INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2008; **58**:897–921
Published online 17 March 2008 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1776

A new volume of fluid method in three dimensions—Part I: Multidimensional advection method with face-matched flux polyhedra

J. Hernández[1,*,†], J. López[2], P. Gómez[1], C. Zanzi[1] and F. Faura[2]

[1] *Dept. de Mecánica, ETSII, UNED, E-28040 Madrid, Spain*
[2] *Dept. de Ingeniería de Materiales y Fabricación, ETSII, Universidad Politécnica de Cartagena, E-30202 Cartagena, Spain*

SUMMARY

A multidimensional advection scheme in 3D based on the use of face-matched flux polyhedra to integrate the volume fraction evolution equation is proposed. The algorithm tends to reduce the formation of 'over/undershoots' by alleviating the over/underlapping of flux polyhedra, thus diminishing the need to use local redistribution algorithms. The accuracy and efficiency of the proposed advection algorithm, which are analyzed using different tests with prescribed velocity field, compare well with other multidimensional advection methods proposed recently. The algorithm is also applied, in combination with a Navier–Stokes solver, to reproduce the impact of a water droplet falling through air on a pool of deep water. The interfacial curvature is calculated using a height-function technique with adaptive stencil adjustment, which provides improved accuracy in regions of low grid resolution. The comparison of the numerical results with experimental results shows a good degree of agreement. Copyright © 2008 John Wiley & Sons, Ltd.

Received 2 August 2007; Revised 30 December 2007; Accepted 2 January 2008

KEY WORDS: volume tracking; volume of fluid method; 3D multidimensional advection; PLIC interface reconstruction; interface curvature; drop impact

1. INTRODUCTION

Among the different methods developed to simulate free surface and interfacial flows [1], the volume of fluid (VOF) method is one of the most widely used. In this method, the interface evolution is described using a discrete function, F, whose value in each cell of the computational mesh is the

[*] Correspondence to: J. Hernández, Dept. de Mecánica, ETSII, UNED, C/Juan del Rosal 12, E-28040 Madrid, Spain.
[†] E-mail: jhernandez@ind.uned.es

Contract/grant sponsor: Spanish Ministerio de Educación y Ciencia; contract/grant numbers: DPI2004-08198, PR2005-0296

fraction of the cell volume occupied by the reference fluid. The function F is a discretized version of a function, f, which is continuous except at the interface, where it jumps from zero to one. The several variants of the method proposed during the last three decades can be distinguished by the characteristics of the algorithms used to determine the position of the interface (reconstruction algorithm) and to integrate in time the volume fraction equation (advection algorithm). Detailed reviews can be found in [2, 3].

For the advection step, it is well known that geometric procedures are more accurate than algebraic formulations. Among these procedures, operator-split schemes require two geometric integrations in 2D and therefore two interface reconstructions per time step (one more in 3D). Some recent examples of operator-split schemes implemented in 3D can be found in [4–7]. In the coupled level set/VOF method of Sussman and Puckett [8], the advection in the level set and volume fraction functions is also made using a second-order operator-split scheme. Unsplit time-integration (multidimensional) schemes are more efficient than split methods for calculating fluxes through cell faces and need to reconstruct the interface only once per time step (in 2D or 3D), although their extension to 3D is more complicated. Miller and Colella [9] developed an extension to 3D of the unsplit advection scheme of Pilliod and Puckett [10, 11], and, more recently, Liovic *et al.* [12] proposed a 3D extension of Rider and Kothe's [2] 2D volume tracking unsplit advection scheme.

The aim of this work is to extend the advection algorithm we proposed in [13] to 3D. The new method involves a multidimensional advection scheme based on the use of face-matched flux polyhedra to integrate the volume fraction evolution equation in order to substantially reduce the 'over/underlapping' of flux polyhedra. The use of a set of simple geometric tasks (specifically developed for this work) makes the method efficient, robust and relatively simple to implement in 3D. Although the algorithms have been implemented in a Cartesian grid, assuming that the normal velocity components are stored at the center of the cell faces (MAC grid), they can easily be extended to other types of grid with no loss of generality.

The proposed advection method is described in Section 2, and its coupling with the Navier–Stokes solver is presented in Section 3. The results are compared in Section 4 with those obtained by other authors for the following tests: advection of a spherical fluid body in simple rotational flow, interface deformation in a flow with non-uniform vorticity and the impact of a water droplet on a deep water pool.

2. MULTIDIMENSIONAL ADVECTION

For 3D incompressible flows, the volume fraction evolution equation

$$\frac{\partial f}{\partial t}+\mathbf{v}\cdot\nabla f=0 \tag{1}$$

can be integrated over a given cell, Ω, of volume V_Ω, and the time interval $\Delta t=t^{n+1}-t^n$, to obtain, at each time step,

$$F^{n+1}=F^n-\frac{1}{V_\Omega}\int_{t^n}^{t^{n+1}}\int_\Omega \nabla\cdot(\mathbf{v}f)\,\mathrm{d}\Omega\,\mathrm{d}t \tag{2}$$

DOI: 10.1002/fld

where the divergence constraint $\nabla\cdot\mathbf{v}=0$ has been taken into account. The integral in Equation (2) represents the net VOF advected out of the cell V_{F_T} and will be solved geometrically using the face-matched flux polyhedron advection (FMFPA-3D) method proposed in this work, which is described below.

Before beginning the description of the proposed advection method, it should be mentioned that, before applying the advection step, the interface will be assumed to have been reconstructed in the previous time step using a PLIC method (such as that proposed in the companion paper [14]), in which the interface is reconstructed in each cell as a plane $\mathbf{n}\cdot\mathbf{x}+C=0$, where $\mathbf{n}$ points to the reference fluid. Since the aim of this paper is to assess the proposed advection method, widely used methods, for which many results are available in the literature, will be used to determine the vector normal to the interface, $\mathbf{n}$, and constant C. Thus, unless otherwise stated, $\mathbf{n}$ will be obtained from the gradient of the volume fraction function, ∇F, using the method of Youngs [15] (see also [4, 16]), and the constant C will be determined from the value of F in the cell and enforcement of local volume conservation using the analytical method of Scardovelli and Zaleski [17] (alternatively, an iterative method, such as Brent's method [18], could also be employed).

The proposed advection method to determine the new VOF fraction at $(t^n+\Delta t)$ and every grid cell, F^{n+1}, consists of the following five steps:

1. Calculate the volumetric flux through every face, Γ, of the cell, which can be expressed as

$$q=\int_{\Gamma}\mathbf{v}\cdot\mathbf{n}_{\Gamma}\,\mathrm{d}\Gamma \tag{3}$$

where $\mathbf{n}_{\Gamma}$ is the unit normal vector to the face pointing out of the cell. The integral in Equation (3) is determined from the velocity components at the face center, using a midpoint rule.

2. Determine the volume

$$V_d=\int_{t^n}^{t^{n+1}} q\,\mathrm{d}t \tag{4}$$

crossing every face Γ of the cell during the time interval Δt. In tests with prescribed velocity fields, the integral in Equation (4) is solved using a trapezoidal quadrature rule. When the Navier–Stokes equations are being solved to calculate the velocity field, the integral in Equation (4) is solved using the face-centered velocity at an intermediate time $t^{n+1/2}$.

3. Construct a flux polyhedron of volume V_d at every face of the cell.
4. Determine the VOF advected during the time interval Δt through every face of the cell, V_F, by truncating the corresponding flux polyhedron with the reconstructed interface.
5. Calculate the total net VOF, V_{F_T}, that leaves (or enters) the cell as the sum of the truncated fluid volumes advected through its six faces, $V_{F_T}=\sum_{p=1}^{6}V_{F_p}$, and the new VOF fraction at $(t^n+\Delta t)$ given by Equation (2), which reduces to

$$F^{n+1}=F^n-\frac{V_{F_T}}{V_{\Omega}} \tag{5}$$

where V_{Ω} is the total volume of the cell.

Steps 3 and 4, which are the key and most time consuming steps of the algorithm, are described in detail in Sections 2.1 and 2.2, respectively.

DOI: 10.1002/fld

2.1. Construction of flux polyhedra

The flux region corresponding to a given cell face is delimited by a polyhedron that consists, in the general case, of six faces, one of which obviously coincides with the cell face considered, whereas the other five are determined, as indicated below, from the velocity vectors at the center of the cell edges and from the condition of Equation (4).

The basic idea is to ensure that the flux polyhedra constructed at cell faces with a common edge have a face with a common orientation. This orientation is determined by the interpolated velocity vector at the center of the common edge. The approach is similar to that proposed in [13] for 2D, in which the flux polygons at cell edges with a common vertex were constructed in such a way that each had one edge with a common orientation, thus avoiding overlapping between flux polygons (see Figures 1(a) and (b)). In 3D, the proposed method avoids the over/underlapping between flux polyhedra constructed at cell faces with a common edge (see Figure 1(c)).

In Figure 2(a), the four faces ($\Gamma_2, \Gamma_3, \Gamma_4$ and Γ_5) of the flux polyhedron that have a common edge with the cell face considered, Γ_1, are constructed parallel to the velocity vectors normal to their respective common edge centers (points a, b, c and d, respectively). For example, the plane that contains face Γ_2 is given by $\mathbf{n}_{\Gamma_2} \cdot \mathbf{x} + C_{\Gamma_2} = 0$, where the normal unit vector $\mathbf{n}_{\Gamma_2} = (\mathbf{x}_2 - \mathbf{x}_1) \times \mathbf{v}_a / |(\mathbf{x}_2 - \mathbf{x}_1) \times \mathbf{v}_a|$ and $C_{\Gamma_2} = -\mathbf{n}_{\Gamma_2} \cdot \mathbf{x}_1$.

It should be pointed out that over/underlapping may still occur between flux polyhedra constructed at cell faces with only a common vertex, as shown in Figure 1(d). Overlapping could be totally avoided if flux polyhedra at cell faces with a common vertex were constructed with an edge with a common orientation (given by the direction of the velocity vector at the common vertex), although at the cost of an additional degree of complexity in the integration (note that the faces of the flux region would, in general, be non-planar surfaces). Such an approach will be considered in future work.

The procedure used to determine the orientation and location of the remaining face requires a more detailed explanation. This is face Γ_6 in Figure 2(c), which is defined by plane $\mathbf{n}_{\Gamma_6} \cdot \mathbf{x} + C_{\Gamma_6} = 0$.

2.1.1. Analytical determination of the orientation of face Γ_6.

Face Γ_6 is constructed parallel to the two segments that join points $a' - c'$ and $b' - d'$ of Figure 2(b). The location of each of these points, ℓ', is defined by the position vector $\mathbf{x}_{\ell'} = \mathbf{x}_\ell - \mathbf{v}_\ell \Delta t$, where $\mathbf{x}_\ell$ is the position vector of the edge center ℓ of face Γ_1 (points a, b, c and d in Figure 2(b)) and $\mathbf{v}_\ell$ is the corresponding velocity vector normal to the edge. In the case of Figure 2(b), for example, the orientation of face Γ_6 is therefore defined by a normal vector perpendicular to both segments $a' - c'$ and $b' - d'$, which can be obtained analytically as

$$\mathbf{n}_{\Gamma_6} = \text{sign}(q) \frac{(\mathbf{x}_{a'} - \mathbf{x}_{c'}) \times (\mathbf{x}_{d'} - \mathbf{x}_{b'})}{|(\mathbf{x}_{a'} - \mathbf{x}_{c'}) \times (\mathbf{x}_{d'} - \mathbf{x}_{b'})|}$$

Note that the sign of the volumetric flux q determines on which side of face Γ_1 the flux polyhedron must be constructed.

For the MAC grid used in this work, each component of $\mathbf{v}_\ell$ normal to the edge is obtained from an average of the corresponding values at the center of the two faces sharing the edge that are perpendicular to the component considered. For the edge center $(i + 1/2, j + 1/2, k)$ of the cell

DOI: 10.1002/fld

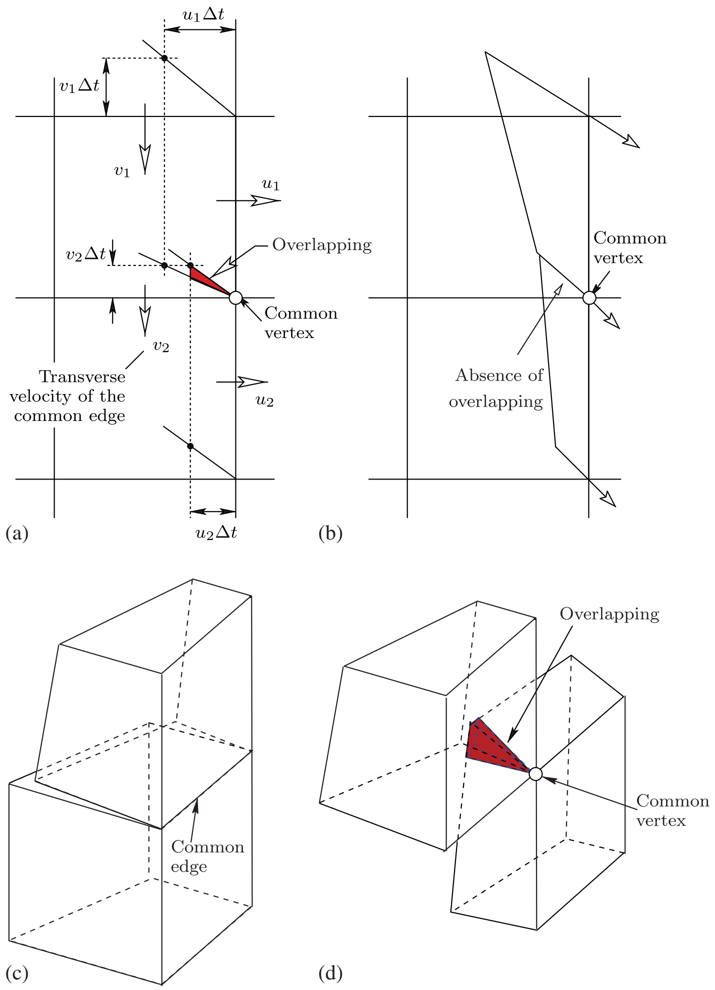

Figure 1. Flux regions constructed at two cell faces with a common edge: (a) Rider and Kothe [2] method; (b) EMFPA method for 2D [13]; (c) FMFPA-3D proposed in this work; and (d) overlapping between flux regions constructed at two cell faces with only one common vertex.

centered at (i, j, k), for example,

$$u_{i+1/2,j+1/2,k}=(u_{i+1/2,j,k}+u_{i+1/2,j+1,k})/2$$

$$v_{i+1/2,j+1/2,k}=(v_{i,j+1/2,k}+v_{i+1,j+1/2,k})/2$$

It was seen that by also considering the velocity component parallel to the edge, the overall improvement in accuracy of the proposed advection method is practically imperceptible, whereas the central processing unit (CPU) time increases by around 5%.

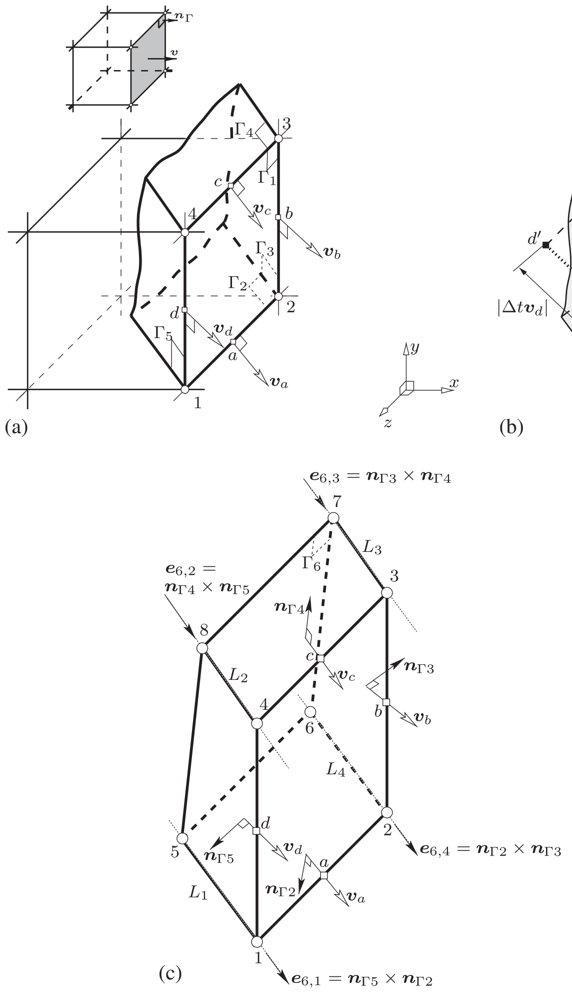

Figure 2. Face-matched flux polyhedron: (a) construction of the four faces that have a common edge with the cell face considered; (b) determination of the orientation of the remaining polyhedron face; and (c) final flux polyhedron after imposing the volume conservation constraint of Equation (4).

2.1.2. Analytical determination of the position of face Γ_6. The position of face Γ_6, given by the constant C_{Γ_6}, is obtained by imposing the volume conservation constraint of Equation (4) and taking into account that the volume of a flux polyhedron of P faces with M_p vertices in each face p can be expressed as [19]

$$V_d = \frac{1}{6}\sum_{p=1}^{P}\left[(\mathbf{n}_{\Gamma_p}\cdot\mathbf{x}_{p,1})\mathbf{n}_{\Gamma_p}\cdot\sum_{m=1}^{M_p}(\mathbf{x}_{p,m}\times\mathbf{x}_{p,m+1})\right] \tag{6}$$

(subscript $m+1$ must be replaced by 1 for $m=M_p$). Here, $\mathbf{n}_{\Gamma_p}$ is the unit length normal to face Γ_p (pointing out of the polyhedron) and $\mathbf{x}_{p,m}$ is the position vector of the counterclockwise (viewed from outside the polyhedron) ordered vertex m of face Γ_p (i.e. an ordered list of vertices for face Γ_2 of the flux polyhedron of Figure 2(c) is the sequence of vertices 2, 1, 5 and 6).

Note that the unknowns in Equation (6) are the position vectors of the four vertices of face Γ_6. Each vertex $\mathbf{x}_{6,m}$ is given by the intersection between the plane $\mathbf{n}_{\Gamma_6}\cdot\mathbf{x}+C_{\Gamma_6}=0$ and the line L_m (parallel to the unit vector, $\mathbf{e}_{6,m}$, determined from the unit length vectors normal to polyhedron faces, as indicated in Figure 2(c)) of intersection between the planes containing the other two polyhedron faces passing through vertex m. For convenience, it can be expressed as a function of the constant C_{Γ_6} as

$$\mathbf{x}_{6,m}=\mathbf{x}^0_{6,m}+\beta_{6,m}C_{\Gamma_6}\mathbf{e}_{6,m} \tag{7}$$

where $\beta_{6,m}=-1/(\mathbf{n}_{\Gamma_6}\cdot\mathbf{e}_{6,m})$ and $\mathbf{x}^0_{6,m}$ is the position vector of the intersection point between line L_m and a plane defined by $\mathbf{n}_{\Gamma_6}\cdot\mathbf{x}=0$ (parallel to face Γ_6 and passing through the origin of the coordinate system), which may be expressed as

$$\mathbf{x}^0_{6,m}=\mathbf{x}_{1,m'}+\beta_{6,m}(\mathbf{n}_{\Gamma_6}\cdot\mathbf{x}_{1,m'})\mathbf{e}_{6,m} \tag{8}$$

where $\mathbf{x}_{1,m'}$ is the position vector of the vertex of the considered cell face (Γ_1 in Figure 2) located on L_m.

Introducing Equation (7) into Equation (6) with $P=6$ and $M_p=4$, equating Equation (6) to Equation (4) and rearranging terms yield the following cubic polynomial function for C_{Γ_6}:

$$\begin{aligned}(\mathbf{Q}_{\Gamma_6}\cdot\mathbf{n}_{\Gamma_6})C^3_{\Gamma_6}&+\left[\mathbf{L}_{\Gamma_6}\cdot\mathbf{n}_{\Gamma_6}+\sum_{p=1}^{5}\mathbf{Q}_{\Gamma_p}\cdot\mathbf{n}_{\Gamma_p}C_{\Gamma_p}\right]C^2_{\Gamma_6}\\&+\left[\mathbf{K}_{\Gamma_6}\cdot\mathbf{n}_{\Gamma_6}+\sum_{p=1}^{5}\mathbf{L}_{\Gamma_p}\cdot\mathbf{n}_{\Gamma_p}C_{\Gamma_p}\right]C_{\Gamma_6}+\sum_{p=1}^{5}\mathbf{K}_{\Gamma_p}\cdot\mathbf{n}_{\Gamma_p}C_{\Gamma_p}+6V_d=0\end{aligned} \tag{9}$$

from which the constant C_{Γ_6} can be obtained analytically [18]. In this equation,

$$\begin{aligned}C_{\Gamma_p}&=-\mathbf{n}_{\Gamma_p}\cdot\mathbf{x}_{p,1}\\ \mathbf{K}_{\Gamma_p}&=\sum_{m=1}^{4}\mathbf{x}^0_{p,m}\times\mathbf{x}^0_{p,m+1}\\ \mathbf{L}_{\Gamma_p}&=\sum_{m=1}^{4}(\mathbf{x}^0_{p,m-1}\times\mathbf{e}_{p,m}+\mathbf{e}_{p,m}\times\mathbf{x}^0_{p,m+1})\beta_{p,m}\\ \mathbf{Q}_{\Gamma_p}&=\sum_{m=1}^{4}(\mathbf{e}_{p,m}\times\mathbf{e}_{p,m+1})\beta_{p,m}\beta_{p,m+1}\end{aligned} \tag{10}$$

(subscripts $m+1$ and $m-1$ must be replaced by 1 and 4 for $m=4$ and $=1$, respectively). For vertices that are not placed on face Γ_6, $\mathbf{x}^0_{p,m}=\mathbf{x}_{p,m}$, $\mathbf{e}_{p,m}=(0,0,0)$ and $\beta_{p,m}=0$.

2.2. Determination of the advected VOF

The VOF advected through each face of a given cell, V_F, which is taken to be positive when the fluid leaves the cell, will depend on the shape of the corresponding flux polyhedron and the location of the reconstructed interface. A set of simple geometric tools, described below in Section 2.2.1, is used to compute the volume of the flux polyhedron truncated by cell faces and by the interface. Firstly, the volumes occupied by the polyhedron in different cells are determined (note that, for the type of polyhedron constructed, donating regions from several cells need to be considered; see Figure 3(a)). Then, these volumes are cut by the reconstructed interface to obtain the corresponding truncated fluid volumes (Figure 3(b) shows, as an example, the truncation of the volume inside the considered cell), which are added to obtain the total truncated fluid volume, V_F, advected through the cell face considered.

2.2.1. Volume truncation. The successive plane–polyhedron intersection operations needed to obtain the VOF advected through a given cell face are performed as follows. The signed distance,

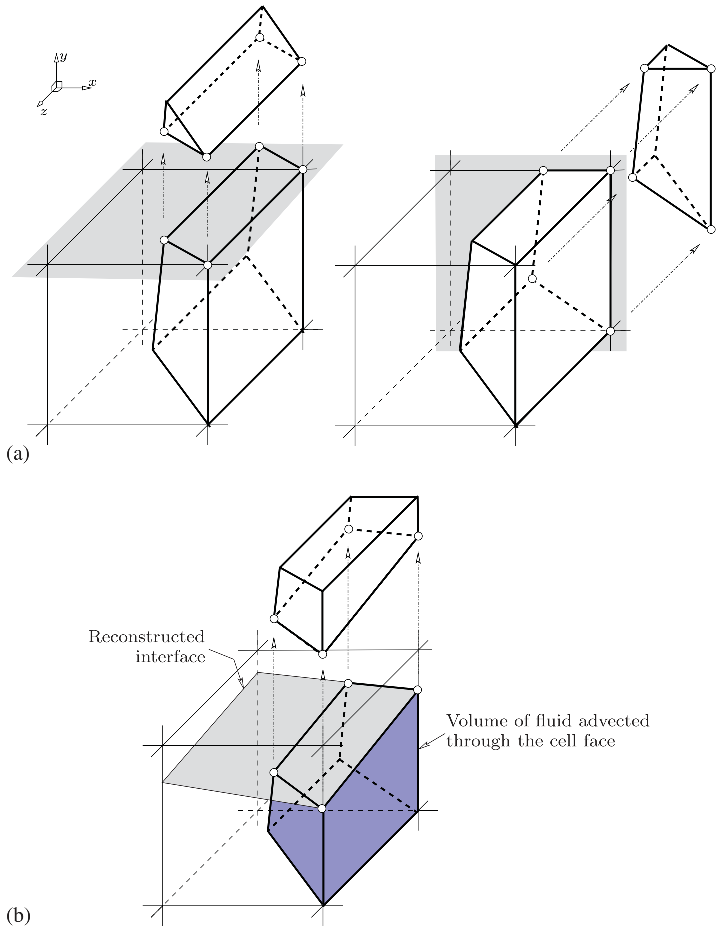

Figure 3. Computation of the fluid volume advected through one face of a given cell: (a) truncation of the flux polyhedron with the cell faces and (b) truncation of the cut flux polyhedron inside the considered cell with the reconstructed interface plane.

ϕ_m, from each flux polyhedron vertex, $\mathbf{x}_m$, to a plane defined by $\mathbf{n}\cdot\mathbf{x}+C=0$ is computed as $\phi_m=\mathbf{n}\cdot\mathbf{x}_m+C$. Note that $\mathbf{n}$ is assumed to point to the reference fluid or to the resulting truncated volume, depending on whether the cutting surface is the interface or a cell face. Also note that if the normal vector, $\mathbf{n}$, points to the vertex, the sign of ϕ will be positive. Whether or not a plane–polyhedron intersection exists can easily be determined by comparing the relative signs of the ϕ_m values of the polyhedron vertices. The truncated polyhedron will be defined by the vertices of the flux polyhedron which have a positive value for the signed distance function and by the intersection points between the plane and the edges of the polyhedron. If, for example, one of these edges is defined by the two adjacent vertices $\mathbf{x}_1$ and $\mathbf{x}_2$ (ϕ_1 and ϕ_2 must have opposite signs), the resulting position vector of the intersection point can be expressed as

$$\mathbf{x}_I=\mathbf{x}_1-\frac{\phi_1}{\phi_2-\phi_1}(\mathbf{x}_2-\mathbf{x}_1)$$

The volume of the truncated polyhedron can finally be obtained from Equation (6).

2.2.2. Special case in which the flux polyhedron volume is lower than V_d. In certain circumstances, the maximum volume of the flux polyhedron for face Γ_1 that can be constructed (determined by the intersection between faces Γ_3 and Γ_5 or Γ_2 and Γ_4; see the example of Figure 4) is lower than the volume V_d calculated from Equation (4); hence, the constraint given by Equation (6) cannot be imposed to construct the polyhedron. In such cases, a flux polyhedron of five faces with a volume lower than V_d is constructed. The VOF advected through the considered cell face is then taken to be equal to that resulting from the truncation of the flux polyhedron by the interface, corrected by the flux factor proposed by López *et al.* [13] for similar situations, which is equal to the ratio between the volume given by Equation (4) and that of the flux polyhedron, V_d':

$$\text{Flux factor}=\frac{V_d}{V_d'} \tag{11}$$

2.2.3. Use of a local redistribution algorithm. The discretization errors introduced in the resolution of Equation (2), along with the mentioned over/underlapping that may occur between polyhedra

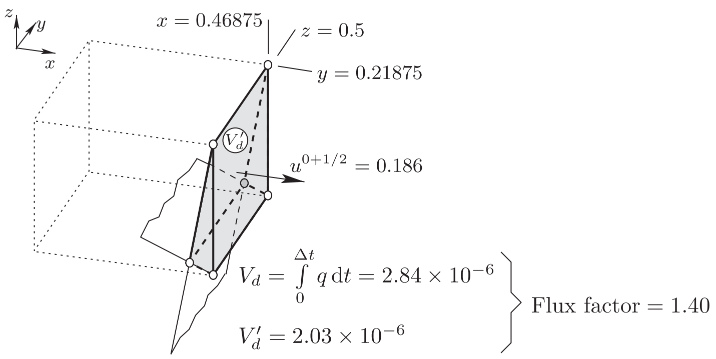
z
y
x
$x = 0.46875$
$z = 0.5$
$y = 0.21875$
V_d'
$u^{0+1/2} = 0.186$
$V_d = \int_0^{\Delta t} q\,\mathrm{d}t = 2.84 \times 10^{-6}$
$V_d' = 2.03 \times 10^{-6}$
Flux factor = 1.40

Figure 4. Flux polyhedron adjustment at the time step from $t=0$ to $\Delta t=\frac{1}{64}$, in the cell with coordinates given in the figure, in the deformation test of LeVeque [20] for a 32^3 grid.

with only one vertex in common, and despite the significant reduction achieved with the use of the FMFPA-3D method proposed in this work, may still cause small overshoots (values of $F>1$) or undershoots ($F<0$) in the volume fraction distribution. When this occurs, a local redistribution algorithm similar to that proposed by Harvie and Fletcher [21] can be applied.

3. COUPLING WITH THE NAVIER-STOKES EQUATIONS

The new VOF method described above was solved in combination with the incompressible Navier–Stokes equations to obtain some of the results presented in the next section. For this purpose, the proposed algorithms were implemented in a code that solves the Navier–Stokes equations on both sides of the interface. The code is an extension to 3D of that proposed by Gómez *et al.* [22] (see also Reference [23]). The projection step has been slightly modified to incorporate a continuous surface tension model based on the balanced-force algorithm proposed by Francois *et al.* [24] and to obtain a discrete divergence-free velocity field at face centers. In this continuous surface tension model, the surface force is calculated at the cell faces as $\sigma\kappa_{\mathrm{f}}(\nabla F)_{\mathrm{f}}$, where σ is the surface tension coefficient, κ_{f} is the interface curvature at the cell face and $(\nabla F)_{\mathrm{f}}$ is the face gradient of the volume fraction (note that the surface force is non-zero only on faces where the face gradient of F is non-zero). The curvature is first computed at all interfacial cell centers using the height-function technique described in Section 3.1 and then interpolated at cell faces. For example, for the cell face $(i+1/2, j, k)$, the following expression is used for interpolation:

$$\kappa_{i+1/2,j,k}=\begin{cases}\frac{1}{2}(\kappa_{i,j,k}+\kappa_{i+1,j,k}) & \text{if } \kappa \text{ is defined in cells } (i,j,k) \text{ and } (i+1,j,k)\\ \kappa_{i,j,k} & \text{if } \kappa \text{ is only defined in cell } (i,j,k) \text{ or}\\ \kappa_{i+1,j,k} & \text{if } \kappa \text{ is only defined in cell } (i+1,j,k)\end{cases}$$

The face-centered velocity at the intermediate time $t^{n+1/2}$, used to construct the convective term of the momentum equation and the flux polyhedron in the proposed FMFPA-3D method, is obtained as a linear combination of the values at the two previous time steps, t^n and t^{n-1}, given by $\mathbf{v}^{n+1/2}=\frac{3}{2}\mathbf{v}^n-\frac{1}{2}\mathbf{v}^{n-1}$ (note that this extrapolation does not introduce any additional non-solenoidal component in the velocity field). The density and viscosity in each computational cell are calculated from

$$\rho=\rho_1+(\rho_2-\rho_1)F, \quad \mu=\mu_1+(\mu_2-\mu_1)F \tag{12}$$

where subscripts 1 and 2 denote fluids at both sides of the interface and F is the volume fraction in the considered cell. In order to take into account the divergence error present in the velocity field, instead of Equation (1), which is used for advection of the volume fraction distribution in tests with prescribed velocity field, we solve

$$\frac{\partial f}{\partial t}+\nabla\cdot(\mathbf{v}f)-f\nabla\cdot\mathbf{v}=0 \tag{13}$$

Rider and Kothe [2] showed that by taking into account the last term of Equation (13) a more precise local and global conservation of the volume fraction can be achieved, along with a reduction (although not complete) in the appearance of overshoots or undershoots.

DOI: 10.1002/fld

3.1. Calculation of the interfacial curvature

In this section, we will briefly describe the method used to calculate the interfacial curvature in the droplet impact problem considered in Section 4.2, which is an especially important issue when using VOF methods. The interface curvature is calculated from the volume fraction distribution at t^{n+1} using the height-function technique [3, 25–28], whose accuracy is increased in this work by introducing the improvements described below.

Let us assume that, at a given interfacial cell, (i,j,k), the largest component of the interface normal vector is aligned with the z-axis (i.e. $|n_z|>|n_x|$ and $|n_z|>|n_y|$). Then, a local distribution of a height function, H, is calculated using a 3×3 stencil on the xy plane centered at cell (i,j,k) (this stencil, which will be used to calculate the interface curvature, will be referred to hereafter as the κ-stencil),

$$H_{r,s}=\sum_{t=-3}^{3} F_{i+r,j+s,k+t}\Delta z_{k+t} \quad \text{for } r=-1,0,1 \quad \text{and} \quad s=-1,0,1 \tag{14}$$

where Δz is the cell height. Note that we need values of F on a $3\times 3\times 7$ stencil surrounding the cell (i,j,k) (which will be referred to hereafter as the HF-stencil).

The curvature of the interface at (i,j,k) is determined as

$$\kappa_{i,j,k}=\frac{H_{xx}+H_{yy}+H_{xx}H_y^2+H_{yy}H_x^2-2H_{xy}H_xH_y}{(1+H_x^2+H_y^2)^{3/2}} \tag{15}$$

where the derivatives are obtained from the distribution of H within the κ-stencil centered at (i,j,k), using standard second-order finite-difference approximations. The interface curvature at cell (i,j,k) is determined using Equation (15) only if the following condition is satisfied: $h_{k-1/2}<H_{0,0}<h_{k+1/2}$, where $h_{k-1/2}$ and $h_{k+1/2}$ are the heights of the lower and upper cell faces over the bottom of the HF-stencil (lower face of cell $(i,j,k-3)$), respectively (see Figure 5(a)). Otherwise, the curvature is made equal to that calculated in the contiguous cell with the same subindices i,j where the mentioned condition is satisfied. For example, at cell $(i,j,k+1)$ in Figure 5(a), $h_{k+1/2}>H_{0,0}$; hence, $\kappa_{i,j,k+1}$ is taken equal to $\kappa_{i,j,k}$.

In order to increase the accuracy of the model in regions with low grid resolution (thin fluid structures or high curvature regions), we introduce the following two improvements in the calculation of the height function, H. First, the size in the height-function direction of the HF-stencil is adapted. For example, if n_z is the largest component of the interface normal vector in cell (i,j,k), the length of the stencil is adjusted within a range from $3\times 3\times 1$ to $3\times 3\times 7$ cells. To this end, the number of cells of the stencil above cell (i,j,k) in the positive z-direction, t_{up}, is gradually increased up to a maximum of 3 cells as long as the following three conditions are satisfied:

$$\text{sign}(n_z)(\alpha_{k+t_{\text{up}}}-\alpha_{k+t_{\text{up}}-1})>0, \quad \alpha_{k+t_{\text{up}}}\neq 0 \quad \text{and} \quad \alpha_{k+t_{\text{up}}}\neq 9(\alpha_{k+t_{\text{up}}}\neq 3 \text{ in 2D})$$

where $\alpha_k=\sum_{r=-1}^{1}\sum_{s=-1}^{1}F_{i+r,j+s,k}$. Note that the above three conditions guarantee that the sum, α_k, of volume fractions along the HF-stencil follows an overall monotonic variation along the height-function direction. Similarly, the number of cells of the stencil below cell (i,j,k), t_{low}, is adjusted within a range from $t_{\text{low}}=0$ to 3.

Figure 5(a) shows an example in 2D in which the HF-stencil is adjusted to only 3 cells along the height-function direction (note that $\alpha_{k+2}=0$ and $\alpha_{k-2}=3$, and thus $t_{\text{up}}=1$ and $t_{\text{low}}=1$). It can be seen that this yields the same curvature result as when a HF-stencil of 7 cells along the

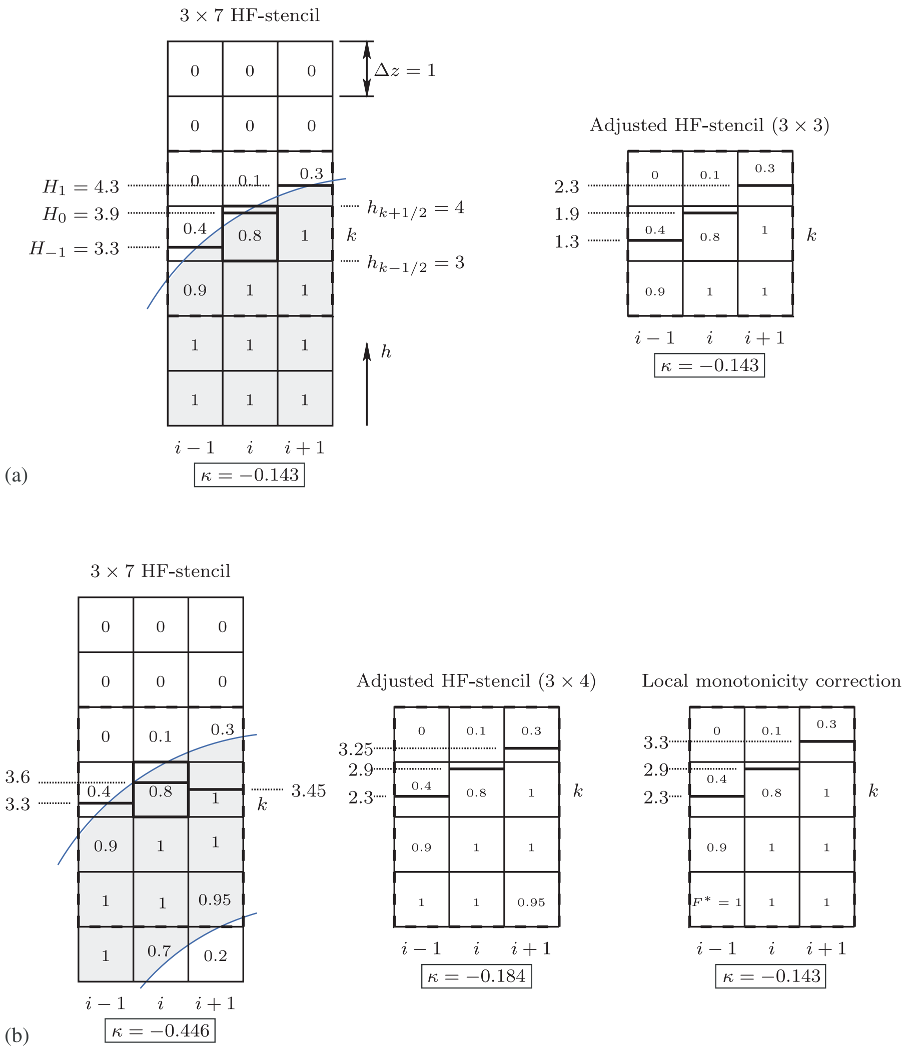

Figure 5. Height function calculation in 2D when the stencil is crossed by (a) one interface and (b) two interfaces.

height-function direction is used, while reducing the number of cells involved in the curvature calculation and, consequently, the CPU time consumed. Figure 5(b) shows a similar example of a case with two interfaces crossing the HF-stencil, in which the upper interface is exactly the same as that in Figure 5(a), and the lower one may affect the accuracy of the calculation of the curvature of the upper one. Note that, after the stencil adjustment (see the central picture of Figure 5(b)), a considerable improvement in the curvature calculation ($\kappa = -0.184$ instead of -0.446, the value of which is obtained using the standard 3×7 HF-stencil) is achieved. Note that the stencil adjustment based on the above overall monotonic conditions ($\alpha_{k-3} < \alpha_{k-2}$ and $\alpha_{k+2} = 0$, and thus $t_{\text{low}} = 2$ and

$t_{\mathrm{up}}=1$) reduces the distortion effects introduced by the presence of an additional interface in the stencil.

An additional improvement consists of introducing a modified distribution of the volume fraction, F^*, used to calculate the height function, which is forced to follow a local monotonic variation along the height direction. For example, if $\mathrm{sign}(n_z)(F_{i+r,j+s,k+t}-F_{i+r,j+s,k+t+1})>0$ for $t=-1,\ldots,-t_{\mathrm{low}}$ or $\mathrm{sign}(n_z)(F_{i+r,j+s,k+t}-F_{i+r,j+s,k+t-1})<0$ for $t=1,\ldots,t_{\mathrm{up}}$, then $F^*_{i+r,j+s,k+t}=\frac{1}{2}(1+\mathrm{sign}(t)\mathrm{sign}(n_z))$; otherwise, $F^*_{i+r,j+s,k+t}=F_{i+r,j+s,k+t}$. Hence, the height function in the κ-stencil is constructed as

$$H_{r,s}=\sum_{t=-t_{\mathrm{low}}}^{t_{\mathrm{up}}} F^*_{i+r,j+s,k+t}\Delta z_{k+t} \quad \text{for } r=-1,0,1 \quad \text{and} \quad s=-1,0,1 \tag{16}$$

Note that, in the case of Figure 5(b), with this additional adjustment $F^*_{i+1,k-2}$ is necessarily unity, producing the same curvature as in the case of Figure 5(a) (see the picture at the right of Figure 5(b)). A similar monotonic correction in the volume fraction was previously suggested by Malik *et al.* [29] for a 2D problem using a fixed 3×7 HF-stencil.

3.1.1. Assessment of the curvature model. Table I shows the maximum $E_{L\infty}=\max|\kappa-\kappa_{\mathrm{exact}}|$ and average $E_{L1}=1/N\sum_{i=1}^{N}|\kappa_i-\kappa_{i\mathrm{exact}}|$ curvature errors for a sphere with diameter equal to 4 and an ellipsoid with axes of length 4.6, 4 and 3.4 in a domain of size $8\times 8\times 8$. The volume fractions are initialized as in [24, 26], using a local mesh refinement technique. The errors were calculated over 100 different situations generated by randomly changing the center of the fluid bodies. The results show that the method reaches a second-order accuracy for both L_∞ and L_1 error norms. On the other hand, the ability of the proposed method to calculate the curvature of the interface in regions of thin fluid filaments (Figure 6(a)) can be clearly seen from Figure 6(b), which shows results corresponding to the problem of a hollow sphere with diameter equal to 4, centered in a domain of 8^3, obtained with a grid resolution of 40^3 cells, in which the fluid has a thickness, e, of only twice the cell size ($2h$). Figure 6(c) represents the maximum $E_{L\infty}$ (left picture) and average E_{L1} (right picture) curvature errors for the outer interface obtained using the standard $3\times 3\times 7$ (dashed lines) and proposed adjusted (continuous lines) HF-stencils, as a function of the ratio between the fluid thickness, e, and the cell size, h. Note that the height-function technique with the proposed adjusted stencil makes it possible to obtain highly accurate results even with a fluid thickness of only twice the cell size, whereas the standard $3\times 3\times 7$ stencil requires the fluid to have a thickness of about four times the cell size at least to produce accurate curvature results.

Table I. Maximum and average curvature error for a sphere with diameter equal to 4 and an ellipsoid with axes of length 4.6, 4 and 3.4 in a domain of size $8\times 8\times 8$.

	Sphere				Ellipsoid			
Grid size	$E_{L\infty}$	$\mathcal{O}$	E_{L1}	$\mathcal{O}$	$E_{L\infty}$	$\mathcal{O}$	E_{L1}	$\mathcal{O}$
20^3	3.34×10^{-1}		2.65×10^{-2}		3.96×10^{-1}		3.31×10^{-2}	
		2.1		2.4		1.6		2.4
40^3	7.81×10^{-2}		5.02×10^{-3}		1.31×10^{-1}		6.49×10^{-3}	
		2.6		2.0		1.8		2.0
80^3	1.31×10^{-2}		1.23×10^{-3}		3.71×10^{-2}		1.61×10^{-3}	

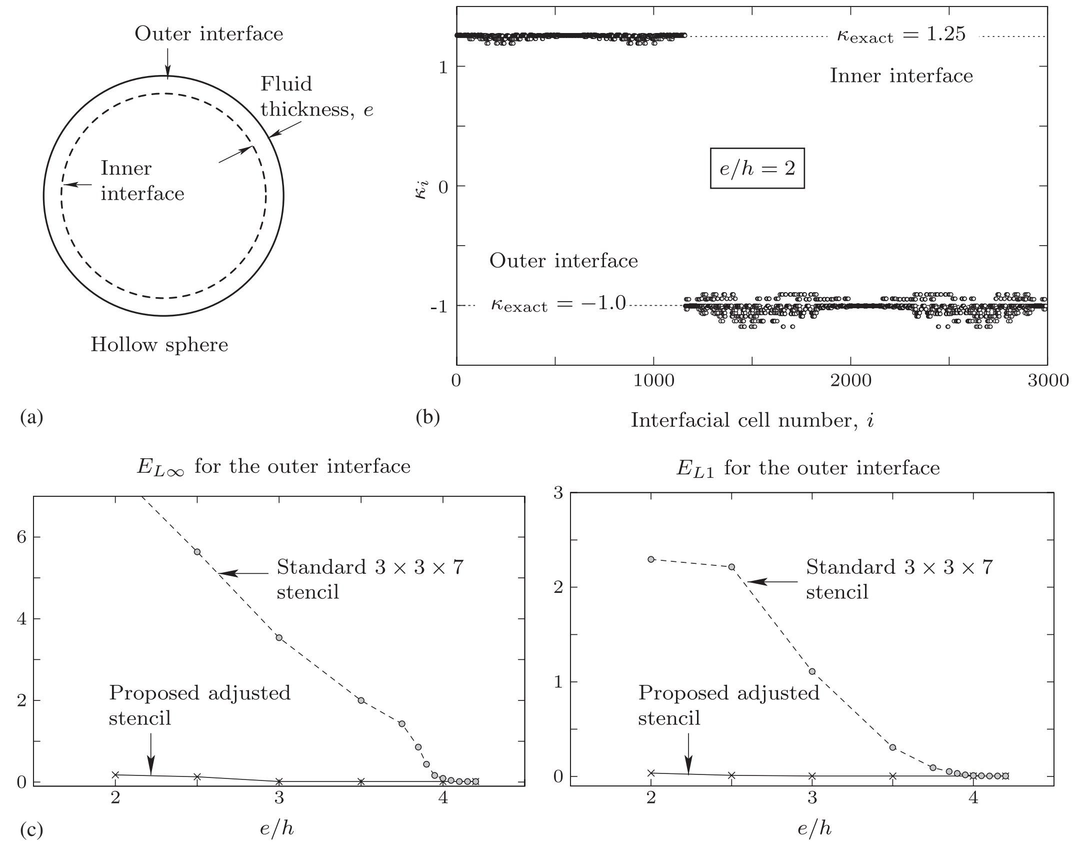

Figure 6. Hollow sphere with outer diameter equal to 4, centered in a domain of 8^3: (a) schematic representation of the problem; (b) curvature results obtained for a case with a fluid thickness equal to $2h$, using a grid resolution of 40^3 cells; and (c) maximum $E_{L\infty}$ and average E_{L1} curvature errors for the outer interface obtained using the standard $3\times3\times7$ stencil and the proposed adjusted stencil, as a function of e/h.

In order to assess the behavior of the curvature model in combination with the Navier–Stokes solver, the equilibrium of an inviscid static spherical drop without gravity, with a radius $R=2$ and located at the center of a domain of size 8^3, is also considered. The densities inside and outside the drop are, respectively, $\rho_1=1$ and $\rho_2=0.1$, and the surface tension coefficient σ is taken to be 73. Since the gravity and other external forces are absent and the flow solver exactly balances the pressure gradient and surface tension force [24], any velocity field produced is a non-physical numerical artifact due to the interfacial curvature errors. To measure the spurious currents, which are estimated by the maximum velocity in the computational domain, $|\mathbf{u}_{\max}|$, this test is computed using different grids and a time step of 0.001. The initial pressure and velocity are set to zero. Figure 7 shows $|\mathbf{u}_{\max}|$ after 1 and 50 time steps. The results of Francois *et al.* [24] and Lörstad *et al.* [25], also obtained with a height-function technique for the curvature calculation, are included in the figure. Note that our results and those obtained by Francois *et al.* [24] are very close because the same scheme was used to incorporate the surface tension force and because the grid used was sufficiently fine to allow the curvature to be accurately calculated with a standard $3\times3\times7$

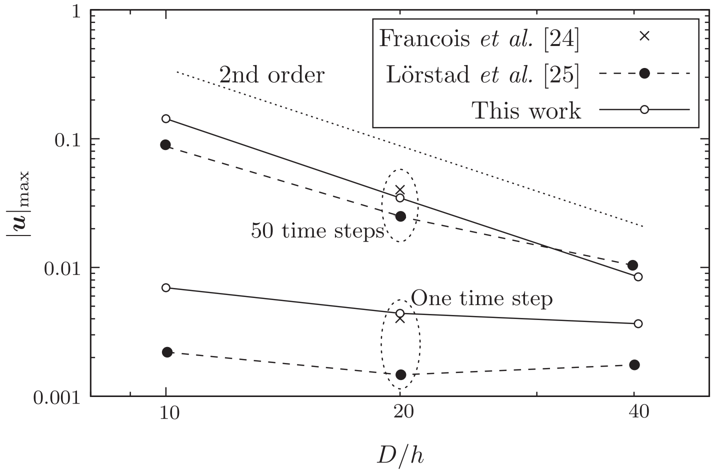

Figure 7. Results of the maximum velocity $|\mathbf{u}|_{\max}$ after 1 and 50 time steps for the inviscid static drop in equilibrium with different grid sizes and $\Delta t = 0.001$. Comparison with the results of Lörstad *et al.* [25] and Francois *et al.* [24].

HF-stencil. Also note that, after 50 time steps, the spurious currents decrease as the grid refinement increases with second-order convergence. For the first time step, the maximum velocity obtained by Lörstad *et al.* [25] is lower, which may be due, in part, to the use of an interpolation kernel to smooth the curvature. However, it is clear that the benefit of curvature smoothing is not evident after 50 time steps.

4. RESULTS AND DISCUSSION

In this section, the accuracy and efficiency of the proposed advection method are analyzed using different tests. The first test corresponds to the advection of a spherical fluid body in simple rotational flow. Then, the ability of the proposed method to handle interfaces that undergo large deformation is analyzed in a flow with non-uniform vorticity. Finally, some results of the impact of water drops on a deep water pool are presented.

4.1. Tests with prescribed velocity field

In order to assess the accuracy and efficiency of the proposed method, the results obtained with it are compared with those obtained with our implementation in 3D of the method proposed by Rider and Kothe [2] for 2D (we will refer hereafter to our extended version as the RK-3D method). In general, the correction of the flux regions at the cell edges proposed in [2], which was made using a donored transverse velocity vector (see Figure 1(a)), causes the faces of the flux polyhedra constructed at cell faces with a common edge to have different orientations in the presence of spatially varying velocity fields. This produces over/underlapping of the flux polyhedra, which tends to increase the formation of overshoots or undershoots and makes the use of a local redistribution algorithm more necessary than with the proposed method. It should be noted that the correction proposed in [2] does not necessarily enforce local volume conservation in

DOI: 10.1002/fld

a divergence-free velocity field, whereas in our extension to 3D of the Rider and Kothe [2] method the polyhedron face opposite the corresponding cell face is appropriately displaced to enforce the volume conservation constraint of Equation (4). All test cases were run on a workstation with dual 2 GHz Intel T7200 processors.

4.1.1. Rotation test. The first test involves the rotation of a sphere of fluid of radius 0.15, initially centered at $(0.5, 0.75, 0.5)$, around an axis parallel to the z-axis and centered in a domain of size $1\times 1\times 1$, in a uniform vorticity field. A Courant–Friedrichs–Lewy (CFL) number based on the maximum velocity component in the domain is approximately equal to 0.5 was used. The error is estimated with an L_1 error norm defined as

$$E=\sum_{i,j,k} V_\Omega^{(i,j,k)}|F^{(i,j,k)}-F_\mathrm{e}^{(i,j,k)}| \tag{17}$$

where $V_\Omega^{(i,j,k)}$ is the volume of cell (i,j,k), and $F^{(i,j,k)}$ and $F_\mathrm{e}^{(i,j,k)}$ are, respectively, the calculated and 'exact' volume fractions at the end of the test. The 'exact' values are accurately obtained from the position of the sphere at the beginning of the test, using a recursive local mesh refinement technique (see, for example, [24, 26]).

Table II shows the error, order of convergence and relative CPU times (required by advection, t_adv, and reconstruction, t_rec) after the sphere of fluid undergoes one complete revolution. The results obtained with the proposed FMFPA-3D method are first compared with those obtained with the RK-3D method, using in both cases the reconstruction method of Youngs. Note that the proposed advection method noticeably increases the accuracy of the results at the cost of a relatively low increase in the required CPU time, which is due to the more complex geometrical operations involved in handling the face-matched flux polyhedra.

It should be emphasized that the overall conservation of fluid volume is considerably improved when the proposed FMFPA-3D method is used. This is a consequence of the reduction in polyhedron

Table II. L_1 error norm, E, order of convergence, $\mathcal{O}$, and relative CPU times (advection, t_adv, and reconstruction, t_rec, times) obtained in the rotation of a spherical fluid body test, using two different advection methods, the reconstruction method of Youngs and three grid sizes.

Advection algorithm	Grid size	E	$\mathcal{O}$	Relative CPU time ($t_\mathrm{adv}+t_\mathrm{rec}$)
RK-3D	32^3	5.99×10^{-4}		1.00 (0.95+0.05)
			1.80	
	64^3	1.72×10^{-4}		13.40 (12.94+0.46)
			1.39	
	128^3	6.57×10^{-5}		198.94 (193.55+5.39)
FMFPA-3D	32^3	4.32×10^{-4}		1.18 (1.13+0.05)
			1.54	
	64^3	1.49×10^{-4}		15.36 (14.92+0.44)
			1.23	
	128^3	6.36×10^{-5}		216.65 (211.24+5.41)

DOI: 10.1002/fld

over/underlapping situations and the increase in accuracy which results from the reduction of discretization errors. In order to illustrate this improvement, the coarse grid case was also solved using both the RK-3D and the FMFPA-3D advection methods implemented in this work, but without using any local redistribution algorithm to reduce the effect of 'over/undershoots'. It can be observed from Figure 8 that the proposed method leads to a considerable reduction in the overshoots and undershoots, which were quantified as

$$F_S = \sum_{i,j,k} [|\min(0.0, F^{(i,j,k)})| + (\max(1.0, F^{(i,j,k)}) - 1.0)] \tag{18}$$

The net change in total volume after one complete revolution was 0.05% for the RK-3D method, and only 8.5×10^{-4}% for the proposed FMFPA-3D method (almost two orders of magnitude smaller). This improvement was found not only in the rotation test but also in the test described below.

Similar results to those of Table II are presented in Table III using, instead of Youngs' method, the CLC-CBIR reconstruction method proposed in [14], for which a second-order accuracy is achieved. Note that the use of a more accurate reconstruction method makes the difference between the RK-3D and the FMFPA-3D advection methods higher (around 30% for all grids). It should be taken into account that, when the CLC-CBIR method is used, the contribution of the reconstruction step to the error E is reduced. A more detailed analysis of the use of the CLC-CBIR reconstruction method in combination with the proposed FMFPA-3D advection method is discussed in the companion paper [14].

Figure 9 shows a comparison between the flux polyhedra constructed using our extension to 3D of the Rider and Kothe [2] method and the face-matched flux polyhedra proposed in this work (Figures 9(a) and (b), respectively). In these figures, the dashed lines represent the limits of the donating flux regions (defined by the locations of the fluid particles that cross the considered cell

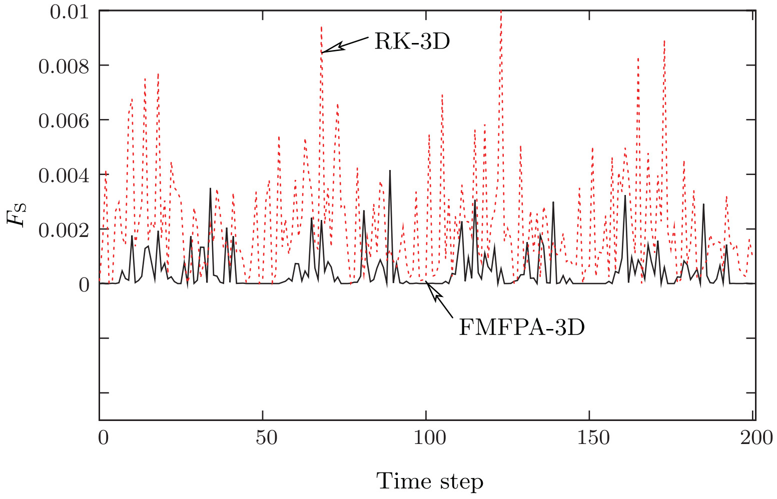

Figure 8. Overshoots and undershoots generated in the rotation test with the RK-3D and the FMFPA-3D algorithms. Results obtained with the reconstruction method of Youngs, a grid size of 32^3 and CFL = 0.5.

DOI: 10.1002/fld

Table III. L_1 error norm, E, and order of convergence, $\mathcal{O}$, obtained for three grid sizes in the rotation of a spherical fluid body test, using the CLC-CBIR reconstruction method proposed in [14] and different advection methods.

Advection algorithm	Grid size	E	$\mathcal{O}$
RK-3D	32^3	6.47×10^{-4}	
			1.95
	64^3	1.67×10^{-4}	
			1.97
	128^3	4.26×10^{-5}	
FMFPA-3D	32^3	4.96×10^{-4}	
			1.94
	64^3	1.29×10^{-4}	
			1.93
	128^3	3.38×10^{-5}	

face during the first integration time step Δt of the test), which were determined accurately from the prescribed velocity field. The differences between both methods can be clearly seen from the figures.

4.1.2. Deformation test. In this test, proposed by LeVeque [20], a sphere of fluid of radius 0.15, initially centered at $(0.35, 0.35, 0.35)$ within a unit computational domain, is deformed in a solenoidal velocity field given by

$$\begin{aligned} u(x,y,z) &= 2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z)\\ v(x,y,z) &= -\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z)\\ w(x,y,z) &= -\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z) \end{aligned} \tag{19}$$

which was modulated in time by multiplying by the factor $\cos(\pi t/T)$, using a period $T=3$.

Table IV shows the error E, order of convergence and relative CPU times at $t=T$ obtained using the two advection methods implemented in this work. A CFL number (based on the maximum velocity component in the domain) equal to 1.0 was used. The proposed FMFPA-3D method is slightly more accurate than the RK-3D method. Note that the large interface distortion produced in this test means that the accuracy of the whole method relies, to a large extent, on the reconstruction step, which makes the differences between both advection methods lower than in the rotation test. Owing to the larger deformation rates, the geometrical operations involved in both the RK-3D and the FMFPA-3D methods reach a similar degree of complexity in this test, considerably diminishing the differences in the CPU time compared with the rotation test. Figure 10 shows the results for the interface shape at the end of the test ($t=T$) obtained using three different grid sizes. The initial sphere of fluid is represented in transparent color. The results obtained with the RK-3D algorithm are similar to those shown in Figure 10.

In order to compare the proposed method with the unsplit VOF advection method recently proposed by Liovic *et al.* [12] (PCFSC), the deformation test was also performed using both the

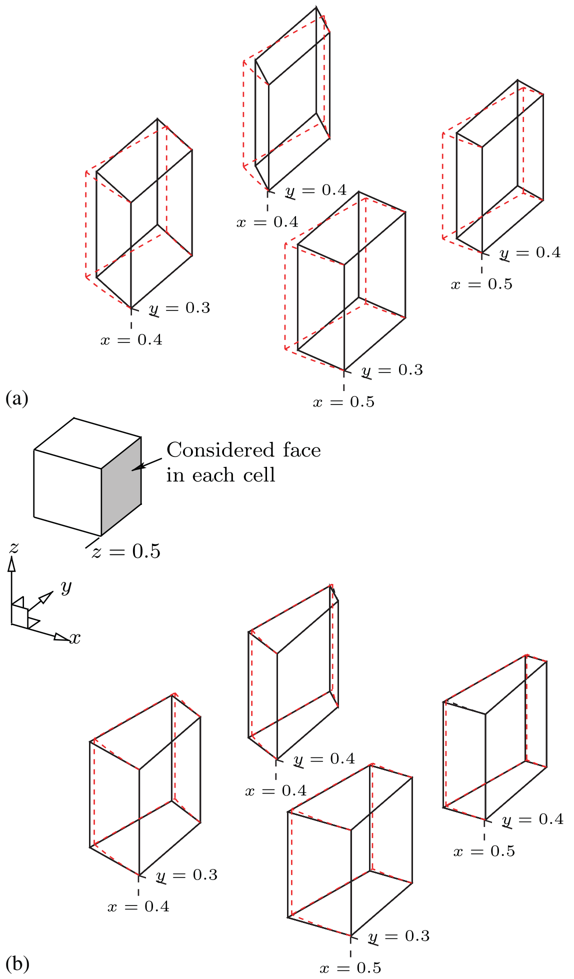

Figure 9. Flux polyhedra constructed in the rotation test of Section 4.1.1 at different cell faces located at $z=0.5$ and during the first time step, using two different advection methods: (a) our implementation in 3D of the method of Rider and Kothe [2] and (b) the method proposed in this work. A grid size of 10^3 and a CFL number equal to 1.0 were used. The exact limits of the donating flux regions are represented with dashed lines.

RK-3D and the FMFPA-3D advection methods and the same reconstruction method of Youngs used by these authors, for the same CFL number. The comparison between the corresponding results, presented in Table V, shows that the accuracy of the FMFPA-3D method is slightly higher than that of the advection method of Liovic *et al.* [12]. On the other hand, the differences between the results obtained with the RK-3D and PCFSC methods are relatively small. This may be due to the fact that, as mentioned by Liovic *et al.* [12], the PCFSC method looks like a 3D extension of the 2D unsplit advection scheme proposed by Rider and Kothe [2]. One important difference between the PCFSC method and our implementation in 3D of Rider and Kothe's [2] scheme (first

DOI: 10.1002/fld

Table IV. L_1 error norm, E, order of convergence, $\mathcal{O}$, and relative CPU times (advection, t_{adv}, and reconstruction, t_{rec}, times) obtained in the deformation test of LeVeque [20] with CFL$=1.0$, using two different advection methods and three grid sizes.

Advection algorithm	Grid size	E	$\mathcal{O}$	Relative CPU time ($t_{\text{adv}}+t_{\text{rec}}$)
RK-3D	32^3	7.53×10^{-3}		1.00 (0.97+0.03)
			1.52	
	64^3	2.62×10^{-3}		11.04 (10.76+0.28)
			1.89	
	128^3	7.09×10^{-4}		132.97 (130.09+2.88)
FMFPA-3D	32^3	7.22×10^{-3}		1.14 (1.11+0.03)
			1.46	
	64^3	2.62×10^{-3}		12.32 (12.05+0.27)
			1.93	
	128^3	6.86×10^{-4}		139.14 (136.26+2.88)

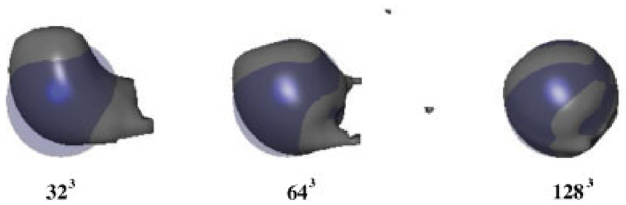

Figure 10. Results for the $F=0.5$ iso-surfaces obtained in the deformation test of LeVeque [20] with CFL$=1.0$, using three different grid sizes.

Table V. L_1 error norm, E, and order of convergence, $\mathcal{O}$, obtained in the deformation test of LeVeque [20], using the RK-3D and the FMFPA-3D methods in combination with Youngs' reconstruction method, for three different grid sizes and CFL$=0.5$. Comparison with the results obtained by Du *et al.* [30] with their front tracking method (LGB) and by Liovic *et al.* [12] with their unsplit VOF advection method (PCFSC) and Youngs' reconstruction method.

	Front tracking (LGB) [30]		PCFSC [12]		RK-3D		FMFPA-3D	
Grid size	E	$\mathcal{O}$	E	$\mathcal{O}$	E	$\mathcal{O}$	E	$\mathcal{O}$
32^3	5.72×10^{-3}		7.86×10^{-3}		7.85×10^{-3}		7.44×10^{-3}	
		3.72		1.43		1.51		1.42
64^3	4.33×10^{-4}		2.91×10^{-3}		2.75×10^{-3}		2.79×10^{-3}	
		1.82		1.98		1.89		1.97
128^3	1.23×10^{-4}		7.36×10^{-4}		7.41×10^{-4}		7.14×10^{-4}	

paragraph of Section 4.1) is that in the PCFSC method the constraint imposed by Equation (4) is not explicitly enforced during the construction of the flux polyhedra. Instead, the flux factor of Equation (11) is applied for all fluxes in spatially varying velocity fields. In the RK-3D method implemented in this work, as in the FMFPA-3D method, this scaling procedure is only applied in the few cases in which the constraint imposed by Equation (4) cannot be geometrically enforced.

Table V also includes results obtained by Du *et al.* [30] using a front tracking method. The accuracy of the results obtained by these authors is generally higher than that of PLIC-VOF methods (an exhaustive comparison with other PLIC-VOF methods for different tests can be found in Du *et al.* [30]). However, the advantages of using PLIC-VOF or similar methods to treat topological changes in the interface still make them competitive from the point of view of computational efficiency and robustness.

4.2. Drop impact test

In this section, a water drop of diameter $D=2.9\,\mathrm{mm}$ impacting a deep water pool with velocities $U=1.55$ and $2.50\,\mathrm{m\,s^{-1}}$ is considered. The Froude and Weber numbers are $Fr=85$ and $We=96$ for $U=1.55\,\mathrm{m\,s^{-1}}$ and $Fr=220$ and $We=248$ for $U=2.50\,\mathrm{m\,s^{-1}}$. Owing to the symmetry of the problem, only one quarter of the physical domain was considered. The computational domain used was $7D\times3.5D\times3.5D$ for $U=1.55\,\mathrm{m\,s^{-1}}$ and $9D\times4.5D\times4.5D$ for $U=2.5\,\mathrm{m\,s^{-1}}$. In both cases the pool depth was $4.5D$, the water drop was initially located at a height equal to $6.0D$ and the domain was discretized on a mesh of $140\times70\times70$ cells. In order to reach the desired impact velocity, a fictitious gravitational force was used to accelerate the drop.

Figures 11(a), for $U=1.55\,\mathrm{m\,s^{-1}}$, and 12(a), for $U=2.50\,\mathrm{m\,s^{-1}}$, show the results of the $F=0.5$ iso-surfaces at different instants after the drop made contact with the pool surface. The time has been made dimensionless with the time D/U. In both simulations, the net change in total volume at the end of the test was lower than $3\times10^{-6}\%$. In the case of the impact velocity of $1.55\,\mathrm{m\,s^{-1}}$, the numerical simulation is able to predict, at $tU/D=10.5$, the entrapment of a bubble with a size of around $0.2D$. For the case of the impact velocity of $2.50\,\mathrm{m\,s^{-1}}$, there is no bubble entrapment and a thick Rayleigh jet is formed. The results at the dimensionless instant 35.8 clearly show the formation of a secondary drop at the jet tip.

The numerical results obtained for the interface shape were assessed using experiments conducted in this work, similar to those carried out by Morton *et al.* [31]. Drops of distilled water of 2.9 mm diameter, formed using a hypodermic needle, were allowed to fall from heights of about 14 and 36 cm on a water pool. These heights were adjusted in order to get the 1.55 and $2.5\,\mathrm{m\,s^{-1}}$ drop impact velocities used in the numerical simulations. A high-speed digital camera (Redlake MotionPro 10 000) was used to capture the images of the impact at a rate of $2000\,\mathrm{frames\,s^{-1}}$, with a shutter speed of 0.5 ms and a resolution of 512×256 pixels. Back lighting was provided by a halogen lamp of 1000 W. The drop impact velocity was measured from the movie images. The dynamics of the impact was found to be relatively sensitive to the time elapsed since the needle and pool were filled with distilled water, one of the causes probably being the heating produced by the halogen lamp. The experimental results presented in Figures 11(b) and 12(b) were obtained just after the needle and pool were filled. In successive experiments, made without replacing the distilled water, the results tended to be more similar to those obtained experimentally by Morton *et al.* [31], which are otherwise very close to those presented here.

A visual comparison between the numerical and experimental results presented in Figures 11 and 12 shows good agreement, even in small interfacial details such as those involved in the bubble detachment from the floor of the crater formed by the drop impact that can be observed in Figure 11, and the break-up of the Rayleigh jet and droplet formation shown in Figure 12. A quantitative comparison is shown in Figure 13, where the numerical predictions for the evolution

DOI: 10.1002/fld

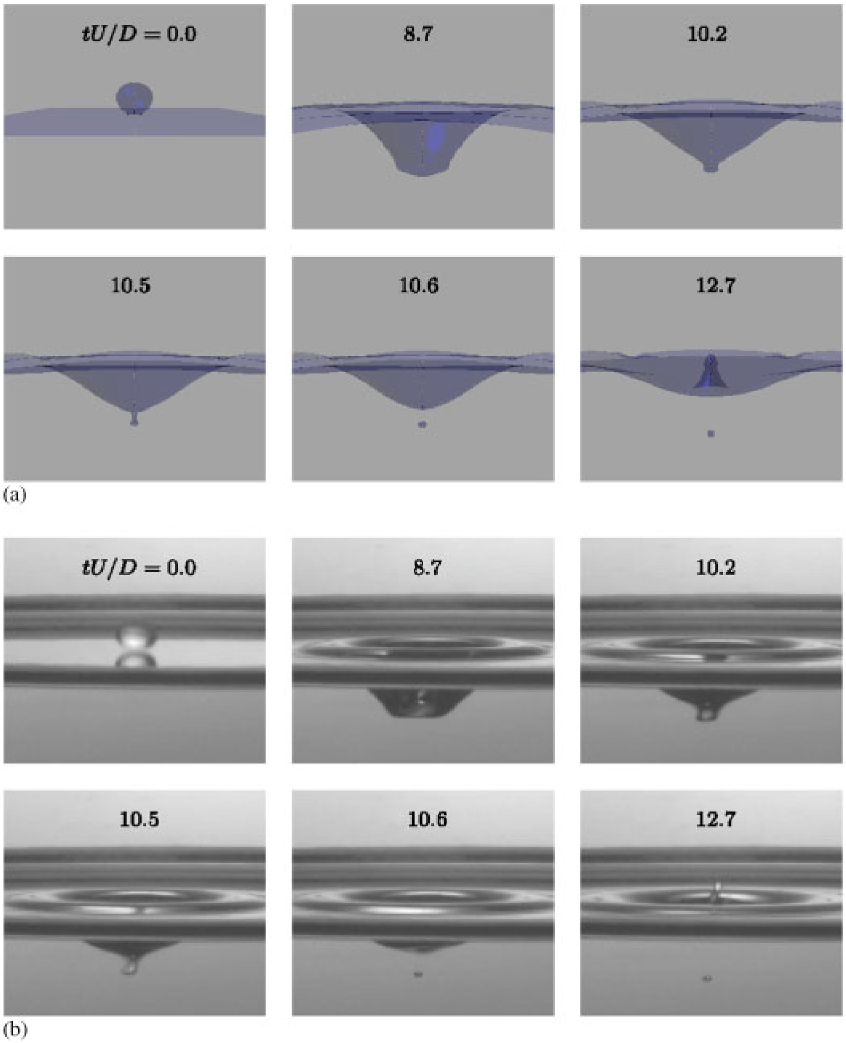

Figure 11. Comparison between (a) numerical predictions for the $F = 0.5$ iso-surfaces and (b) experimental results for the interface shape, at different instants after the drop has made contact with the pool surface, for a drop impact velocity of $U = 1.55\,\mathrm{m\,s}^{-1}$ ($Fr = 85$ and $We = 96$).

of the free-surface depth at the symmetry axis, D_c, are compared with the experimental results for the drop impact cases of Figures 11 and 12. It can be observed that the numerical results reasonably predict the cavity depth evolution during growth and collapse processes.

5. CONCLUSIONS

A volume of fluid method based on a new multidimensional procedure to integrate the volume fraction advection equation has been developed for tracking interfaces in 3D. The fluid volume

DOI: 10.1002/fld

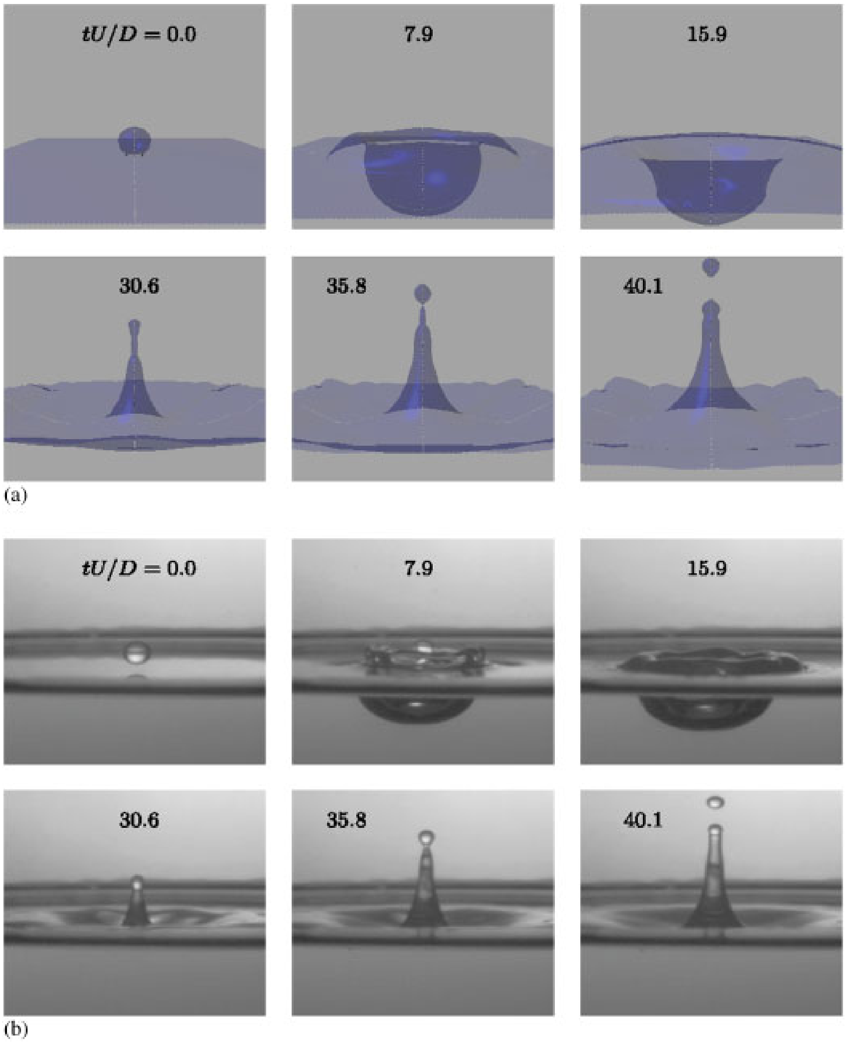

Figure 12. Comparison between (a) numerical predictions for the $F = 0.5$ iso-surfaces and (b) experimental results for the interface shape, at different instants after the drop has made contact with the pool surface, for a drop impact velocity of $U = 2.5\,\mathrm{m\,s^{-1}}$ ($Fr = 220$ and $We = 248$).

fluxes across cell faces are systematically obtained from the intersections between flux polyhedra and the reconstructed interface. For this purpose, a set of simple geometric tasks has been used, which makes the method efficient, robust and relatively simple to implement in 3D. The proposed advection algorithm enforces mass conservation locally through the use of face-matched flux polyhedra and reduces the formation of ‘over/undershoots’, thus diminishing the need to use local redistribution algorithms. The proposed method has been assessed using different tests with prescribed velocity fields. The accuracy of the results is slightly higher than that obtained with other recently proposed multidimensional advection methods. A method to calculate the interfacial

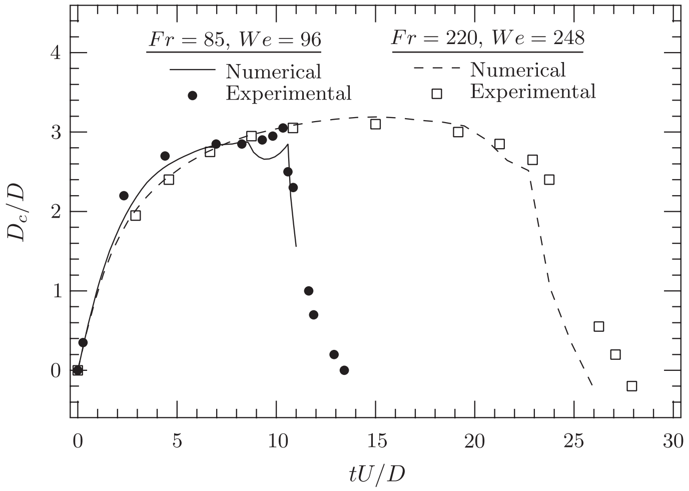

Figure 13. Comparison between numerical predictions and experimental results for the evolution of the free-surface depth at the symmetry axis, D_c, for the drop impact cases of Figures 11 and 12.

curvature based on a height-function technique with adaptive stencil adjustment has also been proposed, which improves accuracy in regions of low grid resolution. A comparison of the numerical results with experimental results for the impact of a water droplet on a deep water pool points to a good degree of agreement.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of the Spanish Ministerio de Educación y Ciencia under Grants DPI2004-08198 and PR2005-0296.

REFERENCES

1. Scardovelli R, Zaleski S. Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics* 1999; **31**:567–603.
2. Rider WJ, Kothe DB. Reconstructing volume tracking. *Journal of Computational Physics* 1998; **141**:112–152.
3. Rudman M. Volume tracking methods for interfacial flow calculations. *International Journal for Numerical Methods in Fluids* 1997; **24**:671–691.
4. Gueyffier D, Li J, Nadim A, Scardovelli R, Zaleski S. Volume of fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *Journal of Computational Physics* 1999; **152**:423–456.
5. Lörstad D, Fuchs L. High-order surface tension VOF-model for 3D bubble flows with high density ratio. *Journal of Computational Physics* 2004; **200**:153–176.
6. Renardy Y, Renardy M. PROST: a parabolic reconstruction of surface tension for the volume-of-fluid method. *Journal of Computational Physics* 2002; **183**:400–421.
7. van Sint Annaland M, Deen NG, Kuipers JAM. Numerical simulation of gas bubbles behaviour using a three-dimensional volume of fluid method. *Chemical Engineering Science* 2005; **60**:2999–3011.
8. Sussman M, Puckett EG. A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. *Journal of Computational Physics* 2000; **162**:301–337.
9. Miller GH, Colella P. A conservative three-dimensional Eulerian method for coupled solid–fluid shock capturing. *Journal of Computational Physics* 2002; **183**:26–82.

DOI: 10.1002/fld

10. Pilliod JE, Puckett EG. Second-order volume-of-fluid algorithms for tracking material interfaces. *Technical Report LBNL-40744*, Lawrence Berkeley National Laboratory, 1997.
11. Pilliod JE, Puckett EG. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *Journal of Computational Physics* 2004; **199**:465–502.
12. Liovic P, Rudman M, Liow J-L, Lakehal D, Kothe D. A 3D unsplit-advection volume tracking algorithm with planarity-preserving interface reconstruction. *Computers and Fluids* 2006; **35**:1011–1032.
13. López J, Hernández J, Gómez P, Faura F. A volume of fluid method based on multidimensional advection and spline interface reconstruction. *Journal of Computational Physics* 2004; **195**:718–742.
14. López J, Zanzi C, Gómez P, Faura F, Hernández J. A new volume of fluid method in three dimensions—Part II: Piecewise-planar interface reconstruction with cubic-Bézier fit. *International Journal for Numerical Methods in Fluids* 2008. DOI: 10.1002/fld.1775.
15. Youngs DL. An interface tracking method for a 3D Eulerian hydrodynamics code. *Technical Report 44/92/35*, AWRE, 1984.
16. Biausser B, Guignard S, Marcer R, Fraunie P. 3D two phase flows numerical simulations by SL-VOF method. *International Journal for Numerical Methods in Fluids* 2004; **45**:581–604.
17. Scardovelli R, Zaleski S. Analytical relations connecting linear interfaces and volume fractions in rectangular grids. *Journal of Computational Physics* 2000; **164**:228–237.
18. Press WH, Teukolsky SA, Vetterling WT, Flannery BP. *Numerical Recipes in Fortran*. Cambridge University Press: Cambridge, U.K., 1986.
19. Schneider PJ, Eberly DH. *Geometric Tools for Computer Graphics*. Morgan Kaufmann Publishers: London, 2003.
20. LeVeque R. High-resolution conservative algorithms for advection in incompressible flow. *SIAM Journal on Numerical Analysis* 1996; **33**:627–665.
21. Harvie DJE, Fletcher DF. A new volume of fluid advection algorithm: the stream scheme. *Journal of Computational Physics* 2000; **162**:1–32.
22. Gómez P, Hernández J, López J. On the reinitialization procedure in a narrow-band locally-refined level set method for interfacial flows. *International Journal for Numerical Methods in Engineering* 2005; **63**:1478–1512.
23. López J, Hernández J, Gómez P, Faura F. An improved PLIC-VOF method for tracking thin fluid structures in incompressible two-phase flows. *Journal of Computational Physics* 2005; **208**:51–74.
24. Francois MM, Cummins SJ, Dendy ED, Kothe DB, Sicilian JM, Williams MW. A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. *Journal of Computational Physics* 2006; **213**:141–173.
25. Lörstad D, Francois MM, Shyy W, Fuchs L. Assessment of volume of fluid and immersed boundary methods for droplet computations. *International Journal for Numerical Methods in Fluids* 2004; **46**:109–125.
26. Cummins SJ, Francois MM, Kothe DB. Estimating curvature from volume fractions. *Computers and Structures* 2005; **83**:425–434.
27. Helmsen J, Colella P, Puckett EG. Non-convex profile evolution in two dimensions using volume of fluids. *Technical Report LBNL-40693*, Lawrence Berkeley National Laboratory, 1997.
28. Torrey M, Cloutman L, Mjolsness R, Hirt C. NASA-VOF2D: a computer program for incompressible flows with free surfaces. *Technical Report LA-10612-MS*, Los Alamos National Laboratory, 1985.
29. Malik M, Fan ES-C, Bussmann M. Adaptive VOF with curvature-based refinement. *International Journal for Numerical Methods in Fluids* 2007; **55**:693–712.
30. Du J, Fix B, Glimm J, Jia X, Li X, Li Y, Wu L. A simple package for front tracking. *Journal of Computational Physics* 2006; **213**:613–628.
31. Morton M, Rudman M, Jong-Leng L. An investigation of the flow regimes resulting from splashing drops. *Physics of Fluids* 2000; **12**:747–763.