



Compressive VOF method with skewness correction to capture sharp interfaces on arbitrary meshes



Fabian Denner*, Berend G.M. van Wachem

Thermofluids Division, Department of Mechanical Engineering, Imperial College London, Exhibition Road, London, SW7 2AZ, United Kingdom

ARTICLE INFO

Article history:

Received 11 March 2014

Received in revised form 1 September 2014

Accepted 3 September 2014

Available online 9 September 2014

Keywords:

Volume of fluid (VOF) method

Two-phase flows

Interface capturing

Advection scheme

Unstructured meshes

ABSTRACT

The accurate and efficient modelling of two-phase flows is at present mostly limited to structured, unskewed meshes, due to the additional topological and numerical complexity of arbitrary, unstructured meshes. Compressive VOF methods which discretize the interface advection with algebraic differencing schemes are computationally efficient and inherently applicable to arbitrary meshes. However, compressive VOF methods evidently suffer severely from numerical diffusion on meshes with topological skewness. In this paper we present a compressive VOF method using a state-of-the-art donor-acceptor advection scheme which includes novel modifications to substantially reduce numerical diffusion on arbitrary meshes without adding computational complexity. The new methodology accurately captures evolving interfaces on any arbitrary, non-overlapping mesh and conserves mass within the limits of the applied solver tolerance. A thorough validation of the presented methods is conducted, examining the pure advection of the interface indicator function as well as the application to evolving interfaces with surface tension. Crucially, the results on equidistant Cartesian and arbitrary tetrahedral meshes are shown to be comparable and accurate.

© 2014 Published by Elsevier Inc.

1. Introduction

The accurate representation and advection of the material interface between two fluids is essential for the predictive quality of immiscible two-phase flow simulations. However, the interface is infinitesimally thin with respect to continuum mechanics and, thus, represents a considerable challenge for finite volume and finite element frameworks. The Volume of Fluid (VOF) [1] method is among the most widely used methods to represent two incompressible, immiscible fluids. Two fundamental types of VOF methods may be distinguished: compressive methods and geometric methods. Both types of VOF methods capture the discrete volume fraction of each phase and transport it based on the underlying fluid. Compressive VOF methods discretize the partial differential equation describing the transport of the volume fraction of each phase using algebraic differencing schemes [2–5]. However, the temporal and spatial discretisation requires special care in order to keep the interface sharp and without distortion. Using a geometric method, an explicit representation of the interface is advected, reconstructed from the VOF volume fraction field. The most notable state-of-the-art reconstruction methods are piecewise linear (PLIC) methods [6–12] and parabolic reconstruction methods [13]. The explicit interface is geometrically fitted to the VOF volume fraction field and advected in an Eulerian, Lagrangian or mixed Eulerian–Lagrangian fashion [14].

* Corresponding author. Tel.: +44 7500 060 053.

E-mail address: fabian.denner@gmail.com (F. Denner).

The major advantages of compressive VOF methods compared to geometric VOF methods are the straightforward implementation on arbitrary meshes and the computational efficiency. Geometric methods advect the interface very accurately, but the three-dimensional reconstruction of the interface requires significantly larger computational resources compared to a two-dimensional reconstruction [12], and even more so on arbitrary meshes, due to the added complexity of the geometric primitives used to reconstruct the interface. Hence, the implementation of geometric VOF methods is considerably more complex than compressive VOF methods, the required computational effort is higher and, as a result, geometric VOF methods are at present, apart from a few recently proposed exceptions [14–17], exclusively available on structured meshes. Despite the greater flexibility and wider applicability of compressive VOF methods, geometric VOF methods such as PLIC methods are in current literature widely preferred over compressive VOF methods, due to their claimed higher accuracy. However, recent studies by Gopala and van Wachem [18], Park et al. [19] and Denner et al. [20] show that compressive VOF methods are generally capable of advecting sharp, evolving interfaces with similar accuracy as state-of-the-art PLIC methods.

A major challenge for compressive VOF methods is to retain the shape and sharpness of the interface. The transient advection of the interface is typically discretized using a second-order temporal discretisation scheme, such as the Crank–Nicolson scheme or the Second-Order Backward Euler scheme [4,21], since first-order schemes are too diffusive to maintain the sharpness of the interface. Studies by Ubbink [21], Jasak [22] and Darwish [23] comprehensively demonstrate that the First-Order Backward Euler scheme as well as the First-Order Forward Euler scheme distort the shape of a circular interface advected at a constant oblique velocity on an equidistant Cartesian mesh. The Crank–Nicolson scheme, on the other hand, is able to preserve the shape of the circular interface of the same test case. In [24], Moukalled and Darwish propose a class of temporal discretisation schemes, which switch between a compressive and a high-resolution scheme based on the angle between the interface normal vector and the velocity vector. Moukalled and Darwish [24] use the Second-Order Backward Euler scheme and a compressive Euler scheme, reporting better results at medium and high Courant numbers than with other commonly used schemes, such as the Crank–Nicolson scheme.

Similarly, the careful spatial discretisation of the interface advection is essential to preserve a sharp interface. Low-order advection schemes are not suitable, as they lead to significant smearing of the interface, whereas high-order schemes result in numerical oscillations and wrinkling of the interface. Compressive VOF methods, therefore, use specifically designed spatial advection schemes based on a donor–acceptor approach. A donor (upwind) and an acceptor (downwind) cell are assigned for every cell face with respect to the underlying flow field. Based on the angle between the local interface normal vector and the normal vector of the cell face, donor–acceptor schemes blend between compressive downwind and diffusive upwind schemes. Following a donor–acceptor approach, a number of spatial discretisation schemes specifically designed for compressive VOF methods have been proposed in recent years, most notably the SURFER scheme [2], the CICSAM scheme [3], the STACS scheme [4], the HRIC scheme [5] and the HiRAC scheme [25]. The application of these schemes to arbitrary meshes is typically straightforward but is also associated with discretisation errors induced by the mesh orientation and arrangement as well as the construction of an artificial upwind node for each cell face. Zhang et al. [26] examined the influence of the upwind node extrapolation and reported a profound impact on the accuracy of the tested advection schemes. Denner and van Wachem [27] observed considerable numerical diffusion simulating the buoyancy-driven rise of a bubble on a tetrahedral mesh, suggesting the numerical diffusion is a result of mesh skewness. A more detailed examination of this issue by Denner [28] supports the assumption that mesh skewness is a major source of error with respect to the interface advection. Ubbink [21] put forward two additional possible reasons for numerical diffusion of the VOF volume fraction using donor–acceptor schemes. First, the extrapolation of the upwind node, required to determine the advection of the colour function on unstructured meshes, and, second, the implicit assumption that an interface is intersecting a cell face if both adjacent cells contain an interface. Numerical diffusion as a result of the applied differencing schemes has previously been extensively discussed in the literature [4,21,29,30]. However, numerical diffusion induced by mesh skewness has, so far, not been the focus of research in the context of interface advection. This is surprising, as mesh skewness is the norm on arbitrary meshes, e.g. boundary-fitted hexahedral meshes or tetrahedral meshes, and leads to a diffusion-like error that severely affects the accuracy of spatial interpolation, as for instance the linear interpolation of face-centred values from adjacent cell-centred values required to discretize the spatial advection of the VOF volume fraction.

In this article we present a revised compressive VOF framework, based on the widely used CICSAM scheme, including modifications aiming at mitigating numerical diffusion due to mesh skewness. In particular, an implicit skewness correction is proposed which substantially decreases numerical diffusion on arbitrary meshes without notably affecting the computational complexity and mass-conservation of the VOF algorithm. It is worth mentioning that the proposed modifications are not limited to the CICSAM scheme but can also be applied to any other donor–acceptor scheme (e.g. HRIC or STACS). The comprehensive and thorough validation demonstrates the capabilities and accuracy of the presented methodology and the results on Cartesian and tetrahedral meshes are shown to be comparable. Moreover, we show that our framework is mass-conserving within the limits of the applied solver tolerance. The presented methodology including the proposed modifications are of significant interest to the two-phase flow modelling community as it improves the predictive quality of interfacial flow simulations considerably and, thus, makes it feasible for applications with complex geometries which cannot be represented by general structured meshes.

The article is structured as follows. In Section 2 the governing equations are outlined and Section 3 briefly explains the applied numerical framework. The discretisation of the applied compressive VOF method is discussed in Section 4. Subsequently, Section 5 is concerned with numerical diffusion caused by mesh skewness and introduces the proposed modifications to mitigate numerical diffusion. The presented compressive VOF methodology including the proposed modifi-

cations are comprehensively tested and validated in Section 6. The findings are summarised and the article is concluded in Section 7.

2. Governing equations

The Volume of Fluid (VOF) method [1] is used to determine the position and behaviour of the interface between two incompressible, isothermal and immiscible fluids. The VOF method applies a volume fraction γ , also called colour function, to every mesh cell, representing the local volume fraction as

$$\gamma(\mathbf{x}, t) = \begin{cases} 0 & \text{fluid A} \\ 1 & \text{fluid B.} \end{cases} \quad (1)$$

Thus, every mesh cell where the colour function value is between 0 and 1 contains an interface. As the interface is transported with the underlying flow field, the colour function is advected by the hyperbolic equation

$$\frac{\partial \gamma}{\partial t} + u_i \frac{\partial \gamma}{\partial x_i} = 0, \quad (2)$$

where \mathbf{u} is the velocity of the flow and t represents time. The density ρ and viscosity μ are defined based on a linear relationship with the colour function γ , given as

$$\rho = \rho_A(1 - \gamma) + \rho_B \gamma \quad (3)$$

$$\mu = \mu_A(1 - \gamma) + \mu_B \gamma, \quad (4)$$

where subscripts A and B denote either of the fluids.

The motion of isothermal fluids is governed by the conservation of momentum and the conservation of mass. The conservation of momentum for a Newtonian fluid is described by the momentum equations as

$$\rho \left(\frac{\partial u_i}{\partial t} + \frac{\partial u_j u_i}{\partial x_j} \right) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] + \rho g_i + f_{s,i}, \quad (5)$$

where p represents pressure, \mathbf{g} is the gravitational acceleration and \mathbf{f}_s is the volumetric force due to surface tension. The conservation of mass is generally defined as

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_i)}{\partial x_i} = \frac{\partial \rho}{\partial t} + u_i \frac{\partial \rho}{\partial x_i} + \rho \frac{\partial u_i}{\partial x_i} = 0. \quad (6)$$

Assuming the fluid is incompressible and isothermal, as considered in this study, the velocity field in each phase is divergence-free, given by

$$\frac{\partial u_i}{\partial x_i} = 0. \quad (7)$$

Hence, Eq. (6) becomes

$$\frac{D\rho}{Dt} = \frac{\partial \rho}{\partial t} + u_i \frac{\partial \rho}{\partial x_i} = 0, \quad (8)$$

which is equivalent to assuming the density of an infinitesimally small fluid particle does not change over time. As explained in Section 3, continuity is implicitly enforced by the deployed numerical framework within the limits of the numerical solver tolerance [28]. Hence, the spatial advection term of Eq. (8) is conservative [31] and, as a result, so is Eq. (8) as a whole. Thus, given an isothermal, incompressible two-phase flow without phase changes and given the linear relationship of density and colour function defined in Eq. (3), i.e. $\rho = \rho(\gamma)$, the mass of each phase is conserved if the VOF advection equation, Eq. (2), is satisfied.

3. Numerical framework

The numerical framework follows a coupled implicit approach for the fluid flow with a segregated interface advection, as presented by Denner and van Wachem [27]. A linear equation system describing the flow is constructed, containing the discretized momentum equations as well as a fourth equation, which provides an additional relationship between pressure and velocity to close the equation system. The fourth equation of the flow equation system is constructed based on the continuity equation, given in Eq. (7), and is defined as

$$\frac{\partial u_i}{\partial x_i} \Big|_p \approx \frac{1}{V_p} \sum_f (\mathbf{u}_f \cdot \mathbf{n}_f) A_f \approx \frac{1}{V_p} \sum_f u_f^n A_f = 0, \quad (9)$$

where f denotes all bounding faces of a given mesh cell P , \mathbf{n}_f is the outward-pointing unit normal vector of face f and A_f is the area of face f . The advecting velocity u^n at cell faces is devised based on the *momentum interpolation method* as detailed in [27]. The fully assembled equation system to describe the fluid flow then follows as

$$\underbrace{\begin{pmatrix} \mathbf{A}_u^x & \mathbf{0} & \mathbf{0} & \mathbf{A}_p^x \\ \mathbf{0} & \mathbf{A}_v^y & \mathbf{0} & \mathbf{A}_p^y \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_w^z & \mathbf{A}_p^z \\ \mathbf{A}_u^c & \mathbf{A}_v^c & \mathbf{A}_w^c & \mathbf{A}_p^c \end{pmatrix}}_{\mathbf{A}} \cdot \underbrace{\begin{pmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{w} \\ \mathbf{p} \end{pmatrix}}_{\boldsymbol{\phi}} = \mathbf{b}, \quad (10)$$

where \mathbf{A} is the coefficient matrix, $\boldsymbol{\phi}$ the solution vector and \mathbf{b} the right-hand side vector. Inside matrix \mathbf{A} , \mathbf{A}_j^i represents the coefficient submatrix for primitive variable j of the i -th equation, with superscripts x , y and z denoting the three momentum equations and c denoting the fourth equation derived using the momentum interpolation method. Solution vector $\boldsymbol{\phi}$ is constituted by the solution subvectors of the three velocity components \mathbf{u} , \mathbf{v} and \mathbf{w} , and pressure \mathbf{p} . The momentum equations, Eq. (5), are discretized using the Second-Order Backward Euler scheme for the transient term and a central differencing scheme for the advection term. If applicable, the volumetric force due to surface tension \mathbf{f}_s acting at the interface is discretized using the continuum surface force (CSF) model [32].

Before the new flow field is computed at any given time instant, the colour function is advected as defined in Eq. (2) based on the flow field resulting from the previous time-step. Advecting the colour function prior to computing the flow field at a given time instant ensures that the pressure field is calculated based on the new interface position. In contrast, updating the interface position after the new flow field has been calculated would result in a lag between pressure distribution and interface position. The discretized VOF advection equation is solved by means of a linear equation system,

$$\mathbf{A} \cdot \boldsymbol{\gamma} = \mathbf{x}, \quad (11)$$

with a subsequent evaluation of the interface curvature if the simulated flow includes surface tension. The interface advection equation is discretized using algebraic discretisation schemes and advanced in a time-marching fashion. The general discretisation of Eq. (2) is described in Section 4 and the proposed modifications to the discretisation are presented in Section 5.

The time-step applied to advect the interface has a crucial influence on the ability to retain the sharpness of the interface. Using the CICSAM scheme, Ubbink [21, Chap. 5] found the interface to stay reasonably sharp for Courant numbers $Co = |\mathbf{u}| \Delta t / \Delta x \leq 0.3$. The case studies of Darwish and Moukalled [4] confirm the trend of significantly improved results for decreasing Courant numbers as presented by Ubbink [21], suggesting viable results for $Co \leq 0.25$. Studies performed by Gopala and van Wachem [18] even suggest a Courant number limit of $Co \leq 0.01$ in order to maintain a sharp interface. In order to overcome the stringent Courant number constraints imposed by the interface advection, different time-steps are applied to advect the interface and the fluid, as previously proposed by Gopala and van Wachem [18]. A similar approach has also been used in a recent study by Heyns et al. [25]. Using a dual time-stepping approach, the Courant number criterion imposed by the interface advection can be fulfilled and, at the same time, efficient use of the computational resources is made while solving the flow equation system. For consistency, the time-step to solve the flow field has to be equal to or a multiple of the interface advection time-step.

4. Compressive VOF method

In the following sections, the discretisation of the interface advection equation is presented and discussed. The numerical schemes presented in this section represent only one specific choice, deemed to be best suited for the discretisation of the interface advection on arbitrary meshes. Other schemes may be used in a similar fashion without affecting the applicability of the proposed modifications presented in Section 5.

4.1. Temporal discretisation

In the presented compressive VOF method, the transient term of Eq. (2) is discretized using the Crank–Nicolson scheme. The discretized interface advection equation is, therefore, given as

$$\frac{(\gamma_P^t - \gamma_P^{t-\Delta t})V_P}{\Delta t} = - \sum_f \frac{\gamma_f^t F_f^t + \gamma_f^{t-\Delta t} F_f^{t-\Delta t}}{2}, \quad (12)$$

where F_f represents the flux through face f . The discretisation presented in Eq. (12) requires the fluxes through the face of two time levels. However, the flux of the new time level is not yet known. Ubbink [21] proposed that if the time-step is sufficiently small, the right-hand side of Eq. (12) can be approximated as

$$- \sum_f \frac{\gamma_f^t F_f^t + \gamma_f^{t-\Delta t} F_f^{t-\Delta t}}{2} \approx - \sum_f \frac{\gamma_f^t + \gamma_f^{t-\Delta t}}{2} F_f^{t-\Delta t}. \quad (13)$$

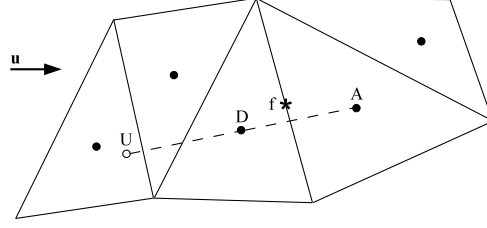


Fig. 1. Example of the arrangement of the face under consideration f , donor cell D , acceptor cell A and the extrapolated upwind node U for an arbitrary, unstructured mesh.

Hence, Eq. (12) simplifies to

$$\frac{(\gamma_P^t - \gamma_P^{t-\Delta t}) V_P}{\Delta t} = - \sum_f \frac{\gamma_f^t + \gamma_f^{t-\Delta t}}{2} F_f^{t-\Delta t}. \quad (14)$$

Analogous to the continuity constraint defined in Eq. (9), it is proposed to determine flux F_f in Eq. (14) based on the advecting velocity u_f^n as

$$F_f = u_f^n A_f. \quad (15)$$

Thus, flux F_f used to advect the interface satisfies continuity and the advection of the interface is defined consistently with the flow advection.

4.2. Spatial discretisation

The CICSAM scheme of Ubbink and Issa [3] is implemented as part of the presented VOF methodology to determine the value of colour function γ_f at mesh face f in Eq. (14). The CICSAM scheme is founded on the *Normalised Variable Diagram* (NVD) of Leonard [33]. By defining an acceptor cell A , a donor cell D and an upwind node U for cell face f , illustrated in Fig. 1, the normalised colour function value is defined as

$$\tilde{\gamma} = \frac{\gamma - \gamma_U}{\gamma_A - \gamma_U}. \quad (16)$$

The *Convective Boundedness Criterion* (CBC) proposed by Gaskell and Lau [34] stipulates that in order to avoid unphysical oscillations in the solution, γ_D has to be bounded by γ_A and γ_U , given as $0 \leq \tilde{\gamma}_D \leq 1$. However, unlike on structured meshes, the upwind value γ_U is not readily available on arbitrary, unstructured meshes. Jasak et al. [35] proposed a general definition of the upwind node for arbitrary meshes following the CBC, defined as

$$\gamma_U = \gamma_A + 2(\mathbf{x}_D - \mathbf{x}_A) \cdot \nabla \gamma|_D, \quad (17)$$

with \mathbf{x}_D and \mathbf{x}_A representing the position vector of donor cell D and acceptor cell A , respectively. Hence, the mesh arrangement and cell shape are irrelevant. This upwind definition has also been used by Ubbink and Issa [3] in the original proposal of the CICSAM scheme.

The CICSAM scheme is based on the Hyper-C (HC) scheme, which follows the upper bound of the CBC, and the ULTIMATE QUICKEST (UQ) scheme of Leonard [33]. Using normalised values as defined in Eq. (16), the HC scheme is given as

$$\tilde{\gamma}_f^{HC} = \begin{cases} \min \{1, \frac{\tilde{\gamma}_D}{Co_D}\} & \text{when } 0 \leq \tilde{\gamma}_D \leq 1 \\ \tilde{\gamma}_D & \text{when } \tilde{\gamma}_D > 0 \text{ and } \tilde{\gamma}_D < 1, \end{cases} \quad (18)$$

and the UQ scheme is defined as

$$\tilde{\gamma}_f^{UQ} = \begin{cases} \min \left\{ \frac{8 Co_D \tilde{\gamma}_D + (1 - Co_D)(6 \tilde{\gamma}_D + 3)}{8}, \tilde{\gamma}_f^{HC} \right\} & \text{when } 0 \leq \tilde{\gamma}_D \leq 1 \\ \tilde{\gamma}_D & \text{when } \tilde{\gamma}_D > 0 \text{ and } \tilde{\gamma}_D < 1, \end{cases} \quad (19)$$

where Co_D is the Courant number with respect to the fluxes leaving the donor cell.

Having defined both discretisation schemes, a blending function ψ_f is required to switch between the two discretisation schemes, so that

$$\tilde{\gamma}_f = \psi_f \tilde{\gamma}_f^{HC} + (1 - \psi_f) \tilde{\gamma}_f^{UQ}. \quad (20)$$

The blending function ψ_f proposed by Ubbink and Issa [3], is based on the angle Θ_f between the interface normal vector \mathbf{m}_f and the vector \mathbf{s}_f , depicted in Fig. 2, connecting the donor cell and acceptor cell, defined as

$$\psi_f = \min \left\{ k_\psi \frac{\cos(2\Theta_f) + 1}{2}, 1 \right\}, \quad (21)$$

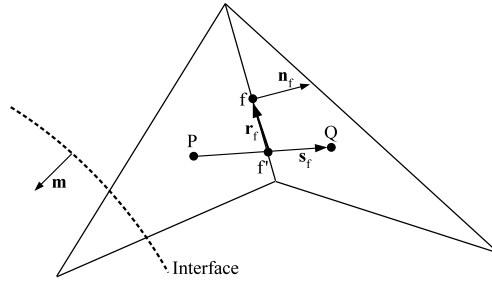


Fig. 2. Mesh element P with neighbour element Q and shared face f of a triangular two-dimensional mesh, including face normal vector \mathbf{n}_f , vector \mathbf{r}_f from interpolation point f' to face centre f and vector \mathbf{s}_f connecting the adjacent cell centres. The dashed line depicts the interface with its normal vector \mathbf{m} .

with angle Θ_f being

$$\Theta_f = \arccos\left(\left|\mathbf{m}_f \cdot \frac{\mathbf{s}_f}{|\mathbf{s}_f|}\right|\right). \quad (22)$$

The constant k_ψ in Eq. (21) controls the dominance of the HC scheme (the compressive scheme), where an increasing value of k_ψ corresponds to a stronger dominance of the HC scheme. Ubbink and Issa [3] recommended $k_\psi = 1$, for which the blending function becomes

$$\psi_f = \frac{\cos(2\Theta_f) + 1}{2} = [\cos(\Theta)]^2. \quad (23)$$

Defining a weighting coefficient β_f using the normalised colour function $\tilde{\gamma}_f$, given as

$$\beta_f = \frac{\tilde{\gamma}_f - \tilde{\gamma}_D}{1 - \tilde{\gamma}_D}, \quad (24)$$

the colour function value at face f follows as [3]

$$\gamma_f = (1 - \beta_f)\gamma_D + \beta_f\gamma_A. \quad (25)$$

As pointed out by Ubbink [21], inserting Eq. (25) in Eq. (14) does not guarantee a bounded solution on non-Cartesian meshes. Because of the implicit implementation, the predicted value γ_f cannot be corrected explicitly but instead the weighting factor β_f has to be corrected by means of a corrector step. For a detailed derivation of the corrector step the interested reader may consult the thesis of Ubbink [21] or the paper of Ubbink and Issa [3]. Because the predictor–corrector approach outlined above can only determine the colour function field within the applied solver tolerance, the colour function value may not fulfill the criteria

$$0 \leq \gamma \leq 1. \quad (26)$$

Thus, any colour function value lower than zero or larger than unity is explicitly set to the respective bound. As demonstrated in Section 6, this does not notably affect the mass-conservation of the method.

5. Skewness correction

Skewness is a commonly encountered complication for non-Cartesian, arbitrary meshes, as for instance depicted in Fig. 2. As a result of skewness, the geometric face centre does not coincide with the interpolation point at the face, as for instance the point where the vector connecting the adjacent cell centres intersects the face, denoted in Fig. 2 as f' . Thus, the linearly interpolated face value becomes inaccurate and the accuracy of the interpolation reduces formally to first order. As demonstrated by Jasak [22], the numerical discretisation error caused by mesh skewness is diffusion-like. The advection error E_s resulting from mesh skewness for cell P is given as

$$E_{s,P} = \sum_f (\mathbf{r}_f \cdot \nabla \gamma|_f) \mathbf{u}_f \mathbf{n}_f A_f, \quad (27)$$

where \mathbf{r}_f is the vector from interpolation point f' to face centre f , as shown in Fig. 2. Defining the diffusion coefficient of the mesh skewness, following Jasak [22], as

$$\Gamma_s = \mathbf{u}_f \cdot \mathbf{r}_f, \quad (28)$$

the error induced by mesh skewness can be formulated as the diffusion term

$$E_s = \nabla \cdot (\Gamma_s \nabla \gamma). \quad (29)$$

Hence, interpolating cell-centred values to face centres without correcting for skewness, as for instance by means of Eq. (25), leads to significant numerical diffusion on meshes with appreciable skewness. Previously published simulation results [4, 21, 27, 28] indicate a considerable impact of mesh skewness on the interface advection on tetrahedral meshes compared to Cartesian meshes. In what follows, three methods to mitigate the numerical diffusion caused by mesh skewness for the compressive VOF method described in the previous section are proposed, discussed and validated.

5.1. Explicit skewness correction

The explicit correction of mesh skewness is widely applied to raise the formal accuracy of spatial interpolation on arbitrary meshes to second order and to improve the accuracy of the results [36–38]. In order to correct the interpolation for mesh skewness, the cell-centred values are interpolated to interpolation point f' , as depicted in Fig. 2. The interpolated value at point f' is then corrected to face centre f using the first derivative at face centre f . Including an explicit skewness correction, the interpolation of the colour function at face f presented in Eq. (25) becomes

$$\gamma_f = (1 - \beta_f) \gamma_D + \beta_f \gamma_A + \nabla \gamma|_f \cdot \mathbf{r}_f, \quad (30)$$

where \mathbf{r}_f is the vector from interpolation point f' to face centre f . Including the explicit skewness correction, the discretized interface advection equation, Eq. (14), follows as

$$\frac{(\gamma_P^t - \gamma_P^{t-\Delta t}) V_P}{\Delta t} = - \sum_f \frac{\gamma_f^t + \gamma_f^{t-\Delta t}}{2} F_f^{t-\Delta t} - \sum_f (\nabla \gamma|_f^{t-\Delta t} \cdot \mathbf{r}_f) F_f^{t-\Delta t}, \quad (31)$$

and Eq. (2) becomes

$$\frac{\partial \gamma}{\partial t} + u_i \frac{\partial \gamma}{\partial x_i} = - \frac{\partial}{\partial x_i} \left(\Gamma_s \frac{\partial \gamma}{\partial x_i} \right). \quad (32)$$

Essentially, the explicit skewness correction subtracts a diffusion term to Eq. (14), which counteracts the numerical diffusion induced by mesh skewness as defined in Eq. (27). As a result, however, the conservation of the colour function as described by Eq. (2) is violated and, hence, the mass of each phase is no longer conserved. The explicit skewness correction is, therefore, expected to increase the mass conservation error.

5.2. Skewness-adapted blending function

One readily available measure to tackle numerical diffusion affecting the interface advection is the application of a more compressive advection scheme. Increasing the coefficient k_ψ of the CICSAM scheme increases the dominance of the compressive Hyper-C scheme over the more diffusive UQ scheme (see Section 4.2). Applying the CICSAM scheme, a notable reduction in numerical diffusion has been observed by Ubbink [21] on triangular meshes and by Denner [28] on tetrahedral meshes for increasing k_ψ . The findings reported by Ubbink [21] and Denner [28] suggest that applying a more compressive advection scheme mitigates the adverse affects of numerical diffusion by overcompressing the interface.

It is, therefore, proposed to define the blending function ψ_f based on the skewness ξ_f of the given cell face, in addition to the angle Θ_f , with the skewness defined as [38]

$$\xi_f = \frac{|\mathbf{r}_f|}{|\mathbf{s}_f|}, \quad (33)$$

where \mathbf{r}_f is the vector from interpolation point f' to face centre f and \mathbf{s}_f is the vector connecting the donor cell and the acceptor cell. The aim is to give the compressive scheme a higher priority for faces with high skewness. The proposed blending function is defined as

$$\psi_f = \begin{cases} 1 & \text{when } \Theta_f < d_f \\ [\cos((\Theta_f - d_f)^{1+k\xi_f})]^2 & \text{when } \Theta_f \geq d_f, \end{cases} \quad (34)$$

where k is a predefined coefficient defining the strength of the compression and d_f is the phase shift of the blending function. In order to assure $\psi_f = 0$ for $\Theta_f = 90^\circ$, the phase shift is given as

$$d_f = \frac{\pi}{2} - \left(\frac{\pi}{2} \right)^{\frac{1}{1+k\xi_f}}. \quad (35)$$

For zero skewness, e.g. on an equidistant Cartesian mesh, or compression $k = 0$, the blending function reduces to the original blending function as defined in Eq. (23). Fig. 3 depicts examples of blending function ψ_f for various magnitudes of skewness ξ_f and compression coefficients k as a function of angle Θ_f . The blending function ψ_f as defined in Eq. (34) ensures the compressive scheme is also applied for larger angles Θ_f with increasing skewness, thus, counteracting numerical diffusion.

Since no additional terms are added to the interface advection equation, the mass conservation in each phase is not affected by the revised blending function. However, caution has to be exercised when overcompressing the interface by

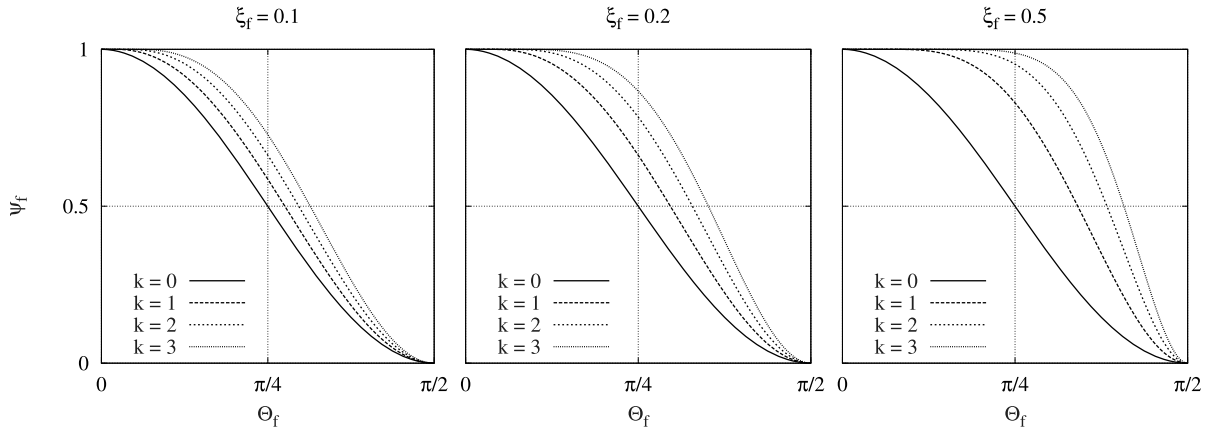


Fig. 3. Examples of the skewness-adapted blending function ψ_f as a function of skewness ξ_f and angle Θ_f for different compression coefficients k .

modifying the blending function. Adapting the blending function based on the mesh skewness in the proposed way does neither account for the magnitude nor the direction of the skewness error E_s as defined in Eq. (27). Furthermore, applying the proposed blending function may lead to local wrinkling of the interface if the compression considerably exceeds the numerical diffusion induced by mesh skewness.

5.3. Implicit skewness correction

As an alternative to the explicit skewness correction presented in Section 5.1 and the skewness-adapted blending function proposed in the previous section, an implicit skewness correction is proposed. Instead of adding an explicit term that accounts for the skewness error, the weighting factor β_f is modified to correct the skewness error. This approach follows a similar rationale as the correction of unbounded values included in the CICSAM scheme [3], where the implicit weighting factor β_f is adapted. Applying the skewness correction given in Eq. (30), the colour function corrected for skewness γ_f^s at face f is given as

$$\gamma_f^s = \gamma_f + \nabla \gamma|_f \cdot \mathbf{r}_f, \quad (36)$$

where γ_f is the original value of the colour function calculated from Eq. (25). Following the definition of the face value presented in Eq. (25), Eq. (36) can be written as

$$(1 - \beta_f^s) \gamma_D + \beta_f^s \gamma_A = (1 - \beta_f) \gamma_D + \beta_f \gamma_A + \nabla \gamma|_f \cdot \mathbf{r}_f, \quad (37)$$

where β_f^s is the skewness-corrected weighting factor for γ_f^s . After rearranging, the skewness-corrected weighting factor β_f^s follows as

$$\beta_f^s = \beta_f + \frac{\nabla \gamma|_f \cdot \mathbf{r}_f}{\gamma_A - \gamma_D}. \quad (38)$$

In order to obtain a bounded solution, β^s is explicitly bounded to $0 \leq \beta^s \leq 1$. The revised weighting factor β^s , therefore, corrects the face value of the colour function implicitly if skewness is present, taking into account the values at the donor cell and the acceptor cell as well as the magnitude and the direction of the colour function gradient. Applying the revised weighting factor β_f^s instead of β_f in Eq. (25) is expected to significantly reduce numerical diffusion as a result of mesh skewness. Moreover, since no additional terms are added to the interface advection equation, Eq. (2), by applying the implicit skewness correction, the discretisation retains its original mass-conserving properties, as demonstrated in Section 6.

6. Results

The compressive VOF methodology described in Section 4, as well as the remedies for mesh skewness proposed in Section 5 are validated and compared using three representative test cases. The pure advection of a fluid particle in a constant velocity field, see Section 6.1, and in a shear flow, see Section 6.2, is simulated. These test cases allow the assessment of the proposed methods regarding accuracy and mass conservation without the influence of velocity gradients, pressure gradients or surface tension. Subsequently, the rise of a bubble in an initially quiescent fluid is presented in Section 6.3, focusing on the shape evolution of the bubble and the development of its rise velocity. Two bubbles with different properties are considered. The accurate advection of the colour function is crucial in this test case, since the acting surface tension as well as the buoyancy force and the viscous stresses are directly dependent on the colour function field. In a previous study,

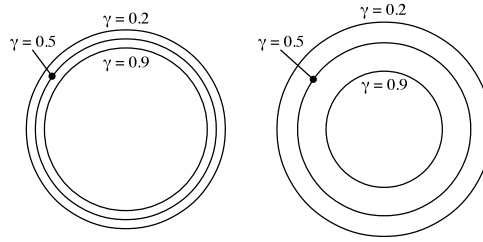


Fig. 4. Example of the spatial spreading of the colour function as a result of numerical diffusion. The depicted iso-contours represent $\gamma = 0.2$, $\gamma = 0.5$ and $\gamma = 0.9$.

Denner and van Wachem [27] observed significant differences in the predicted behaviour between Cartesian and tetrahedral meshes for this test case with respect to the bubble rise velocity and the numerical diffusion of the colour function.

Two reference values are defined to quantify the conservation of mass and the numerical diffusion of the tested methods. Assuming a mass-conserving interface capturing/tracking method is deployed in a closed computational domain, the total volume of each phase must be conserved. Hence, the volume of colour function in the domain,

$$V_\gamma = \sum_{p=1}^N \gamma_p V_p, \quad (39)$$

where N is the total number of mesh cells, has to be conserved as long as no colour function $\gamma \neq 0$ is advected out of the domain, regardless of the presence or extend of numerical diffusion. The conservation error of the colour function $E(V_\gamma)$ is defined as

$$E(V_\gamma) = \frac{|V_\gamma - V_{\gamma,0}|}{V_{\gamma,0}}, \quad (40)$$

where $V_{\gamma,0}$ is the initial volume of the colour function. However, even given perfect conservation of the colour function, the distribution of the colour function is affected by numerical diffusion. Numerical diffusion means that iso-contours of the colour function are spreading out over time and, as a consequence, the volume enclosed by a given iso-contour, defined as

$$V(\gamma) = \sum_{p=1}^N \begin{cases} 0 & \text{if } \gamma \leq a \\ \gamma_p V_p & \text{if } \gamma > a, \end{cases} \quad (41)$$

changes, as depicted in Fig. 4. Tracking the temporal evolution of the relative volume V^* for a representative colour function value,

$$V^*(\gamma) = \frac{V(\gamma)}{V_0(\gamma)}, \quad (42)$$

with $V_0(\gamma)$ being the initial volume enclosed by the iso-contour of γ , provides a means to quantify numerical diffusion.

6.1. Advection of a circular interface in a constant velocity field

The advection of a circular interface on a tetrahedral mesh is simulated to examine the ability of the presented methods to preserve the shape of the interface. The domain has a size of $2 \text{ m} \times 1 \text{ m} \times 0.05 \text{ m}$ and is resolved with a tetrahedral mesh of approximately 4.0×10^5 cells. The circular interface has an initial diameter of 0.1 m and its centre is initially positioned at $\mathbf{x}(t = 0 \text{ s}) = (0.15, 0.15, 0) \text{ m}$, as depicted in Fig. 5. The velocity field is uniform with $\mathbf{u} = (1.7, 0.7, 0.0) \text{ ms}^{-1}$ and the time-step corresponds to a maximum Courant number of 0.1 . Since the advection is constant and no forces are acting, the interface should keep its original shape. Given the domain dimensions and the applied velocity, the interface centre should be positioned at $\mathbf{x}(t = 1 \text{ s}) = (1.85, 0.85, 0) \text{ m}$ at the end of the simulation.

Fig. 6 shows the final colour function distribution for the original method as presented in Section 4 and the results including the modifications proposed in Section 5. Even though the temporal and spatial velocity gradients are zero, this simple convection test unveils considerable difficulties of state-of-the-art compressive VOF methods in preserving the shape of the interface on the applied tetrahedral mesh. As observed in Fig. 6(a), the shape of the fluid particle is notably distorted if no numerical treatment for mesh skewness is applied. The fluid particle disintegrates when the explicit skewness correction presented in Section 5.1 is applied, shown in Fig. 6(b). The application of the skewness-adapted blending function, proposed in Section 5.2, does not result in considerable differences compared to the original method. As observed in Figs. 6(c)–(e), a higher compression coefficient k leads to a better preservation of the fluid particle shape. Applying the implicit skewness correction proposed in Section 5.3, the interface is advected accurately and the final shape of the fluid particle is in very good agreement with its initial shape, as shown in Fig. 6(f).

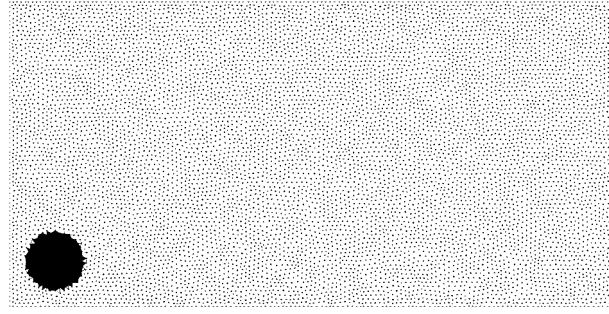


Fig. 5. Mesh and initial colour function distribution for the constant advection case at $t = 0$ s.

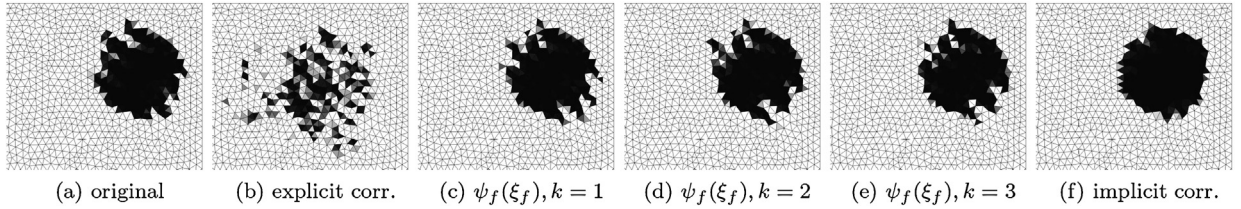


Fig. 6. Final distribution of the colour function for the constant advection case at $t = 1$ s.

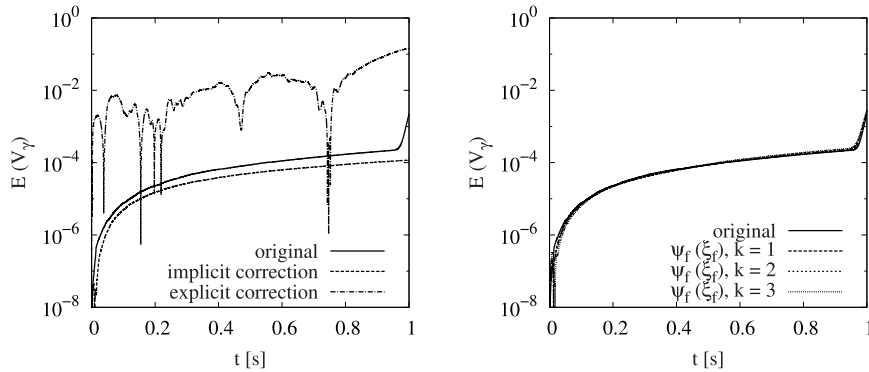


Fig. 7. Conservation error of the total volume of the colour function $E(V_\gamma)$ as a function of time for the constant advection case.

With respect to mass conservation, given an applied solver tolerance of 10^{-6} and 7716 time-steps performed in the simulation, the error in colour function volume of the order of 10^{-4} , as shown in Fig. 7, for the original methodology without modifications as well as including the implicit skewness correction or the skewness-adapted blending function is well within the solver tolerance. The explicit skewness correction, however, leads to an unacceptable error in volume conservation larger than 10%. The sharp rise in conservation error at the end of the simulation observed in Fig. 7 for the original method without modifications and including the modified blending function indicates that in these cases a small amount of colour function is already advected out of the domain, indicating considerable numerical diffusion since the interface should remain entirely in the domain. Considering the volume occupied by the iso-contour representing $\gamma = 0.9$ as a function of time in Fig. 8, a similar conclusion with regards to the impact of the modifications to the accuracy of the results can be drawn as from the qualitative assessment of Fig. 6. Applying the implicit skewness correction clearly results in the smallest diffusion of the colour function field, evident by the almost constant volume covered enclosed in the iso-contour. The explicit skewness correction causes the largest errors, whereas the skewness-adapted blending function and the original method are of similar accuracy.

6.2. Advection of a circular interface in shear flow

A circular interface in a shear flow is simulated to assess the accuracy of the proposed methods for shear-driven interface advection. This case represents a more challenging test of the interface advection algorithm as the advection in a constant velocity field presented in the previous section, and has been used in a number of similar studies (e.g. [3,18,39]). The domain has a size of $\pi \text{ m} \times \pi \text{ m} \times 0.1\pi \text{ m}$ and the applied velocity field is given as

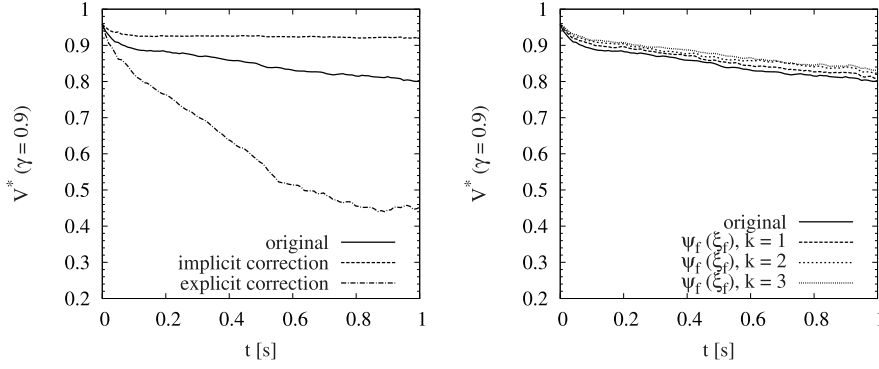


Fig. 8. Relative volume V^* enclosed by the iso-contour of $\gamma = 0.9$ as a function of time for the constant advection case.

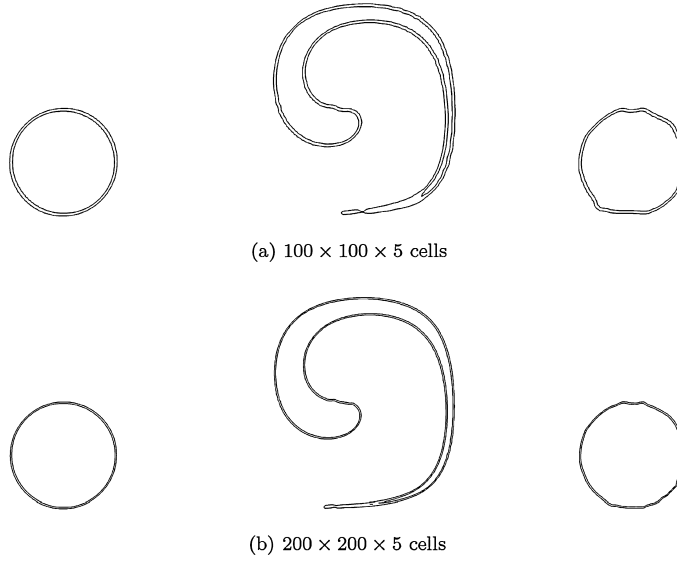


Fig. 9. Initial ($t = 0$ s), intermediate ($t = 12.5$ s) and final ($t = 25$ s) result of the shear advection test on two equidistant Cartesian meshes. The iso-contour lines represent $\gamma = 0.25$ and $\gamma = 0.75$.

$$\mathbf{u}(x, y, z) = \begin{pmatrix} \sin(x) \cos(y) \\ -\cos(x) \sin(y) \\ 0 \end{pmatrix} \text{ m s}^{-1}. \quad (43)$$

The applied tetrahedral mesh resolves the domain by approximately 2.38×10^5 cells. Reference results on Cartesian meshes of $100 \times 100 \times 5$ cells and $200 \times 200 \times 5$ cells are also provided. Initially, the circular interface has a diameter of 0.2π m and is positioned at $(0.5\pi, 0.25\pi, 0)$ m. The interface is advected with the velocity given in Eq. (43) for 12.5 s with a Courant number of 0.1. Subsequently, the flow field is inverted and the interface is advected for another 12.5 s. Hence, assuming a perfect advection of the interface, the interface should assume its initial shape at the end of the simulation.

The results obtained on the Cartesian meshes, depicted in Fig. 9, show the initialised interface at $t = 0$ s as well as the intermediate result at $t = 12.5$ s and the final result at $t = 25$ s. The tail of the stretched interface at the shown intermediate stage ($t = 12.5$ s) is naturally more detailed on the finer mesh. The final interface shape at $t = 25$ s on both meshes is in excellent agreement with the initial interface shape and only minor differences are noticeable. The major difference between the two Cartesian meshes is the interface thickness, which directly depends on the resolution of the mesh.

Noticeable deficiencies are observed for the results obtained on the tetrahedral mesh, shown in Figs. 10 and 11, compared with the results of the Cartesian meshes. Using the compressive VOF method in its original form without modifications, the tail of the sheared interface at the intermediate stage is thicker and shorter than on the Cartesian meshes, as shown in Fig. 10(a). The interface in the final result, given in Fig. 11(a), has some resemblance of the initially circular interface but also shows significant distortion. The explicit correction is distorting the colour function field to an unrecognisable extent, as seen in Figs. 10(b) and 11(b), and is clearly inapplicable in this particular test case. The results obtained using the skewness-adapted blending function, shown in Figs. 10(c)–(e) and 11(c)–(e), retain the major interface features but show an increased wrinkling and distortion of the interface with increased coefficient k . The wrinkling may be a result of the overcompression of the interface. Applying the implicit skewness correction proposed in Section 5.3, the thickness of the

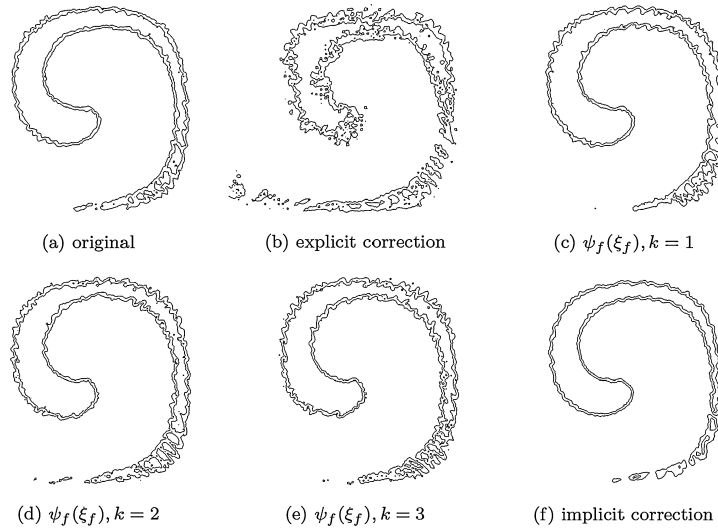


Fig. 10. Intermediate result at $t = 12.5$ s of the shear advection test on the tetrahedral mesh. The iso-contour lines represent $\gamma = 0.25$ and $\gamma = 0.75$.

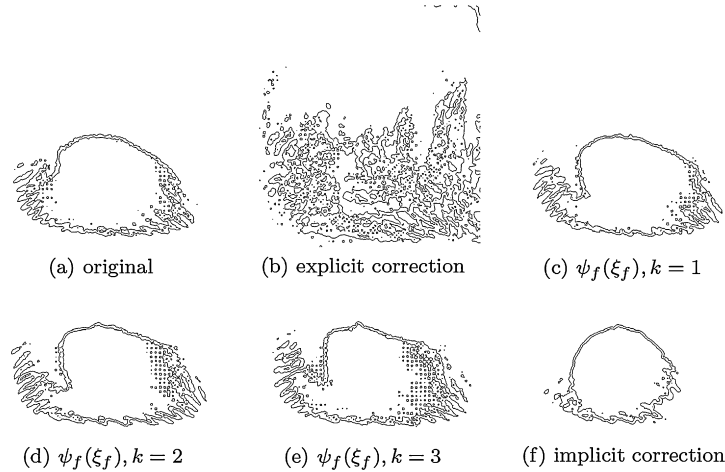


Fig. 11. Final result at $t = 25.0$ s of the shear advection test on the tetrahedral mesh. The iso-contour lines represent $\gamma = 0.25$ and $\gamma = 0.75$.

tail in the intermediate result, see Fig. 10(f), is in better agreement with the result of the Cartesian meshes. The final result obtained with the implicit correction, shown in Fig. 11(f), shows considerably less distortion than the results obtained with the unmodified VOF method and the result is also in reasonable agreement with the initial interface shape.

6.3. Rising bubble

A bubble rising in a heavier fluid due to buoyancy is simulated, characterised by its Morton number $Mo = g\mu_o^4/\rho_o\sigma^3$ and its Eötvös number $Eo_d = \rho_o g d_0^2/\sigma$. Two non-dimensional sets of properties are considered: a) $Eo_d = 40$, $Mo = 0.056$ and b) $Eo_d = 243$, $Mo = 266$. The influence of surface tension and inertia is stronger for bubble a) than for bubble b), whereas viscous stresses play a more pertinent role for bubble b).

For both bubbles the fluid of the continuous phase has a density of $\rho_o = 1000 \text{ kg m}^{-3}$. The viscosity of the continuous phase is $\mu_o = 0.2736 \text{ Pas}$ for the bubble with $Eo_d = 40$, $Mo = 0.056$ and $\mu_o = 0.5869 \text{ Pas}$ for the bubble with $Eo_d = 243$, $Mo = 266$. The properties of the dispersed phase are set according to the chosen density and viscosity ratios. Following the suggestions of Lebaigue et al. [40] and to make the results comparable to previously reported results of the chosen bubble [20,41], the density and viscosity ratio of the bubble with $Eo_d = 40$ and $Mo = 0.056$ is $\rho_i/\rho_o = \mu_i/\mu_o = 10^{-2}$. The ratio of density and viscosity of the bubble with $Eo_d = 243$ and $Mo = 266$ are $\rho_i/\rho_o = 10^{-3}$ and $\mu_i/\mu_o = 10^{-2}$, in order to make the results comparable to previously reported experimental [42] and numerical studies [41,43,44]. In both cases the bubble is initially spherical with a diameter of $d_0 = 0.02 \text{ m}$ and a surface tension coefficient defined by the non-dimensional parameters given above. The gravitational acceleration of $g = 10 \text{ ms}^{-2}$ is acting in negative y -direction. Both fluids are initially at rest and the motion of the bubble is induced by buoyancy only. The domain is of size $5d_0 \times 7d_0 \times 5d_0$ and is resolved with a tetrahedral mesh of approximately 1.4×10^6 cells. Reference results are also provided using an equidistant

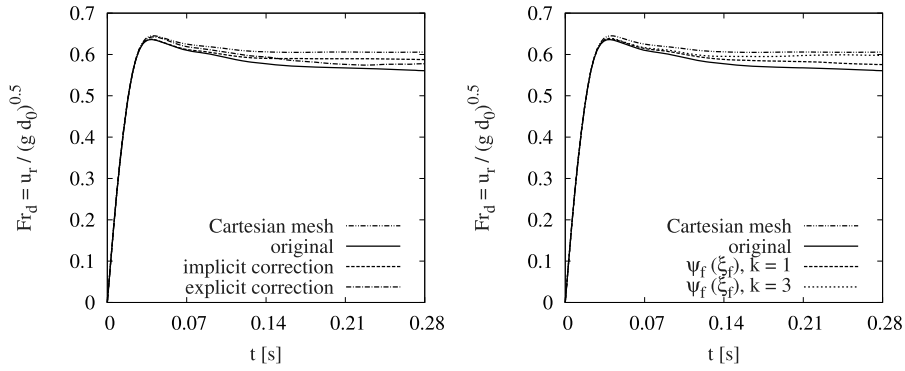


Fig. 12. Rise velocity of the bubble with $Eu_d = 40$, $Mo = 0.056$ as a function of time on the equidistant Cartesian mesh and the tetrahedral mesh, including the proposed modifications.

Cartesian mesh of $100 \times 140 \times 100$ cells. For all cases, the applied time-step fulfills the capillary time-step constraint of Brackbill et al. [32], given as

$$\Delta t \leq \sqrt{\frac{(\rho_i + \rho_o) \Delta x^3}{4\pi\sigma}}, \quad (44)$$

where Δx represents the mesh spacing. For cases simulated on the tetrahedral mesh, the average distance between neighbouring cell centres in the interfacial region is taken as the representative mesh spacing to determine the capillary time-step constraint. This time-step corresponds to a Courant number of $Co \leq 0.83$ in all cases. Using the dual time-stepping approach described in Section 3, the time-step applied to advect the VOF colour function is dynamically adapted to satisfy $Co = 0.01$. The boundary at the top of the domain is considered to be an outlet boundary, all other boundaries are free-slip walls. The interface curvature is evaluated using the CELESTE method [27] and the fluid properties are convoluted at the interface as described in [45].

The finite size of the fluid domain affects the rise velocity of the bubble, which is important for the correct interpretation of the simulation results. Following the work of Harmathy [46], the expected rise velocity in a finite domain is

$$\mathbf{u}_r \approx \mathbf{u}_r^\infty \left[1 - \left(\frac{d_0}{L} \right)^2 \right], \quad (45)$$

where L is the domain size perpendicular to the rise axis and \mathbf{u}_r^∞ is the rise velocity in a domain of infinite extend. Given the domain width of $5d_0$, the rise velocity in the deployed computational domain is expected to be 96% of the rise velocity observed in a domain of infinite extend.

6.3.1. $Eu_d = 40$, $Mo = 0.056$

Empirical studies by Clift et al. [47, Fig 2.5] suggest a terminal Reynolds number of $Re_d = |\mathbf{u}_r| \rho_o d_0 / \mu_o \approx 20.5\text{--}21.0$ for a bubble with $Eu_d = 40$ and $Mo = 0.056$. This Reynolds number represents a Froude number of $Fr_d = |\mathbf{u}_r| / \sqrt{d_0 g} \approx 0.626\text{--}0.642$ with respect to the fluid properties given above. Hence, considering the finite extend of the computational domain, the expected terminal Froude number is $Fr_d \approx 0.601\text{--}0.616$. Numerical studies of Lubin et al. [48], using a computational domain of unspecified extend, resulted in a terminal Froude number of 0.60. Denner et al. [20] reported a terminal Froude number of 0.58 using a VOF-PLIC method in a domain of $5d_0$ width and Pivello et al. [41] obtained a terminal Froude number $Fr_d = 0.606$ with a front-tracking method in a domain of $8d_0$ width.

Fig. 12 depicts the temporal evolution of the rise velocity of the bubble for the presented method and the proposed modifications on the tetrahedral mesh in comparison with the results obtained on the equidistant Cartesian mesh. Applying the presented compressive VOF method without modifications on the tetrahedral mesh results in a continuously decreasing rise velocity after the initial acceleration and does not reach a terminal rise velocity in the simulated time frame, as observed on the Cartesian mesh. Based on the similar evolution of the bubble shapes, shown in Fig. 13, on the Cartesian mesh and the tetrahedral mesh applying the compressive VOF method without modifications, a difference in drag force can be disregarded as a potential explanation for the difference in rise velocity. Denner and van Wachem [27] attributed the progressively decreasing rise velocity to a gradual reduction of the effective density difference between the fluids as a result of numerical diffusion of the colour function. Each of the three skewness correction methods improves the prediction of the rise velocity, as observed in Fig. 12. However, the explicit skewness correction distorts the bubble considerably, as shown in Fig. 13, whereas the implicit correction as well as the skewness-adapted blending function do not notably affect the bubble shape. Interestingly, the skewness-adapted blending function with a compression coefficient $k = 3$ provides an accuracy very similar to the implicit skewness correction for this test case. As mentioned in Section 5.2, adapting the blending function based on the local mesh skewness compresses the interface regardless of the direction or magnitude of the skewness error. Thus, the success of adapting the blending function strongly depends on the local mesh errors and the flow characteristics.

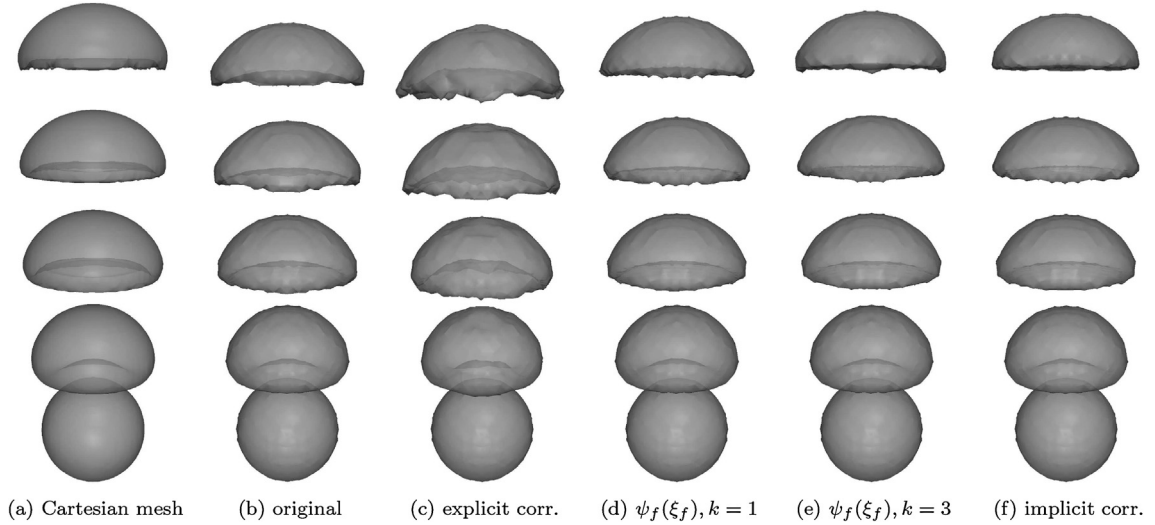


Fig. 13. Shape evolution of the rising bubble with $Eu_d = 40$, $Mo = 0.056$ on the Cartesian mesh and the tetrahedral mesh, including the proposed modifications. The bubble shapes are illustrated every 0.07 s, starting from the initial position, with the iso-contours representing $\gamma = 0.5$.

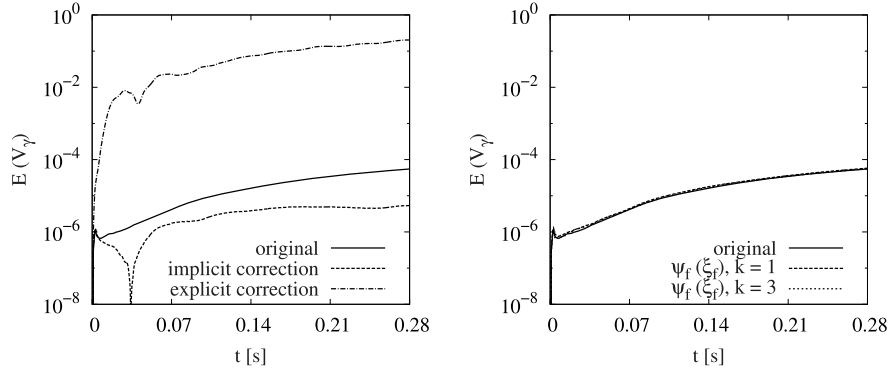


Fig. 14. Conservation error of the total volume of the colour function $E(V_\gamma)$ as a function of time for the rising bubble ($Eu_d = 40$, $Mo = 0.056$).

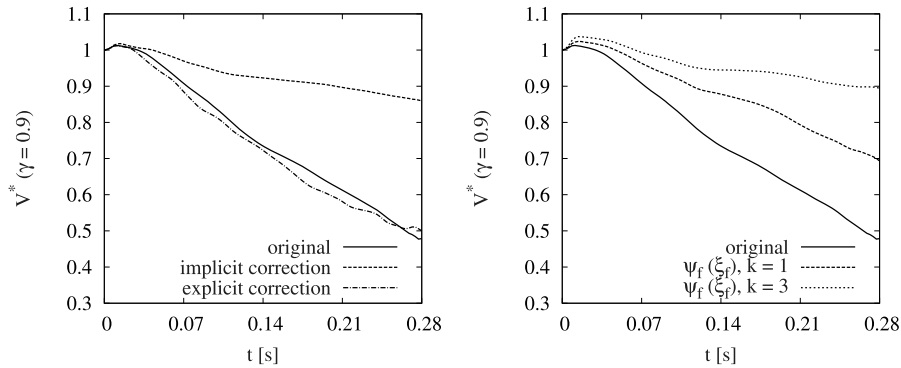


Fig. 15. Relative volume V^* enclosed by the iso-contour of $\gamma = 0.9$ as a function of time for the rising bubble ($Eu_d = 40$, $Mo = 0.056$).

The conservation error of the colour function, depicted in Fig. 14, presents a familiar picture. The conservation error of the presented VOF method without modifications as well as with the implicit skewness correction or the skewness-adapted blending function is negligible, whereas the explicit skewness correction leads to a substantial conservation error. The gain in colour function observed for the explicit skewness correction increases the bubble volume and, as a result, increases the acting buoyancy and the rise velocity of the bubble. According to the evolution of the volume enclosed by the iso-contour of $\gamma = 0.9$, shown in Fig. 15, the implicit skewness correction and the skewness-adapted blending function reduce the numer-

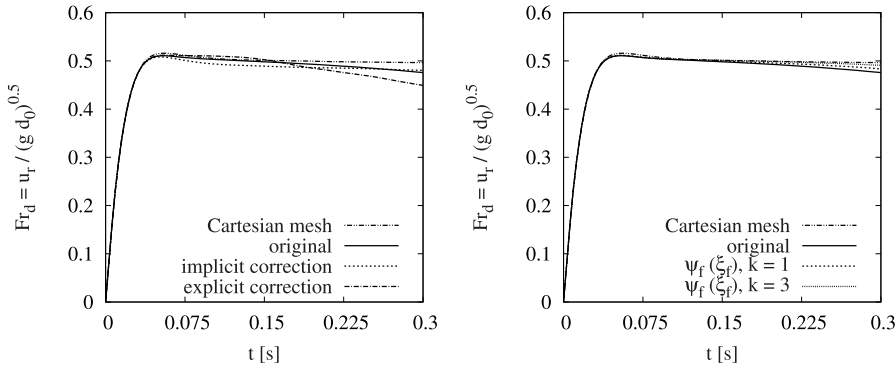


Fig. 16. Rise velocity of the bubble with $Eo_d = 243$, $Mo = 266$ as a function of time on the equidistant Cartesian mesh and the tetrahedral mesh, including the proposed modifications.

ical diffusion considerably compared to the unmodified compressive VOF method and the explicit skewness correction. This corresponds very well with the predicted rise velocities, especially the continuous reduction in rise velocity for the compressive VOF method without modifications, and, more generally, further highlights the importance of mitigating numerical diffusion on arbitrary, unstructured meshes.

The simulation of the rising bubble on the tetrahedral mesh without skewness correction requires on average 22.6% more computational time to complete each time-step than on the Cartesian mesh. With respect to the three skewness correction methods, the required computational time on the tetrahedral mesh increases by 32.2% using the explicit correction, by 26.5% using the skewness-adapted blending function and by 15.3% using the implicit correction, compared to the simulation without skewness correction.

6.3.2. $Eo_d = 243$, $Mo = 266$

Experiments conducted by Bhaga and Weber [42] for a bubble with $Eo_d = 243$ and $Mo = 266$ rising freely due to buoyancy resulted in a terminal Reynolds number of $Re_d = 7.77$, equivalent to $Fr_d = 0.51$, given the fluid properties applied in the present study. The experimental apparatus used by Bhaga and Weber [42] can be assumed to represent a domain of infinite size and the optical method used to determine the rise velocity of the bubble has a reported tolerance of 5–7%, subject to the shape of the bubble. The experimental results of Bhaga and Weber corresponds well with the empirical studies of Cliff et al. [47, Fig. 2.5], suggesting a terminal Reynolds number of $Re_d \approx 8$. Numerical simulations using a front-tracking method reported by Hua et al. [43] predicted $Re_d = 7.605$ and $Fr_d = 0.499$ in a domain of $8d_0$ width with a resolution of 20 cells per bubble diameter. For the same domain width, with a mesh resolution of 32 cells per bubble diameter and also using a front-tracking method, Pivello et al. [41] reported a terminal Reynolds number of $Re_d = 7.67$, representing a Froude number of $Fr_d = 0.503$, for the density ratio applied in the present study. Considering the domain width of $8d_0$, the results of Hua et al. [43] and Pivello et al. [41] are in very good agreement with the expected Froude number $Fr_d = 0.984$ $Fr_{d,\infty} = 0.502$ following Eq. (45), with $Fr_{d,\infty}$ being the Froude number based on the rise velocity in a domain of infinite size. Axisymmetric simulations using a coupled level-set/moment-of-fluid method published by Jemison et al. [44] predicted a terminal Reynolds number of $Re_d = 7.77$ in a domain of $4d_0$ width and a mesh resolution of 64 cells per bubble radius, precisely matching the experimental result. Jemison et al. [44] also reported consistently exact solutions of the rise velocity for other bubbles with respect to the experiments of Bhaga and Weber [42], which can be considered a rather unexpected outcome given the tolerance of the experiments and the impact of the finite width of the computational domain used by Jemison and co-workers.

The rise velocity as a function of time, depicted in Fig. 16, shows a similar trend, albeit smaller differences between the tested methods, as the rise velocity of the bubble with $Eo_d = 40$ and $Mo = 0.056$. As previously mentioned, the motion of this bubble is less governed by inertia as the bubble discussed in the previous section, which may explain the different impact of the discretisation of the spatial advection and the applied correction of mesh skewness. Apart from the Froude number obtained by applying the explicit skewness correction on the tetrahedral mesh, the predicted Froude numbers on the Cartesian mesh and the tetrahedral mesh are in very good agreement with the expected terminal Froude number of 0.4895, based on the experimental results and the size of the computational domain. The rise velocity recorded on the tetrahedral mesh without correcting for skewness continuously decreases. A similar development can be observed when the explicit skewness correction or the skewness-adapted blending function are applied. The implicit correction, on the other hand, provides a stable and accurate prediction of the rise velocity. Fig. 17 shows the shape of the bubble on the equidistant Cartesian mesh and the tetrahedral mesh. Apart from the result on the tetrahedral mesh with the explicit skewness correction, the bubble shapes obtained on the tetrahedral mesh are in excellent agreement with the bubble shapes predicted on the equidistant Cartesian mesh. Similar as for the rising bubble discussed in the previous section, the explicit skewness correction distorts the bubble considerably.

The conservation of the colour function, shown in Fig. 18, is very similar to the conservation of the colour function for the test cases presented above. An excellent conservation of the colour function is observed for the compressive VOF method

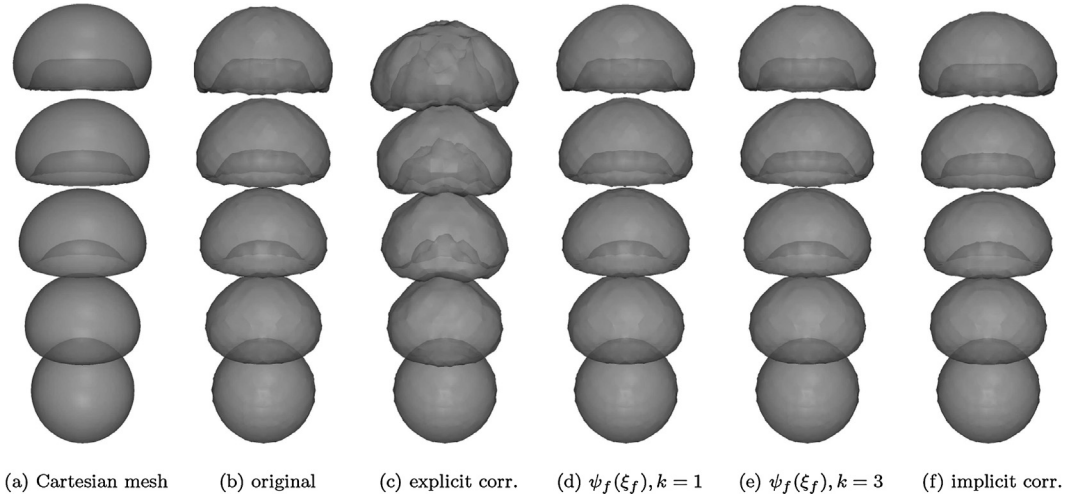


Fig. 17. Shape evolution of the rising bubble with $Eo_d = 243$, $Mo = 266$ on the Cartesian mesh and the tetrahedral mesh, including the proposed modifications. The bubble shapes are illustrated every 0.075 s, starting from the initial position, with the iso-contours representing $\gamma = 0.5$.

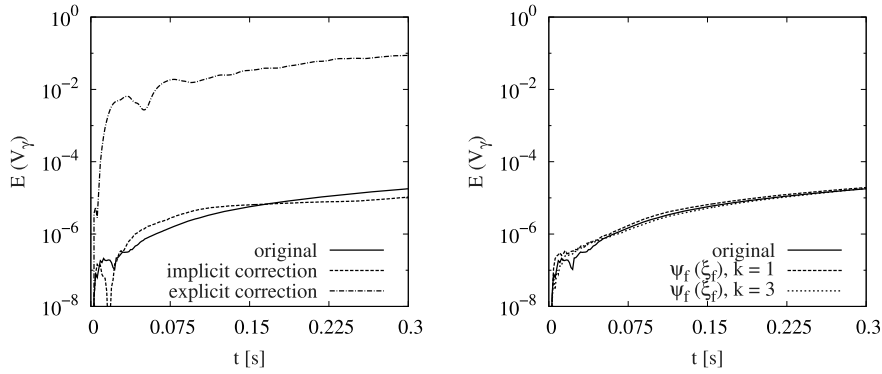


Fig. 18. Conservation error of the total volume of the colour function $E(V_\gamma)$ as a function of time for the rising bubble ($Eo_d = 243$, $Mo = 266$).

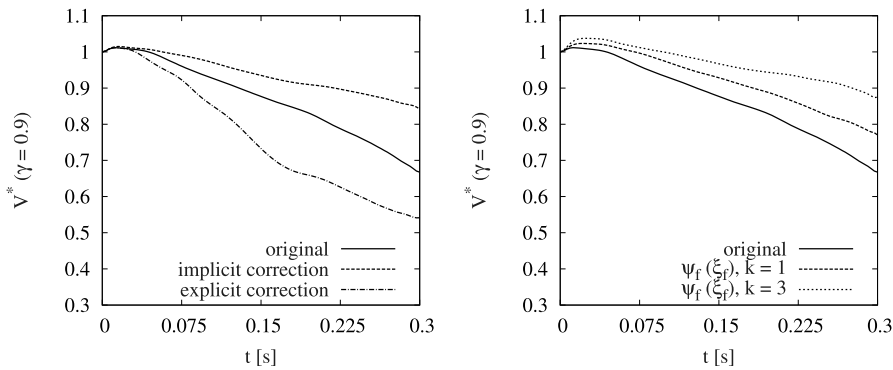


Fig. 19. Relative volume V^* enclosed by the iso-contour of $\gamma = 0.9$ as a function of time for the rising bubble ($Eo_d = 243$, $Mo = 266$).

without modifications as well as including the implicit skewness correction or the skewness-adapted blending function, whereas the conservation of the colour function is clearly violated by the explicit skewness correction. The evolution of the volume inside the iso-contour of $\gamma = 0.9$, presented in Fig. 19, is also very similar to the results obtained for the bubble with different properties discussed in the previous section. The implicit correction as well as the skewness-adapted blending function evidently reduce the numerical diffusion compared to the unmodified compressive VOF method. The differences between the modifications are smaller than seen for the other bubble in the previous section, which corresponds well with the smaller differences in rise velocity observed in Fig. 18.

Contrary to the simulation of the bubble with $Eo = 40$ presented in the previous section, the simulation of the bubble with $Eo = 243$ is faster on the tetrahedral mesh without skewness correction than on the Cartesian mesh, requiring on average 18.3% less computation time per time-step. Applying the explicit skewness correction on the tetrahedral mesh increases the simulation time of each time-step by 66.7%, by 83.8% using the skewness-adapted blending function and by 18.3% using the implicit skewness correction.

7. Conclusion

In this article, the discretisation and implementation of a compressive VOF method for three-dimensional arbitrary meshes has been presented. The presented compressive VOF methodology is applicable to arbitrary, unstructured meshes and conserves the mass of each phase within the limits of the applied solver tolerance. However, as reported in previous studies [4,21,27,28], the predictive quality of compressive VOF methods on unstructured meshes is severely challenged by numerical diffusion. Although numerical diffusion of the VOF colour function has been comprehensively studied with respect to the applied differencing schemes, the numerical diffusion of the VOF colour function induced by an inferior mesh quality, *i.e.* mesh skewness, has not been the subject of published research yet.

In order to mitigate the adverse effects of mesh skewness, three skewness correction methods have been proposed and validated. The presented results demonstrate the profound and consistent improvement in accuracy achieved by the proposed implicit skewness correction. The implicit skewness correction does not affect the mass-conserving properties of compressive VOF methods and, in terms of accuracy, the implicit skewness correction brings the results obtained on tetrahedral meshes in reach of simulations conducted on equidistant Cartesian meshes. The results also demonstrate that the explicit correction of skewness is not a reliable remedy for the numerical diffusion of the VOF colour function and that it compromises the accurate mass conservation originally provided by compressive VOF methods. The adaptation of the blending function of the spatial discretisation schemes in response to the local mesh skewness reduces numerical diffusion, but the indiscriminate compression of the interface, without accounting for the direction and magnitude of the skewness error, affect the consistency of the solution accuracy.

The presented study also highlights the capabilities of compressive VOF methods in general as well as their applicability to arbitrary meshes, and further supports previous studies [18–20] which suggest that compressive VOF methods are able to predict two-phase flows with an accuracy comparable to interface reconstruction methods, such as PLIC methods. Evidently, discretisation errors induced by the computational mesh have a crucial influence on the predictive quality of two-phase flow simulations and, as demonstrated in this article, accounting for these errors is essential to make two-phase flow modelling on tetrahedral and polyhedral meshes accurate and reliable.

Acknowledgement

The authors are grateful to the Engineering and Physical Sciences Research Council (EPSRC) for their financial support (grant EP/K008595/1).

References

- [1] C. Hirt, B. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, *J. Comput. Phys.* 39 (1981) 201–225.
- [2] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, G. Zanetti, Modelling merging and fragmentation in multiphase flows with SURFER, *J. Comput. Phys.* 113 (1994) 134–147.
- [3] O. Ubbink, R. Issa, A method for capturing sharp fluid interfaces on arbitrary meshes, *J. Comput. Phys.* 153 (1999) 26–50.
- [4] M. Darwish, F. Moukalled, Convective schemes for capturing interfaces of free-surface flows on unstructured grids, *Numer. Heat Transf., Part B, Fundam.* 49 (2006) 19–42.
- [5] S. Muzaferija, M. Perić, P. Sames, P. Schellin, A two-fluid Navier–Stokes solver to simulate water entry, in: *Proceedings of the Twenty-Second Symposium on Naval Hydrodynamics*, Washington DC, USA, 1999, pp. 638–649.
- [6] D.L. Youngs, Time-dependent multi-material flow with large fluid distortion, in: K. Morton, M. Baines (Eds.), *Numerical Methods for Fluid Dynamics*, Academic Press, New York, 1982, p. 273.
- [7] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, *J. Comput. Phys.* 141 (1998) 112–152.
- [8] B. van Wachem, J. Schouten, Experimental validation of 3-D Lagrangian VOF model: bubble shape and rise velocity, *AIChE J.* 48 (2002) 2744–2753.
- [9] J.E. Pilliod, E.G. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, *J. Comput. Phys.* 199 (2004) 465–502.
- [10] E. Aulisa, S. Manservigi, R. Scardovelli, S. Zaleski, Interface reconstruction with least-squares fit and split advection in three-dimensional Cartesian geometry, *J. Comput. Phys.* 225 (2007) 2301–2319.
- [11] J. López, J. Hernández, Analytical and geometrical tools for 3D volume of fluid methods in general grids, *J. Comput. Phys.* 227 (2008) 5939–5948.
- [12] C. Wu, D. Young, H. Wu, Simulations of multidimensional interfacial flows by an improved volume-of-fluid method, *Int. J. Heat Mass Transf.* 60 (2013) 739–755.
- [13] Y. Renardy, M. Renardy, PROST: a parabolic reconstruction of surface tension for the volume-of-fluid method, *J. Comput. Phys.* 183 (2002) 400–421.
- [14] T. Maric, H. Marschall, D. Bothe, voFoam – a geometrical volume of fluid algorithm on arbitrary unstructured meshes with local dynamic adaptive mesh refinement using OpenFOAM, arXiv:1305.3417, 2013, pp. 1–30.
- [15] J. Mencinger, I. Žun, A PLIC–VOF method suited for adaptive moving grids, *J. Comput. Phys.* 230 (2011) 644–663.
- [16] Z. Wang, J. Yang, F. Stern, A new volume-of-fluid method with a constructed distance function on general structured grids, *J. Comput. Phys.* 231 (2012) 3703–3722.
- [17] M. Huang, L. Wu, B. Chen, A piecewise linear interface-capturing volume-of-fluid method based on unstructured grids, *Numer. Heat Transf., Part B, Fundam.* 61 (2012) 412–437.
- [18] V.R. Gopala, B.G.M. van Wachem, Volume of fluid methods for immiscible-fluid and free-surface flows, *Chem. Eng. J.* 141 (2008) 204–221.

- [19] I. Park, K. Kim, J. Kim, A volume-of-fluid method for incompressible free surface flows, *Int. J. Numer. Methods Fluids* 61 (2009) 1331–1362.
- [20] F. Denner, D. van der Heul, G. Oud, M. Villar, A. da Silveira Neto, B. van Wachem, Comparative study of mass-conserving interface capturing frameworks for two-phase flows with surface tension, *Int. J. Multiph. Flow* 61 (2014) 37–47.
- [21] O. Ubbink, Numerical prediction of two fluid systems with sharp interfaces, Ph.D. thesis, Imperial College London, 1997.
- [22] H. Jasak, Error analysis and estimation for the finite volume method with applications to fluid flow, Ph.D. thesis, Imperial College London, 1996.
- [23] M. Darwish, Development and testing of a robust free-surface finite volume method, Technical Report, Faculty of Engineering and Architecture, American University of Beirut, Beirut, Lebanon, 2003.
- [24] F. Moukalled, M. Darwish, Transient schemes for capturing interfaces of free-surface flows, *Numer. Heat Transf., Part B, Fundam.* 61 (2012) 171–203.
- [25] J. Heyns, A. Malan, T. Harms, O. Oxtoby, Development of a compressive surface capturing formulation for modelling free-surface flow by using the volume-of-fluid approach, *Int. J. Numer. Methods Fluids* 71 (2013) 788–804.
- [26] D. Zhang, C. Jiang, C. Yang, Y. Yang, Assessment of different reconstruction techniques for implementing the NVSF schemes on unstructured meshes, *Int. J. Numer. Methods Fluids* 74 (2014) 189–221.
- [27] F. Denner, B. van Wachem, Fully-coupled balanced-force VOF framework for arbitrary meshes with least-squares curvature evaluation from volume fractions, *Numer. Heat Transf., Part B, Fundam.* 65 (2014) 218–255.
- [28] F. Denner, Balanced-force two-phase flow modelling on unstructured and adaptive meshes, Ph.D. thesis, Imperial College London, 2013.
- [29] H. Rusche, Computational fluid dynamics of dispersed two-phase flows at high phase fractions, Ph.D. thesis, Imperial College London, 2002.
- [30] Y.-Y. Tsui, S.-W. Lin, T.-T. Cheng, T.-C. Wu, Flux-blending schemes for interface capture in two-fluid flows, *Int. J. Heat Mass Transf.* 52 (2009) 5547–5556.
- [31] Y. Morinishi, T. Lund, O. Vasilyev, P. Moin, Fully conservative higher order finite difference schemes for incompressible flow, *J. Comput. Phys.* 143 (1998) 90–124.
- [32] J.U. Brackbill, D. Kothe, C. Zemach, Continuum method for modeling surface tension, *J. Comput. Phys.* 100 (1992) 335–354.
- [33] B.P. Leonard, The ULTIMATE conservative difference scheme applied to unsteady one-dimensional advection, *Comput. Methods Appl. Mech. Eng.* 88 (1991) 17–74.
- [34] P. Gaskell, A. Lau, Curvature-compensated convective transport: SMART, a new boundedness-preserving transport algorithm, *Int. J. Numer. Methods Fluids* 8 (1988) 617–641.
- [35] H. Jasak, H. Weller, A. Gosman, High resolution NVD differencing scheme for arbitrarily unstructured meshes, *Int. J. Numer. Methods Fluids* 31 (1999) 431–449.
- [36] I. Demirdžić, S. Muzaferija, Numerical method for coupled fluid flow, heat transfer and stress analysis using unstructured moving meshes with cells of arbitrary topology, *Comput. Methods Appl. Mech. Eng.* 125 (1995) 235–255.
- [37] S. Mathur, J. Murthy, A pressure-based method for unstructured meshes, *Numer. Heat Transf., Part B, Fundam.* 31 (1997) 195–215.
- [38] F. Juretić, A. Gosman, Error analysis of the finite-volume method with respect to mesh type, *Numer. Heat Transf., Part B, Fundam.* 57 (2010) 414–439.
- [39] M. Rudman, Volume-tracking methods for interfacial flows calculations, *Int. J. Numer. Methods Fluids* 24 (1997) 671–691.
- [40] O. Lebaigue, C. Duquennoy, S. Vincent, Test-case No 1: rise of a spherical cap bubble in a stagnant liquid (PN), in: H. Lemonnier, D. Jamet, O. Lebaigue (Eds.), *Validation of Advanced Computational Methods for Multiphase Flow*, Begell House Inc., 2005, p. 260.
- [41] M. Pivello, M. Villar, R. Serfaty, A. Roma, A. Silveira-Neto, A fully adaptive front tracking method for the simulation of two phase flows, *Int. J. Multiph. Flow* 58 (2014) 72–82.
- [42] D. Bhaga, M. Weber, Bubbles in viscous liquids: shapes, wakes and velocities, *J. Fluid Mech.* 105 (1981) 61–85.
- [43] J. Hua, J. Stene, P. Lin, Numerical simulation of 3D bubbles rising in viscous liquids using a front tracking method, *J. Comput. Phys.* 227 (2008) 3358–3382.
- [44] M. Jemison, E. Loch, M. Sussman, M. Shashkov, M. Arienti, M. Ohta, Y. Wang, A coupled level set-moment of fluid method for incompressible two-phase flows, *J. Sci. Comput.* 54 (2012) 454–491.
- [45] F. Denner, B. van Wachem, On the convolution of fluid properties and surface force for interface capturing methods, *Int. J. Multiph. Flow* 54 (2013) 61–64.
- [46] T.Z. Harmathy, Velocity of large drops and bubbles in media of infinite or restricted extent, *AIChE J.* 6 (1960) 281–288.
- [47] R. Clift, J. Grace, M. Weber, *Bubbles, Drops and Particles*, Academic Press, New York, 1978.
- [48] P. Lubin, S. Vincent, S. Abadie, J.-P. Caltagirone, Three-dimensional large eddy simulation of air entrainment under plunging breaking waves, *Coast. Eng.* 53 (2006) 631–655.