\Delta\Omega}, \tag{3}$$

where $\Delta\Omega \equiv \int_\Omega \mathrm{d}v$ is the volume of the cell. The field f is called the volume fraction, and bounds on it follow immediately from Eqs. (1) and (3):

$$0 \leqslant f \leqslant 1 \tag{4}$$

which state that a cell cannot be less than empty or more than full. An additional benefit of VOF is that conservation of f ensures conservation of the volume of each fluid type in the system, by definition. This important property is lost when using level-set capturing methods.

There are, however, still fundamental open problems in computing dynamic free-surfaces with the VOF method. First is the problem of interface reconstruction, wherein the values of the volume fraction in a local group of cells are used to reconstruct the explicit location of the interface. First-order assumptions were used in the original VOF methods which simplified the reconstruction and flux calculation process, but they result in extremely poor performance [6]. In [7], a set of analytic equations is presented which relate the intercept of a linear interface with the volume fraction once the normal is known. However, determining the normal direction is not trivial and [6] shows that a poor estimation can limit a linear reconstruction to first-order convergence.

Ref. [6] suggests two second-order methods for determining the normal, the so-called LVIRA (Least-square Volume-of-fluid Interface Reconstruction Algorithm) and ELVIRA (Efficient Least-square Volume-of-fluid Interface Reconstruction Algorithm) methods. Both are based on generating a linear interface which minimizes the volume fraction error in the surrounding cells (3×3 in 2D and $5 \times 5 \times 5$ in 3D) and has zero error in the central cell. While LVIRA and ELVIRA are second-order reconstruction methods, they both require repeated generation and inversion of 'trial interfaces'. LVIRA iteratively minimizes the squared error using a nonlinear minimization technique. As such it involves an unknown number of inversions for the volume fraction in each of the cells in the local block. In ELVIRA, a finite set of equations (six in 2D) generate the 'trial interfaces'. Ref. [6] shows that at least one of these equations reconstructs a linear interface exactly, however, that reference does not fundamentally relate the form of these equations to the interface normal, nor does it deal with extensions to 3D or non-uniform grids.

The second open problem in VOF free-surface simulation is the conservative advection of the volume fraction field f. The field c and other fields based on it, such as f and ρ, are discontinuous across the interface. Smoothing the color-function is not an option because it results in an uncontrollable widening of the transition between water and air, as discussed in [8]. This issue, along with condition (4) make proper transport of the volume fraction nontrivial. In fact, there are no conservative VOF transport methods for three-dimensional (3D) flows in the literature [9]. Volume conservation is a zeroth-order condition (equivalent to mass conservation in the flow solver) and its violation results in many non-physical characteristics such as spurious flotsam and jetsam and high-frequency fluctuations in interface pressure.

References such as [6,10] present unsplit advection algorithms based on geometric flux integrations along approximated characteristics of the velocity field. These are, however, algorithmically complex particularly for three-dimensional flows. In addition, such methods often violate conservation drastically [11], due to overshooting of condition (4). Algorithms which split the advection problem along each Cartesian component are prevalent [10,8] and much less complex but still have not produced a conservative 3D method. A recent paper [12] investigates this problem in detail and [9] develops a two-dimensional algorithm EI–LE (Eulerian Implicit–Lagrangian Explicit) which conserves the volume fraction exactly. Unfortunately, this is dependent upon a two-dimensional divergence-free velocity field and the mathematical analysis in that work does not naturally extend to 3D.

Solutions of the reconstruction and advection problems, in an otherwise mature and commonly used method such as VOF, are therefore overdue. A recent paper by Weymouth et al. [13] briefly presents new efforts in this direction and this paper is a continuation of that work presenting practical solutions to both the reconstruction and advection problems.

In Section 2 the transport problem is presented in further detail. Section 2.1 demonstrates the nonlinear connection between gradients of the volume fraction and the interface normal in the reconstruction problem. Utilizing this, a second-order and computationally efficient reconstruction method is developed which requires no inversions of 'trial interfaces'. Section 2.2 discusses the stretching of the volume fraction field f in an operator-split algorithm which is the primary difficulty in

conservative algorithms of this type. A new and simple advection algorithm is presented which anticipates such stretching, enabling it to be fully conservative. Both the reconstruction algorithm and advection algorithm are easily applicable to two or three-dimensional flows on uniform or nonuniform Cartesian background grids. Section 2.3 presents a series of canonical tests of the advection method including a new corner flow test which highlights the need for test cases with flow stretching. Finally, in Section 3, the complete VOF method is used in a Cartesian-grid flow solver to simulate two test flows, a simple linear wave and a large-amplitude breaking wave in 3D.

2. Volume fraction transport equation

The transport equation for the color-function is based on the conservation of fluid type for any material fluid point. The color-function conservation equation is simply

$$\frac{Dc}{Dt} = \frac{\partial c}{\partial t} + \vec{u} \cdot \vec{\nabla} c = 0. \tag{5}$$

As discussed in Section 1, the gradients of c are not defined across the interface (and are zero elsewhere), but such gradients can be avoided by using an integral form of the transport equation. This is derived by integrating (5) over each cell, integrating the second term by parts and using the divergence theorem to obtain:

$$\frac{\partial}{\partial t}\int_{\Omega} c\,\mathrm{d}v + \oint_{\partial\Omega} c u_n\,\mathrm{d}s = \oint_{\Omega} c\vec{\nabla}\cdot\vec{u}\,\mathrm{d}v, \tag{6}$$

where the term on the right hand side is the dilatation of dark volume, equal to zero in an incompressible flow. Eq. (6) is in fact a conservation equation for the volume fraction f instead of the color-function. Thus, the transport equation is written compactly as

$$\frac{\partial f}{\partial t}\Delta\Omega + F_{net} = \oint_{\Omega} c\vec{\nabla}\cdot\vec{u}\,\mathrm{d}v \tag{7}$$

where F_{net} is the net flux of dark fluid *out of* Ω.

The volume fraction is updated in time by solving (7) in two steps. First, the reconstruction step in which the explicit interface location is locally approximated from the volume fraction field. Second, the advection step in which the fluxes are computed and (7) is integrated forward in time.

2.1. Reconstruction method

In order to accurately compute the fluxes in (7) the free-interface must be locally reconstructed. A 2D example is sketched in Fig. 1. On the left is the true free-interface location, however, the only information available is the value of the volume fraction f for each cell.

As stated above, a local linear reconstruction is required of the form

$$\vec{m}\cdot\vec{x} = \alpha \tag{8}$$

where $\vec{m}$ is the surface normal vector and α is the intercept. Because the descriptions are local, the surfaces do not generally match on the boundaries as shown in Fig. 1. Determining the value of α and $\vec{m}$ from f for each cell on the interface $(0 < f < 1)$

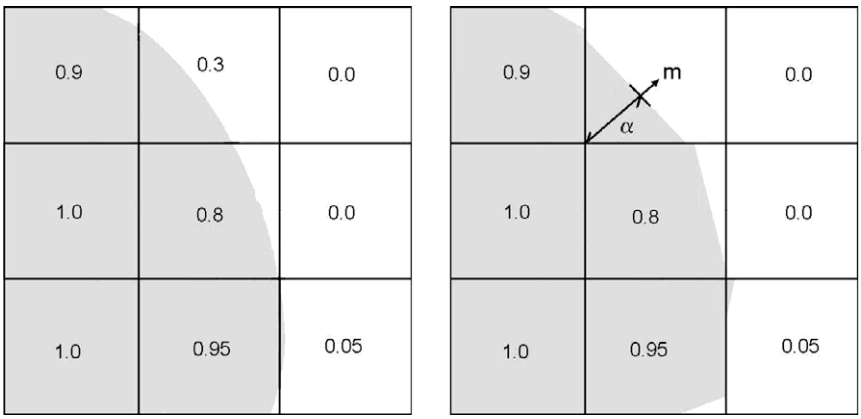


Fig. 1. Schematic of a typical 3×3 block of cells in 2D. On the left is the exact interface and on the right is the linear reconstruction which maintains f. Determining the local linear interface from the volume fractions defines the reconstruction problem.

is the reconstruction problem. As noted in the introduction, closed form solutions for $\alpha = \alpha(f, \vec{m})$ and $f = f(\alpha, \vec{m})$ are available in 2D and 3D but determining the interface normal as a function of the local volume fractions is nontrivial.

In level-set methods, the unit normal vector is computed simply by taking the gradient of the interface distance function. Such a function is not available in VOF. Typical reconstruction methods make use of an analogous thinking in estimating the normal $\vec{m}$ as some finite difference of the local volume fraction field. However, this analogy is confused as gradients of f are not analytically defined across the interface. Pilliod and Puckett [6] demonstrate that such methods do not result in second-order estimates of the normal, although the ELVIRA method is a successful amalgamation of such difference formulae.

Examination of the interface equation is the key to uncovering an exact relationship between the normal and the volume fraction field. As the length of $\vec{m}$ is arbitrary, (8) for 2D can be written as

$$y = \alpha - mx, \tag{9}$$

where the y-component of the normal has been set to 1. If values of y on the interface were known, the normal could be easily established using

$$m = -\frac{\partial y}{\partial x}. \tag{10}$$

These values of y are not known (we have no distance function in VOF), however, we can utilize the mean-value theorem to estimate y from the only information available, the finite volume f. We define the mean-value of the interface as the integral of that function divided by the integration width,

$$\bar{y} = \frac{\int_a^b y(x)\,\mathrm{d}x}{\int_a^b \mathrm{d}x} \tag{11}$$

which is equal to $y(a/2 + b/2)$ for a linear interface. When the interface does not pass through the top or bottom boundary of a cell, (11) may be related to the volume fraction by

$$\bar{y} = \frac{\int_\Omega c\,\mathrm{d}v}{\int_{\partial\Omega} \mathrm{d}x} = f\Delta y. \tag{12}$$

Taking a cue from previous reconstruction methods, we increase the chances of the interface staying within the upper and lower boundary by tripling the height of the cell as shown in Fig. 2. In that figure the original cells are summed vertically resulting in three concatenated cells. As shown in the figure, this ensures that at least two extended cells accurately estimate the interface position ($\bar{y} \neq f\Delta y$ in the left concatenated cell because the interface crosses the upper boundary of that extended cell). Using finite differences on those two accurate interface heights reproduces the normal exactly for a linear interface, the requirement for a second-order reconstruction method.

With Eq. (12) and the understanding of its range of validity, we develop a general reconstruction process on Cartesian grids without inverting 'trial interfaces'. First, we use central differences of f to estimate m_x, m_y (and m_z in 3D). This does not generally lead to the correct value of m but it does give us an estimate of the interface orientation. Define the major axis

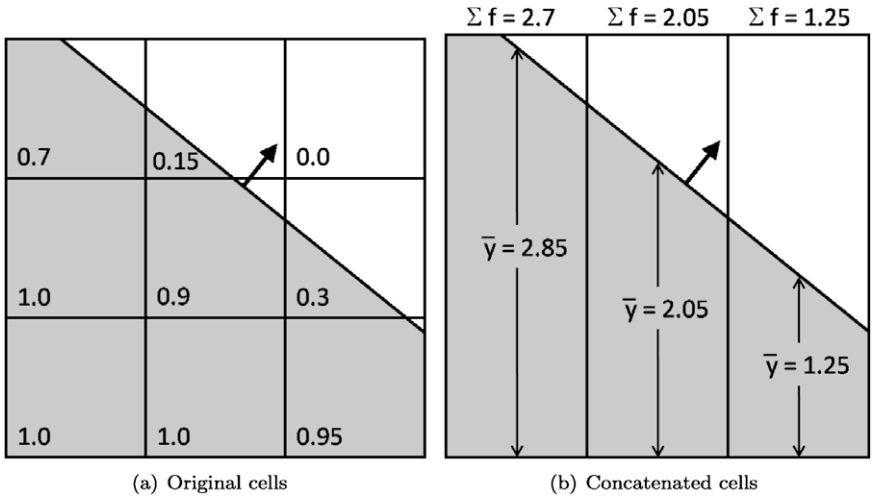


Fig. 2. (a) Illustration of determining the interface height $\bar{y}$ for a linear interface and unit square cells. When the interface does not cross the top or bottom of the cell, the volume fraction f is related to $\bar{y}$ through Eq. (12). Concatenating cells, as in (b), insures that this condition is met for at least two volumes (the center and right concatenated cells in this example) allowing the exact mean values and normal to be determined from f directly.

as the one with the largest component of $\vec{m}$. For discussion, assume this is the z axis. Next, estimate the interface height by summing in that direction, i.e.,

$$\bar{z}_{i,j,k} = \sum_{l=-1}^{1} f_{i,j,k+l} \Delta z_{i,j,k+l}. \tag{13}$$

As shown in the figure, the height of the central cell is always estimated with second-order accuracy. If the central cell is more than half full then height in the cell 'downwind' of the normal (the right cell in the figure) is also estimated to second-order. If the center cell is less than half full the 'upwind' cell value is estimated to second-order. Finally, compute

$$m_x = \frac{\partial \bar{z}}{\partial x}; \quad m_y = \frac{\partial \bar{z}}{\partial y} \tag{14}$$

using forward or backward differences as appropriate.

The new method allows exact reconstruction of a linear interface in 2D or 3D using only the local $3 \times 3(\times 3)$ block of cells and no inversions. As such it represents a significant savings in computation time. On a 2D uniform grid we require only seven arithmetic operations and two conditional statements to determine a cell's interface normal. In comparison ELVIRA requires 6 or 7 operations to determine each of the 6 trial normals, plus coordinate transformations of the trial normals into each of the 8 surrounding cells, plus inversions of the volume fraction (around 12 operations and 3 conditionals) for all 9 local cells for all 6 trial normals, plus the operations and conditionals to choose the best of the trial normals. The total computational cost of the current method is thus around two orders of magnitude less than the ELVIRA method which itself is an improvement on the LVIRA method. Because the new method reproduces a linear interface exactly, the method is assured to be second-order, and this is verified in the following sections. It is used for the remainder of this work.

2.2. Advection method

Once the reconstruction step is complete, Eq. (7) is used to update the volume fractions in the advection step. As noted above, standard VOF methods use an operator-split advection method wherein the transport problem is split into sequential updates of the volume fraction in each of the $\mathcal{N}$ spatial dimensions. The split version of (7) is:

$$\Delta f'_{ijk} \frac{\Delta\Omega}{\Delta t} = F_{i+1/2} - F_{i-1/2} + \int_\Omega c \frac{\partial u}{\partial x} \, \mathrm{d}v \tag{15}$$

$$\Delta f''_{ijk} \frac{\Delta\Omega}{\Delta t} = G_{j+1/2} - G_{j-1/2} + \int_\Omega c \frac{\partial v}{\partial y} \, \mathrm{d}v \tag{16}$$

$$\Delta f'''_{ijk} \frac{\Delta\Omega}{\Delta t} = H_{k+1/2} - H_{k-1/2} + \int_\Omega c \frac{\partial w}{\partial z} \, \mathrm{d}v \tag{17}$$

where G and H are the fluxed dark fluid in the second and third directions; i.e., the fluid is first transported along x, then y, and then (for 3D flows) z. For brevity, we use a new notation to rewrite (15)–(17) simply as:

$$\Delta f \frac{\Delta\Omega}{\Delta t} = \Delta_d F_d + \int_\Omega c \frac{\partial u_d}{\partial x_d} \, \mathrm{d}v \quad \text{for } d = 1, \ldots, \mathcal{N}; \tag{18}$$

where d is the Cartesian index and $\Delta_d F_d$ is the net flux in that direction. By treating only one velocity component at a time, the calculation of the fluxed dark volume can be analytically determined using the relations in [7]. A typical two-dimensional advection problem is shown in Fig. 3. In this figure the velocities are scaled by $\Delta t/\Delta x$ making them local Courant numbers. With this scaling a donating region upwind of each face can be defined with width equal to the velocity magnitude. The dark fluid within each region is fluxed into the next cell. In Fig. 3, the fluid in the bottom-right corner of cell (i,j) is in two donating regions. This illustrates that in order to avoid the possibility of fluxing the same fluid into two different cells, the surface must be reconstructed after each sweep of an operator-split method.

The dilatation term in (18) is required because, at each step of an operator-split algorithm, the volume fraction is advected in a one-dimensional flow *which is not divergence-free*. Without accounting for the dilatation of dark volume there is no way to ensure that condition (4) is met after each step of the algorithm. For example, in Fig. 3 more fluid is being fluxed in from cell $(i-1,j)$ than space left after fluxing out to cell $(i+1,j)$, meaning that cell (i,j) overfills in the first sweep. A recent paper [12] investigates this effect in detail and shows that no operator-split method in the literature is truly volume conserving. The conservative advection method developed in that work is mathematically elegant, but unfortunately does not extend to 3D or nonuniform grids.

The problem of conservation can be stated in terms of a short set of concurrent requirements (see Weymouth et al. [13]). If for a given algorithm:

1. The flux terms are conservative, *and*
2. the divergence term sums to zero, *and*
3. no clipping or filling of a cell is needed due to violation of (4) at any stage;

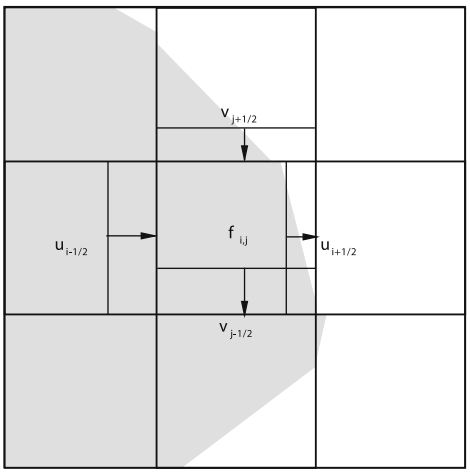


Fig. 3. 2D diagram of a typical linear VOF surface reconstruction on a 3×3 block of cells with scaled velocity components. The donating regions are depicted as rectangles upwind of each cell face. The dark fluid in each donating region is fluxed to the neighboring cell. While the local velocity field for the center cell is divergence-free, stretching would cause the cell to overfill in the first step unless the 1D divergence is accounted for.

then the algorithm *must* conserve f exactly (to machine precision). The first requirement ensures that any fluid going into one cell is coming out of another and the second ensures that there is no net source term added to the advection equation. Along with the third requirement, it is guaranteed that there is no net change in the dark fluid volume regardless of the dimensionality of the system.

A simple operator-split method can be designed to meet these requirements for 2D or 3D flows. The fluxes and dilatation term in (18) must be integrated explicitly to satisfy the first and second conditions. This only leaves flexibility in the estimate of the integral in the dilatation term. Previous VOF advection methods estimate the integral using the volume fraction f. On the other hand, the cell center value of the color-function c

$$c_c = \begin{cases} 1 & \text{if} \quad f > 1/2 \\ 0 & \text{else} \end{cases} \tag{19}$$

follows from simple geometry and is equally valid. As long as the field c_c is treated fully explicitly (using f from the previous time step) the simple advection method:

$$\Delta f = \frac{\Delta t}{\Delta \Omega}\left(\Delta_d F_d + c_c \frac{\partial u_d}{\partial x_d}\Delta\Omega\right) \quad \text{for } d = 1,\ldots,\mathcal{N} \tag{20}$$

satisfies the first two requirements. The third condition, of no overfilling or over-emptying, can be enforced with a simple Courant restriction of the form:

$$\Delta t \sum_{d=1}^{\mathcal{N}} \left|\frac{u_d}{\Delta x_d}\right| < \frac{1}{2}; \tag{21}$$

guaranteeing that the transport method is fully conservative for general flows. Note that in addition to handling 3D flows, the algorithm is also applicable to both uniform and clustered Cartesian-grids. The proof of the third condition relies on bounding the possible flux through each face, and is supplied in Appendix A. The proof also demonstrates the critical importance of using Eq. (19) in the dilatation term. Thus, the advection algorithm given above enables strict conservation of fluid volume while matching (or exceeding) ease of implementation and computational efficiency found in the documented operate-split methods.

2.3. Transport method verification

There are standard tests in the literature for the numerical verification of VOF transport algorithms. The general method involves starting with an analytic interface and transporting the resulting volume fraction in a given constant velocity field. The result is then compared to the known analytic solution. Ref. [6] presents a set of such tests for uniform flow with an arbitrary orientation and rotational flow for shapes such as circles, crosses and notched circles. A common feature of these canonical VOF benchmarks is that the test flows have no stretching. This simplifies the calculation of the error metrics but artificially ensures that the methods conserve mass. All of the split-operator advection methods in the literature (including the present Eq. (20)) collapse to the simple update:

Table 1
Error results for the circle geometry translation and rotation tests. The metric is the L_1 error in the volume fraction as given by (23). The results show second-order convergence with increasing resolution. $N = R/\Delta x$ is the number of points resolving the circle.

Error of 100 random translations, $\Delta x/\Delta t = 1/2$				
N	8	16	32	64
E	1.5324e−2	4.7428e−3	1.5086e−3	4.2818e−4
Error of 100 random translations, $\Delta x/\Delta t = 1/32$				
N	8	16	32	
E	1.9208e−2	7.9863e−3	2.6550e−3	
Error of one rotation, $\Delta x/\Delta t = \pi/6$				
N	8	16	32	64
E	1.5756e−2	3.7770e−3	9.7135e−4	1.6705e−4

$$\Delta f \frac{\Delta \Omega}{\Delta t} = \Delta_d F_d \quad \text{for } d = 1, \ldots, \mathcal{N}, \tag{22}$$

for flows without stretching. Thus the present method should perform identically to previous methods for translation and rotation cases. To verify this (and the second-order convergence of the method), the translation and rotation tests for a circle geometry are repeated and the results are shown in Table 1. The error metric for these tests is simply:

$$E = \sum_{i,j,k} \left| f_{i,j,k} - \tilde{f}_{i,j,k} \right| \tag{23}$$

where $\tilde{f}$ is the exact volume fraction solution.[1] In the table $N = R/\Delta x$ where R is the circle radius. As Table 1 shows, the current method is second-order for translation and rotation cases for CFL numbers $\Delta x/\Delta t = 1/2,\ 1/32$. This is expected since the methods are effectively identical for flows with no stretching.

To highlight the true mass conservation properties of the proposed transport method relative to existing methods, we consider background flows with stretching. A simple choice is a corner flow, defined by:

$$u = x; \quad v = -y, \tag{24}$$

which has uniform stretching throughout the flow and is divergence-free. Fig. 4 and Table 2 show the results for three advection algorithms. The first, labeled 'baseline' is the advection method with no stretching term, given by (22). The second is the 2D split method of Pilliod and Puckett [6]. The third is the present method, given by (20). Fig. 4(a) shows the starting position of the circle centered at $(x/R, y/R) = (3/2, 4)$. Fig. 4(b) shows the reference solution, computed using the current method with a fine grid using $N = 64$ points. Figures (c) and (d) show solutions using the baseline and current method with $N = 32$. Clearly, the baseline method which does not account for the 1D stretching is invalid, resulting in a flow full of flotsam.

Table 2 further verifies that result. In this table, E is calculated as above and compared to the reference solution. The L_1 metric is the mean dark fluid loss at each step of the transport algorithm. Defining the volume of dark fluid as V we have:

$$L_1 = \frac{1}{T} \sum_{t=1}^{T} |V_t - V_{t-1}|. \tag{25}$$

The second loss metric is the total percent change of volume after transport:

$$Change = \frac{V_0 - V_T}{V_0}. \tag{26}$$

The table shows that the loss of volume for the baseline case is around 15%, and not mitigated with increased resolution. The Pilliod 04 method fares much better, but still features volume losses of around 0.01% even on the finest grid. The current method conserves volume exactly for every resolution. The corner flow test thus highlights the need for VOF methods to be tested in flows with stretching. The performance of the second method appears adequate for the corner flow, however, this velocity field is 2D and the stretching is very mild $\frac{\partial u}{\partial x} = 1$. In a more violent flow, such as the flow associated with breaking waves, the conservation problem is more severe. This is demonstrated in the following section.

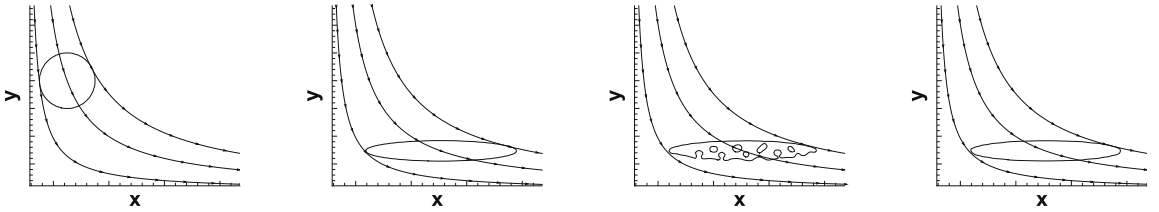
3. Two-phase flow solver verification

The VOF transport method developed and tested in the previous section is incorporated into a general flow solver to simulate general 2D and 3D two-phase flows. The solver is then tested for two free-surface flows.

[1] Ref. [6] uses a more advanced metric based on analytic integrations of the color-function. While this is necessary for establishing reconstruction errors (as $f = \tilde{f}$ for any valid reconstruction) it is not required when measuring the transport error.

Table 2
Error and dark fluid loss for the corner flow test. The L_1 metric is the average dark fluid volume loss per unit time and the % change metric is the start to finish change in dark volume as defined by (25) and (26), respectively. The current method is second-order accurate and is the only method to conserve volume exactly.

Method	N	L_1	% Change	E
Baseline	8	3.0079e−4	14.85	2.0397e−1
Baseline	16	1.4734e−4	14.15	1.9188e−1
Baseline	32	7.6070e−5	14.62	1.9631e−1
Pilliod 04	8	1.4549e−5	0.084	3.8764e−2
Pilliod 04	16	2.7978e−6	-0.036	1.1548e−2
Pilliod 04	32	5.3006e−7	0.013	2.5441e−3
Current	8	0.	0.	4.0853e−2
Current	16	0.	0.	1.1261e−2
Current	32	0.	0.	2.4661e−3



(a) Starting position (b) Reference solution, $N = 64$ (c) Baseline solution, $N = 32$ (d) Current solution, $N = 32$

Fig. 4. Contours of $f = 1/2$ and streamlines for the corner flow test case. (a) Shows the starting position, and (b) shows the reference solution obtained on a fine grid ($N = 64$). (c) Shows the result when 1D-stretching is not accounted for. (d) Shows the result when stretching is accounted for using the current method with $N = 32$.

3.1. Description of the flow solver

The fluid velocity $\vec{u}$ satisfy the conservation of momentum for an incompressible inviscid fluid, given by the Euler equation:

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u}\cdot\vec{\nabla})\vec{u} = -\frac{1}{\rho}\vec{\nabla}p - \vec{g}, \tag{27}$$

where p is the total pressure, $\vec{g}$ the gravitational acceleration vector, and ρ the local fluid density. Taking the divergence of (27) results in a variable coefficient Poisson equation for the pressure of the form:

$$\vec{\nabla}\cdot\left(\frac{1}{\rho}\vec{\nabla}p\right) = -\vec{\nabla}\cdot\left(\frac{\partial \vec{u}}{\partial t} + (\vec{u}\cdot\vec{\nabla})\vec{u} + \vec{g}\right). \tag{28}$$

The pressure field resulting from the solution of this equation is used to project the velocity field onto one satisfying the divergence-free constraint:

$$\vec{\nabla}\cdot\vec{u} = 0. \tag{29}$$

Our basic implementation of these equations follows that in the Numerical Flow Analysis (NFA) code [5]. The discrete forms of (27) and (28) are posed on a Cartesian-grid covering the fluid domain. Staggered variable placement is used. The time derivatives are treated with an explicit low storage second-order Runge–Kutta method [5]. The pressure terms are treated conservatively using central differences and a preconditioned conjugate-gradient method is used to iteratively solve the Poisson equation. The convective terms are treated with a slope-limited QUICK scheme [14] for stability and accuracy.

In many free-interface problems the effects of surface tension are significant and in these cases a forcing term must be added to the momentum equations on the interface. Such forcing terms can be based upon the volume fraction f or an interface distance function such as those commonly used in level-set methods [2]. While this class of flows have their own difficulties (such as efficient determination of the pressure field) the waves in this work are assumed to be sufficiently large that surface tension is negligible. As such, the sole adjustment to the single-phase flow solvers is the inclusion of spatially variable fluid properties, such as Eq. (2).

Two free-surface standing wave tests are presented here to verify the proper implementation of the new reconstruction and advection treatments in the VOF method: a 2D small amplitude wave, and a 3D large-amplitude breaking wave.

3.2. 2D linear wave test

The first test of the two-phase solver is a simple small amplitude standing wave, given by the interface elevation $\eta(x,t) = A\cos(2\pi x/\lambda)\cos(\omega t)$, where A is the wave amplitude, λ the wave length, and $\omega = \sqrt{2\pi g/\lambda}$ the wave frequency (in deep-water). This analytic benchmark is used to assess the performance of the numerical method but it is important to note that the correspondence is not exact. Firstly, the VOF method is a general two-phase flow solution method, whereas the analytic solution assumes the effect of the air is negligible and the free-surface BCs may be linearized around the mean free-surface. Additionally, the QUICK treatment of the fluxes introduces numerical damping on the order of $O(\Delta x^3)$, inconsistent with the inviscid wave form. Because of these factors, the VOF solution will not converge exactly to the analytic solution.

We mitigate this discrepancy by using a large density ratio $\rho_w/\rho_a = 1000$, a small wave height $A/\lambda = 0.01$, and relatively fine numerical grids $N = \lambda/\Delta x = 25,\ 50,\ 100,\ 200$. The computational domain chosen is a rectangle with width λ which extends above and below the mean free-surface by height λ with reflection boundary conditions set on all walls.

Table 3 shows the error E in the volume fraction compared to the analytic solution. The error on the finest grid is $O(10^{-4})$ and the error is monotonically decreasing with increased resolution, but the convergence rate is less than second-order because of the modeling errors discussed above. To measure the convergence rate more precisely, we compare the error at each refinement level with the finest solution, labeled E_{ref} in the table. The results are approximately second-order and the VOF flow solver is numerically verified and analytically validated.

3.3. 3D nonlinear wave test

A large-amplitude 3D standing air-water wave is used for the next test. Because this wave is nonlinear, there is no analytic solution and we instead compare our results with those from existing VOF methods.

The same Cartesian-grid solver described in the previous section is run on a cubic computational domain, with $N = \Delta x/\lambda = \Delta y/\lambda = \Delta z/\lambda = 128,\ \Delta x/\Delta t = 0.05$ and reflection boundary conditions set on all walls. An initial doubly sinusoidal free surface, $\eta(x,y,t=0) = A\cos(2\pi x/\lambda)\cos(2\pi y/\lambda)$ is prescribed with $A = 0.25\lambda$. The slope is well above the Stokes limit and the three-dimensional free-interface quickly evolves into a nonlinear high-energy breaking wave as seen in Fig. 5.

Four VOF transport methods are tested, the first is a baseline test, Eq. (22). The second two are extensions of the methods of [8,10,6] to three-dimensional flows. The first (E–E–E) uses all explicit integrations, and the second (I–I–E) uses implicit integration for the first two steps. These are all compared to the current method defined by (20) and the results are shown in Table 4. The first two columns give the L_1 and L_∞ norms of the global volume loss of water in the domain after each use of the advection method. The third column gives the percentage change in total water-volume in the domain after one period of the standing wave. The baseline gives by far the worst performance, with a net mass loss of 12% after only one period. The E–E–E and I–I–E results have $O(1)\%$ error after one period with I–I–E slight better than E–E–E. The current method demonstrates conservation of water-volume after each step and after one period to machine precision.

Note that the results in Table 4 are around two orders of magnitude worse than the corner flow test. There are two reasons for this decreased performance. First, the violent breaking has more extreme stretching than the corner flow. Instead of $\partial u/\partial x = 1$ the local stretching could be as large as $u_{\max}/\Delta x$. For this breaking wave test this value is two orders larger than the corner flow, around 100. The second reason is that the flow is 3D. As detailed in [13], sources of error in these methods approximately cancel in a 2D divergence-free flow. Lagrangian flux calculation (as in the EI–LE scheme of [9]) is required for the cancelation to be exact but partial cancelation still mitigates the error in 2D. In 3D the additional advection step makes this impossible.

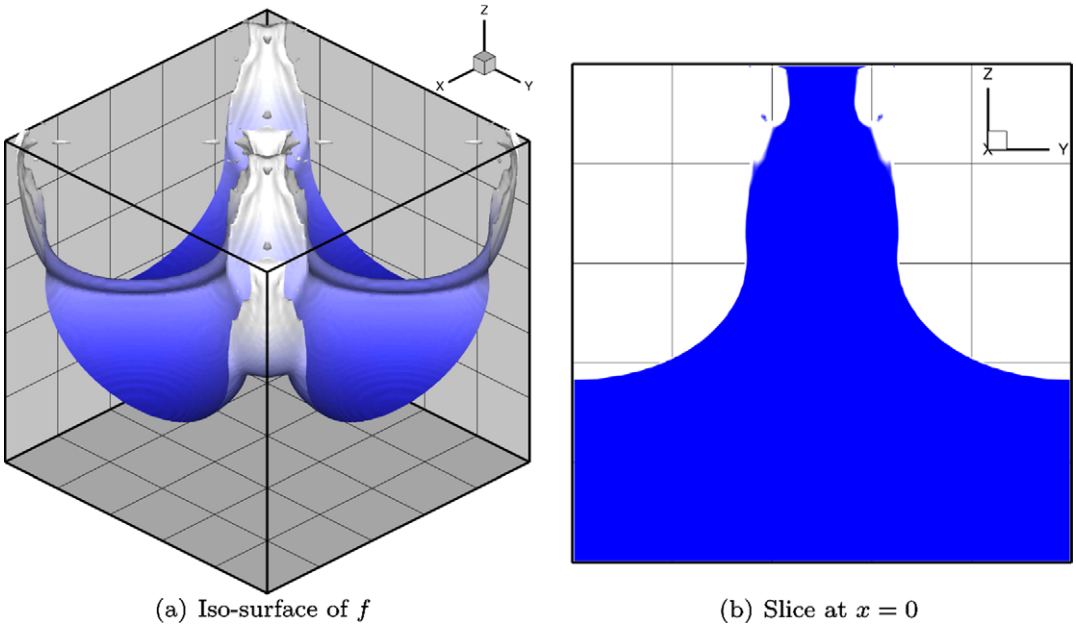
In light of these problems it may seem somewhat surprising that the E–E–E and I–I–E schemes are an order more conservative than the baseline case. Recall that the baseline case has no stretching term and as discussed in Section 2.2 this means that the method overfills during advection and requires excess volume to be 'clipped' at each iteration. This accounts for the drastic mass loss in both 2D and 3D tests. On the other hand, while both the E–E–E and I–I–E methods overfill, *improper inclusion of the dilatation term also allows them to over-empty*. The process could be crudely modeled as a random walk. By allowing steps to be taken in both directions, the 'average distance' from the starting point is decreased but still grows with the iteration number. The only way to avoid this situation is through exact conservation at each step, as achieved by the current method.

Table 3
Error norms for the linear wave test. The E metric is with respect to the linear solution $\eta(x)$, while the E_{ref} metric is with respect to the $N = 200$ reference solution. Both are monotonically convergent and E_{ref} shows second-order convergence with h.

$N = \lambda/\Delta x$	25	50	100	200
E	1.145e−2	4.399e−3	1.115e−3	9.521e−4
E_{ref}	1.162e−2	3.277e−3	5.921e−4	–

Table 4
Global mass loss measurements for a high slope 3D standing wave test case. The first two columns refer to the volume loss at each step, and the final column gives the total change after one wave period.

Algorithm	L_1 norm	L_∞ norm	% Change
Baseline	2.53e−3	1.40e−2	12.63
E–E–E	3.03e−4	2.51e−3	1.42
I–I–E	1.22e−4	1.34e−3	−0.81
Current	1.34e−12	6.83e−12	0.00



(a) Iso-surface of f (b) Slice at $x = 0$

Fig. 5. Visualization of the free-surface for the 3D wave test at $t/T = 0.5$. At this time the nonlinear wave has impacted the top of the computational domain: (a) is the iso-surface of $f = 0.5$ with contours colored by elevation, (b) is a cross section at $x = 0.05$ with contours colored by f.

4. Conclusions

This paper describes the development, verification, and validation of improvements of the VOF method for 2D and 3D flows. A new method for reconstructing the interface location from a local $3 \times 3(\times 3)$ set of volume fractions is presented which requires no inversions, reducing computational costs, and which is second-order for 2D and 3D problems on uniform or clustered grids. In addition, analysis of the advection equation for the color-function c was used to develop an operator-split transport equation for the volume fraction which is exactly conservative in 2D and 3D. The new method is simple to implement and its conservative property makes it ideal for the study of highly energetic free-surface flows such as breaking waves.

Acknowledgments

This work was supported by the Office of Naval Research under Grant N00014-01-1-0124 through Dr. L. Patrick Purtell. The computational resources for this work were provided through a Challenge Project grant from the Depart of Defense High Performance Computing Modernization Office (Project C1V).

Appendix A. Proof of VOF volume conservation

This appendix details the proof of fluid volume conservation for the Volume-of-Fluid (VOF) transport equation presented in Section 2.2 where three basic requirements for a conservative scheme are stated: (1) conservative treatment of the volume flux; (2) zero sum divergence term; and (3) enforcement of the volume fraction constraint $0 \leqslant f \leqslant 1$ at every point in the algorithm. The first two requirements are easily met (Section 2.2) but demonstrating the third is non-trivial, particularly in three spacial dimensions.

The general explicitly integrated transport algorithm is

$$(f - f_0)\frac{\Delta\Omega}{\Delta t} = \sum_{d=1}^{\mathcal{N}} \left(\Delta_d F_d + \int_\Omega c \frac{\partial u_d}{\partial x_d} \, \mathrm{d}v \right) \tag{30}$$

where f is the volume fraction, $\Delta\Omega$ the cell volume, Δt the time step, $\mathcal{N}$ the number of spacial dimensions, F the flux of dark fluid, c the color-function, $\Delta_d F_d$ the net flux in the d direction, and u the velocity. Eq. (30) has conservative flux treatment and a zero sum divergence term. The 1D-stretching integral must be modeled with the volume fraction as the color-function value is not known. Assuming a model of the form

$$\int_\Omega c \frac{\partial u_d}{\partial x_d} \, \mathrm{d}v = g \frac{\partial u_d}{\partial x_d} \Delta\Omega \tag{31}$$

we wish to chose $g = g(f)$ which ensures

$$0 \leqslant f \leqslant 1 \tag{32}$$

at all times in accordance with the third requirement above. Sections 2.2 and 3 demonstrate that setting $g = 0$ (the baseline method) results in a transport equation which cannot conserve fluid volume in a flow with stretching. Those sections also assert and verify that setting $g = c_c$, the color-function at cell center, ensures constraint (32) and produces a volume conservative VOF scheme. This appendix proves this asertion by bounding the flux F in terms of the velocity u and local volume fraction f.

General bounds on the flux of dark fluid F must accommodate general interface orientations and velocity fields. As in Section 2.2, we will scale the velocities and fluxes by $\Delta t/\Delta x$, making them local Courant numbers. To simplify the initial analysis we first consider only the flux through the right face of the cell when transported by a positive velocity. For this case the lower bound on the flux of dark fluid is

$$F \geqslant \max(0, u - \hat{f}) \tag{33}$$

where $\hat{f} \equiv 1 - f$ is the volume of light fluid. Fig. 6(a) illustrates the lower bound, which states that if $u - \hat{f} > 0$ then at least that volume of dark fluid is fluxed. For instance, if $u = 0.4$ and $f = 0.9$ then $\hat{f} = 0.1$ and $F \geqslant 0.3$ regardless of the interface orientation. The upper bound for the same case is

$$F \leqslant \min(u, f) \tag{34}$$

which is illustrated in Fig. 6(b) and states that the flux must not exceed the lesser of f and u. For instance, if $f = 0.1$ and $u = 0.4$ then $F \leqslant 0.1$.

Eqs. (33) and (34) are valid for any interface orientation and positive velocity on the right face. The next task is to extend these bounds to more general velocity conditions. The first case is simple; when the left face velocity is negative the same bounds apply with negatives multiplying the velocities. In the case of a negative right face velocity (u_r) or positive left face velocity (u_l) the flux depends on the unknown volume fraction of the neighboring cells, and the tightest available bounds are

$$0 \leqslant |F| \leqslant |u| \tag{35}$$

Expanding the analysis to consider the net flux gives four general cases: (a) $u_r, u_l > 0$; (b) $u_r > 0, u_l < 0$; (c) $u_r < 0, u_l > 0$, and (d) $u_r, u_l < 0$. The bounds for the net flux $\Delta F \equiv F_r + F_l$ in each case are derived by combining (33)–(35) as appropriate, giving

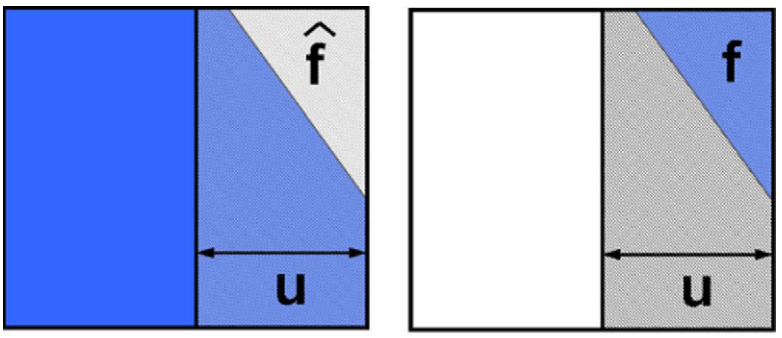


Fig. 6. Illustrations of the lower and upper bound on the flux F of dark fluid for positive velocity components on the right face. The fluxes in these examples are on the limits of bound Eqs. (33) and (34) ($F = u - \hat{f}$ and $F = f$) but the bounds are valid for any f, u and surface orientation.

$$\text{case(a)} \qquad -\min(u_r,f) \leqslant \Delta F \leqslant u_l - \max(0,u_r-\hat{f}) \tag{36a}$$

$$\text{case(b)} \qquad -\min(u_r-u_l,f) \leqslant \Delta F \leqslant -\max(0,u_r-u_l-\hat{f}) \tag{36b}$$

$$\text{case(c)} \qquad 0 \leqslant \Delta F \leqslant u_l - u_r \tag{36c}$$

where case (d) is the same as case (a) swapping u_l and u_r. As the cases are equivalent we do not consider case (d) separately. Note that applying (33) and (34) blindly to case (b) gives

$$-\min(u_r,f)-\min(-u_l,f) \leqslant \Delta F \leqslant -\max(0,u_r-\hat{f})-\max(0,u_l-\hat{f}) \tag{37}$$

but this bound is not tight. Combining it with $-f \leqslant \Delta F \leqslant 0$ results in the bound given earlier.

As stated above these bounds are used to test (30) against constraint (32). In other words, we determine if

$$f_o + \Delta F_{\min} + g(u_r-u_l) \geqslant 0 \quad \text{and} \quad f_o + \Delta F_{\max} + g(u_r-u_l) \leqslant 1 \tag{38}$$

for all u,f and some choice of g. First we choose $g=0$ which give the 'baseline' method. Substitution of the bounds on the net flux from Eqs. (36) into these inequalities and setting $g=0$ results in bounds which must be satisfied for the 'baseline' method to conserve fluid volume,

$$f_0 \geqslant \min(u_r,f_0); \qquad \hat{f}_0 \geqslant u_l - \max(0,u_r-\hat{f}_0) \tag{39a}$$

$$f_0 \geqslant \min(u_r-u_l,f_0); \quad \hat{f}_0 \geqslant -\max(0,u_r-u_l-\hat{f}) \tag{39b}$$

$$f_0 \geqslant 0; \qquad \hat{f}_0 \geqslant u_l - u_r. \tag{39c}$$

For example, the first bound states that if $f_0 \geqslant \min(u_r,f_0)$ then case (a) cannot over-empty the cell. This bound holds for any f_0 and u_r and therefore choosing $g=0$ never results in over-emptying due to case (a). However, the second and last inequalities cannot be enforced because u_r can be either greater or smaller than u_l. For example, in case (a) if $\hat{f}_0=\epsilon,\ u_r=0.1$ and $u_l=0.3$ then the cell is allowed to overfill. A time step restriction will reduce the Courant-scaled velocities but cannot enforce the bounds for arbitrarily small ϵ. Thus, the baseline case without a dilatation term allows overfilling and does not generally conserve fluid volume.

Next we naively choose $g=1$ which gives

$$f_0 \geqslant \min(u_r,f_0) + u_l - u_r; \qquad \hat{f}_0 \geqslant u_r - \max(0,u_r-\hat{f}_0) \tag{40a}$$

$$f_0 \geqslant \min(u_r-u_l,f_0) + u_l - u_r; \quad \hat{f}_0 \geqslant -\max(0,u_r-u_l-\hat{f}) + u_r - u_l \tag{40b}$$

$$f_0 \geqslant u_l - u_r; \qquad \hat{f}_0 \geqslant 0. \tag{40c}$$

Now the first and fifth inequalities are not enforceable. For example, if $f_0=\epsilon,\ u_l=0.3$ and $u_r=-0.1$ the cell becomes less than empty. Thus, choosing $g=1$ allows a volume to 'over-empty' due to excessive modeling of the 1D-stretching term. Similarly, choosing $g=f_0$ results in case (a) bounds given by

$$f_0 \geqslant \min(u_r,f_0) - f_0(u_l-u_r); \quad \hat{f}_0 \geqslant u_r - \max(0,u_r-\hat{f}_0) - \hat{f}_0(u_r-u_l) \tag{41a}$$

neither of which can be guaranteed in a general flow, resulting in both overfilling and over-emptying with this choice of g.

As discussed in Section 2.2, the natural choice for g is the cell centered value of the color-function c as it derives from a volume integral quadrature. In this case

$$g = \begin{cases} 1 & \text{if} \quad f_0 > 1/2 \\ 0 & \text{else} \end{cases} \tag{42}$$

To use this g, we split the cases based on the value of f_0; where $f_0 \leqslant 1/2$ we use the notation f^-, and where $f_0 > 1/2$ we use the notation f^+. From Eq. (42), we see the bounds for f^- and f^+ will be similar to those given for $g=0,1$ above. However, the unenforceable bounds in those cases now include restrictions on the values of f_0^+ and f_0^-

$$f_0^+ \geqslant \frac{1}{2} \geqslant \min(u_r,f_0) + u_l - u_r; \quad \hat{f}_0^- \geqslant \frac{1}{2} \geqslant u_l - \max(0,u_r-\hat{f}_0) \tag{43a}$$

$$f_0^+ \geqslant \frac{1}{2} \geqslant u_l - u_r; \quad \hat{f}_0^- \geqslant \frac{1}{2} \geqslant u_l - u_r. \tag{43b}$$

which are all achievable through a time step restriction.

Effectively, the use of $g=c_c$ activates the dilation term only when there is a sufficient amount of dark fluid in the cell to assure no over-emptying. This result is easily extended to $\mathcal{N}$ spatial dimensions by further restriction of the time step. A Courant restriction of the form $\sum_{d=1}^{\mathcal{N}}|u_d| \leqslant \frac{1}{2}$ for cells on the interface guarantees the bounds (32) are met at all times resulting in exact conservation of volume of both dark and light fluids.

References

[1] S. Muzaferija, M. Peric, Comuputation of free-surface flows using the finite-volume method and moving grids, Numer. Heat Transfer 32 (Part B) (1997) 369–384.
[2] K. Hendrickson, D.K.-P. Yue, Navier–Stokes simulations of unsteady small-scale breaking waves at a coupled air-water interface, in: 26th Symposium on Naval Hydrodynamics, 2006.
[3] M. Sussman, K. Smith, M. Hussaini, M. Ohta, R. Zhi-Wei, A sharp interface method for incompressible two-phase flows, J. Comput. Phys. 221 (2) (2007) 469–505.
[4] C. Hirt, B. Nicholas, Volume of fluid (vof) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201–225.
[5] D.G. Dommermuth, M. Sussman, R.F. Beck, T.T. O'Shea, D.C. Wyatt, K. Olson4, P. MacNeice, The numerical simulation of ship waves using Cartesian grid methods with adaptive mesh refinement, in: 25th Symposium on Naval Hydrodynamics, 2004.
[6] J. Pilliod, E. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, J. Comput. Phys. 199 (2004) 465–502.
[7] R. Scardovelli, S. Zaleski, Analytical relations connecting linear interfaces and volume fractions in rectangular grids, J. Comput. Phys. 164 (2000) 228–237.
[8] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, J. Comput. Phys. 141 (1998) 112–152.
[9] E. Aulisa, S. Manservisi, R. Scardovelli, S. Zaleski, A geometrical area-preserving volume-of-fluid advection method, J. Comput. Phys. 192 (2003) 355–364.
[10] E. Puckett, A. Almgren, J. Bell, D. Marcus, J. Rider, A high-order projection method for tracking fluid interfaces in variable density incompressible flows, J. Comput. Phys. 130 (2) (1997) 269–282.
[11] D.J.E. Harvie, D.F. Fletcher, A new volume of fluid advection algorithm: the defined donating region scheme, Int. J. Numer. Meth. Fluids 35 (2001) 151172.
[12] R. Scardovelli, S. Zaleski, Interface reconstruction with least-square fit and split Eulerian Lagrangian advection, Int. J. Numer. Meth. Fluids 41 (2003) 251274.
[13] G.D. Weymouth, D.G. Dommermuth, K. Hendrickson, D.K.-P. Yue, Advancements in cartesian-grid methods for computational ship hydrodynamics, in: 26th Symposium on Naval Hydrodynamics, 2006.
[14] B.P. Leonard, S. Mokhtari, Beyond first-order upwinding: the ultra-sharp alternative for non-oscillatory steady-state simulation of convection, Int. J. Numer. Meth. Eng. 30 (1990) 729–766.