Journal of Computational Physics **153,** 26–50 (1999)
Article ID jcph.1999.6276, available online at http://www.idealibrary.com on IDEAL®

# A Method for Capturing Sharp Fluid Interfaces on Arbitrary Meshes

O. Ubbink* and R. I. Issa†,[1]

**FLAMENGRO: Fluid and Mechanical Engineering Group, Private Bag X337, Pretoria, 0001, South Africa; and †Mechanical Engineering Department, Imperial College of Science Technology & Medicine, Exhibition Road, London SW7 2BX, United Kingdom*
E-mail: ubbink@global.co.za and r.issa@ic.ac.uk

Received April 1, 1998; revised March 30, 1999

The paper describes a high resolution method (CICSAM) for the accurate capturing of fluid interfaces on meshes of arbitrary topology. It is based on the finite-volume technique and is fully conservative. The motion of the interface is tracked by the solution of a scalar transport equation for a phase-indicator field that is discontinuous at the interface and uniform elsewhere; no explicit interface reconstruction, which is perceived to be difficult to implement on unstructured meshes, is needed. The novelty of the method lies in the adaptive combination of high resolution discretisation schemes which ensure the preservation of the sharpness and shape of the interface while retaining boundedness of the field. The special implicit implementation presented herein makes it applicable to unstructured meshes and an extension to such grids is presented. The method is capable of handling interface rupture and coalescence. The paper outlines the methodology of CICSAM and its validation against academic test cases used to verify its accuracy. © 1999 Academic Press

*Key Words:* volume of fluid; volume fraction; free surface flow; interface tracking; interface capturing; unstructured meshes.

## 1. INTRODUCTION

The prediction of the flow of two immiscible fluids separated by a well-defined interface is of interest to several industrial applications such as two-phase flow in the chemical industry and the prediction of waves in the marine environment. An efficient numerical scheme should not only resolve the flow field but also predict the position of the interface accurately, while maintaining its sharpness.

[1] Present address: FLAMENGRO: Fluid and Mechanical Engineering Group, Private Bag X337, Pretoria, 0001, South Africa.

The available numerical methods for the prediction of two-phase flows with discrete interfaces can be classified into *surface* and *volume* methods [1, 2]. *Surface methods* mark and track the interface explicitly, either (i) with a set of marker particles or line segments or (ii) by associating the interface with a set of nodal points. Methods in (ii) are associated with moving grids and are sometimes referred to as *interface fitted* methods. Surface methods maintain sharp interfaces for which the exact position is known throughout the calculation. A disadvantage of these methods is that special treatment needs to be employed to deal with interfaces which are exposed to large deformations or stretching [3]. Extensive reviews of this type of methods can be found in [1, 2].

In *volume methods*, the different fluids are marked either by massless particles [4, 5] or by an indicator function which may in turn be a volume fraction [6, 7] or a level set [8–10]. The advantage of these methods is their ability to deal with arbitrarily shaped interfaces and to cope with large deformations as well as interface rupture and coalescence in a natural way. In the massless particle method, individual particles are tracked in a Lagrangian manner across an Eulerian mesh; such schemes are non-conservative in general. On the other hand, methods which use indicator functions solve a scalar transport equation in an Eulerian manner, thereby enabling the enforcement of conservation. The choice of indicator function can be either a scalar step function representing the volume fraction of the space occupied by one of the fluids (often referred to generically as VOF) or a smooth but arbitrary function (level set) encompassing a prespecified iso-surface which identifies the interface.

The choice of volume fraction as a phase indicator is a popular one but is prone to problems associated with the advection of a step function across a mesh that have to be overcome, namely: how to advect the interface without diffusing, dispersing, or wrinkling it. The present paper offers a scheme which goes a long way toward meeting these requirements and can also be used on unstructured meshes, without recourse to the more sophisticated, but computationally complex, surface reconstruction schemes.

Different researchers have proposed various techniques for capturing a well-defined interface with the use of volume fractions. One of the first methods is known as the SLIC method (Simple Line Interface Calculation) [6], with several extensions and improvements [11–14], and can be interpreted as relying on an explicit Lagrangian advection of reconstructed interface segments. The basic idea of these schemes is that a predefined set of rules based on volume fraction values of the neighbouring cells is used to reconstruct the fluid distribution for a cell. The local velocities move the fluid distributions in a Lagrangian way and the new volume fraction values are updated accordingly. In these methods the cell shapes, usually rectangular, are implicitly included in the reconstruction of the interface. Consequently, it is difficult to extend these methods to arbitrary complex meshes. Extensions to three-dimensional calculations pose similar difficulties. An extensive review of this type of methods can be found in [15].

An alternative approach to the problem of preserving interface resolution is to discretise the convective scalar transport equation for the volume fraction with a differencing scheme that guarantees physical (bounded) volume fraction values while preventing the smearing of the transitional area over several mesh intervals.

The widely used donor–acceptor formulation of the volume of fluid (VOF) method [7] (sometimes referred to as the original VOF) is such a scheme. However, it has been derived as a piecewise constant volume tracking method, with the appealing feature that its volume fluxes can be formulated algebraically without reconstructing the interface. It takes the interface orientation into account when calculating the amount of fluid fluxed over the face of a control volume, ensures physical volume fraction values (overall boundedness between zero and one), and keeps the transitional area over one control volume. However, it does not preserve local boundedness; i.e., a volume fraction value which initially lies between the values of its neighbours does not necessarily preserve this property when advected in the absence of shear. This feature numerically introduces new local minima and maxima in the volume-fraction field and leads to instances of non-physical deformation of the interface shape [14, 16–18].

High resolution differencing schemes such as TVD methods, Flux Corrected Transport (FCT) schemes, and techniques using the Normalised Variable diagram (NVD) [19] offer another alternative. Several attempts have been made recently to apply these schemes, but all turn out to be too diffusive [20, 21]. FCT schemes are non-diffusive in nature, but create areas of unphysical flotsam (floating wreckage) or jetsam (jettisoned goods) [16]. Furthermore, these schemes are based on one-dimensional derivations with extensions to multi-dimensional flow by operator splitting [13, 22]. This limits their implementation to structured meshes where control volume faces are aligned with the coordinate axes.

In this paper a new compressive discretisation scheme called CICSAM (Compressive Interface Capturing Scheme for Arbitrary Meshes) is developed. It makes use of the NVD concept [19] and switches between different differencing schemes to yield a bounded scalar field, but one which preserves both the smoothness of the interface and its sharp definition (over one or two computational cells). The derivation of the scheme is based on the recognition that no diffusion of the interface (whether physical or numerical) can occur; thus it is only appropriate for sharp fluid interfaces. The scheme is developed in the context of multi-dimensional applications, avoiding the need to use operator splitting. The resulting scheme turns out to be implicit by necessity and is theoretically of second-order temporal accuracy according to a Taylor series analysis of an equivalent linearised equation system.

## 2. THE GOVERNING EQUATIONS

The different fluids are modelled as a single continuum obeying the same set of governing equations, with the different fluids identified locally by a volume fraction field. The present paper deals with incompressible fluids for which the governing equations for continuity and momentum are

$$\nabla \cdot \mathbf{u} = 0 \tag{1}$$

and

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} - \mathbf{T}) = \rho \mathbf{g} + \mathbf{f}_{\dot{\sigma}}, \tag{2}$$

where $t$ is the time, $\mathbf{u}$ the velocity, $\mathbf{g}$ the gravitational acceleration, $\mathbf{T}$ the stress tensor which contains the pressure $P$, and $\mathbf{f}_\sigma$ the force due to surface tension. The local density $\rho$ and viscosity $\mu$ of the fluid are defined as

$$\rho = \alpha\rho_1 + (1 - \alpha)\rho_2 \tag{3}$$

and

$$\mu = \alpha\mu_1 + (1 - \alpha)\mu_2, \tag{4}$$

where the subscripts 1 and 2 denote the different fluids and $\alpha$ is the volume fraction defined as

$$\alpha = \begin{cases} 1 & \text{for the point } (\mathbf{x}, t) \text{ inside fluid 1} \\ 0 & \text{for the point } (\mathbf{x}, t) \text{ inside fluid 2.} \end{cases} \tag{5}$$

Finally, the conservative form of the scalar convection equation for the volume fractions,

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot \alpha \mathbf{u} = 0, \tag{6}$$

closes the set of equations.

On a mesh, the Continuum Surface Force (CSF) formulation of the surface tension force is given by [23]

$$\mathbf{f}_\sigma = -(\sigma \nabla \alpha) \nabla \cdot \left( \frac{\nabla \alpha}{|\nabla \alpha|} \right), \tag{7}$$

where $\sigma$ is the surface tension.

The set of equations (1) through (7) is solved by employing a finite-volume procedure in a time-marching fashion. The discretisation of the continuity and momentum equations (1) and (2) follows standard practices [18], using the PISO algorithm [24], and does not, therefore, warrant presentation here. However, the discretisation of Eq. (6), which governs the motion of the interface, is the key to the accurate capturing of the interface, and hence will be the focus of the remainder of the paper.

The method of solution used in the present work is a sequential one, wherein the volume fraction equation (6) is solved first, at the beginning of each time step. The new volume fraction field is then used to compute the new density and viscosity according to Eqs. (3) and (4). The momentum and continuity equations are then solved by the PISO algorithm [24], which executes a sequence of predictor–corrector steps utilising a pressure equation that is derived from a combination of Eqs. (1) and (2).

## 3. THE PHASE INDICATOR EQUATION

### 3.1. *Discretisation of the Equation*

The finite-volume discretisation of the volume fraction convection equation is based on the integral form of Eq. (6) over each control volume and the time interval $\delta t$. If P denotes the centre of the control volume (with volume $V_P$), the Crank–Nicolson discretisation, second-order accurate in time, leads to

$$\left(\alpha_P^{t+\delta t} - \alpha_P^t\right) V_P = -\sum_{f=1}^{n} \frac{1}{2} \left( (\alpha_f F_f)^t + (\alpha_f F_f)^{t+\delta t} \right) \delta t, \tag{8}$$

where $f$ is the centroid of the cell face and $F_f$ is the volumetric flux defined as

$$F_f = \mathbf{A}_f \cdot \mathbf{u}_f, \tag{9}$$

where $\mathbf{A}$ is the outward-pointing face area vector normal to the face. The summation in Eq. (8) is over all cell faces.

It should be noted that the Crank–Nicolson scheme embodied in Eq. (8) is a necessary practice (as will be shown later) if operator splitting (which is inextricably linked to regular meshes) is to be avoided and if the solution is to be free from numerical diffusion in all flow directions [18]. The scheme as defined above is more expensive in terms of computer storage because it needs both the old and the new time level's values for the volumetric flux F at the faces. However, this can be overcome, because for a small enough time step the variation of $F$ is negligible in comparison with the larger variation of $\alpha$ and therefore it is reasonable to use only the most recent value of $F$. Thus Eq. (8) reduces to

$$\alpha_P^{t+\delta t} = \alpha_P^t + \frac{\delta t}{V_P}\sum_{f=1}^{n} \alpha_f^* F_f, \tag{10}$$

where $\alpha_f^*$ is the approximation of the time-averaged volume fraction face value defined as

$$\alpha_f^* = \tfrac{1}{2}\left(\alpha_f^t + \alpha_f^{t+\delta t}\right). \tag{11}$$

For a cell-centred method, the cell centre values are used to interpolate the values on the faces in Eq. (11). It is this interpolation, which can guarantee a bounded solution while maintaining the sharpness of the interface, that will be presented next.

The interpolation of the face value is the same for all faces and it is therefore sufficient to present the derivation for a single face only. A schematic representation for a one-dimensional control volume is given in Fig. 1. The centre cell (donor cell), referred to with a subscript D, has two neighbours, known as the acceptor cell (the cell receiving fluid), referred to with a subscript A, and the upwind cell, referred to with a subscript U. Thus the flow direction must be used to determine the location of the neighbours. The face between the donor and acceptor cells, referred to with a subscript *f*, is the face under consideration.

It is reasonable to assume that the interpolated value at the face (face value) should lie between the values of the donor and acceptor cells. Linear interpolation of the face value (central differencing), is second-order accurate, but results in an unbounded solution for convection dominated problems. The use of the donor cell's value (upwind differencing), guarantees a bounded solution but is diffusive and smears the transitional area between the fluids over several cells. The use of the acceptor cell value (downwind differencing) does not preserve boundedness but maintains the resolution of the interface. The problem of interface tracking therefore boils down to the selection of a combination of differencing schemes which will preserve both the boundedness of the volume fraction distribution and the sharpness of the interface.

The original VOF [7] is a compressive scheme which has been developed for the capturing of well-defined interfaces. The name "donor–acceptor" already describes the process: the

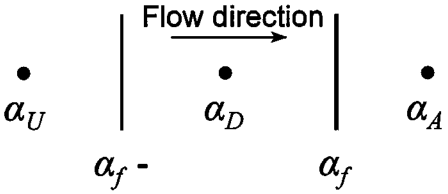


**FIG. 1.** One-dimensional control volume.

amount of volume fraction fluxed over the cell face is calculated by taking into account the volume fraction values in both the donor and the acceptor cell.

The downwind (acceptor) cell's value of a certain face is used to maintain (or steepen) the resolution of the interface. It is well known that the downwind scheme violates the boundedness criteria, unless used under strictly specified conditions. In order to maintain a bounded volume fraction field, a blending between the donor and the acceptor cell's values is used. This blended value depends on availability of the different fluids in the donor cell.

An undesirable feature of the downwind scheme is its tendency to wrinkle the interface when the flow is almost tangential to the interface [17, 18]. The original VOF [7] compensates for this by switching to upwind (donor-cell) differencing when the interface orientation is more likely to be tangential to the flow direction than normal—this switch is activated abruptly when the angle between the interface and direction of motion is 45°. An extensive study on the conditions for switching to upwind differencing was carried out [17] and revealed that the accuracy of the methodology depends to a great extent on this angle. This will be elaborated on later. Thus, a problem associated with both the original VOF [7] and SURFER [17] is that the donor–acceptor formulation used deforms the interfacial shape numerically [14, 16, 17, 18].

An analysis of the donor–acceptor formulation used in the original VOF [7] and SURFER [17] has shown that this non-physical deformation of the interface originates because the methodology used does not comply with the local boundedness criteria and because of the sudden switch between the controlled downwind differencing and upwind differencing [18]. The switch should be between two high resolution schemes which comply with the local boundedness criteria: e.g., a bounded compressive scheme when the interface orientation is more likely to be normal to the direction of motion and a more accurate interpolation scheme, such as bounded central differencing or bounded quadratic upwind interpolation, when the interface is more likely to be tangential to the direction of motion. Furthermore, it is desirable to switch more gradually between these schemes than to switch suddenly, as proposed by the original VOF and SURFER.

The high resolution schemes used in the present study and the mechanism of switching between them are described next.

### 3.2. *Normalised Variable Diagram*

The normalised variable [19] forms the basis on which the high resolution schemes are constructed and is defined as

$$\tilde{\alpha} = \frac{\alpha - \alpha_U}{\alpha_A - \alpha_U}. \tag{12}$$

The normalised variable can be used to give expressions for $\tilde{\alpha}_D$ and $\tilde{\alpha}_f$:

$$\tilde{\alpha}_D = \frac{\alpha_D - \alpha_U}{\alpha_A - \alpha_U} \tag{13}$$

$$\tilde{\alpha}_f = \frac{\alpha_f - \alpha_U}{\alpha_A - \alpha_U}. \tag{14}$$

Reference [25] presents a convection boundedness criterion (CBC) for one-dimensional implicit flow calculations. The CBC uses the normalised variable and stipulates bounds on

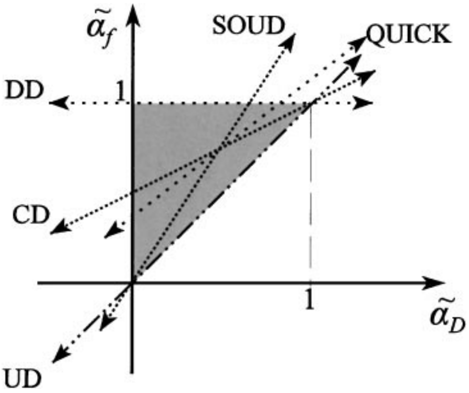


**FIG. 2.** Normalised Variable Diagram (NVD) with the convection boundedness criteria (CBC). UD, Upwind Differencing; CD, Central Differencing; DD, Downwind Differencing; SOUD, Second-Order Upwind Differencing; QUICK, Quadratic Upwind Interpolation.

$\tilde{\alpha}_f$ for which an implicit differencing scheme in 1D always preserves the local boundedness criteria:

$$\begin{aligned} &\tilde{\alpha}_f = \tilde{\alpha}_D && \text{for } \tilde{\alpha}_D < 0 \text{ or } \tilde{\alpha}_D > 1 \\ &\tilde{\alpha}_D \le \tilde{\alpha}_f \le 1 && \text{for } 0 \le \tilde{\alpha}_D \le 1. \end{aligned} \tag{15}$$

Figure 2 shows the Normalised Variable Diagram (NVD), which plots the normalised face value as a function of the normalised donor-cell value. The downwind, upwind, second-order upwind, and central differencing schemes are also represented in the diagram. The CBC, given by Eq. (15), defines the shaded area together with the line representing the upwind differencing scheme. Reference [19] showed that the various differencing schemes and the CBC can easily be constructed for one-dimensional explicit flow calculations. This is done by introducing a linear weighting based on the Courant number $c$,

$$\tilde{\alpha}_f = (1 - c)\tilde{\alpha}_f^* + c\tilde{\alpha}_D, \tag{16}$$

where $\tilde{\alpha}_f^*$ is the normalised face value for the implicit implementation. With this linearisation $\tilde{\alpha}_f \rightarrow \tilde{\alpha}_f^*$ if $c \rightarrow 0$ and $\tilde{\alpha}_f \rightarrow \tilde{\alpha}_D$ if $c \rightarrow 1$; thus a point to point transfer of the upwind nodal value occurs if $c_f = 1$. The CBC for explicit flow calculations reduces to

$$\begin{aligned} &\tilde{\alpha}_f = \tilde{\alpha}_D && \text{for } \tilde{\alpha}_D < 0 \text{ or } \tilde{\alpha}_D > 1 \\ &\tilde{\alpha}_D \le \tilde{\alpha}_f \le \min\left\{1, \tfrac{\tilde{\alpha}_D}{c}\right\} && \text{for } 0 \le \tilde{\alpha}_D \le 1. \end{aligned} \tag{17}$$

Figure 3 shows the CBC for the explicit implementation with an arbitrary Courant number $c$. In multi-dimensional flow the worst-case conditions are applied to the CBC by defining the Courant number c to be the Courant number of the donor cell, defined as

$$c = \sum_{\text{faces}} \left| c_f^{\text{out}} \right|, \tag{18}$$

where $c_f^{\text{out}}$ is the Courant number for each outflow face of the donor cell [18, 26].

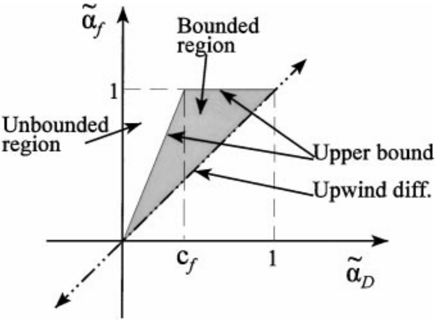


**FIG. 3.** The convection boundedness criteria for explicit flow calculations.

A differencing scheme which follows the upper bound of the CBC for explicit flow calculations is shown to be very compressive because it turns every finite gradient in a scalar field into a step profile [19]. Named as HYPER-C in [19], it was dismissed because of its unsuitability for advecting continuous profiles. However, it is precisely this characteristic that is needed for the present purpose (at least when the interface is more likely to be normal to the direction of motion).

The CBC defined above, together with the knowledge gained from the original VOF [7] and SURFER [17] on taking into account the interface orientation to the direction of motion, will be used in the next section for the construction of CICSAM, the proposed compressive differencing scheme.

### 3.3. *The Basis of the CICSAM Differencing Scheme*

In the previous section it was mentioned that the upper bound of the CBC, defined as

$$\tilde{\alpha}_{f_{\text{CBC}}} = \begin{cases} \min\left\{\frac{\tilde{\alpha}_D}{c}, 1.0\right\} & \text{when } 0 \le \tilde{\alpha}_D \le 1 \\ \tilde{\alpha}_D & \text{when } \tilde{\alpha}_D < 0, \tilde{\alpha}_D > 1, \end{cases} \tag{19}$$

is the most compressive differencing scheme which complies with the local boundedness criteria and would therefore be the most suitable scheme for the advection of a step profile, at least in one dimension (i.e., when the step is normal to the flow direction).

Unfortunately, on its own a scheme defined by Eq. (19) which is derived from one-dimensional considerations is not suitable for direct application to multi-dimensions, because it tends to wrinkle the interface [7, 17]; numerical results in [18] demonstrate this behaviour. This is because downwinding tends to compress any gradient into a step profile, even if the orientation of the interface is almost tangential to the flow direction.

This problem is addressed in several works, all of which switch from the controlled downwind formulation to upwind differencing [6, 17, 27] under certain conditions. The most elementary procedure is to switch to upwinding when $|\alpha_D - \alpha_A| < k_s$, where $k_s$ is a small prescribed constant [27].

The original VOF [7] determines the slope of the interface and switches to upwind differencing if the smallest angle between the interface and the face of the control volume is more than 45°. An extensive study on the conditions for switching to upwind differencing was carried out in [17], which presents numerical results for a square volume fraction profile convected in a uniform oblique velocity field. The results for different critical angles show that the square can end up with any shape, from a badly smeared sphere to a rotated square with a tail of flotsam, or a bullet shape aligned with the flow direction.

The defects highlighted by [17] would also be encountered with the present method if one merely switched to upwind differencing at some point. This prompted a reexamination of the problem which led to the realisation that the question should not be *when* to switch, but rather *how* to switch, to upwinding. It was also realised that upwind differencing is the worst scheme to switch to because it does not actually preserve the shape of an interface which lies almost tangential to the flow direction. Thus, it is necessary to switch to another high order scheme which will preserve the interface shape better. In the present study this is done with the aid of the NVD and without much additional computational effort, because $\tilde{\alpha}_D$ is already available.

A comparison of several higher order differencing schemes for the convection of various profiles in a uniform flow field reveals that ULTIMATE-QUICKEST (UQ), the transient one-dimensional explicit bounded version of QUICK [28], performs the best [19]. It is this scheme which is chosen for CICSAM, although other high order schemes would probably be equally suitable. It should be emphasised however that on its own UQ is too diffusive to apply everywhere, hence the need to switch between schemes at the appropriate point. The mathematical formulation of UQ used in the present study is obtained by applying Eq. (16) to QUICK:

$$\tilde{\alpha}_{f_{\mathrm{UQ}}} = \begin{cases} \min\left\{\frac{8c\tilde{\alpha}_D + (1-c)(6\tilde{\alpha}_D + 3)}{8}, \tilde{\alpha}_{f_{\mathrm{CBC}}}\right\} & \text{when } 0 \le \tilde{\alpha}_D \le 1 \\ \tilde{\alpha}_D & \text{when } \tilde{\alpha}_D < 0, \tilde{\alpha}_D > 1. \end{cases} \tag{20}$$

As mentioned earlier, a key issue here is how to switch between Eq. (19) and Eq. (20). Here, a weighting factor $0 \le \gamma_f \le 1$, based on the angle between the interface and the direction of motion, is introduced for the prediction of the normalised face value of CICSAM. This weighting factor is used to switch smoothly between the upper bound of the CBC (Eq. (19)) and the less compressive differencing scheme, Eq. (20):

$$\tilde{\alpha}_f = \gamma_f \tilde{\alpha}_{f_{\mathrm{CBC}}} + (1 - \gamma_f)\tilde{\alpha}_{f_{\mathrm{UQ}}}. \tag{21}$$

The value $\gamma_f = 1$ is used when the interface orientation is normal to the direction of motion and $\gamma_f = 0$ when the interface is tangential to it. This implies that UQ operates where CBC fails to preserve the gradient in the interface and that CBC operates where UQ fails to maintain the sharpness of the interface.

The basic derivation of CICSAM is completed with the definition of the weighting factor $\gamma_f$, which is based on the cosine of the angle $\theta_f$ between $(\nabla\alpha)_D$, the vector normal to the interface, and the vector $\mathbf{d}_f$, which connects the centres of the donor and acceptor cells. Thus,

$$\theta_f = \arccos\left|\frac{(\nabla\alpha)_D \cdot \mathbf{d}_f}{|(\nabla\alpha)_D|\,|\mathbf{d}_f|}\right| \tag{22}$$

$$\gamma_f = \min\left\{k_\gamma \frac{\cos(2\theta_f) + 1}{2}, 1\right\}, \tag{23}$$

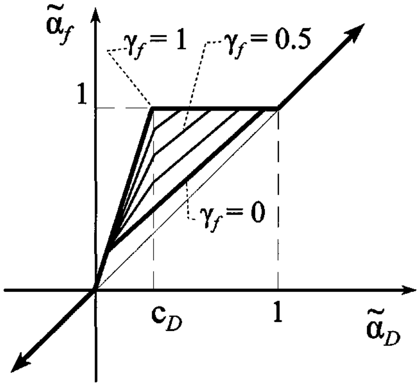


**FIG. 4.** NVD for CICSAM differencing scheme.

where $k_\gamma \geq 0$ is a constant introduced to control the dominance of the different schemes (the recommended value is $k_\gamma = 1$). The NVD of CICSAM is shown in Fig. 4.

To summarise, the normalised face value predicted with the CICSAM differencing scheme for explicit one-dimensional uniform flow is defined by Eq. (21). The actual face value can be obtained by an algebraic manipulation of Eqs. (13) and (14) to give

$$\alpha_f = (1 - \beta_f)\alpha_D + \beta_f \alpha_A, \tag{24}$$

where

$$\beta_f = \frac{\tilde{\alpha}_f - \tilde{\alpha}_D}{1 - \tilde{\alpha}_D}. \tag{25}$$

The weighting factor $\beta_f$, which implicitly contains the upwind value $\alpha_U$, carries all the information regarding the fluid distribution in the donor, acceptor, and upwind cells as well as the interface orientation relative to the direction of motion. The formulation of the face value in Eq. (24) is useful in enabling the multi-dimensional implicit implementation of CICSAM on arbitrary meshes, as shown in the next section.

It is of interest to compare CICSAM to the original VOF scheme. For each cell face CICSAM needs two explicit face values, namely the Hyper-C value and the Ultimate-Quickest value. Depending on the gradient of the volume fraction field (orientation of the interface) a blend of these two values is taken. The original VOF also needs two face values, namely downwind value bounded to the availability of the fluids in the donor cell and an upwind value. Here one of the two values is taken to be the face value, depending on the interface orientation. Both schemes need the upwind value: the original VOF uses it in the switch decision and CICSAM uses it for the third-order Ultimate-Quickest face value and to bound the first-order downwind scheme.

The difference in the bounding mechanism of the original VOF and CICSAM first-order downwind scheme can be explained with the aid of Fig. 5. Panel (a) shows an arbitrary fluid configuration in three neighbouring cells. The middle cell is the donor cell and the face under consideration is that between the donor and acceptor cells. Panel (b) shows the fluid distribution predicted with the donor–acceptor formulation of the original VOF. It shows that the level of the fluid in the upwind cell is ignored in the prediction of the

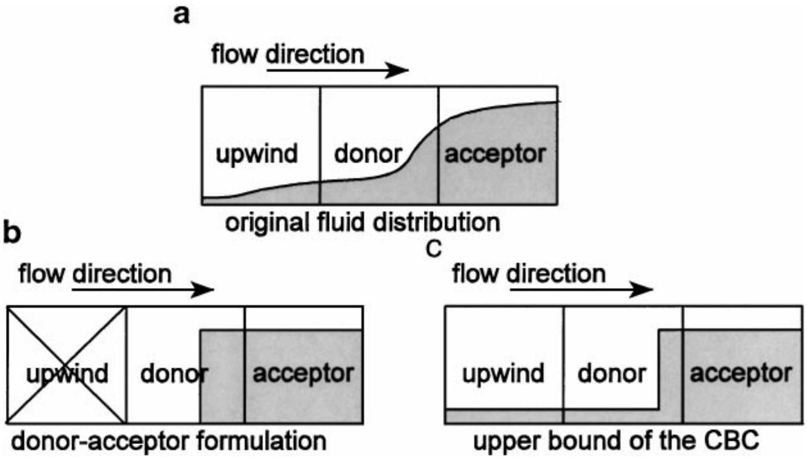


**FIG. 5.** Comparison between the donor–acceptor formulation and the upper bound of the CBC.

fluid distribution in the donor cell. This results in an incorrect steepening of the interface gradient and eventually in the non-physical deformation of the interface. Panel (c) shows the fluid distribution predicted with Hyper-C, the scheme which follows the upper bound of the CBC. The level of the fluid in the upwind cell is taken into account, producing a more realistic fluid distribution in the donor cell.

From the NVD in Fig. 4, it is clear that the formal order of accuracy is not uniform. It can vary from first order (upwind or downwind) to second order (centred) to third order (UQ). It can be argued that formal order of accuracy is not an appropriate measure of the performance of a scheme in the vicinity of a step function. Indeed in one dimension the best scheme for preserving the sharpness of the step is the first-order downwind scheme. Temporally, however, the discretisation is formally second-order accurate, as discussed in the following.

## 4. GENERALISATION OF CICSAM

### 4.1. *Extension to Multi-dimensions*

Most, if not all, bounded differential schemes are derived to preserve local boundedness in one-dimensional flow and an implementation in multi-dimensions may not necessarily maintain this characteristic. In multi-dimensional applications, an explicit split operator technique [13, 22] is usually resorted to. The basic idea of operator splitting is to apply the one-dimensional equation in separate steps for each of the coordinate directions. This limits the implementation to structured meshes in which the faces of the control volume are aligned with the coordinate axes. In the literature no reference to compressive schemes implemented in ways other than the explicit split operator technique could be found. A possible reason for this is that the donor–acceptor equation, used in earlier methods, consists of a complicated *min*/*max* principle which makes it almost impossible to separate in a linear weighting between the two cells sharing the same face.

The aim of the development of CICSAM is the ability to implement it on arbitrary meshes, and therefore it is necessary to find an alternative to the split operator technique. To this end

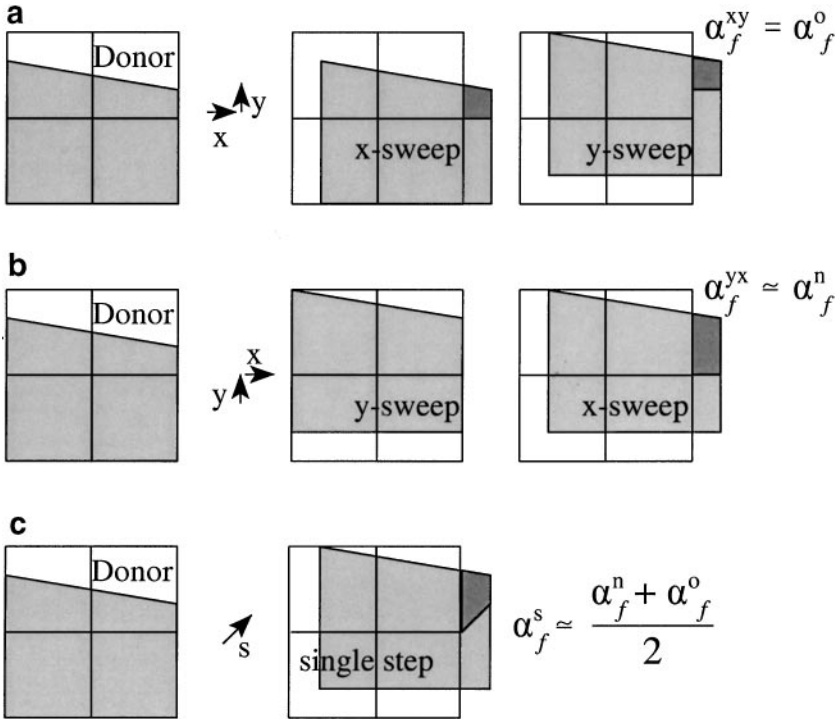

a
Donor
x
y
x-sweep
y-sweep
$\alpha_f^{xy} = \alpha^o_f$
b
Donor
x
y
y-sweep
x-sweep
$\alpha_f^{yx} \simeq \alpha^n_f$
c
Donor
s
single step
$\alpha^s_f \simeq \frac{\alpha^n_f + \alpha^o_f}{2}$

**FIG. 6.** Prediction of the face value in multiple dimensions.

an analysis of the mechanism of the split operator technique is used for the construction of the new, alternative technique to be presented herein.

As mentioned before, the operator splitting technique calculates the new volume fraction values in different sweeps, one for each coordinate direction. For symmetric differencing schemes, the new volume fraction values should be independent of the order in which the sweeps are performed. The amount of volume fraction convected over a particular cell face, however, varies with the sequence in which the sweeps are performed.

Figure 6 shows a donor cell with some of its neighbours on a two-dimensional mesh and is used to support the derivation of the new implicit implementation. The right face of the donor cell is under consideration here. The donor cell contains two fluids moving diagonally across the mesh. Figure 6a shows the original and new positions of the fluids contained in the four cells after an x-sweep followed by a y-sweep. Figure 6b shows the same process for a y-sweep followed by an x-sweep. The dark shaded area shows the amount of fluid 1 which has crossed the right cell face of the donor cell. Clearly the new fluid distributions are exactly the same as they should be, but the amount of fluid 1 convected over the cell face during this time step differs. Figure 6c shows the new fluid distribution after a single step and it is clear that the amount of fluid 1 which has crossed the right cell face is the average of the amounts shown in Figs. 6a and 6b. The explicit split operator discretisation for the xy-sweep shown in Fig. 6a is

$$\alpha^i_D = \alpha^t_D + c_{f_w}\alpha^t_{f_w} - c_{f_e}\alpha^t_{f_e} \tag{26}$$

$$\alpha^{t+\delta t}_D = \alpha^i_D + c_{f_s}\alpha^i_{f_s} - c_{f_n}\alpha^i_{f_n}, \tag{27}$$

where the subscripts *n, e, s, w* indicate the north, east, south, and west cell faces, respectively. The superscript *i* indicates an intermediate stage after a sweep in one direction has been completed. A combination of the two sweeps reduces to

$$\alpha_D^{t+\delta t} = \alpha_D^t + c_{f_w}\alpha_{f_w}^t - c_{f_e}\alpha_{f_e}^t + c_{f_s}\alpha_{f_s}^i - c_{f_n}\alpha_{f_n}^i. \tag{28}$$

A similar discretisation applies for the yx-sweep shown in Fig. 6b

$$\alpha_D^{t+\delta t} = \alpha_D^t + c_{f_w}\alpha_{f_w}^i - c_{f_e}\alpha_{f_e}^i + c_{f_s}\alpha_{f_s}^t - c_{f_n}\alpha_{f_n}^t. \tag{29}$$

In order to remove the dependence on the sequence in which the sweeps are performed, the average of the xy-sweep and yx-sweep (Eqs. (28) and (29)) is taken and this results in

$$\begin{aligned}\alpha_D^{t+\delta t} = \alpha_D^t &+ \tfrac{1}{2}c_{f_w}\left(\alpha_{f_w}^t + \alpha_{f_w}^i\right) - \tfrac{1}{2}c_{f_e}\left(\alpha_{f_e}^t + \alpha_{f_e}^i\right)\\ &+ \tfrac{1}{2}c_{f_s}\left(\alpha_{f_s}^t + \alpha_{f_s}^i\right) - \tfrac{1}{2}c_{f_n}\left(\alpha_{f_n}^t + \alpha_{f_n}^i\right).\end{aligned} \tag{30}$$

In order to solve Eq. (30) in a single step it is necessary to make an assumption about the intermediate face value of the volume fraction denoted by the superscript *i*. From inspection of Figs. 6b and 6c it is evident that the choice

$$\alpha_f^i = \alpha_f^{t+\delta t} \tag{31}$$

is a good approximation, and this is supported by a truncation error analysis which shows that Eq. (30) becomes second-order accurate in time (equivalent to using the Crank–Nicolson scheme). Indeed, numerical experimentation reveals that this scheme must be employed (as opposed to the Explicit or Euler implicit schemes) if deformation of the interface is to be avoided [18].

With this definition of the face values another problem arises. The face values are calculated according to Eq. (24) but $\beta_f$, the weighting factors for the new time, are not available. Earlier it was mentioned that the weighting factors represent the slope and orientation of the interface. If the time step is small enough, the interface slope and orientation will vary little and it is therefore reasonable to assume that the old weighting factors can be used. Thus, the time-averaged face value is defined as

$$\alpha_f^* = (1-\beta_f)\frac{\alpha_D^t + \alpha_D^{t+\delta t}}{2} + \beta_f\frac{\alpha_A^t + \alpha_A^{t+\delta t}}{2}. \tag{32}$$

However, use of Eq. (32) does not guarantee that overall boundedness is always satisfied. Occasionally there may be volume fraction values slightly less than zero or slightly greater than unity, especially when the mesh consists of very badly shaped cells. A predictor–corrector solution procedure which removes all the non-physical values is needed and this is presented next. Thus, the value of $\alpha$ in Eq. (32) should be regarded as an initial guess which may have to be updated to result in a fully bounded field.

### 4.2. *Predictor–Corrector Procedure*

As mentioned, compressive differencing schemes are derived to operate at the upper limit of the boundedness criteria and non-physical volume fraction values occur from time to time. Reference [7] overcomes these occurrences by resetting the non-physical values back to zero or one, claiming the accumulated changes of the fluid volumes are small enough. However, it was found in the present study that the resetting of the non-physical volume fraction values introduces a conservation error which affects the momentum equations and manifests itself in a spurious velocity field.

It should be emphasised that the occurrences of non-physical values are rare. However, they do appear from time to time and therefore need to be corrected. The procedure for achieving this is to solve the $\alpha$ equation in two steps: a predictor and a corrector sequence. The latter need only be carried out when non-physical volume fraction values occur.

The predictor step consists of the calculation of the CICSAM weighting factor $\beta_f$ and the new volume fractions. If any of the new volume fraction values have non-physical values, then $\beta_f^*$ is set equal to $\beta_f$ and the correction procedure described below is effected.

For an explicit split operator calculation the face values of the donor cells with unbounded values could be adjusted in correspondence with the degree of unboundedness in the donor cell. For the implicit scheme, however, the face values cannot be corrected directly, because they are implicitly defined by the weighting factors and the new volume fraction values. Thus it is the weighting factors which need to be corrected instead.

Furthermore, with an explicit split operator calculation the non-physical values remain localised. With the implicit implementation however, the localised non-physical values are convected throughout the domain and it is difficult to trace their origin. When correcting the weighting factors, it is therefore necessary to distinguish between non-physical values caused by too much downwinding and those caused by convection into a cell during the implicit calculation. The correction procedure described here automatically distinguishes between the two.

The correction procedure consists of a visit to all cell faces. If the donor cell contains a non-physical volume fraction value, a test is performed to evaluate whether too much downwinding at the face under consideration has caused the unboundedness. If so, the weighting $\beta_f^*$ is reduced with $\beta_f'$, an amount proportional to the degree of unboundedness, as described below.

First, the negative volume fraction case will be discussed. A negative volume fraction implies that more of fluid 1 has left the donor cell than is available in it. The new amount of fluid 1 to be convected over the face is determined by subtracting the unboundedness error from the original amount of fluid convected over the face, reducing to

$$\alpha_f^{**} c_f V_D = \alpha_f^* c_f V_D - \mathrm{E}^- V_D \tag{33}$$

$$\Rightarrow \alpha_f^{**} = \alpha_f^* - \frac{\mathrm{E}^-}{c_f}, \tag{34}$$

where $\alpha_f^{**}$ is the new face value and $\mathrm{E}^-$ is the magnitude of the unbounded volume fraction value, defined as

$$\mathrm{E}^- = \max\left\{-\alpha_D^{t+\delta t}, 0\right\}. \tag{35}$$

The new face value $\alpha_f^{**}$ is substituted in the face value equation (32) to obtain a new approximation for the $\beta$ weighting factor

$$\alpha_f^{**} = (1 - \beta_f^{**}) \frac{\alpha_D^t + \left(\alpha_D^{t+\delta t} + \mathrm{E}^-\right)}{2} + \beta_f^{**} \frac{\alpha_A^t + \left(\alpha_A^{t+\delta t} - \mathrm{E}^-\right)}{2}, \tag{36}$$

where

$$\beta_f^{**} = \beta_f^* - \beta'_f. \tag{37}$$

According to Eq. (36) the corrected weighting factor $\beta_f^{**}$ should always be less than or equal to the previous weighting factor. If not, the contribution of the downwind cell starts to increase and so does the degree of unboundedness. The lower limit on $\beta_f^{**}$ remains zero and this is applied to Eq. (37) to obtain bounds for $\beta'_f$:

$$0 \leq \beta'_f \leq \beta_f^*. \tag{38}$$

Equations (36) and (37) are substituted into Eq. (34) to obtain

$$(1 - (\beta_f^* - \beta'_f))\frac{\alpha_D^t + \alpha_D^{t+\delta t} + \mathrm{E}^-}{2} + (\beta_f^* - \beta'_f)\frac{\alpha_A^t + \alpha_A^{t+\delta t} - \mathrm{E}^-}{2} = \alpha_f^* - \frac{\mathrm{E}^-}{c_f} \tag{39}$$

$$\Rightarrow \beta'_f = \frac{\mathrm{E}^-(2 + c_f - 2c_f\beta_f^*)}{2c_f(\Delta\alpha^* - \mathrm{E}^-)}, \tag{40}$$

where

$$\Delta\alpha^* = \frac{\alpha_A^t + \alpha_A^{t+\delta t}}{2} - \frac{\alpha_D^t + \alpha_D^{t+\delta t}}{2}. \tag{41}$$

Equation (40) needs to be bounded with the bounds defined in Eq. (38):

$$\beta'_f = \begin{cases} \min\left\{\frac{\mathrm{E}^-(2+c_f-2c_f\beta_f^*)}{2c_f(\Delta\alpha^*-\mathrm{E}^-)}, \beta_f^*\right\} & \text{when } \Delta\alpha^* > \mathrm{E}^- \\ 0 & \text{when } \Delta\alpha^* \leq \mathrm{E}^-. \end{cases} \tag{42}$$

A similar corrector for the case in which the volume fraction value exceeds unity can be derived,

$$\beta'_f = \begin{cases} \min\left\{\frac{\mathrm{E}^+(2+c_f-2c_f\beta_f^*)}{2c_f(-\Delta\alpha^*-\mathrm{E}^+)}, \beta_f^*\right\} & \text{when } \Delta\alpha^* < -\mathrm{E}^+ \\ 0 & \text{when } \Delta\alpha^* \geq -\mathrm{E}^+, \end{cases} \tag{43}$$

where

$$\mathrm{E}^+ = \max\left\{\alpha_D^{t+\delta t} - 1, 0\right\}. \tag{44}$$

Equations (42) and (43) are now used to update the weighting factors $\beta_f^{**}$ (Eq. (37)) and these are then used to solve Eq. (8), the discretised $\alpha$ equation, for the new volume fraction distribution. If non-physical values still exist, the corrector procedure is repeated. On an orthogonal mesh the corrector sequence is seldom needed but on an arbitrary mesh with badly shaped control volumes it may be necessary to apply it more than once.

### 4.3. *Unstructured Meshes*

As noted above, the upwind value is required for the calculation of the normalised variable defined by Eq. (12). On arbitrary meshes the upwind value $\alpha_U$ is not necessarily

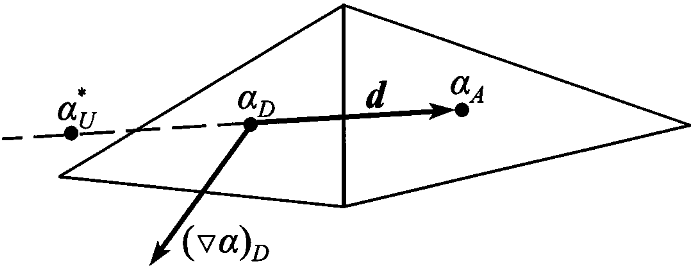


**FIG. 7.** The prediction of the upwind value for an arbitrary cell arrangement.

readily available and therefore high resolution schemes are restricted to rather simple mesh structures. Reference [29] overcomes this problem with a new definition of $\tilde{\alpha}_D$ which does not contain $\alpha_U$ directly, but the gradient $(\nabla\alpha)_D$, calculated with the aid of Gauss's theorem. However, numerical results on the convection of step profiles on arbitrary meshes have shown that this formulation does not necessarily guarantee a bounded solution.

To ensure boundedness, a new method for the calculation of $\tilde{\alpha}_D$ is presented. This method also uses the gradient over the cell, but it is used to obtain a bounded approximation $\alpha_U^*$ for the upwind value, instead. This approximated upwind value is then used in Eq. (13) for the calculation of $\tilde{\alpha}_D$.

Figure 7 gives a schematic representation of two cells with an arbitrary arrangement. The gradient $(\nabla\alpha)_D$ is a vector quantity defined over the donor cell and points in the direction in which $\alpha$ changes the most. The dot product of this gradient with the unit vector tangential to **d**,

$$(\nabla\alpha)_{D_{\mathbf{d}}} = (\nabla\alpha)_D \cdot \frac{\mathbf{d}}{|\mathbf{d}|}, \tag{45}$$

gives the gradient of $\alpha$ over the donor cell in the direction of **d**. The same gradient can also be approximated with central differencing:

$$(\nabla\alpha)_{D_{\mathbf{d}}} = \frac{\alpha_A - \alpha_U^*}{2|\mathbf{d}|}. \tag{46}$$

Equations (45) and (46) are used to obtain an expression for $\alpha_U^*$:

$$\alpha_U^* = \alpha_A - 2(\nabla\alpha)_D \cdot \mathbf{d}. \tag{47}$$

The above approximation does not guarantee a bounded $\alpha_U^*$ and therefore it is necessary to bound it with known bounds of $\alpha$. The bounds can be either the maximum and minimum values of the whole flow or local values derived from the cell's nearest neighbours:

$$\alpha_U^* = \min\{\max\{\alpha_U^*, \alpha_{\min}\}, \alpha_{\max}\}. \tag{48}$$

In the present study the physical bounds of the volume fractions (zero and one) are used for the lower and upper bounds, respectively. On a regular orthogonal grid this formulation will always return the true upwind value.

## 5. TEST CASES

The test cases in this section focus on the advection of interfaces of different shapes exposed to translation, rotation, and shear flow. Prescribed velocity fields are used and no attempt is made to couple the advection of the volume fraction field with the momentum equations. A comparison of the behaviours of different volume tracking schemes, namely SLIC, the original VOF, FCT-VOF, and Youngs-VOF, made for a number of academic cases is available [16] and those cases will be repeated here for CICSAM on both structured and unstructured meshes. For the purpose of comparison the solution errors presented by Ref. [16] are repeated here. The solution error is defined as

$$E = \frac{\sum_i^{\text{all cells}} \left\| \alpha_i^n Vol_i - \alpha_i^a Vol_i \right\|}{\sum_i^{\text{all cells}} \alpha_i^o Vol_i}, \tag{49}$$

where $\alpha^n$ is the calculated solution after $n$ time steps, $\alpha^a$ the analytical solution, and $\alpha^o$ the initial condition. This definition differs slightly from that given in [16], to make provision for variable volume sizes. However, in the case of constant cell volumes the two definitions are equivalent.

Several meshes are employed for this series of tests. For the cases presented in Sections 5.1 to 5.3 four meshes have been employed, namely:

—100 × 100 uniformly spaced cells for $x, y \in [0, \pi]$,

—200 × 200 uniformly spaced cells for $x, y \in [-2, 2]$,

—an unstructured mesh consisting of 10,040 cells shown in Fig. 8 (to correlate with the uniformly spaced mesh of 100 × 100 cells), and

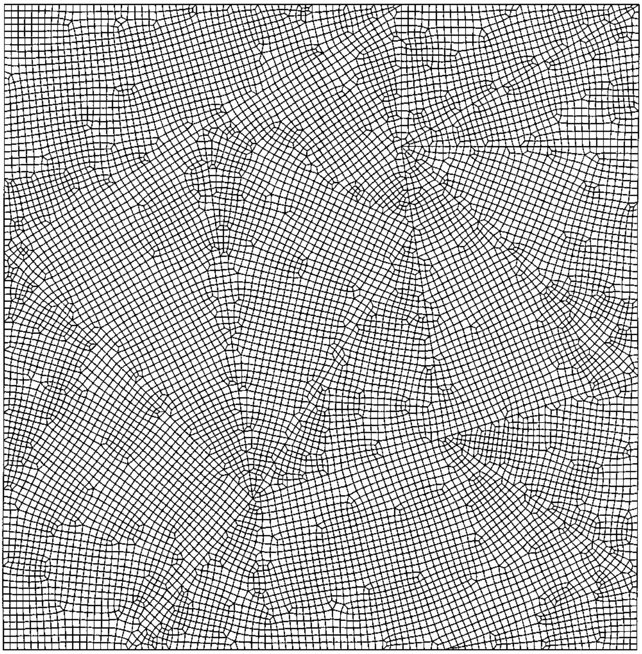


**FIG. 8.** Unstructured computational mesh (10,040 cells).

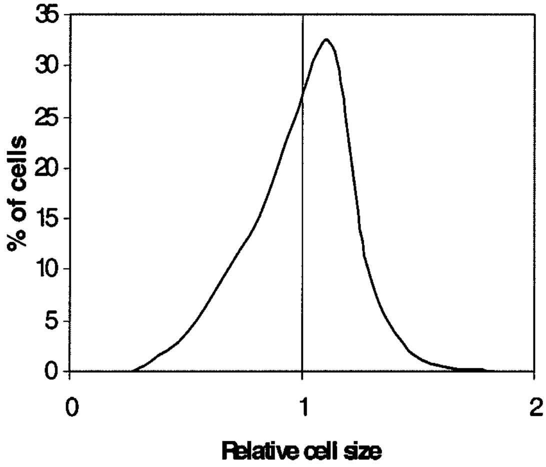


**FIG. 9.** Cell size distribution of the 10,040-cell unstructured mesh compared with that of the uniform mesh of 100 × 100 cells.

—an unstructured mesh of 40,341 cells (to correlate with the uniformly spaced mesh of 200 × 200 cells).

An analysis of the cell sizes on the unstructured meshes is given in Figs. 9 and 10. Approximately a quarter of the cells on the 10,040-cell mesh are the same size as those in the case of the uniformly spaced mesh, ±30% of the cells have a finer resolution, and ±45% of the cells have a coarser resolution. The cell sizes range between 0.5 and 1.5 times the cell sizes of the uniformly spaced mesh. Approximately 30% of the cells on the 40,341-cell mesh are the same size as those in the case of the uniformly spaced mesh, ±35% of the cells have a finer resolution, and ±35% of the cells have a coarser resolution, ranging between 0.4 and 1.7 times the cell sizes of uniformly spaced mesh. In all cases the time step is chosen to maintain a maximum mesh (material) Courant number of 0.25. These time steps are kept

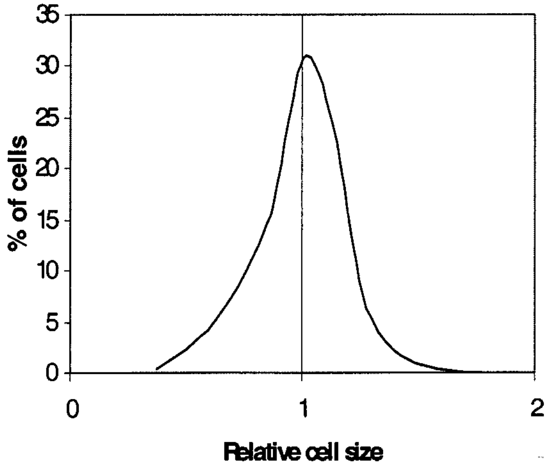


**FIG. 10.** Cell size distribution of the 40,341-cell unstructured mesh compared with that of the uniform mesh of 200 × 200 cells.

the same for the runs on the unstructured meshes, resulting in higher maximum Courant numbers on these meshes.

Results for successive mesh refinement are presented in Section 5.4 to illustrate the convergence behaviour of the scheme.

### 5.1. *Constant, Oblique Velocity Field*

The first series of tests is the advection of different scalar fields with an oblique unidirectional velocity field of (2, 1). The different volume fraction fields are:

—a hollow square aligned with the coordinate axes,
—a hollow square rotated at 26.57° to the x-coordinate axis, and
—a hollow circle.

In the case of the structured mesh, the side lengths of outer and inner interfaces of the square are 40 and 20 cells, respectively, and for the circle the outer and inner radii are 20 and 10 cells, respectively. Initially the centres of the shapes are at $(-1.2, -1.2)$ of the $200 \times 200$ mesh and after advection the theoretical new centres are at (1.3, 0.05). The same initial conditions are mapped onto the unstructured mesh by simple area-overlapping calculations.

Close-up views of the final shapes are given in Fig. 11, with the analytical solution shown in the top row, the structured mesh results in the middle row, and unstructured mesh results in the bottom row. The solution errors, together with those calculated by Ref. [16], are given in Table I. Figure 12 gives the graphical correlation of the solution errors. The results obtained with the structured mesh are denoted by CICSAM-S and those with the unstructured mesh by CICSAM-U.

Compared to the shapes presented by Ref. [16], CICSAM on both structured and unstructured meshes performs on average much better than SLIC and the original VOF, is an

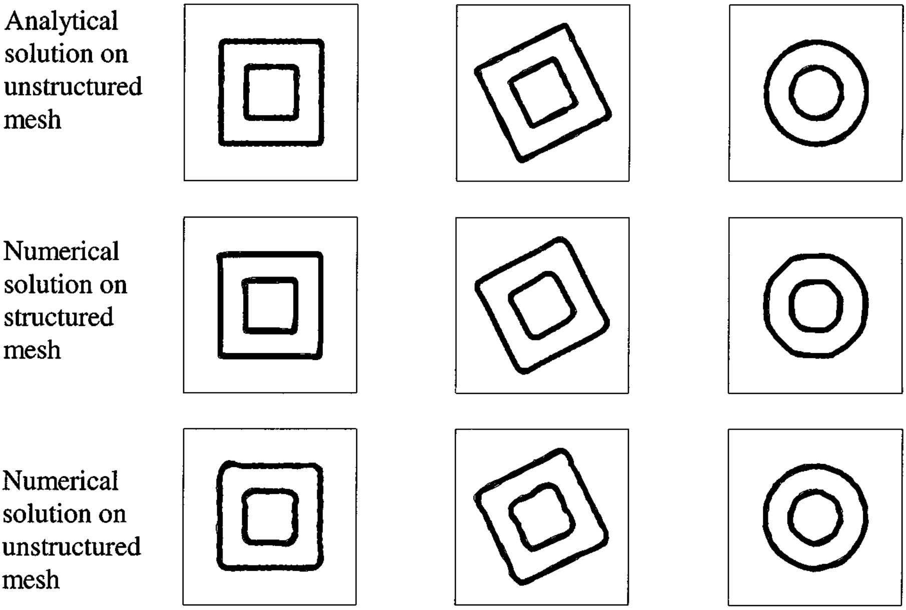

Analytical solution on unstructured mesh
Numerical solution on structured mesh
Numerical solution on unstructured mesh

**FIG. 11.** Advection with an oblique unidirectional velocity field.

**TABLE I**
**Errors for Translation**

| | SLIC[a] | Hirt-Nicols[a] | FCT-VOF[a] | Youngs-VOF[a] | CICSAM-S | CICSAM-U |
|---|---|---|---|---|---|---|
| Square | 1.32E-01 | 6.86E-03 | 1.63E-08 | 2.58E-02 | 2.50E-02 | 3.97E-02 |
| Rotated Square | 1.08E-01 | 1.60E-01 | 8.15E-02 | 3.16E-02 | 4.00E-02 | 4.00E-02 |
| Circle | 9.18E-02 | 1.90E-01 | 3.99E-02 | 2.98E-02 | 4.33E-02 | 2.84E-02 |

[a] Errors obtained from Ref. [16].

improvement on FCT-VOF, and closely follows Youngs' method. The latter is more consistent regarding the magnitude of the error for the three different shapes while the magnitudes of the errors of CICSAM and SLIC vary slightly for the different runs but not to the same degree as in the case of the original VOF and FCT-VOF. What is significant is that the results of CICSAM on an unstructured mesh are better than those of all the other methods on structured meshes with the exception of Youngs' method, which is much more complex because it is based on a reconstruction methodology.

### 5.2. *Solid Body Rotation*

In the present case the volume fraction field, in the shape of a circle with a slot as illustrated in Fig. 13, is exposed to a circular velocity field. The same computational grids of the previous section are used, with the centre of the circle at (0.0, 0.75) and the centre of rotation at (0, 0). On the structured mesh the diameter of the slotted circle is 50 cells, the width of the slot 6 cells, and the depth of the slot 30 cells. The time steps have been chosen such that one rotation corresponds to 2524 time steps (the same as that used by Ref. [16]).

Results after one revolution are shown in Fig. 13. The solution errors, including those presented by Ref. [16], are given in Table II and a graphical representation of these errors is given by Fig. 14. The relative performance of the different schemes is similar to that in the

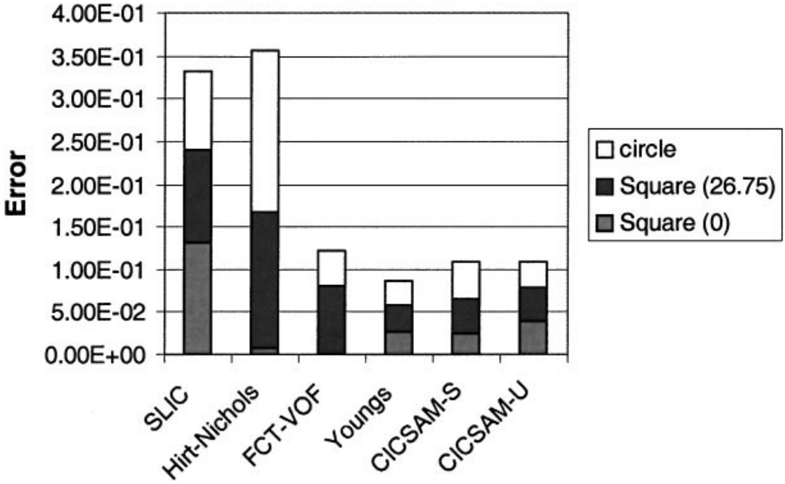
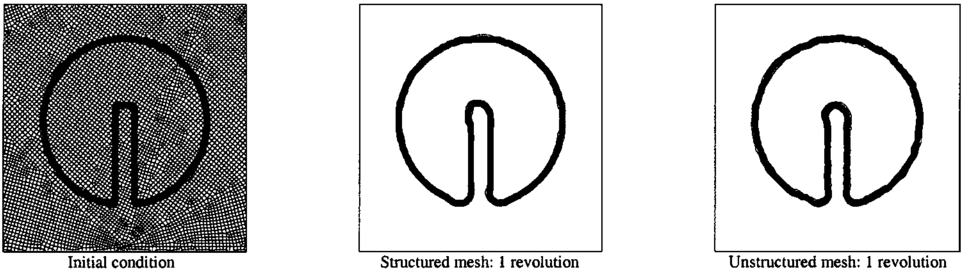


**FIG. 12.** Graphical correlation of the solution errors for the translation test.



Initial condition

Structured mesh: 1 revolution

Unstructured mesh: 1 revolution

**FIG. 13.** Solid body rotation for a split circle.

previous example. Here again the unstructured mesh results from CICSAM are superior to the results of other methods on structured meshes, apart from Youngs' method.

### 5.3. *Shear Flow*

In most real interfacial flow cases the situation is far more complicated than merely preserving the initial shape, as the interface deforms considerably. As argued by Ref. [14], it is important to evaluate the performance of volume tracking schemes in the presence of fluid shear. Such a velocity field is given by [16],

$$\mathbf{u} = (\sin(x)\cos(y), -\cos(x)\sin(y), 0)\,, \tag{50}$$

where $x, y \in [0, \pi]$. The structured mesh of $100 \times 100$ cells and its corresponding unstructured mesh are used here, with the initial volume fraction condition defined as a circle with radius $0.2\pi$ and centre $(0.5\pi, 0.2(1+\pi))$. The volume fraction field is exposed to the above velocity field for a certain time (say $N$ time steps), after which the velocities are reversed for the same length of time in order to return the volume fraction field to the initial condition. A perfect advection scheme should result in the same initial volume fraction field.

The results just before reversing the velocities and at the end of the calculations for $N = 1000$ and $N = 2000$ time steps are given in Fig. 15. Solution errors for $N = 250$, $N = 500$, $N = 1000$, and $N = 2000$ for CICSAM together with errors presented by Ref. [16] are presented in Table III. The graphical correlation of these solution errors is given in Fig. 16. The error trends are very similar to those presented in the previous two examples.

Theoretically, breakup of the tail should never occur for this case. In the case $N = 2000$ the interface is stretched to such an extent that two parts of the interface pass through a single cell (especially in the case of the unstructured mesh, which has a coarser mesh resolution at some places), a situation which is a challenge to most interface capturing techniques. Despite this limitation, CICSAM performs well, even on unstructured meshes.

**TABLE II**
**Errors for Rotation**

| | SLIC[a] | Hirt-Nicols[a] | FCT-VOF[a] | Youngs-VOF[a] | CICSAM-S | CICSAM-U |
|---|---|---|---|---|---|---|
| Split circle | 8.38E-02 | 9.62E-02 | 3.29E-02 | 1.09E-02 | 1.62E-02 | 2.02E-02 |

[a] Errors obtained from Ref. [16].

**TABLE III**
**Errors for Shear Flow**

| *N* | SLIC[a] | Hirt-Nicols[a] | FCT-VOF[a] | Youngs-VOF[a] | CICSAM-S | CICSAM-U |
|---|---|---|---|---|---|---|
| 250 | 2.72E-02 | 3.24E-02 | 1.94E-02 | 2.61E-03 | 1.63E-02 | 9.39E-03 |
| 500 | 3.30E-02 | 4.00E-02 | 2.35E-02 | 5.12E-03 | 2.09E-02 | 1.10E-02 |
| 1000 | 4.59E-02 | 6.60E-02 | 3.14E-02 | 8.60E-03 | 2.90E-02 | 1.82E-02 |
| 2000 | 9.02E-02 | 1.09E-01 | 1.44E-01 | 3.85E-02 | 5.67E-02 | 4.17E-02 |

[a] Errors obtained from Ref. [16].

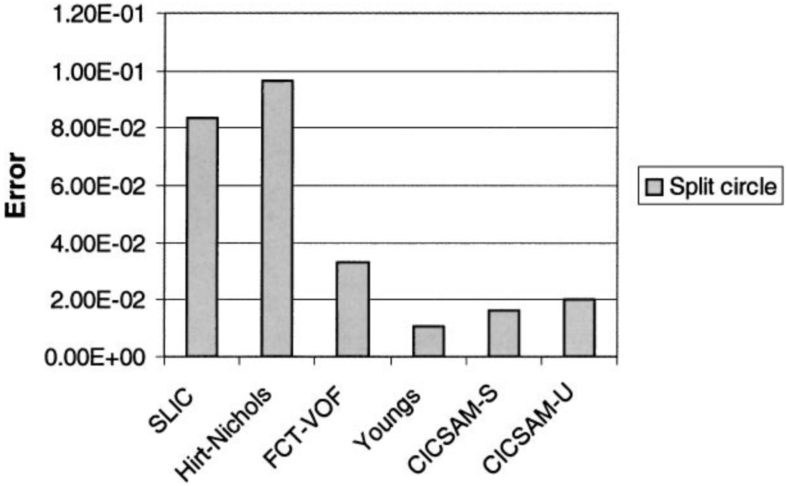


**FIG. 14.** Graphical correlation of the solution errors for the rotation test.

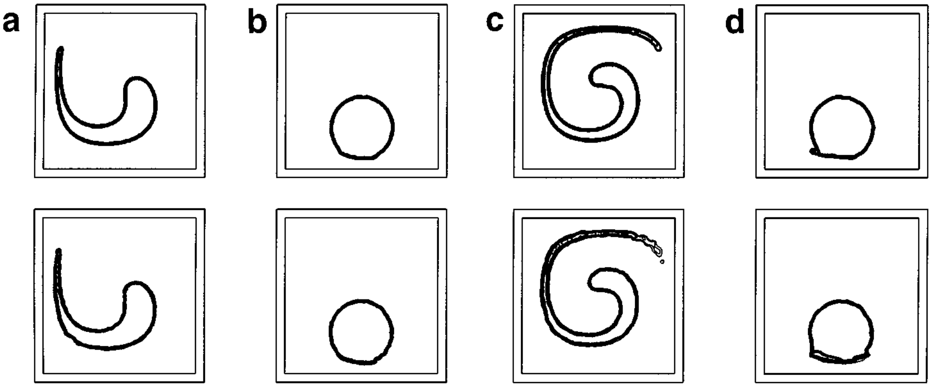


**FIG. 15.** Results for the shear flow case. Top row, on the structured mesh; bottom row, on the unstructured mesh: (a) after 1000 steps forward, (b) after 1000 steps forward followed by 1000 steps backward, (c) 2000 steps forward, (d) 2000 steps forward followed by 2000 steps backward.

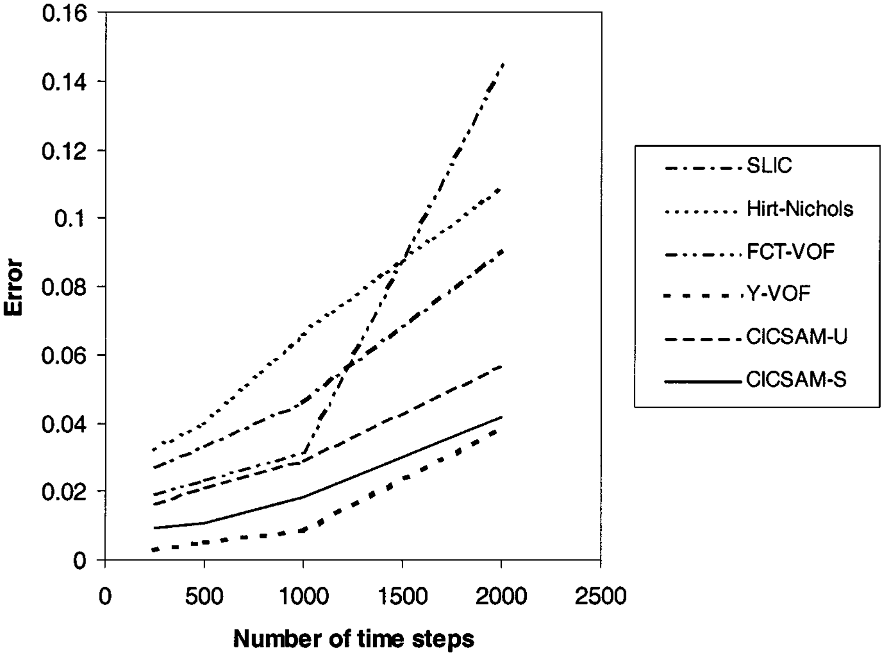


**FIG. 16.** Graphical correlation of the solution errors for the shear flow test.

## 5.4. *Convergence*

The case studies are completed with a convergence test for the rotation of a circular volume fraction field on a regular mesh. A circle with radius 1.6 and centre [0, 0] is placed in a computational domain of $x, y \in [-2, 2]$ and is exposed to a rotational velocity field. Solution errors as defined by Eq. (49) after one revolution for different meshes (10 × 10, 20 × 20, 40 × 40, 80 × 80, 160 × 160) are given in the middle column of Table IV. The order of convergence for two consecutive runs is given in the last column, indicating a first-order convergence rate as the mesh becomes fine. This is not surprising since the interface is captured over two to three cells at all times irrespective of the mesh density. Thus the error in capturing the interface can at best diminish proportionally to the reduction in the cell width, i.e., a linear convergence rate with mesh refinement. This indeed supports the argument made earlier that the formal order of accuracy can be misleading in the context of step functions.

**TABLE IV**
**Errors and Order of Convergence for Rotation of a Circle**

| Mesh | Error | Order |
|---|---|---|
| 10 × 10 | 1.94E-01 | |
| | | 2.90E+00 |
| 20 × 20 | 2.60E-02 | |
| | | 1.14E+00 |
| 40 × 40 | 1.18E-02 | |
| | | 1.10E+00 |
| 80 × 80 | 5.48E-03 | |
| | | 1.04E+00 |
| 160 × 160 | 2.66E-03 | |

## 6. CONCLUSIONS

A method for capturing sharp fluid interfaces on arbitrary meshes has been presented. It is based on the solution of a transport equation for a fluid-indicator function which is chosen to be the volume fraction. To circumvent the problem associated with the advection of a step profile across Eulerian meshes, a new discretisation scheme CICSAM, which combines both sharp resolution of the interfaces and boundedness of the volume fraction field, has been developed. The scheme relies on the Normalised Variable Diagram and switches smoothly between two high resolution schemes depending on the orientation of the interface to the flow direction, a procedure which ensures optimum preservation of the integrity of the interface. The method is developed in multi-dimensions to avoid the need for operator splitting, which would limit usefulness to regular meshes. The resulting scheme is implicit, is theoretically of second-order accuracy in time, and is applicable to unstructured meshes. It is demonstrated that this scheme can outperform, even on unstructured meshes, most other interface capturing schemes with the exception of the more complex interface reconstruction methods.

## ACKNOWLEDGMENTS

The authors thank Mr H. Weller for the use of his FOAM code, Mr. LA le Grange for the use of the Flo$^{++}$ code, and Dr. H. Jasak for helpful discussions.

## REFERENCES

1. J. M. Hyman, *Physica D* **12**, 396 (1984).
2. J. H. Ferziger and M. Perić, *Computational Methods for Fluid Dynamics* (Springer-Verlag, Heidelberg, 1996).
3. S. O. Unverdi and G. Tryggvason, *J. Comput. Phys.* **100**, 25 (1992).
4. F. H. Harlow and J. E. Welch, *Phys. Fluids* **8**, No. 12, 2182 (1965).
5. B. J. Daly, *Phys. Fluids* **10**, No. 2, 297 (1967).
6. W. F. Noh and P. Woodward, *Lect. Notes in Phys.* **59**, 330 (1976).
7. C. W. Hirt and B. D. Nichols, *J. Comput. Phys.* **39**, 201 (1981).
8. S. Osher, *J. Comput. Phys.* **79**, 12 (1988).
9. M. Sussman, P. Smereka, and S. Osher, *J. Comput. Phys.* **114**, 146 (1994).
10. J. A. Sethian, *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Material Sciences* (Cambridge Univ. Press, Cambridge, UK, 1996).
11. A. J. Chorin, *J. Comput. Phys.* **35**, 1 (1980).
12. P. Lötstedt, *J. Comput. Phys.* **47**, 211 (1982).
13. D. L. Youngs, Time-dependent multi-material flow with large fluid distortion, in *Numerical Methods for Fluid Dynamics*, edited by K. W. Morton and M. J. Baines (Academic Press, London, 1982), p. 273.
14. N. Ashgriz and J. Y. Poo, *J. Comput. Phys.* **93**, 449 (1991).
15. W. J. Rider and D. B. Kothe, *J. Comput. Phys*. **141**, 112 (1998).
16. M. Rudman, *Int. J. Numer. Methods Fluids*. **24**, 671 (1997).
17. B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, and G. Zanetti, *J. Comput. Phys*. **113**, 134 (1994).
18. O. Ubbink, *Numerical Prediction of Two Fluid Systems with Sharp Interfaces*, Ph.D. thesis, University of London (1997).
19. B. P. Leonard, *Comput. Methods Appl. Mech. Engrg.* **88**, 17 (1991).
20. M. S. Darwish, *Numer. Heat Transfer Part B* **24**, 353 (1993).

21. K. A. Pericleous and K. S. Chan, The sea method for free surface problems with heat transfer and change of phase, in *Proc. Numerical Methods in Multiphase Flows, 1994 ASME Fluids Engineering Division Summer Meeting, Lake Tahoe, Nevada, June 19–23,* edited by C. T. Crowe *et al*. (1994).

22. B. P. Leonard, A. P. Lock, and M. K. MacVean, *Monthly Weather Rev*. **124**, 2588 (1996).

23. J. U. Brackbill, D. B. Kothe, and C. Zemach, *J. Comput. Phys*. **100**, 335 (1992).

24. R. I. Issa, *J. Comput. Phys*. **62**, 40 (1986).

25. H. Gaskell and A. K. C. Lau, *Int. J. Numer. Methods Fluids* **8**, 617 (1988).

26. B. P. Leonard, Bounded higher-order upwind multidimensional finite-volume convection–diffusion algorithms, in *Advances in Numerical Heat Transfer*, edited by W. J. Minkowycz and E. M. Sparrow (Taylor & Francis, London, 1997), p. 432.

27. J. D. Ramshaw and J. A. Trapp, *J. Comput. Phys*. **21**, 438 (1976).

28. B. P. Leonard, *Comput. Methods Appl. Mech. Engrg*. **19**, 59 (1979).

29. H. Jasak, *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows*, Ph.D. thesis, University of London (1996).