Journal of Computational Physics 228 (2009) 9107–9130

Contents lists available at ScienceDirect

Journal of Computational Physics

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

A Quadratic Spline based Interface (QUASI) reconstruction algorithm for accurate tracking of two-phase flows

S.V. Diwakar, Sarit K. Das, T. Sundararajan *

Department of Mechanical Engineering, IIT Madras, Chennai 600036, India

ARTICLE INFO

Article history:
Received 13 October 2008
Received in revised form 10 September 2009
Accepted 11 September 2009
Available online 17 September 2009

Keywords:
Multiphase
Interface reconstruction
Volume of fluid
Quadratic spline
Curvature effects

ABSTRACT

A new Quadratic Spline based Interface (QUASI) reconstruction algorithm is presented which provides an accurate and continuous representation of the interface in a multiphase domain and facilitates the direct estimation of local interfacial curvature. The fluid interface in each of the mixed cells is represented by piecewise parabolic curves and an initial discontinuous PLIC approximation of the interface is progressively converted into a smooth quadratic spline made of these parabolic curves. The conversion is achieved by a sequence of predictor–corrector operations enforcing function (C^0) and derivative (C^1) continuity at the cell boundaries using simple analytical expressions for the continuity requirements. The efficacy and accuracy of the current algorithm has been demonstrated using standard test cases involving reconstruction of known static interface shapes and dynamically evolving interfaces in prescribed flow situations. These benchmark studies illustrate that the present algorithm performs excellently as compared to the other interface reconstruction methods available in literature. Quadratic rate of error reduction with respect to grid size has been observed in all the cases with curved interface shapes; only in situations where the interface geometry is primarily flat, the rate of convergence becomes linear with the mesh size. The flow algorithm implemented in the current work is designed to accurately balance the pressure gradients with the surface tension force at any location. As a consequence, it is able to minimize spurious flow currents arising from imperfect normal stress balance at the interface. This has been demonstrated through the standard test problem of an inviscid droplet placed in a quiescent medium. Finally, the direct curvature estimation ability of the current algorithm is illustrated through the coupled multiphase flow problem of a deformable air bubble rising through a column of water.

1. Introduction

Flows with moving interfaces and the associated heat and mass transfer form an integral part of many natural and industrial processes. Most multiphase flows are characterized by the presence of fluid structures over a variety of scales. A common example of this is seen in the disintegration of a liquid jet in air where the characteristic dimension varies from the jet diameter to the size of fine droplets. These diverse flow behaviors are the consequence of interfacial phenomena and related processes, whose understanding is of paramount importance in the design of devices such as injectors. Although numerical modeling offers promising tools in this regard, the added non-linearity of interfacial phenomena makes the modeling of these flows much more challenging.

* Corresponding author. Tel.: +91 44 22574683; fax: +91 44 22574652.
E-mail address: tsundar@iitm.ac.in (T. Sundararajan).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.09.014

Broadly, the available multiphase modeling techniques can be classified as moving-grid and fixed-grid methods based on the relative deformation of the grid with respect to interface dynamics. In a moving-grid method, the computational grid for each fluid sub-domain can be distinctly identified and the grid points located on the interface move along with it to accurately predict the temporal evolution of the interface. However, at large interfacial deformations, the method requires continuous remeshing effort to account for the changes in topology. On the other hand, fixed-grid methods effectively predict the dynamics of the interface using special schemes on a non-deformable grid with less computational effort. The two important classifications of fixed-grid methods are the front-tracking and the front-capturing techniques, which involve explicit and implicit treatment of the interface respectively. The application of a front-tracking technique to represent the interface explicitly by a distribution of marker points connected by spline interpolation was demonstrated by Popinet and Zaleski [1] where the interface dynamics was effectively tracked by Lagrangian advection of these marker points in the flow field. The marker points easily resolve the sub-grid level fluid structures and the use of spline curves helps in the direct estimation of interface attributes like curvature which are required for the evaluation of surface tension forces. In spite of these advantages, difficulties arise with regard to accurate mass conservation and in simulating flows with interface merger and breakup. The front-capturing techniques like volume of fluid (VOF) and level set (LS), on the other hand, effectively resolve macro-fluid structures larger than the grid spacing [2,3]. They use characteristic phase functions for an implicit description of the interface, which automatically handles interface merger and breakup.

The level set method uses the signed distance (ϕ) of grid points from the interface for an implicit representation. The advection of the distance function 'ϕ' and its subsequent re-initialization leads to the transient evolution of the interface. Even though the level set method is attractive due to the differentiable properties of distance function, a complete conservation of mass in the domain using level set is still a challenging issue. On the contrary, the VOF method enforces strict volume conservation of various constituent phases and uses a discrete volume fraction distribution (F) to represent the interface. The cells are assigned with F value between 0 and 1, based on the volume fraction of the primary fluid (in the case of a binary system) in the cell. The discontinuous nature of the volume fraction 'F' necessitates two geometric procedures, namely: interface reconstruction and fluid advection, to track the fluid volumes. The interface reconstruction procedure involves estimation of the approximate orientation and position of the interface within the cell and the advection process geometrically evaluates the material volume flux through the cell faces based on the reconstructed interface.

Issues in VOF method primarily arise due to the non-uniqueness of interface reconstruction procedure. Earlier reconstruction methods like DeBar's piecewise linear approximation [4], the stair-step approximation of Hirt and Nichols [2] and SLIC (Simple Line Interface Calculation) reconstruction of Noh and Woodward [5] resulted in unphysical 'flotsam' and 'jetsam' even for simple flow configurations. Notable improvement in the reconstruction procedure was brought in by the Youngs' PLIC [6] algorithm (Piecewise Linear Interface Calculation) which allowed the interfaces to be represented by straight line segments normal to the gradient of volume fraction (F). The precise location of these line segments within each cell is then obtained based on the volume of primary fluid in the cell. In spite of its simplicity and robustness, the sensitivity of the PLIC reconstruction on the algorithm used to calculate the volume fraction gradients, renders it non-unique. Also, the PLIC representation suffers from discontinuity of interface shape across the cell boundaries and the linear approximation of the interface is insufficient to allow for the direct estimation of attributes like the local interfacial curvature. These factors have given rise to the widespread use of special curvature calculation methods based on the volume fraction distribution. The convolution (CV) technique of Williams et al. [7] uses a mollified volume fraction distribution obtained from the application of various smoothing kernels. Other accurate methods include Height Function (HF) technique of Cummins et al. [8] where the discrete sums of volume fractions are used to obtain the curvature information.

Alternatively, direct estimation of curvature using higher order approximations of the interface within mixed cells have been considered by a few researchers. The Piecewise Parabolic Interface (PPIC) reconstruction was first presented by Price et al. [9], where a second order equation was used to represent the interface. The method involves an iterative non-linear optimization scheme to obtain the coefficients of quadratic expression. A similar generalized three-dimensional approach was presented by Renardy and Renardy [10] and a direct estimate of curvature was used to calculate the surface tension force in the interfacial cell. Although these methods possess higher interface reconstruction accuracy and lower magnitudes of spurious currents (which are generally caused by error in the curvature estimation and the imbalance between pressure gradients and surface tension force), the iterative error minimization procedure employed in these methods renders them computationally intensive. Moreover, the parabolic interfaces obtained from these methods still suffer from discontinuities at the cell boundaries. The other class of higher order methods proposed by Ginzburg and Wittum [11] and Lo'pez et al. [12] involve cubic spline based interface reconstruction. In [11], an initial spline passing through the midpoints of the PLIC interface is evolved into a cubic spline which conserves mass exactly in each of the mixed cells. The computations involve the simultaneous solution of '3n' equations (n – number of mixed cells forming the spline) subjected to volume constraint in each cell. The cubic spline aids in the direct estimation of curvature using a continuous form of interface representation in the whole domain; however, it tends to produce wavy interfaces in some cases due to the non-locality of errors.

In the current work, a new two-dimensional Quadratic Spline based Interface (QUASI) reconstruction algorithm is presented, where piecewise quadratic curves are used to represent the interface within each cell. A continuous quadratic spline across the mixed cells ($0 < F < 1$) is then evolved from a sequence of predictor–corrector operations, where the end points of an initial PLIC interface approximation are corrected to enforce function and derivative (C^0 and C^1) continuities at the cell boundaries. Simple analytical expressions have been derived for all the steps involved which effectively minimize the computational effort required for the current algorithm. The use of higher order curves with constraints results in unique and

accurate representation of the interface and aids in the direct estimation of the local interface curvature. Here, the QUASI reconstruction procedure has been used in conjunction with the EMFPA fluid advection algorithm of Lo'pez et al. [12]. The accuracy of the QUASI reconstruction over the other methods has been clearly demonstrated using standard reconstruction and advection test cases.

2. The QUASI reconstruction

A general quadratic spline $Q_i(X)$ using second order polynomial interpolation and satisfying C^0 and C^1 continuities at all points (Fig. 1) can be expressed as

$$Q_i(X) = a_i X^2 + b_i X + c_i, \quad X \in [X_i, X_{i+1}], \quad i = 1, 2, \ldots, n. \tag{1}$$

Such a spline (Eq. (1)) passing through 'n + 1' nodes would require simultaneous solution of '$3n$' equations to obtain the coefficients a_i, b_i and c_i $(i = 1, 2, \ldots, n)$. The conditions of function continuity $(Q_i(X_P) = Y_p; Q_{i+1}(X_P) = Y_p;\ i = 1, 2, \ldots, n)$ and slope continuity $(dQ_i(X_P)/dX = dQ_{i+1}(X_P)/dX;\ i = 2, \ldots, n \text{ or } n+1)$ at all the interior points, provide the required '$3n$' equations for a self-connected spline and '$3n - 1$' equations for a non-self-connected spline. Closure in the latter case can be achieved by setting $a_i = 0$ (straight line approximation) for either the first or the last spline function.

The use of spline function to represent the interface in a VOF problem is very challenging, as the data points which define the spline function are not known a priori. But, with the knowledge of volume fraction data for each cell, it is possible to convert this tedious problem into a systematic predictor–corrector procedure, which forms the essence of the current paper. The overall procedure of the current QUASI reconstruction algorithm is briefly summarized below.

1. Initially, an approximate interface is reconstructed in all the mixed cells using the PLIC algorithm. In each mixed cell, a straight line perpendicular to the gradient of phase function F is generated and it is located within the cell such that the volume fraction constraint is satisfied. The end points of the interfaces formed in this manner are generally discontinuous across the mixed cells, (Fig. 2(b)).
2. The next step involves the achievement of C^0 continuity from the initial PLIC representation, where the end points from the consecutive mixed cells are forced to coincide on the common edge of the cells.
3. A piecewise quadratic curve passing through these C^0 continuous end points is then obtained in each cell (Fig. 2(c)) subjected to the volume constraint in the corresponding cell. Analytical expressions relating the end points, volume fraction and the coefficients of quadratic curve have been derived in Section 2.3.
4. For each pair of consecutive mixed cells, C^1 continuous interfaces are obtained by moving the end points along the common edges of the cells such that the slopes of the parabolas on either side of a common edge are equal. The direct estimation of the C^1 continuous point for the given pair of cells is analytically possible using the relations derived in Section 2.4.
5. In the cases where it is not possible to obtain an explicit C^0 continuous interface (ex: domain boundary), the above C^1 correction procedure cannot be implemented. Under these circumstances, the discontinuous end point is moved along the cell edge so that the curvatures at the midpoint of the interfaces for the consecutive mixed cells are matched.
6. The application of the above C^0 and C^1 correction procedure (steps 3–5) to all the mixed cells over a few iterations renders a smooth quadratic spline representation of all the interface segments in the domain.

Fig. 2 shows the implementation of the above steps in reconstructing the sector of a circle. The details of the initial PLIC prediction and the different hierarchy of corrections involved are given elaborately in the remainder of this section.

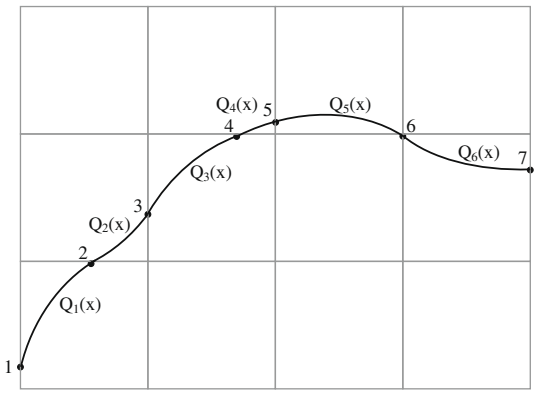

Fig. 1. Quadratic spline representation of the interface.

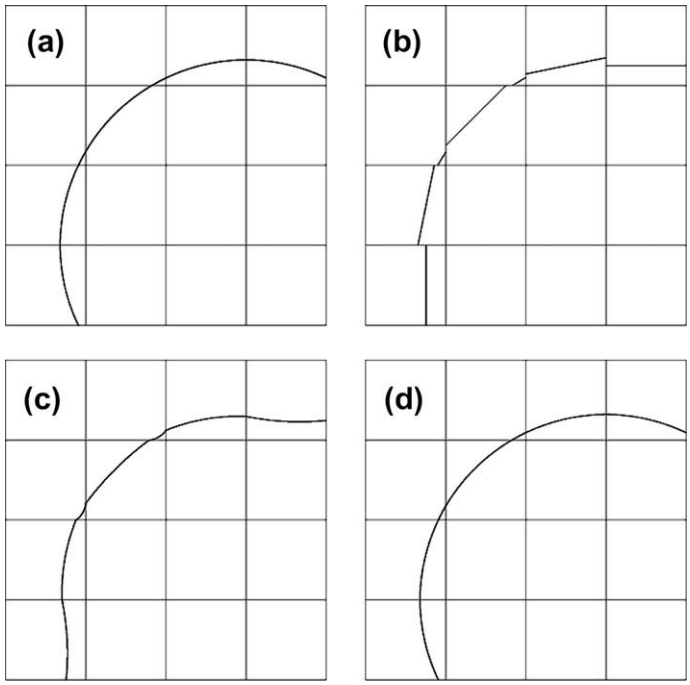

Fig. 2. (a) Original shape of the sector of a circle; (b) initial interface approximation using PLIC reconstruction; (c) after implementing C^0 continuity and (d) after implementing C^1 correction.

2.1. Initial PLIC reconstruction

The PLIC reconstruction procedure basically involves two steps namely: the estimation of interface orientation (normal) and the exact positioning of interface subjected to volume constraint in the cell. A variety of normal estimation techniques have been used in literature and these include Youngs' [6] multi-dimensional stencil, LVIRA of Puckett et al. [13], ELVIRA of Pilliod and Puckett [14] and the least-square linear and quadratic fitting methods of Scardovelli and Zaleski [15]. In the current work, Youngs' multidimensional stencil method [6] has been used for the estimation of interface normal, because of its relative simplicity. Even though the other techniques possess higher order interface reconstruction accuracy [13–15], the iterative C^0 and C^1 corrections employed in the QUASI algorithm reduce the sensitivity of the final interface shape on the initial approximation. With the interface normal orientation thus obtained, the position of the linear interface segment can be precisely [16,17] located based on the volume of primary fluid in the cell.

2.2. Establishing C^0 continuity

The PLIC interfaces thus generated in the mixed cells are generally discontinuous, in contrast to the realistic fluid interface configuration. The current QUASI algorithm resolves this problem by forcing C^0 continuity of the interface segments in the consecutive mixed cells. Generally, three scenarios of interface alignment arise from an initial PLIC representation. In the first scenario, the end points of the PLIC interface in consecutive mixed cells lie on the common edge of the cells (Fig. 3(a)) and the C^0 continuous point is obtained by averaging the coordinates of the two discontinuous end points. In the second case, the end point of interface in one of the cells alone may lie on the common edge (Fig. 3(b)) and hence, any one end point of the interface in the neighbouring cell is moved to the common edge to obtain the C^0 continuity. The use of the quadratic form of interface representation allows arbitrary movement of interface end points and preserves the volume constraint criterion in the cell by adjusting the midpoint of the interface segment suitably. The criterion for selecting the end point to be moved to the common edge is explained in the following Section 2.2.1. The third scenario generally arises in the case of mixed cells which are diagonal (Fig. 3(c)) to each other. Here, obtaining an explicit C^0 continuity is not possible without introducing a new mixed cell. Sometimes, similar problems also arise for mixed cells located on the boundary of the domain. This situation is left to be dealt with the curvature correction procedure explained in Section 2.3.2.

2.2.1. Selection of movable end point

Establishment of C^0 continuity in situations where only one interfacial end point in consecutive mixed cells lies on the common edge requires special care. Fig. 4(a) illustrates such a situation where the end point b_I of the interface in cell I alone lies on the common edge 1–2. Such an initial PLIC interface configuration is fundamentally inconsistent, as vertex 2 lies in

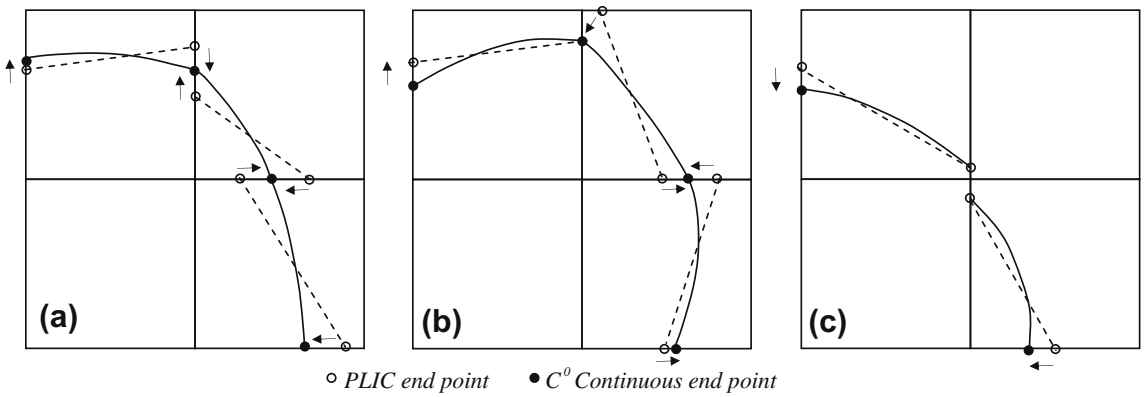

Fig. 3. C^0 continuity from the initial PLIC interface: (a) end points on the common edge; (b) end points not on the common edge and (c) diagonal mixed cells with un-connectable end points.

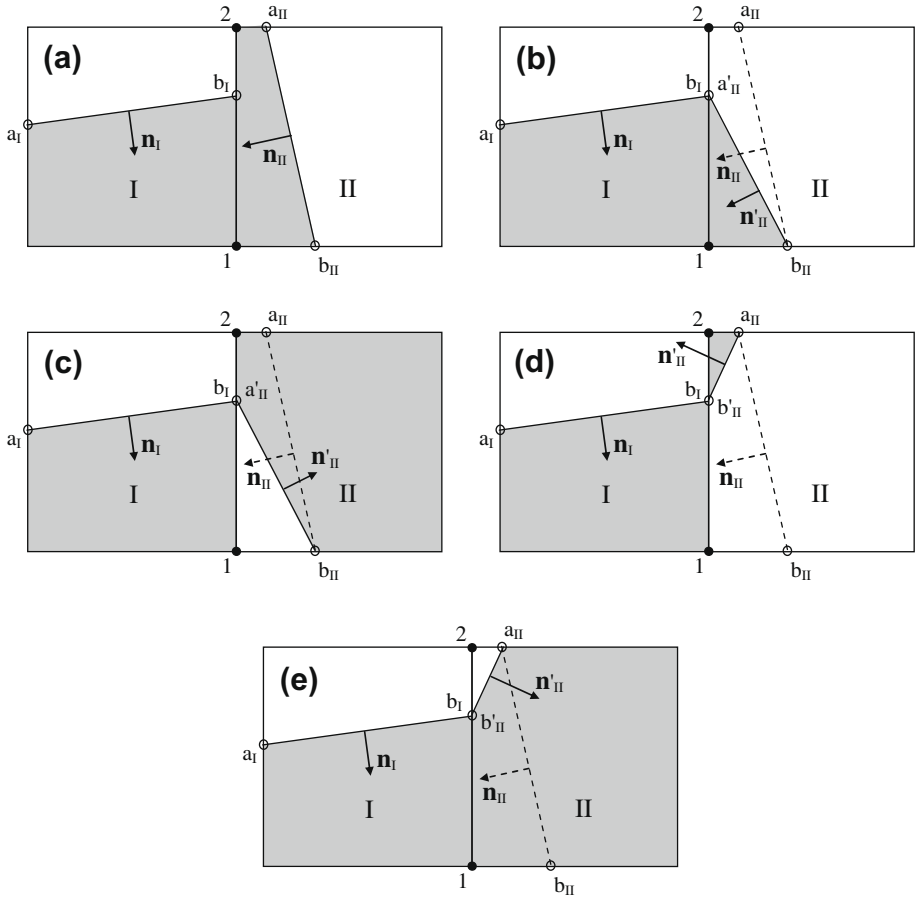

Fig. 4. C^0 Continuity from PLIC Interfaces with only one end point on the common edge.

the secondary fluid zone with respect to cell I and in primary fluid zone with regard to cell II. Proper establishment of C^0 continuity would alleviate such inconsistencies and this aspect is used as the first and main guiding criterion to select the end point of cell II that will merge with point b_I on the common edge. Also the interface obtained with the new end points in cell II cannot have orientation which is at odds with the original orientation obtained from the gradients of volume fraction since, any large variation would impose fluid topological inconsistencies at other locations in the domain. This second criterion is functionally represented by the scalar product $\vec{n}_{II} \cdot \vec{n}'_{II}$, where $\vec{n}_{II}$ is the normal of the interface obtained from

the volume fraction gradients and $\vec{n}'_{II}$ is the orientation (normal) of the new interface obtained after moving one end point of interface to the common edge in cell II. Ideally, the value of this dot product should be close to one for the configuration that is least deviant from the original interface. In order to identify the appropriate C^0 merger of end points, the different possibilities of end-point relocation and primary fluid filling configurations need to be considered. Figs. 4(b)–4(e) show all possible ways of end point movement and primary fluid relocation for the initial PLIC configuration shown in Fig. 4(a). The figures show only the tentative fluid zones and not the actual fluid areas as further adjustments for slope continuity and volume constraint need to be applied. In the cases (c) and (d), the vertex 2 is still in an inconsistent position with respect to fluid zones of cells I and II. Out of the remaining configurations, case (b) is the most appropriate one since it has the maximum value for the dot product of interface normals and also it implicitly maintains fluid consistencies at all the vertices of cell II.

2.3. Piecewise parabolic representation

After obtaining the C^0 continuous end points in all the mixed cells, piecewise quadratic curves are fitted in each of the cells, subjected to the volume constraint criteria. In the current work, analytical expressions have been derived which relate the coefficients of the quadratic function to the local volume fraction and the locations of the end points lying on the cell boundary. The quadratic function in each interfacial cell can be represented by a parabola of the form,

$$Y = aX^2 + bX + c \tag{2}$$

when the axis of the parabola is close to the vertical (Y) direction. Since the real fluid interface can have arbitrary orientation within a cell, it is better to introduce a coordinate transformation for each mixed cell and express the equation of the parabola in terms of the local coordinates. Price et al. [9] placed the local coordinate system at the center of the interfacial cell with angle of rotation 'θ' defined with respect to the global coordinate system. The parameter 'θ' and the coefficients a, b and c, were determined using the method of least squares. However, in the current work, the local coordinate system is placed at the cell vertex in the primary fluid region that is far away from the line connecting the new end points (Fig. 5) and the orientation is fixed based on the normal of the connecting line. This guarantees that all the points and respective projected areas lie in the positive ordinate region of the local coordinate system.

The relations between the global (X,Y) and the local coordinate (x,y) system are given as,

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} n_X & n_Y \\ -n_Y & n_X \end{bmatrix} \left\{ \begin{bmatrix} X \\ Y \end{bmatrix} - \begin{bmatrix} X_p \\ Y_p \end{bmatrix} \right\}, \tag{3}$$

where (X_p, Y_p) is the location of the reference vertex in the global coordinate system and (n_X, n_Y) are the direction cosines of the normal for the line connecting the end points of the parabola.

Fig. 5 shows a typical interfacial cell (volume fraction F and area A_{cell}) in the global coordinate system (X–Y) with end points 1 and 3 lying on the cell boundary. The local coordinate system (x–y) is such that the origin is at vertex P and the orientation of the y-coordinate is perpendicular to the line connecting the end points 1 and 3. A_{ph} is the total area (formed by two triangles here) between the abscissa of the local coordinate system and the cell periphery ($1 - P - 3$) connecting the end points in anti-clockwise orientation. Since three points are required for the complete description of parabola defined in Eq. (2), point 2 is selected such that

$$x_2 = \frac{x_1 + x_3}{2}. \tag{4}$$

Hence for the fixed end points (1 and 3), finding the ordinate of the point 2 (y_2) subjected to the volume constraint provides the parabolic function for the interface in the cell. The coefficients of the parabola (a, b and c) passing through the points 1, 2 and 3 can be determined as

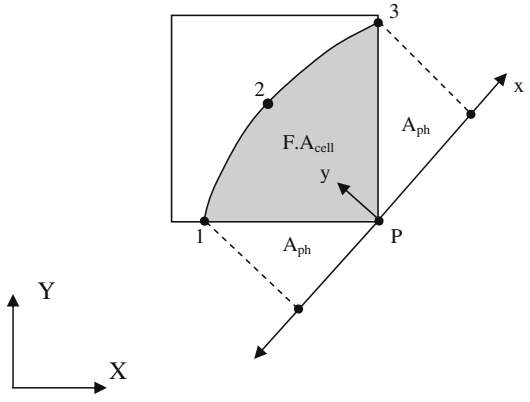

Fig. 5. Piecewise parabolic interface in a cell.

$$a = \frac{2(y_1 + y_3 - 2y_2)}{(x_3 - x_1)^2}, \tag{5a}$$

$$b = \frac{x_1^2(y_1 + 3y_3 - 4y_2) - x_3^2(3y_1 + y_3 - 4y_2) + 2x_1x_3(y_1 - y_3)}{(x_3 - x_1)^3}, \tag{5b}$$

$$c = \frac{(x_1 + x_3)(x_1y_3 + x_3y_1) - 4x_1x_3y_2}{(x_3 - x_1)^2}. \tag{5c}$$

The total area under the parabola between the end points 1 and 3 can be obtained as

$$\int_1^3 y\,dx = \int_1^3 (ax^2 + bx + c)\,dx = \frac{a(x_3^3 - x_1^3)}{3} + \frac{b(x_3^2 - x_1^2)}{2} + c(x_3 - x_1). \tag{6}$$

By substituting for the coefficients of a, b and c from Eqs. (5a)–(5c), this area expression can be simplified as

$$\int_1^3 y\,dx = \frac{(y_1 + y_3 + 4y_2)(x_3 - x_1)}{6}. \tag{7}$$

From Fig. 5 it is seen that the area under the parabola is equal to the sum of primary fluid area within the cell and the area (A_{ph}) under the cell boundary within the endpoints of the parabola. i.e.

$$\int_1^3 y\,dx = F \cdot A_{\mathrm{cell}} + A_{\mathrm{ph}}(x_1, x_3), \tag{8}$$

where

$$A_{\mathrm{ph}}(x_1, x_3) = \left[(x_P - x_1)\left(\frac{y_P + y_1}{2}\right)\right] + \left[(x_3 - x_P)\left(\frac{y_3 + y_P}{2}\right)\right]. \tag{9}$$

The functional form of the above area will change based on the number of vertices lying in the primary fluid zone of the cell. From Eqs. (7) and (8), the ordinate of the point '2' is obtained as,

$$y_2 = \frac{1}{4}\left\{\frac{6[F \cdot A_{\mathrm{cell}} + A_{\mathrm{ph}}(x_1, x_3)]}{(x_3 - x_1)} - (y_1 + y_3)\right\}. \tag{10}$$

With Eq. (10), the complete description of the parabola is obtained as a function of the end points and the volume of the primary fluid in the cell.

2.4. C^1 correction

After obtaining the piecewise parabolic form of the interface passing through the C^0 continuous end points in each of the mixed cells, the next step involves the enforcement of C^1 continuity, where the slopes of the parabolas on either side of the common edge are made the same. In the current work, analytical expressions have been derived which directly estimate the location of the new end point on the common edge for a given set of consecutive mixed cells based on the volume constraint in the respective cells. Generally, the slopes of parabola at the two end points (1 and 3) are given as

$$\left.\frac{dy}{dx}\right|_3 = 2ax_3 + b = \frac{(x_3 - x_1)(2y_1 + 4y_3) - 6[F \cdot A_{\mathrm{cell}} + A_{\mathrm{ph}}(x_1, x_3)]}{(x_3 - x_1)^2}, \tag{11a}$$

$$\left.\frac{dy}{dx}\right|_1 = 2ax_1 + b = \frac{(x_1 - x_3)(4y_1 + 2y_3) + 6[F \cdot A_{\mathrm{cell}} + A_{\mathrm{ph}}(x_1, x_3)]}{(x_3 - x_1)^2}. \tag{11b}$$

In order to achieve C^1 continuity at the common edge, a common coordinate system is considered for each pair of the interfacial cells. A typical configuration of such system is shown in Fig. 6. Here the cells I and II have the common edge $(i - j)$ where the C^0 continuous initial end point of the parabolas in the cells is $3'$. The volume of the primary fluid as per the C^0 continuous interface approximation is shown as the shaded zone.

Now the slopes of the two parabolas of cells I and II at point $3'$ are given by

$$\left.\frac{dy}{dx}\right|_{3'}^{\mathrm{I}} = (2ax_{3'} + b)_{\mathrm{I}} = \frac{(x_{3'} - x_1)(2y_1 + 4y_{3'}) - 6[(F \cdot A_{\mathrm{cell}}) + A_{\mathrm{ph}}(x_1, x_{3'})]_{\mathrm{I}}}{(x_{3'} - x_1)^2}, \tag{12a}$$

$$\left.\frac{dy}{dx}\right|_{3'}^{\mathrm{II}} = (2ax_{3'} + b)_{\mathrm{II}} = \frac{(x_{3'} - x_2)(4y_{3'} + 2y_2) + 6[(F \cdot A_{\mathrm{cell}}) + A_{\mathrm{ph}}(x_{3'}, x_2)]_{\mathrm{II}}}{(x_2 - x_{3'})^2}. \tag{12b}$$

Now parametrizing the point $3'$ in terms of distance (α) along the common edge, we obtain

$$\begin{aligned} x_{3'} &= x_i + \alpha(x_j - x_i), \\ y_{3'} &= y_i + \alpha(y_j - y_i). \end{aligned} \tag{13}$$

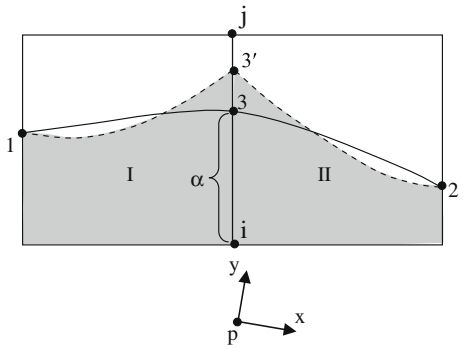

Fig. 6. Slope continuity at the common edge from function continuity.

Substituting for the coordinates of 3‘ and area (A_{ph}) from Eqs. (13) and (9) respectively, Eq. (12a) can be rewritten in terms of ‘α’ as

$$\left.\frac{dy}{dx}\right|_{\alpha}^{I} = \frac{A_I\alpha^2 + B_I\alpha + C_I}{(D_I\alpha + E_I)^2}, \tag{14}$$

where the coefficients are

$$A_I = (x_j - x_i)(y_j - y_i),$$
$$B_I = [2(x_j - x_i)(y_1 - y_i) + 4(y_j - y_i)(x_i - x_1)],$$
$$C_I = (x_i - x_1)(2y_1 + 4y_i) - 6[F.A_{cell} + A_{ph}(x_1, x_i)]_I,$$
$$D_I = (x_j - x_i),$$
$$E_I = (x_i - x_1).$$

Similarly, Eq. (12b) can be rewritten as

$$\left.\frac{dy}{dx}\right|_{\alpha}^{II} = \frac{A_I\alpha^2 + B_{II}\alpha + C_{II}}{(D_I\alpha + E_{II})^2}, \tag{15}$$

where

$$B_{II} = [2(x_j - x_i)(y_2 - y_i) + 4(y_j - y_i)(x_i - x_2)],$$
$$C_{II} = (x_i - x_2)(4y_i + 2y_2) + 6[F.A_{cell} + A_{ph}(x_i, x_2)]_{II},$$
$$E_{II} = (x_i - x_2).$$

Eqs. (14) and (15) can be equated to obtain the point 3 at which the slopes of parabola on either side of the cell are equal. This yields a cubic equation in ‘α’ of the form

$$f(\alpha) = P\alpha^3 + Q\alpha^2 + R\alpha + S = 0, \tag{16}$$

where the coefficients are given as

$$P = D_I^2(B_I - B_{II}) + 2A_ID_I(E_{II} - E_I),$$
$$Q = D_I^2(C_I - C_{II}) + A_I\left(E_{II}^2 - E_I^2\right) + 2D_I(B_IE_{II} - B_{II}E_I),$$
$$R = \left(B_IE_{II}^2 - B_{II}E_I^2\right) + 2D_I(C_IE_{II} - C_{II}E_I),$$
$$S = C_IE_{II}^2 - C_{II}E_I^2.$$

Eq. (16) can be solved directly to obtain the C^1 correction for the given pair of interfacial cells. The application of the above C^1 correction procedure to all the interfacial cells iteratively leads to the smooth and continuous quadratic spline representation of the interface as shown in Fig. 2(d). Typically, the optimum number of iterations for the overall C^1 correction has been found to be 10 in most cases, as further operations do not seem to alter the interface end points significantly.

2.5. Curvature correction

For the cases where a C^0 continuous end point cannot be determined by a simple merger of PLIC interface end points (Fig. 3(c) or in the case of a boundary cell), the application of the above C^1 correction procedure is not feasible. However, such an uncorrected end point would result in interface representation which is no better than the PLIC reconstruction locally. This necessitates an alternative correction procedure in which the discontinuous end point is moved along the cell boundary in such a way that the curvature at the mid point of the interface matches the curvature of a selected neighboring

mixed cell. The selected interfacial cell for this purpose should not be close to empty or full (i.e. $0.02 < F < 0.98$), as the points which define the parabolic curve in these cells are generally close to each other in the physical domain and hence, can lead to erroneous extrapolation as compared to the cells which are half-filled. Once again analytical expressions have been derived to aid in locating the end point where the curvature of these interfaces match.

In terms of the coefficients (a, b and c), the curvature (κ) of a parabola at any point is given by

$$\kappa = \frac{-2a}{(1+(2ax+b)^2)^{3/2}}. \tag{17}$$

By substituting for coefficients a and b, the curvature at the mid point of the parabola (point 2) can be obtained as a function of the end points 1 and 3 as

$$\kappa = \frac{-2\{3(y_1+y_3)(x_3-x_1) - 6[F\cdot A_{\text{cell}} + A_{\text{ph}}(x_1,x_3)]\}}{((x_3-x_1)^2+(y_3-y_1)^2)^{3/2}}. \tag{18}$$

Fig. 7 shows the configuration of a parabola for which the point 1 is C^0 continuous and point 3′ of cell II is discontinuous and hence correcting this end point involves satisfying the following condition,

$$\kappa|^{\text{I}}_{2'} = \kappa|^{\text{II}}_{2}, \tag{19}$$

where $\kappa|^{\text{I}}_{2'}$ is the midpoint curvature of the neighbouring interfacial cell. The curvature $\kappa|^{\text{II}}_{2}$ can be parameterized, based on either of the end points using the relations described in Eq. (13). For a given cell, the dependence of the midpoint curvature on the end point 3′ can be expressed as

$$\kappa(x_3)|^{\text{II}}_{2} = \frac{A\alpha + B}{\{C\alpha^2 + D\alpha + E\}^{3/2}}, \tag{20}$$

where

$$\begin{aligned}
A &= -6[(y_j-y_i)(x_i-x_1) + (y_1-y_i)(x_j-x_i)],\\
B &= -6(y_1+y_i)(x_i-x_1) + 12[F\cdot A_{\text{cell}} + A_{\text{ph}}(x_1,x_i)],\\
C &= [(x_j-x_i)^2 + (y_j-y_i)^2],\\
D &= 2[(x_j-x_i)(x_i-x_1) + (y_i-y_1)(y_j-y_i)],\\
E &= [(x_i-x_1)^2 + (y_i-y_1)^2].
\end{aligned}$$

In certain cases, the midpoint curvature may have to be parameterized in terms of the end point 1 and in such circumstances, the resulting coefficients of Eq. (20) will be

$$\begin{aligned}
A &= 6[(y_j-y_i)(x_i-x_3) + (y_3-y_i)(x_j-x_i)],\\
B &= 6(y_3+y_i)(x_i-x_3) + 12[F.A_{\text{cell}} + A_{\text{ph}}(x_i,x_3)],\\
C &= [(x_j-x_i)^2 + (y_j-y_i)^2],\\
D &= 2[(x_j-x_i)(x_i-x_3) + (y_3-y_i)(y_j-y_i)],\\
E &= [(x_i-x_3)^2 + (y_i-y_3)^2].
\end{aligned}$$

From Eq. (20), the analytical relation for matching curvature of the neighbouring cells is obtained as

$$f(\alpha) = \frac{A\alpha + B}{\{C\alpha^2 + D\alpha + E\}^{3/2}} - \kappa|^{\text{I}}_{2'} = 0. \tag{21}$$

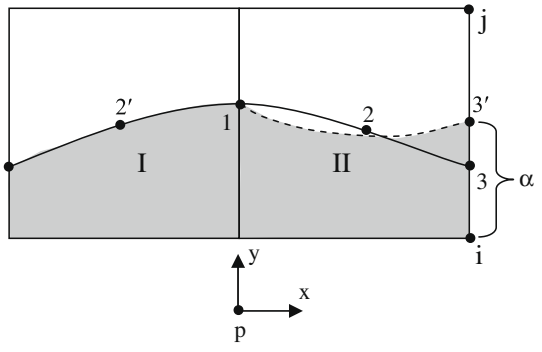

Fig. 7. Matching curvature at discontinuous and boundary interfaces.

The above equation can be solved iteratively to obtain the new location of the discontinuous end point at which the desired curvature conditions are fulfilled. In cases where the boundaries of the domain are defined by solid walls, the end point correction can also be achieved by matching the slope of the curve (Eq. (14) a and b) with the known contact angle for the fluid interface. The iterative use of Eq. (21) in conjunction with Eq. (16) applied to all interface segments results in a smooth quadratic spline representation of the entire interface.

During the above correction through Eq. (21), if any end point of the interface does not lie on the correct cell edge (due to the initial PLIC approximation), the value of α obtained will become out of bounds ($\alpha < 0$ or $\alpha > 1$). In such cases, the end point is allowed to jump over to the consecutive edge of the cell, provided the resultant configuration does not end up having both the interface end points of the cell located on the same edge. This scenario is illustrated in Fig. 8 where the initial PLIC end points (Fig. 8(b)) for the fluid interface configuration shown in Fig. 8(a) cannot be forced to meet at the common edge (2–4) of cells I and II using the C^0 correction procedure described in Section 2.2. As a result, these end points are adjusted through the curvature correction procedure explained above, which will result in the value of α exceeding the allowable limits ($\alpha < 0$ or $\alpha > 1$). This has to be generally constrained to lie within the extents of the cell edge (edge 2–1 for cells I and edge 3–2 for the cell II) by making $\alpha = \varepsilon$ if $\alpha < 0$ and $\alpha = 1 - \varepsilon$ if a > 1, where ε is very small. However, such a constrained QUASI interface shown in Fig. 8(c) is still inconsistent with respect to fluid zones and will result in erroneous curvature estimates. This problem can be easily rectified by forcing (appropriately setting $\alpha = 0$ or $\alpha = 1$) the end points to meet on vertex 2, which will automatically form a C^0 continuous point on the common edge 2–4. In other words, the end point of interface lying on the edge 1–2 of cells I and on edge 2–3 of cell II jumps over the vertex 2 to get associated with the common edge 2–4. Subsequent C^1 corrections result in an interface configuration (Fig. 8(d)) which is almost identical to the shape of the original fluid structure.

2.6. Remarks

The iterative correction steps involved in the QUASI algorithm significantly augment the robustness of initial PLIC representation for the interface. In fact, a proper establishment of C^0 continuity across the cells as described here actually enhances the consistency of adjacent fluid zones in the domain, which a typical PLIC representation generally misses out. Also, the current implementation of various constraining procedures ensures localization of errors and aids in a faster attainment of stable and consistent interface configuration. Any correction process involving a pair of mixed cells does not affect the volume conservation in their neighborhood since the movement of an end point of a parabola is compensated by altering the location of central point '2' to satisfy volume constraint in the cell. This process just induces a small slope imbalance at the other end point and does not mandate a large displacement of the point. Hence, any correction process is highly localized and this feature of the algorithm has been observed to ensure faster convergence of the overall interface geometry. On the other hand, representation in the form of a spline using two points per interfacial cell as in [11] requires significant change in

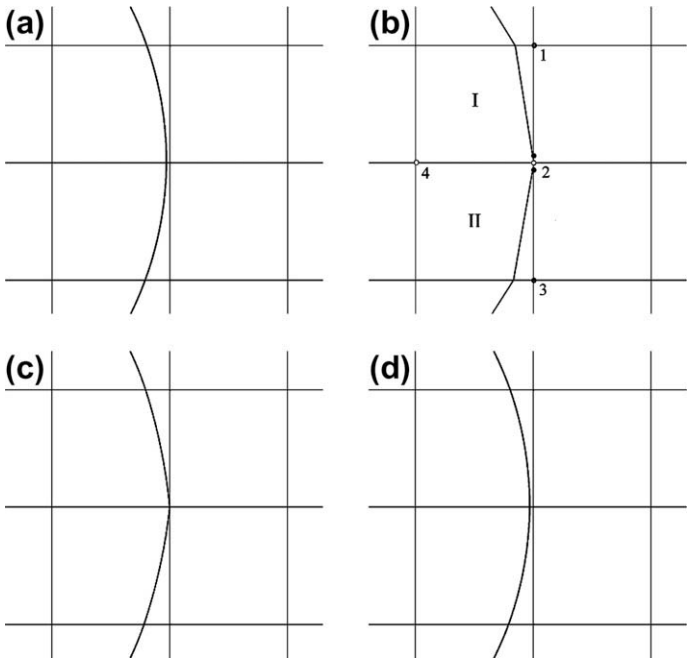

Fig. 8. Vertex jumping of discontinuous end points to obtain C^0 continuity: (a) original shape of the sector of a circle; (b) discontinuous PLIC interface; (c) QUASI interface without vertex jump and (d) QUASI interface with vertex jump.

the position of the other end point to satisfy volume conservation. This in turn, necessitates altering the end point positions in the subsequent cells also. Hence any local error in interface representation propagates to the whole spline, often leading to interfacial oscillations.

3. Static interface reconstruction using QUASI algorithm

The efficacy of the QUASI algorithm can be assessed using reconstruction tests of standard interface geometries like straight lines, circles and squares. All the tests carried out here use a square domain of size 1×1 with the center of the circle/square or the base point of the straight line located randomly within the diagonal limits of (0.4, 0.4) and (0.6, 0.6). The truncated volume of primary fluid in each cell is estimated analytically and a cell is defined as interfacial when $\varepsilon < F < 1 - \varepsilon$, where ε is 10^{-6}. The L_1 norm of the error between the original interface $Q_{ex}(x, y)$ and the reconstructed interface $Q_{rec}(x, y)$ in the mixed cells is used to quantify the accuracy of various reconstruction procedures

$$L_1 = \frac{1}{l} \sum_{i=1}^{N} \int |Q_{ex}(x, y) - Q_{rec}(x, y)| dx, \tag{22}$$

where 'l' refers to the total length of interface segment. The summation (Eq. (22)) described above is carried out over all the mixed cells (N) in the domain and the integral inside the summation actually corresponds to the area between the original and the reconstructed interface representation. In order to avoid accidental favoring of any particular interface orientation by the reconstruction method, an average error of over 1000 randomly generated sample geometries has been reported for each test problem. The results of the current method have been compared with those obtained by the PLIC methods of Youngs' [6] and Pilliod and Puckett [14]. The convergence rate of each algorithm is estimated based on the power law variation of truncation error with the grid spacing. The functional form of the convergence rate is expressed as,

$$\Theta = \frac{ln\left(L_1^{k+1}/L_1^{k}\right)}{ln(0.5)}, \tag{23}$$

where the superscript k represents the level of mesh refinement.

3.1. Reconstruction of random circles

Structures of circular shape have great relevance in bubble/droplet dynamics and hence understanding the reconstruction accuracy of circles is of vital interest. In the current test, the radius of the circle has been randomly varied between 0.2 and 0.25 keeping in mind the need for the radius to be at least twice larger as compared to the cell size for a successful reconstruction using VOF. Although the parameters selected above restrict the entire circle to lie bounded within the domain, the curvature correction procedure of the QUASI algorithm allows for accurate reconstruction of circles extending outside the domain as well (Fig. 2). Table 1 shows the average reconstruction error of 1000 random circles generated within the domain. The results reveal that the QUASI algorithm has error of two to five orders of magnitude lower than any typical PLIC method. The table also brings out the error reduction from the progressive corrections (PLIC $\rightarrow C^0$ continuity $\rightarrow C^1$ continuity) employed in the current QUASI algorithm. The improved accuracy of QUASI reconstruction is mainly due to the quadratic representation of the interface with the iterative implementation of the C^0 and C^1 continuities further adding to its advantage. A comparison of the QUASI interface with the original circle (Fig. 9) reveals its closeness to the exact shape even on a coarse mesh. On the contrary, the interface obtained from the PLIC reconstruction is discontinuous.

3.2. Reconstruction of random lines

Understanding the ability of the reconstruction algorithm to represent interfaces with straight line shapes is essential as flat interfaces form part of most stratified multiphase flow regimes. In this test, straight lines cutting across the domain are generated randomly with their base point located close to the domain center. The results of the error analysis for 1000 random straight lines are presented in Table 2. As expected, the ELVIRA method [14] reconstructs the straight line interface exactly and using ELVIRA as an initial PLIC estimator for the current QUASI reconstruction would have had a similar effect. Since Youngs' stencil [6] is used to estimate the initial interface shape, the error from the QUASI algorithm is not close to machine zero even though it is very low as compared to the Youngs' algorithm. The error in the current case primarily occurs due to the definition of the truncated mixed cell ($\varepsilon < F < 1 - \varepsilon$) which sometimes results in discontinuity of the interface within the domain. For a generalized and robust QUASI algorithm this truncation becomes necessary and even upon including the error due to this truncation, the level of accuracy obtained in the current test case substantiates the suitability of the QUASI method for all practical problems. It is interesting to see that the asymptotic error reduction is linear with step size due to the straight line geometry being approximated here; in other words, the quadratic approximation of the present algorithm becomes ineffective for this case.

Table 1
Reconstruction error averaged for 1000 random circles.

Grid	Youngs' method (PLIC)		ELVIRA (PLIC)		QUASI (after C^0 correction)		QUASI (after C^1 corrections)	
	Error (L_1)	Order (Θ)	Error (L_1)	Order (Θ)	Error (L_1)	Order (Θ)	Error (L_1)	Order (Θ)
10^2	1.37E−03		2.63E−03		7.96E−04		3.82E−05	
		1.93		1.66		1.58		4.17
20^2	3.60E−04		8.34E−04		2.66E−04		2.13E−06	
		1.58		1.68		2.20		4.02
40^2	1.20E−04		2.59E−04		5.79E−05		1.32E−07	
		1.26		1.82		1.46		3.99
80^2	5.04E−05		7.39E−05		2.10E−05		8.31E−09	
		1.12		1.91		0.94		3.96
160^2	2.32E−05		1.97E−05		1.09E−05		5.34E−10	
		1.05		1.94		0.92		3.31
320^2	1.12E−05		5.13E−06		5.77E−06		5.39E−11	

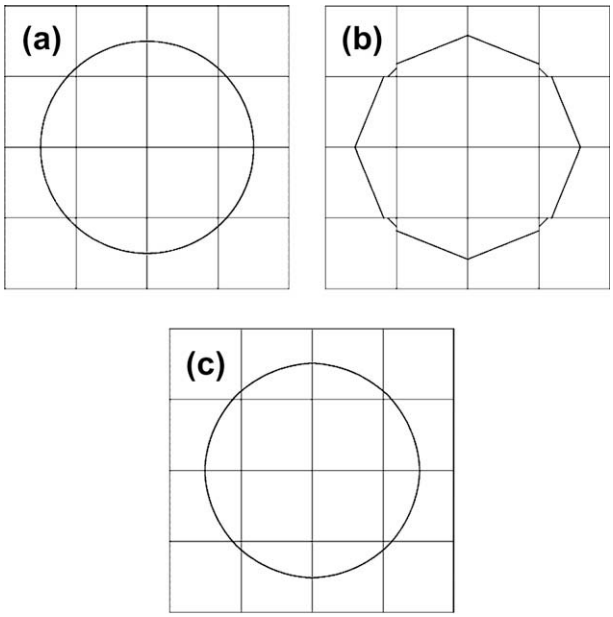

Fig. 9. Reconstruction of a circle: (a) original shape; (b) PLIC reconstruction and (c) QUASI reconstruction.

Table 2
Reconstruction error averaged for 1000 random lines.

Grid	Youngs' method (PLIC)		ELVIRA (PLIC)		QUASI (current)	
	Error (L_1)	Order (Θ)	Error (L_1)	Order (Θ)	Error (L_1)	Order (Θ)
10^2	2.89E−04		∼0		7.04E−12	
		0.82		–		1.29
20^2	1.64E−04		∼0		2.87E−12	
		0.93		–		1.70
40^2	8.61E−05		∼0		8.83E−13	
		0.96		–		1.72
80^2	4.43E−05		∼0		2.68E−13	
		0.98		–		1.01
160^2	2.24E−05		∼0		1.33E−13	

3.3. Reconstruction of random squares

Even though the presence of surface tension rules out the possibility of sharp corners in real fluid structures, it is of academic interest to understand the reconstruction of this singularity by various methods. Bounded squares of random orientation are generated within the domain such that the center of the square lies in the vicinity of the domain center. Also

the side length of the square is randomly varied between 0.4 and 0.6 such that the fluid structure is sufficiently resolved. Fig. 10 shows the PLIC and QUASI interfaces obtained for a square of side length 0.5 aligned with the coordinate axes. The QUASI interface closely follows the actual square, except in the cells which contain the corner singularity. Although the order of curve used in QUASI algorithm is insufficient for an accurate corner representation, the continuous interface configuration obtained here is more reasonable as compared to the PLIC interface. This effect is corroborated from the quantitative L_1 error (Table 3) estimated for 1000 random squares considered within the domain. Since the corner singularity forms the major contributor of error for this geometry, the lower magnitude of error for the QUASI algorithm reveals better representation of the corner due to higher order interface approximation and implementation of C^0 and C^1 continuity constraints on the interface.

3.4. Computational time metrics

Along with the assessment of interface reconstruction accuracy, an estimate of the computational cost involved in using the QUASI method is essential to thoroughly weigh the overall performance of the algorithm. In this regard, the total computational time involved in the reconstruction of 1000 random circles as described in Section 3.2 is reported in Table 4. Understandably, the computational effort required for one C^0 correction and 10 C^1 corrections steps per sample is much higher than the simple gradient calculation procedure involved in the Youngs' PLIC algorithm. This is exactly reflected in the metrics of Table 4 where the ratio of computational time for pure reconstruction process between the current QUASI algorithm and the Youngs' PLIC method is found to vary between 57 and 30. Even with the error minimization procedure, the ELVIRA reconstruction method is 3–7 times faster than the current QUASI algorithm. However, in most practical problems, the interface reconstruction process forms a small part of the multiphase simulation procedure. Hence, any increase in the computational effort of reconstruction does not induce a proportional increase in the total simulation time. In fact, it would be shown later that the computational time increase for the QUASI algorithm in a typical multiphase flow problem (air bubble rise in water) is around 5% of the computational time required by the PLIC method; thus offering a high benefit to cost ratio.

4. Dynamic interface reconstruction using the QUASI algorithm

The dynamic tracking of the fluid structures in a flow situation requires an advection algorithm to be used in conjunction with the interface reconstruction method. Even though the functional accuracies of the interface reconstruction method and the advection process are mutually independent, the cumulative performance of these methods plays an important role in real flow simulations. The details of the advection algorithm used in the current work are briefly described in the remainder of the section. Also two test problems have been discussed which effectively bring out the enhancement in accuracy while using the QUASI interface reconstruction.

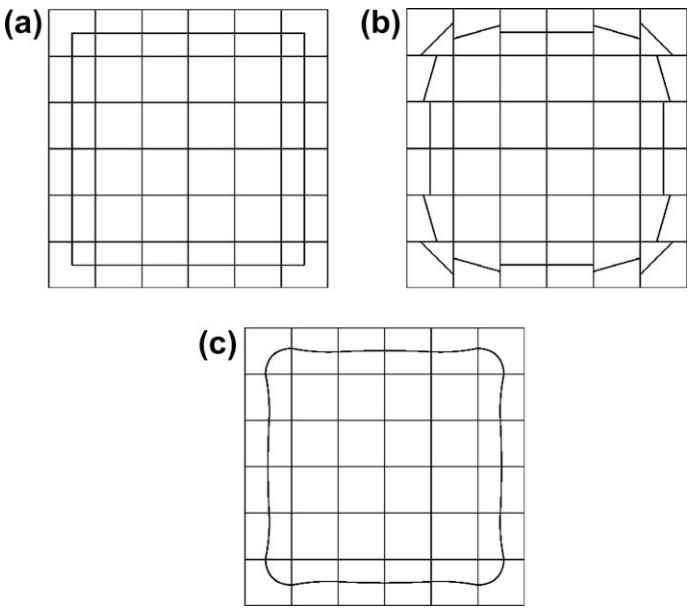

Fig. 10. Reconstruction of a square: (a) original shape; (b) PLIC reconstruction and (c) QUASI reconstruction.

Table 3
Reconstruction error averaged for 1000 random squares.

Grid	Youngs' method (PLIC)		ELVIRA (PLIC)		QUASI (current)	
	Error (L_1)	Order (Θ)	Error (L_1)	Order (Θ)	Error (L_1)	Order (Θ)
10^2	4.97E−03		4.87E−03		2.77E−03	
		1.90		2.00		1.86
20^2	1.33E−03		1.22E−03		7.63E−04	
		1.82		2.02		2.14
40^2	3.76E−04		3.02E−04		1.73E−04	
		1.68		1.99		1.98
80^2	1.17E−04		7.60E−05		4.37E−05	
		1.52		2.00		1.99
160^2	4.06E−05		1.90E−05		1.10E−05	
		1.36		2.00		2.02
320^2	1.58E−05		4.76E−06		2.71E−06	

Table 4
Relative computational time (s) of various reconstruction methods.

Grid	Youngs' method (PLIC)	ELVIRA (PLIC)	QUASI (current)
40^2	0.44	6.74	25.25
80^2	0.92	13.54	52.79
160^2	3.26	28.19	151.01
320^2	13.25	59.38	401.41

4.1. VOF advection

In VOF method, the transient evolution of interface is basically obtained by conserving the primary fluid volume fraction in the domain using the equation

$$\frac{\partial F}{\partial t} + \vec{U} \cdot \vec{\nabla} F = 0. \tag{24}$$

The discrete nature of F necessitates geometric estimation of the convection term in the above equation, which involves calculation of material fluxes through the cell faces based on the reconstructed interface. Conventionally, the time integration of the above equation is carried out by two methods namely, the operator-split and unsplit algorithms. In the current work, the unsplit time integration algorithm named Edge-Matched Flux Polygon Advection (EMFPA) of Lo'pez et al. [12] has been used due to its relative simplicity and better accuracy over other methods. The method primarily constructs non-intersecting polygons which represent the total fluid zones that will advect through each face of the cell during the given time. With the known location and orientation of interface in the cell, the estimation of actual material flux implies identification of primary fluid zones within these flux polygons. In view of these considerations, the scalar conservation equation (Eq. (24)) can be simplified as

$$F^{n+1} = F^n - \frac{1}{A_{\text{cell}}} \sum_{i=1}^{4} A_{Fi}, \tag{25}$$

where A_{Fi} is the area of the primary fluid leaving the cell face 'i' in time Δt (Fig. 11).

The detailed steps for estimating this area for a PLIC configuration have been discussed elaborately by Lo'pez et al. [12]. In the current work, slightly more computations are involved since the interface is represented by a parabola. The steps involved in the area calculation are briefly given below.

1. Construct the flux polygon based on the volumetric efflux through the cell face [12].
2. Find the relative position of the flux polygon vertices with respect to the parabolic interface (i.e. location with respect to the primary fluid zone).
3. Find the points of intersection of the parabolic interface with the flux polygon.
4. Using the flux polygon vertices lying in the primary fluid zone and the points of interface intersection, form a clockwise/counter-clockwise hybrid polygon whose edges are either straight lines or parabolas. For the clockwise hybrid polygon (1–2–5–6) shown in Fig. 9, edges 1–2, 2–5 and 6–1 are straight lines and edge 5–6 is a part of parabola.
5. The area bounded by the hybrid polygon (with n sides) can be obtained by summing the area under these edges taken in proper order. i.e.

$$A_{Fi} = \sum_{p=1}^{n} A_{pq}. \tag{26}$$

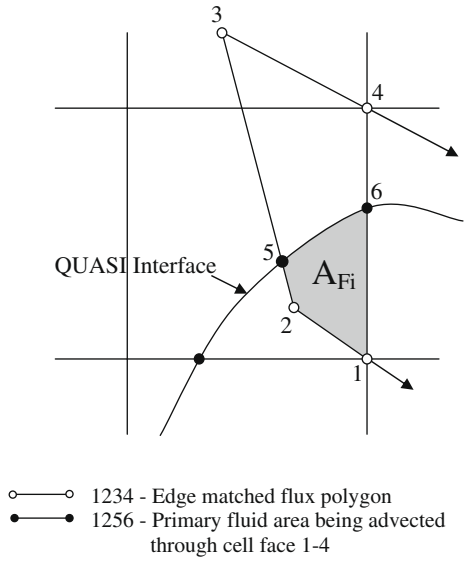

Fig. 11. EMFPA algorithm using QUASI reconstruction.

Here, 'q' is the consecutive vertex of the polygon taken in clockwise or anti-clockwise order and the area A_{pq} under the edge $p-q$ is defined as

$$A_{pq} = \begin{cases} (x_q - x_p)\left(\frac{y_q+y_p}{2}\right) & \text{Edge } p-q - \text{Straight line}, \\ \frac{a\left(x_q^3-x_p^3\right)}{3} + \frac{b\left(x_q^2-x_p^2\right)}{2} + c(x_q - x_p) & \text{Edge } p-q - \text{Parabola}. \end{cases}$$

Using the above EMFPA algorithm implementation, we now quantify the enhancements obtained by the usage of QUASI reconstruction method in some standard benchmark cases with advection.

4.2. Zalesak slotted disk rotation

The first benchmark case is the problem suggested by Zalesak [18] where a slotted circle of the primary fluid is rotated for one complete revolution in a uniform vorticity field. The parameters for the current tests have been adopted from the work of Rudman [19] and the error is defined as

$$E^n = \frac{\sum_{i,j}|F^n_{i,j} - F^0_{i,j}|}{\sum_{i,j}F^0_{i,j}}, \tag{27}$$

where 'n' represents the number of revolutions of the slotted disk.

Table 5 shows the comparison of E^1 errors (Eq. (27)) for various reconstruction and advection techniques after one complete anti-clockwise revolution of the slotted disk. It is evident that the current combination of QUASI reconstruction and EMFPA advection has the lowest magnitude of error among the methods compared in the table. The interface profiles obtained using Youngs' PLIC-EMFPA and QUASI-EMFPA techniques after 1, 5 and 10 complete revolutions are shown in

Table 5
Errors in slotted circle rotation after one complete revolution.

Reconstruction – advection algorithm	Error (E^1)
Youngs – stream [20]	1.07E−02
Puckett – stream [20]	1.00E−02
Young – EMFPA [12/current work]	1.06E−02
SIR – EMFPA [12]	8.74E−02
ACLSVOF (uniform triangular grid) [21]	7.19E−03
ACLSVOF (adaptive triangular grid) [21]	1.25E−02
Linear least square fit – split Lagrangian [15]	9.42E−03
Quadratic fit – split Lagrangian [15]	5.47E−03
Quadratic fit + continuity – split Lagrangian [15]	4.16E−03
QUASI – EMFPA (current work)	2.69E−03

Fig. 12. Zalesak slotted circle after (a) one; (b) five and (c) ten complete revolutions in a uniform vorticity field.

nodes. With the inclusion of advection process in the above test, the proportion of total time spent on the reconstruction process decreases and this reduces the computational time disparity between these methods to a factor of 3 from the high values shown in Table 4.

4.3. Rider–Kothe reversed single vortex

The assessment of accuracy in a rotational field alone does not effectively bring out the usability of reconstruction methods as these flow situations are hardly experienced in real fluids. A better test of interface reconstruction and advection was suggested by Bell et al. [22] and was subsequently modified by Rider and Kothe [23], where the deformation of a circle placed in a time-reversed non-uniform vorticity field was studied. The solenoidal velocity distribution in the domain of size 1×1 is obtained by defining the stream function field as

$$\Psi(x,y,t) = \frac{1}{\pi}\sin^2(\pi x)\sin^2(\pi y)\cos\left(\frac{\pi t}{T}\right). \tag{28}$$

By the above formulation the fluid structure gets deformed to the maximum extent at time $t = T/2$ and returns back to the original position at time $t = T$. The error (defined below) due to the reconstruction/advection algorithm for the given grid spacing 'h' is quantified by comparing the initial and final volume fraction (F) distributions.

$$E^1 = \sum_{i,j} h^2 |F_{i,j}^{\text{final}} - F_{i,j}^{\text{initial}}|. \tag{29}$$

In the current test problem, a circle of radius 0.15 initially placed at (0.5,0.75) is allowed to deform in the above solenoidal velocity field (Eq. (28)) with a time period of $T = 2$ s. Table 6 shows the comparison of E^1 errors obtained by various methods at different CFL numbers (1,0.1) and grid sizes (32×32, 64×64, 128×128). Once again the QUASI/EMFPA algorithm has the lowest error as compared to any other combination of available methods. The improvement in the interface reconstruction is clearly evident from the errors calculated at the two different CFL numbers. For the given time duration, the decrease in the CFL number increases the number of reconstruction and advection steps involved to track the interface and an increase in the error is a cumulative effect of each reconstruction/advection step. Incidentally, the Youngs' stencil in conjunction with the same EMFPA algorithm shows an increase in error at lower CFL number in contrast to the QUASI/EMFPA algorithm; this illustrates the accurate nature of the QUASI reconstruction. Even the least square fitting method of Scardovelli and Zaleski [15] shows an increase in error and only the hybrid marker and VOF technique of Aulisa et al. [21] shows a behavior similar to the QUASI reconstruction. Fig. 13 shows the QUASI and PLIC reconstructed interface profiles of the deformed circle at time $t = T/2$ and $t = T$ for the grids of 32×32 and 128×128.

5. Curvature estimation and surface tension modeling

Modeling of flows with dominant surface tension effects involves the estimation of attributes like local interface curvature for the force evaluation. This generally requires the use of special curvature calculation methods [7,8] which operate directly on the volume fraction distribution in the domain. Alternatively, with the use of higher order interface representation [9–12], the interface curvature can also be obtained directly from the predicted shape of the interface. Such a convenience is available in the current QUASI reconstruction algorithm also, where the use of parabolic interface segments with enforced C^0 and C^1 continuities enables a direct and accurate estimate of the curvature in each mixed cell.

5.1. Accuracy of curvature estimation

In order to quantify the curvature estimation accuracy of the QUASI reconstruction, a simple test suggested by Cummins et al. [8] is used here. A circle of radius 0.25 is placed at (0.5,0.5) in the domain of size 1×1. The mesh size is varied between 0.005 and 0.05 and the L_2 norm of the error in curvature is defined as

Table 6
E^1 errors for a circle placed in reversed single vortex for various reconstruction/advection algorithm ($T = 2$).

Reconstruction – advection algorithm	CFL = 1.0			CFL = 0.1		
	$n_x = 32$	$n_x = 64$	$n_x = 128$	$n_x = 32$	$n_x = 64$	$n_x = 128$
Youngs – stream [20]	2.49E−03	7.06E−04	2.23E−04	-	–	–
Youngs – EMFPA (current)	2.31E−03	6.89E−04	2.25E−04	3.15E−03	8.96E−04	3.16E−04
Youngs – EMFPA [12]	2.32E−03	6.72E−04	2.24E−04	–	–	–
Linear least square fit – EI – LE advection [15]	1.75E−03	4.66E−04	1.02E−04	2.22E−03	–	–
Quadratic least square fit – EI – LE advection [15]	1.88E−03	4.42E−04	9.36E−05	2.00E−03	–	–
Quadratic least square fit + continuity – EI – LE advection [15]	1.09E−03	2.80E−04	5.72E−05	1.14E−03	–	–
Mixed markers and VOF method [21]	1.00E−03	2.69E−04	5.47E−05	6.38E−04	1.57E−04	1.61E−05
SIR – EMFPA [12]	8.62E−04	2.37E−04	5.62E−05	–	–	–
QUASI – EMFPA (current)	6.65E−04	1.57E−04	4.33E−05	2.72E−04	5.85E−05	1.08E−05

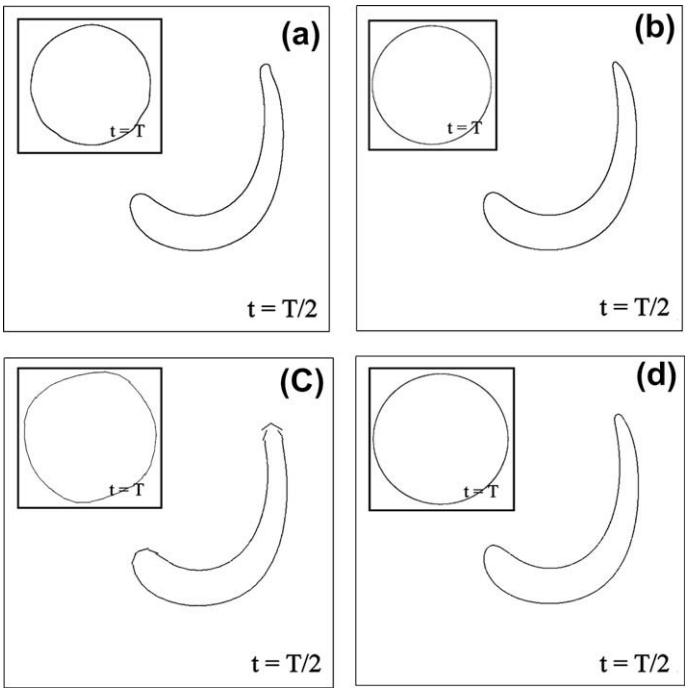

Fig. 13. Reconstructed interface profile at time $t = T/2$; (Inset: $t = T$) at different grid sizes (a) QUASI – 32×32; (b) QUASI – 128×128; (c) PLIC – 32×32 and (d) PLIC – 128×128.

$$L_2 = \sqrt{\frac{\sum_{i=1}^{N} (\kappa_i - \kappa_{\text{exact}})^2}{N}}. \tag{30}$$

Here, the performance of the QUASI reconstruction is compared with the convolution technique of Williams et al. [7] and the Height Function (HF) method described by Cummins et al. [8].

In the convolution technique used here, mollified volume fractions ($\widetilde{F}$) are obtained by convolving the original discrete F field using a radially symmetric sixth-order kernel (K_6) as defined in [24] with a smoothing length of $d = 1.9h^{1/2}$. The continuous nature of this mollified volume fraction allows for the finite difference estimation of the interface normal ($\vec{n} = \nabla \widetilde{F}$) and the curvature ($\kappa = \vec{\nabla} \cdot \vec{n}$). The height function method uses a stencil-based estimation, where discrete sums (volume integrals) of volume fractions along the direction of largest F gradient are differentiated to calculate the local interface curvature. The steps involved in the height function algorithm are discussed elaborately by Cummins et al. [8] and Francois et al. [24]. In the current QUASI algorithm, the curvature is directly estimated at the mid point of the parabolic interface using Eq. (17).

Fig. 14 shows the variation of L_2 norm of curvature error obtained using various curvature estimation methods for the mixed cells with F between 0.01 and 0.99. It is evident that both the height function and the QUASI reconstruction methods possess second order error convergence as compared to the convolution technique which exhibits only first order accuracy. However, the QUASI reconstruction procedure has the lowest magnitude of error as compared to the other algorithms. Although the height function method has second order convergence, it requires the fluid scales to be adequately resolved by the grid and hence the method is unattractive for coarser meshes. The convolution technique, on the other hand, is devoid of this problem; but it is sensitive to the size of the kernel support, whose general behaviour is not well understood. In view of these facts, the current QUASI algorithm offers a promising alternative by possessing both robustness and second order accuracy.

5.2. Modelling of Surface tension effects in multiphase flow

The inclusion of surface tension effects in a fixed-grid multiphase flow model is not a straightforward task, since these effects manifest as local surface forces near the fluid interface. The fundamental difficulties in modelling these local phenomena have been conventionally overcome by the use of Continuum Surface Force (CSF) approximation [25] which represents surface tension as a continuous volumetric force in the vicinity of the interface. In spite of the apparent simplicity, problems arise while modeling flows with dominant surface tension effects, since the CSF approximation tends to produce spurious or anomalous flow currents near the interface. The major reasons for such spurious currents include improper balance of surface tension force and pressure gradients in the CSF implementation and the usage of erroneous curvature estimates to

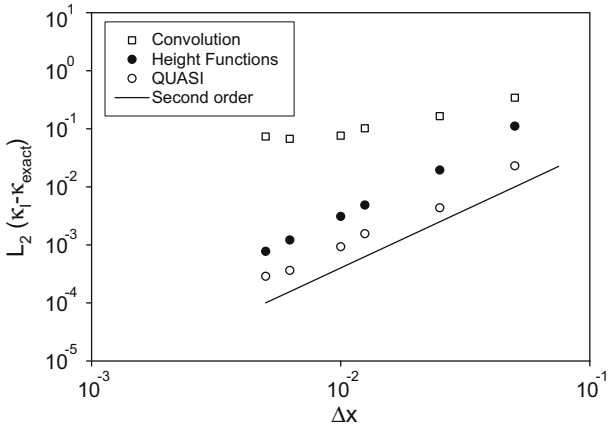

Fig. 14. L_2 norm of error in estimation of curvature for a circular interface.

calculate the body force. While the recent flow algorithms [10,24] have meticulously addressed the former issue by identically discretizing the pressure gradients and the surface tension force at various locations of interest, the special curvature calculation procedure such as the Height function method, the reconstructed distance function technique of Cummins et al. [8], PROST of Renardy and Renardy [10], SIR of Lo'pez et al. [12] and the current QUASI algorithm rigorously attempt to tackle the latter issue.

The flow algorithm used in the current work extends the implementation of Renardy and Renardy [10] to a semi-staggered grid (Fig. 15) based finite volume scheme which allows similar discretization of pressure gradient and surface tension terms at the cell faces and cell centres. The semi-staggered grid arrangement used here stores the velocity information (both u and v) on the cell vertices and the pressure information on the cell centres, thereby forming two types of cells (continuity and momentum cells) to ensure proper pressure–velocity coupling. The volume fraction information is also assigned to the centres of continuity cells since it is intimately coupled with mass balance. A single momentum conservation equation (given below) is applicable for both the phases and it is evaluated using the momentum cell shown in Fig. 15(a)

$$\frac{\partial(\rho\vec{u})}{\partial t} + \nabla\cdot(\rho\vec{u}\vec{u}) = -\nabla(P_1+P_2) + \nabla.(\mu(\nabla\vec{u}+\nabla^T\vec{u})) + \rho\vec{g} + \vec{F}^c_{ST}. \tag{31}$$

Here $\vec{u}$ and $\vec{g}$ represent the velocity field and the acceleration due to gravity, respectively and $\vec{F}^c_{ST}$ is the surface tension force acting at the centre of a momentum cell (in the form of a body force). Properties like density ρ and viscosity μ are evaluated based on the volume fraction of primary fluid in a cell using the expressions

$$\rho = F\rho_{\text{primary}} + (1-F)\rho_{\text{secondary}}, \tag{32}$$

$$1/\mu = F/\mu_{\text{primary}} + (1-F)/\mu_{\text{secondary}}. \tag{33}$$

The current algorithm uses two components of pressure (P_1 and P_2) to satisfy the overall conservation equations. First, the exact balance between the capillary pressure component P_1 and the singular surface tension force $\vec{F}_{ST}$ is achieved by forcing $\vec{F}_{ST} - \nabla P_1$ to be divergence free within the domain. With regard to the current finite volume scheme, this condition can be suitably converted into an integral form of surface force balance equation (given below) which is applicable over each continuity control volume as shown in Fig. 15(b)

$$\oint \frac{\partial P_1}{\partial n}\, dl = \int\int \nabla\cdot\vec{F}^f_{ST}\, dA. \tag{34}$$

Here, the pressure gradients and the surface tension force are identically estimated at the continuity cell faces using the pressure nodes (which also store volume fraction information) on either side of the face. This in turn requires a separate surface tension $\left(\vec{F}^f_{ST}\right)$ formulation in order to obtain proper force balance at these cell faces. For the face $(i+1/2,j)$ of cell (i,j), the surface tension force can be evaluated as

$$\vec{F}^f_{ST} = -\sigma\kappa_{i+1/2,j}\nabla F_{i+1/2,j}, \tag{35}$$

where σ is the coefficient of surface tension and the curvature κ (at the cell face) is obtained as a weighted average [11] of cell centred curvatures in related continuity cells through the expression

$$\kappa_{i+1/2,j} = \frac{\omega_{ij}\kappa_{ij} + \omega_{i+1,j}\kappa_{i+1,j}}{\omega_{ij}+\omega_{i+1,j}}. \tag{36}$$

For any cell 'x', the weighing factor can be obtained as $\omega_x = F_x(1-F_x)$. Also, the volume fraction gradients at the cell faces can be obtained by setting

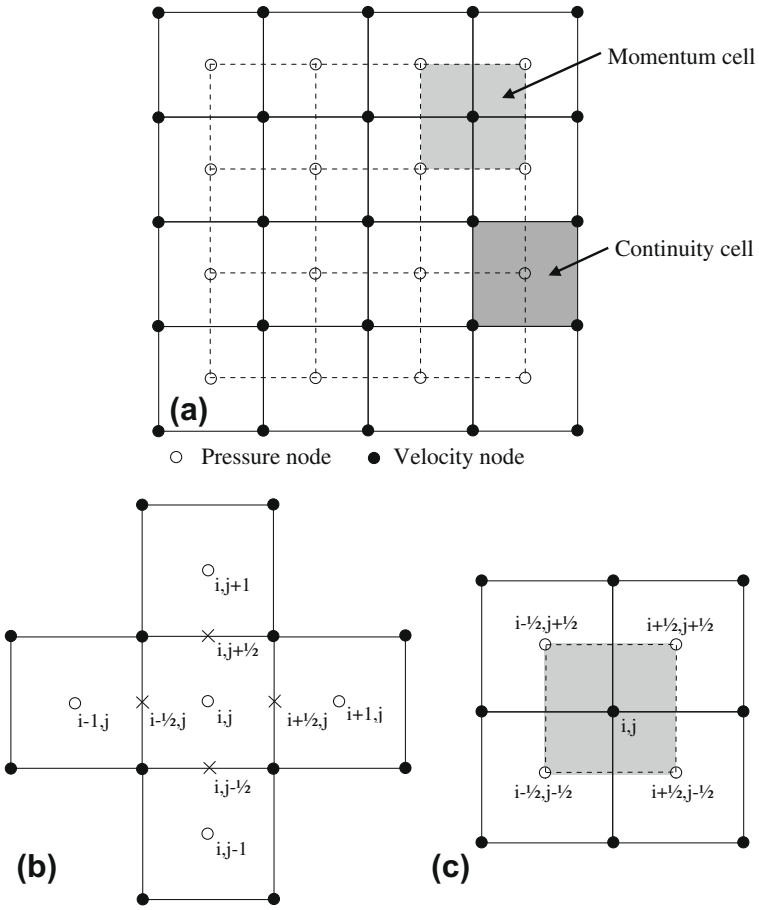

Fig. 15. Schematic of semi-staggered grid arrangement.

$$\frac{\partial F}{\partial x} = (F_{i+1,j} - F_{i,j})/\Delta x; \quad \frac{\partial F}{\partial y} = 0. \tag{37}$$

Using the capillary pressure thus obtained, momentum conservation equation (Eq. (31)) is solved over the momentum cells (Fig. 15(c)) to evaluate the guessed velocity field in the domain. However, any attempt to calculate the surface tension force $\vec{F}^c_{ST}$ at the momentum cell centre using interpolation of values from the continuity cell faces would undermine the premise of perfect force balance. Hence, a distinct formulation of surface tension force (shown below) is used to obtain proper force balance at the momentum cell centres

$$\vec{F}^c_{ST} = -\sigma\kappa\nabla F. \tag{38}$$

Here, the curvature κ (for the momentum cell) is again evaluated as the weighted average of cell centred curvatures in related continuity cells as

$$\kappa = \frac{\omega_{i-1/2,j-1/2}\kappa_{i-1/2,j-1/2} + \omega_{i+1/2,j-1/2}\kappa_{i+1/2,j-1/2} + \omega_{i+1/2,j+1/2}\kappa_{i+1/2,j+1/2} + \omega_{i-1/2,j+1/2}\kappa_{i-1/2,j+1/2}}{\omega_{i-1/2,j-1/2} + \omega_{i+1/2,j-1/2} + \omega_{i+1/2,j+1/2} + \omega_{i-1/2,j+1/2}}. \tag{39}$$

Since the pressure and volume fraction values are collocated at the centre of continuity cells (which are also the vertices of momentum cells), the gradient of pressure in Eq. (31) and volume fraction in Eq. (38) can be identically estimated (as shown below) and thus, a proper force balance can be enforced.

$$\begin{aligned} \frac{\partial F}{\partial x} &= \frac{F_{i+1/2,j+1/2} + F_{i+1/2,j-1/2} - F_{i-1/2,j+1/2} - F_{i-1/2,j-1/2}}{2\Delta x}, \\ \frac{\partial F}{\partial y} &= \frac{F_{i+1/2,j+1/2} + F_{i-1/2,j+1/2} - F_{i+1/2,j-1/2} - F_{i-1/2,j-1/2}}{2\Delta y}. \end{aligned} \tag{40}$$

Finally, the other pressure component P_2 is evolved to satisfy mass conservation, using the Poisson equation given below:

$$\nabla \cdot \left(\frac{1}{\rho}\nabla P_2\right) = \frac{\nabla \cdot \vec{u}}{\Delta t}. \tag{41}$$

In the current work, Incomplete Lower Upper (ILU) preconditioned Bi-Conjugate Gradient Stabilized (BICGSTAB, [26]) linear solver has been used to solve the pressure correction equation and the implicitly discretized momentum equations.

5.3. Inviscid static droplet in equilibrium

The efficacy of the above flow solver implementation can be best tested using the standard problem of an inviscid static droplet placed in a quiescent fluid medium. In this test, usage of exact force estimates in the flow algorithm should yield a velocity field which is close to machine zero and any anomalous behavior is a direct indicator of the persistent force imbalance. The test parameters used here are adopted from the work of Francois et al. [24] where a droplet of radius 2 is placed at the center of an 8×8 domain which is discretized with a mesh of size 40×40 spatially and time step of 10^{-6} s temporally. In order to quantify the effectiveness of force balance implementation, the exact value of interfacial curvature for the above droplet is first used in the surface tension term. The effect of using various other curvature estimates is then evaluated for different density ratios (surrounding medium density (ρ_1) is maintained at 1 and the fluid density (ρ_2) inside the droplet is varied between 1 and 10^5). The error in pressure jump across the interface is evaluated using three different norms ($E(\Delta P_{\text{total}}), E(\Delta P_{partial})$ and $E(\Delta P_{\max})$) as defined by Francois et al. [24].

Table 7 shows the magnitudes of spurious currents and the various pressure error norms for different curvature estimation methods and density ratios. Here, the magnitude of spurious currents obtained for an exact curvature specification comes out to be of the order of round-off error irrespective of the density ratio. This is a consequence of the perfect balance between the pressure gradient and the surface tension force obtained through the current implementation of the flow solver. Although similar inferences can be obtained from the partial and the maximum norm of pressure jump across the droplet, the value of total norm calculated is significantly large even upon using the exact value of curvature in surface tension estimation. This norm quantifies the error involved in small pressure transition region obtained due to the CSF method which generally smoothens the sharp pressure jump across the interface. The magnitude of spurious currents and pressure error norms for other curvature estimation methods is a reaffirmation of the results from Section 5.1 where the current QUASI algorithm was shown to exhibit the lowest magnitude of error with respect to curvature estimation. The decrease in the spurious current velocities with the increase in density ratio conforms to the dimensional analysis of Francois et al. [24] which showed inverse dependence between the velocity magnitude and the droplet density.

Table 8 shows the convergence behavior and the time-wise evolution of spurious currents in the same domain for different curvature evaluation techniques. The order of convergence for the error in pressure jump closely replicates the results obtained for curvature estimation in Section 5.1 where both the Height functions method and the QUASI technique exhibited second order behavior. Also, these methods exhibit monotonic convergence of spurious velocity with respect to mesh size and similar temporal growth rate (compared after 50 and 100 time steps). With its better curvature estimation capability, the QUASI algorithm offers lower magnitude of error in all the situations compared.

5.4. Air bubble rise in water

In order to illustrate the use of the current QUASI algorithm for a complex flow situation, the test problem of an air bubble rising in water column has been studied. This study demonstrates the robustness and applicability of the present interface

Table 7
Magnitude of spurious currents and error in pressure jump for an inviscid static droplet in equilibrium using various curvature prediction methods.

ρ_2/ρ_1	$\lvert\vec{u}\rvert_{\max}$	$E(\Delta P_{\text{Total}})$	$E(\Delta P_{\text{Partial}})$	$E(\Delta P_{\text{Max}})$
(a) Exact				
10^1	1.47E−17	2.92E−02	1.17E−15	3.55E−13
10^3	1.65E−17	2.92E−02	1.17E−15	5.07E−13
10^5	1.65E−17	2.92E−02	7.79E−16	5.05E−13
(b) Convolution				
10^1	3.09E−06	5.01E−02	2.20E−02	3.57E−02
10^3	8.51E−09	4.94E−02	2.17E−02	2.79E−02
10^5	8.54E−11	4.94E−02	2.17E−02	2.79E−02
(c) Height functions				
10^1	8.29E−08	2.45E−02	4.90E−03	6.03E−03
10^3	2.18E−10	2.45E−02	4.89E−03	6.10E−03
10^5	2.19E−12	2.45E−02	4.89E−03	6.10E−03
(d) QUASI				
10^1	2.73E−08	3.03E−02	1.11E−03	9.92E−04
10^3	9.86E−11	3.03E−02	1.10E−03	9.86E−04
10^5	9.94E−13	3.03E−02	1.10E−03	9.86E−04

Table 8
Convergence of error and temporal evolution of spurious currents for different curvature estimation methods.

$R/\Delta x$	$E(\Delta P_{\text{Partial}})$	$\lvert\vec{u}\rvert_{\max,1}$	$\lvert\vec{u}\rvert_{\max,50}$	$\lvert\vec{u}\rvert_{\max,100}$
(a) Convolution				
5	8.598E−02	6.848E−07	3.423E−05	6.839E−05
10	3.758E−02	9.028E−07	4.485E−05	8.969E−05
20	2.032E−02	7.311E−07	3.669E−05	7.338E−05
(b) Height functions				
5	2.152E−02	5.638E−08	2.813E−06	5.644E−06
10	4.924E−03	2.892E−08	1.338E−06	2.645E−06
20	1.201E−03	9.520E−09	4.683E−07	9.419E−07
(c) QUASI				
5	2.167E−03	2.799E−08	1.429E−06	2.883E−06
10	6.639E−04	1.391E−08	6.958E−07	1.372E−06
20	1.683E−04	6.974E−09	3.782E−07	7.473E−07

algorithm for a practical flow problem having large interfacial deformation. The problem uses a domain of size 2×4 with an air bubble of radius 0.5 placed at location (1, 1) initially. The properties of the two fluids and other parameters used for the simulations are listed in Table 9. Due to the lack of experimental and analytical data for the current problem, the temporal evolution of the air–water interface obtained using QUASI reconstruction has been compared with the simulation obtained by PLIC algorithm with height functions employed for curvature estimation.

Fig. 16 shows the transient evolution of the interface obtained using the PLIC and the QUASI reconstruction methods. Although the two interface profiles show a close match initially, the difference between them emerges with time, as clearly observed at $t = 0.75$ s. This fact is corroborated from the values of pressure jump across the interface listed in Table 10 at various locations marked in Fig. 16. The transient thinning and leg formation of the air bubble structure depicted in Fig. 16(c) renders both PLIC and QUASI algorithms ineffective for these under-resolved scales at later times. The liquid–air interfaces after the breakup process ($t = 1.3$ s) are shown in Fig. 17 where similar fluid structures are predicted by these methods. However the QUASI algorithm, despite its robustness at the macro scale, tends to produce minor interfacial oscillations as compared to the PLIC technique involving Height functions for curvature estimation. Possible ways of overcoming this difficulty include local mesh refinement and further sophistication of the reconstruction algorithm which are currently being attempted by the authors.

The extra computational time involved for the above problem using QUASI algorithm is not more than 5% of the time required for the solution using the PLIC method. Even though the time involved in the iterative predictor–corrector operations of the reconstruction process is high as compared to that of a typical PLIC algorithm, this does not have a serious implication

Table 9
Physical properties and simulation parameters for the bubble rise simulation.

Density of air	1.29 kg/m^3
Density of water	1000 kg/m^3
Viscosity of air	1.8×10^{-5} Ns/m^2
Viscosity of water	1.0×10^{-3} Ns/m^2
Coefficient of surface tension	0.072 N/m
Mesh	50×100
Time step	0.001 s

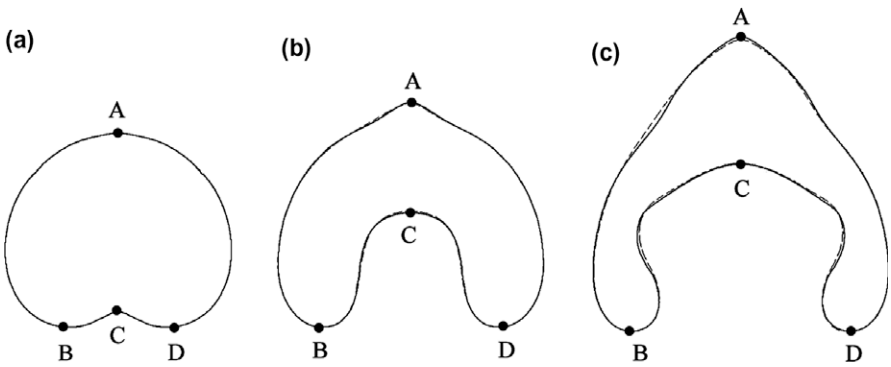

Fig. 16. Exact interface of the rising air bubble in water at (a) $t = 0.25$ s; (b) $t = 0.5$ s; (c) $t = 0.75$ s dotted line – PLIC interface; continuous line – QUASI interface.

Table 10
Pressure jump across the interface at specific locations for a rising air bubble.

Time (s)	A		B		C		D	
	PLIC	QUASI	PLIC	QUASI	PLIC	QUASI	PLIC	QUASI
0.25	432	431.2	1755.83	1753.05	729.45	715.05	1759.13	1753.66
0.5	596.7	562.35	2312.4	2314.2	122.2	108.05	2312.8	2315
0.75	360.7	232.9	1717.4	1678	128.5	113.1	1686.5	1678.2

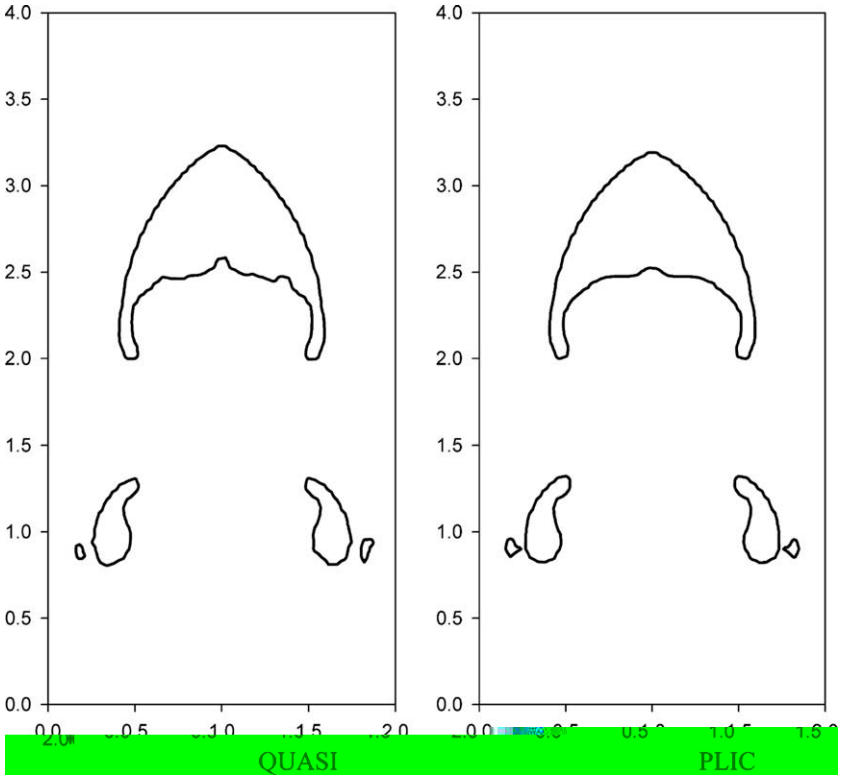

Fig. 17. Liquid–air interface ($F = 0.5$) after 1.3 s from the initial condition.

in the simulation of a coupled multiphase flow problem as the major share of the computational effort is consumed by the flow solver.

Extension of current QUASI algorithm to three dimensions may not be a straight forward task, but is certainly achievable. Possible ways of extension include representation of interfaces by quadratic surfaces, which can be evolved from an initial piecewise planar approximation into a spline-surface representation of the interface. Similar to the approach of the current QUASI algorithm, characteristic end points of the quadratic surfaces on the common cell faces can be adjusted to satisfy various levels of continuity.

6. Summary and conclusions

A new quadratic spline based interface reconstruction algorithm for continuous representation of the interface has been presented. In this method, the initially discontinuous interface obtained by PLIC is progressively converted into to a smooth and continuous quadratic spline curve, by the sequential enforcement of C^0 and C^1 continuities between each pair of neighboring interfacial segments. Since all major changes in interfacial node locations occur during the C^0-continuity stage, only minor corrections are needed in the end point locations during iterative interface slope matching at the cell boundaries. The availability of a free- floating mid- point in the quadratic spline approximation of each interfacial segment aids in the localization of errors and virtually eliminates the global transmission of errors between segments. Employing analytical expressions for predicting the locations of interfacial nodes at all stages of the interface reconstruction significantly reduces the computational requirement. Although the present interface reconstruction procedure involves higher computational time than a PLIC based solution, in a typical multiphase flow simulation exercise where the flow solver requires maximum

computational effort, the simulation time increases only marginally. The continuous representation of the interface facilitates a direct estimate of the interface curvature with a low magnitude of error as well as monotonic second order error convergence. Furthermore, in problems involving surface tension effects, the magnitudes of spurious currents are minimized due to two sequential evaluations of the pressure field, catering to the effects of capillary forces and the evolving flow fields. The basic QUASI algorithm is devoid of any assumption concerning the nature of the grid and hence it can also be coupled with flow solver codes using unstructured meshes.

Acknowledgments

The authors wish to thank Dr. Shaligram Tiwari for the productive discussions they had with him. The authors also acknowledge the financial support of Indira Gandhi Center for Atomic Research (IGCAR), Kalpakkam, India for this work.

References

[1] S. Popinet, S. Zaleski, A front-tracking algorithm for accurate representation of surface tension, Int. J. Numer. Methods Fluids 30 (1999) 775–793.
[2] C.W. Hirt, B.D. Nicholls, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201–225.
[3] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.
[4] R. DeBar, Fundamentals of the Kraken Code, Technical Report UCIR-760, LLNL, 1974.
[5] W.F. Noh, P.R. Woodward, SLIC (simple line interface method), in: A.I. van de Vooren, P.J. Zandbergen (Eds.), Lecture Notes in Phys, vol. 59, Springer-Verlag, Berlin/New York, 1976, p. 330.
[6] D.L. Youngs, Time-dependent multi-material flow with large fluid distortion, in: K.W. Morton, M.L. Norman, (Eds.), Numerical Methods for Fluid Dynamics, 1986.
[7] M.W. Williams, D.B. Kothe, E.G. Puckett, Accuracy and convergence of continuum surface-tension models, in: W. Shyy, R. Narayanan (Eds.), Fluid Dynamics at Interfaces, Cambridge University Press, Cambridge, 1998, pp. 294–305.
[8] S.J. Cummins, M.M. Francois, D.B. Kothe, Estimating curvature from volume fractions, Comput. Struct. 83 (2005) 425–434.
[9] G.R. Price, G.T. Reader, R.D. Rowe, J.D. Bugg, A piecewise parabolic interface calculation for volume tracking, in: Proceedings of the Sixth Annual Conference of the Computational Fluid/Dynamics Society of Canada, University of Victoria, Victoria, British Columbia, 1998.
[10] Y. Renardy, M. Renardy, PROST: a parabolic reconstruction of surface tension for the volume-of-fluid method, J. Comput. Phys. 183 (2002) 400–421.
[11] I. Ginzburg, G. Wittum, Two-phase flows on interface refined grids modeled with VOF, staggered finite volumes, and spline interpolants, J. Comput. Phys. 166 (2001) 302–335.
[12] J. Lo'pez, J. Herna'ndez, P. Go'mez, F. Faura, A volume of fluid method based on multidimensional advection and spline interface reconstruction, J. Comput. Phys. 195 (2004) 718–742.
[13] E.G. Puckett, A.S. Almgren, J.B. Bell, D.L. Marcus, W.J. Rider, A high-order projection method for tracking fluid interfaces in variable density incompressible flows, J. Comput. Phys. 130 (1997) 269–282.
[14] J.E. Pilliod, E.G. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, J. Comput. Phys. 199 (2004) 465–502.
[15] R. Scardovelli, S. Zaleski, Interface reconstruction with least-square fit and split Eulerian–Lagragian advection, Int. J. Numer. Methods Fluids 41 (2003) 251–274.
[16] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, S. Zaleski, Volume of fluid interface tracking with smoothed surface stress methods for three-dimensional flows, J. Comput. Phys. 152 (1999) 423–456.
[17] R. Scardovelli, S. Zaleski, Analytical relations connecting linear interfaces and volume fractions in rectangular grids, J. Comput. Phys. 164 (2000) 228–237.
[18] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, J. Comput. Phys. 31 (1979) 335–362.
[19] M. Rudman, Volume-tracking methods for interfacial flow calculations, Int. J. Numer. Methods Fluids 24 (1997) 671–691.
[20] D.J.E. Harvie, D.F. Fletcher, A new volume of fluid advection algorithm: the stream scheme, J. Comput. Phys. 162 (2000) 1–32.
[21] E. Aulisa, S. Manservisi, R. Scardovelli, A mixed markers and volume-of-fluid method for the reconstruction and advection of interfaces in two-phase and free-boundary flows, J. Comput. Phys. 188 (2003) 611–639.
[22] J.B. Bell, P. Colella, H.M. Glaz, A second-order projection method for the incompressible Navier–Stokes equations, J. Comput. Phys. 85 (1989) 257–283.
[23] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, J. Comput. Phys. 141 (1998) 112–152.
[24] M.M. Francois, S.J. Cummins, E.D. Dendy, D.B. Kothe, J.M. Sicilian, M.W. Williams, A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework, J. Comput. Phys. 213 (2006) 141–173.
[25] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, J. Comput.Phys. 100 (1992) 335–354.
[26] H.A. van der Vorst, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear systems, SIAM J. Sci. Statist. Comput. 13 (1992) 631–644.