INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids **28**: 357–378 (1998)

A VOLUME-TRACKING METHOD FOR INCOMPRESSIBLE MULTIFLUID FLOWS WITH LARGE DENSITY VARIATIONS

MURRAY RUDMAN*

Commonwealth Scientific and Industrial Research Organisation (CSIRO), Division of Building, Construction and Engineering, PO Box 56, Highett, Victoria 3190, Australia

SUMMARY

A numerical technique (FGVT) for solving the time-dependent incompressible Navier–Stokes equations in fluid flows with large density variations is presented for staggered grids. Mass conservation is based on a volume tracking method and incorporates a piecewise-linear interface reconstruction on a grid twice as fine as the velocity–pressure grid. It also uses a special flux-corrected transport algorithm for momentum advection, a multigrid algorithm for solving a pressure-correction equation and a surface tension algorithm that is robust and stable. In principle, the method conserves both mass and momentum exactly, and maintains extremely sharp fluid interfaces. Applications of the numerical method to prediction of two-dimensional bubble rise in an inclined channel and a bubble bursting through an interface are presented. © 1998 John Wiley & Sons, Ltd.

KEY WORDS: interfacial flow; multigrid; surface tension; flux-corrected transport

1. INTRODUCTION

In the numerical computation of immiscible multifluid problems with large density variation (e.g. splashing liquid drops and bubbles bursting at a surface), there is need of an accurate representation of the interface separating the fluids; an accurate and robust representation of surface forces; and a means to accommodate the stiffness introduced into the solution procedure due to discontinuous (and large) spatial density variations.

Several numerical methods have been designed to simulate gas–liquid flows by calculating the liquid flow and neglecting the dynamics of the gas phase. The most well-known is the marker and cell (MAC) method [1], in which Lagrangian marker particles are advected with the local fluid velocity, with their distribution determining the instantaneous fluid configuration. A variant of MAC in which two different sets of marker particles allowed immiscible multifluid problems to be simulated was also presented in Reference [1]. Although the MAC method allows arbitrary, time dependent flows to be simulated, difficulties with the particle representation of the fluid arise. The number density of particles is finite (and typically fairly

* Correspondence to: Division of Building, Construction and Engineering, CSIRO, PO Box 56, Highett, Victoria 3190, Australia.

CCC 0271–2091/98/120357–22$17.50

Received April 1997
Revised November 1997

low), therefore false regions of void can be created and incorrect densities are easily interpolated in flows with high shear (and hence high fluid extension). Although variations of the basic method that incorporate a set of linked marker particles (e.g. Reference [2]) can alleviate some of these problems in two dimensions, its extension to three dimensions is difficult and coalescence and fragmentation of fluid regions cause logical difficulties in the algorithm.

For single valued interfaces, the use of a height function (e.g. Reference [3]), in which the distance between the interface and a reference level is calculated, offers a simple and robust method for simulating interfacial flow in both two and three dimensions. However, the restriction to single valued interfaces rules out a wide class of interesting and important problems.

Boundary integral methods and Lagrangian finite element methods can be difficult to adapt to general interfacial flow simulations, primarily because they cannot easily follow fragmentation and coalescence events that are an essential part of many multifluid problems. A recent Eulerian finite element method for free surface flows has been developed [4], although its ability to treat complex geometries is the only significant advantage over more basic methods such as MAC.

In some important problems (for example the rise of a gas bubble in a liquid), the gas phase dynamics cannot be neglected. The problem then becomes one of an incompressible fluid in which large fluid distortions and large density variations exist (typically three orders of magnitude).

Techniques for simulating the full multifluid case have been developed, for example the SOLA-VOF algorithm of Nichols *et al.* [5] and the interface tracking method of Unverdi and Tryggvason [6]. The SEA algorithm of Pericleous *et al.* [7] also solves the full variable density problem, but uses van Leer flux limiting instead of a special interface treatment to minimise interface diffusion. More recently, techniques for second-order time-accurate simulation of such flows have been based on Godunov-type schemes on collocated grids (e.g. Howell [8], Bell and Marcus [9], Rider *et al.* [10] and Pilliod and Puckett [11]). In References [8] and [9] advection of density proceeds using high-order, flux-limited advection techniques without any special attempt to maintain discontinuous interfaces. In References [10] and [11] the same basic methodology is followed, except that a second-order piecewise-linear interface construction (PLIC) technique is used to ensure that very sharp interfaces are maintained throughout the computation.

In contrast to collocated grids, staggered grids do not lend themselves as easily to Godunov-type schemes because the number of characteristic interpolations required increases significantly, especially in three dimensions (although it is certainly possible to implement these schemes on staggered grids, see Tau [12]). Staggered grids, however, have some advantages over collocated grids. Importantly, the discrete pressures remain coupled and the chequerboard pressure patterns typical of collocated grids do not arise. Additionally, the use of multigrid algorithms for the pressure solution remain straightforward, unlike those discussed in Reference [8].

In this paper, a technique is presented for the simulation of multifluid flows with large density variation on staggered grids. In principle, the method conserves both mass and momentum, uses a fully explicit in time update and uses MAC-type momentum advection techniques.

2. EQUATIONS OF MOTION AND NUMERICAL ALGORITHM

The scaled equations of motion are

$$\frac{\partial C}{\partial t} + \nabla \cdot (\boldsymbol{U}C) = 0, \tag{1}$$

$$\rho = \rho_1 C + \rho_2 (1 - C), \tag{2}$$

$$\frac{\partial \rho \boldsymbol{U}}{\partial t} + \nabla \cdot (\rho \boldsymbol{U}\boldsymbol{U}) = -\nabla P + \frac{1}{Fr^2}\rho \hat{\boldsymbol{g}} + \frac{1}{We}\boldsymbol{F_S} + \frac{1}{Re}\nabla \cdot \tau, \tag{3}$$

$$\nabla \cdot \boldsymbol{U} = 0, \tag{4}$$

where C is a fractional volume function, $\hat{\boldsymbol{g}}$ is a unit vector pointing in the direction of gravity and $\boldsymbol{F_S}$ is the surface force arising from interfacial effects. The fractional volume function C is advected with the local velocity $\boldsymbol{U}$. The density (and other scalar quantities such as viscosity) are obtained as shown in Equation (2). When simulating N fluids ($N > 2$), Equations (1) and (2) are extended in the obvious manner to include $N-1$ fractional volume functions (C_k, $k = 1, N-1$). For simplicity, only the case of two fluids will be considered below.

The dimensionless parameters in Equations (1)–(4) are the Froude, Weber and Reynolds numbers respectively:

$$Fr = \frac{\mathscr{U}}{(g\mathscr{L})^{1/2}}, \qquad We = \frac{\rho_0 \mathscr{U}^2 \mathscr{L}}{\sigma}, \qquad Re = \frac{\rho_0 \mathscr{U} \mathscr{L}}{\mu_0}, \tag{5}$$

where $\mathscr{U}$ and $\mathscr{L}$ are suitably chosen velocity and length scales, ρ_0 and μ_0 are density and viscosity scales for the major component fluid, σ is the coefficient of surface tension between the major and minor component fluids and g is the acceleration due to gravity.

2.1. Time integration

The second-order algorithm used for time integration of Equations (1)–(4) is most easily explained by describing a simpler first-order algorithm that is based on the projection algorithm first introduced by Chorin [13]:

1. Estimate values of C by solving

   $$C^{n+1} = C^n - \delta t \nabla \cdot (C^n \boldsymbol{U}^n),$$

   and estimate new $(n+1)$ densities and viscosities from

   $$\rho^{n+1} = \rho_1 C^{n+1} + \rho_2 (1 - C^{n+1}), \qquad \mu^{n+1} = \mu_1 C^{n+1} + \mu_2 (1 - C^{n+1}).$$

2. Find intermediate values of the velocity field by solving

   $$\rho^{n+1}\boldsymbol{U}^* = \rho^n \boldsymbol{U}^n + \delta t \left\{ -\mathscr{A}(C^n, \boldsymbol{U}^n) - \nabla P^n + \rho^n \hat{\boldsymbol{g}} + \frac{1}{We}\boldsymbol{F}_{\boldsymbol{S}}^n + \mathscr{D}(\mu^n, \boldsymbol{U}^n) \right\},$$

   where the symbols $\mathscr{A}$ and $\mathscr{D}$ refer to the advective and diffusive operators respectively (discussed below).
3. Calculate the pressure correction, δP, from the discrete form of the Helmholtz equation,

   $$\nabla \cdot \left(\frac{1}{\rho^{n+1}} \nabla \delta P \right) = \frac{1}{\delta t} \nabla \cdot \boldsymbol{U}^*,$$

4. Update new estimates of velocity and pressure,

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^* - \frac{\delta t}{\rho^{n+1}} \nabla \delta P,$$

$$P^{n+1} = P^n + \delta P.$$

The second-order improved Euler time stepping scheme involves repetition of the steps above, once with a half time step and once with a full time step, but with the source terms on the right-hand-side of each equation (except for C in Equation (1)) determined using half-time estimates of $\boldsymbol{U}$, ρ and P. It is not possible to use $C^{n+1/2}$ in the source term on the right-hand-side of the C advection equation because of the geometric nature of the flux calculation. However, because the Youngs' fluxes are estimates of the average values of the fractional volume fluxed over the edge of a cell during time δt, they may be considered time-centred. Thus, use of C^n in the source terms in Equation (1) in the second step does not reduce the order of the time integration.

There are a number of stability criterion that must be satisfied to ensure the stability of the numerical solution. They are

$$\delta t < \min(\delta x/U_{\max}, \delta y/V_{\max}), \quad \delta t < \frac{\delta x^2 \delta y^2 Re}{4(\delta x^2 + \delta y^2)}, \quad \delta t < \left(\frac{We}{\rho_g + \rho_f} \frac{\min(\delta x, \delta y)^3}{\pi} \right)^{1/2}. \tag{6}$$

Courant condition Diffusion time step Capillary time step

2.2. Advection of the colour function

The method used to advect C is the volume tracking method of Youngs [14] (see also Reference [15]). Unlike the method of Youngs [14], C is advected here on a computational grid that is twice as fine as the one used for momentum and pressure. Hence the numerical method is referred to as the FGVT (fine grid volume tracking) method. The $N_x \times N_y$ grid used for momentum and pressure solution is referred to as the standard grid, and the one used for C advection is the fine grid. The fine grid C is denoted c, and is defined as

$$\exists c_{i,j} \forall (i,j) \ni (i,j) \in \{1 \le i \le 2N_x, 1 \le j \le 2N_y\}.$$

A coarse-grid cell and its four fine-grid cells and the cell indexing are shown in Figure 1. Use of a fine grid for C is essential in ensuring that mass and momentum advection can be made consistent, and that errors in the discrete momentum, $(\rho \boldsymbol{U})^{n+1}$, do not give rise to large errors in the velocity when momentum is divided by density.

To ensure total mass conservation, a velocity field $\boldsymbol{u} = (u, v)$ is required on the fine grid which must satisfy a discrete form of the divergence condition (Equation (4)) on this grid. For the x-component of the fine-grid velocity u, piecewise linear interpolation of the standard-grid velocity U in the x-direction and piecewise constant interpolation in the y-direction (and vice versa for the y-direction velocity v) yields a suitable velocity field:

$$\begin{cases} u_{2i+1/2,2j} = U_{i+1/2,j} \\ u_{2i+1/2,2j-1} = U_{i+1/2,j} \\ u_{2i-1/2,2j} = 0.5(U_{i+1/2,j} + U_{i-1/2,j}) \\ u_{2i-1/2,2j-1} = 0.5(U_{i+1/2,j} + U_{i-1/2,j}) \end{cases} \quad \begin{cases} v_{2i,2j+1/2} = V_{i,j+1/2} \\ v_{2i-1,2j+1/2} = V_{i,j+1/2} \\ v_{2i,2j-1/2} = 0.5(V_{i,j+1/2} + V_{i,j-1/2}) \\ v_{2i-1,2j-1/2} = 0.5(V_{i,j+1/2} + V_{i,j-1/2}) \end{cases} . \tag{7}$$

The zeroth-order interpolation defined by Equation (7) is necessary to ensure mass conservation on the fine grid, however it does not affect the overall spatial order of the scheme. (The reasons for using c and $\boldsymbol{u}$ are discussed in detail in Sections 2.3 and 2.4.) The algorithm used here to advect c with $\boldsymbol{u}$ is based on that of Youngs [14]. It is described in detail in Reference [15] and has been shown by Pilliod and Puckett [11] to be only first-order-accurate in space. The second-order-accurate algorithm discussed in Reference [11] has not been implemented here.

On the standard grid, cell-centred values of C are required as well as cell-centred and cell-edged values of the density. Standard grid values of C are given by

$$C_{i,j} = 0.25(c_{2i,2j} + c_{2i-1,2j} + c_{2i,2j-1} + c_{2i-1,2j-1}),$$

$$C_{i+1/2,j} = 0.25(c_{2i+1,2j} + c_{2i,2j} + c_{2i+1,2j-1} + c_{2i,2j-1}),$$

$$C_{i,j+1/2} = 0.25(c_{2i,2j+1} + c_{2i,2j} + c_{2i-1,2j+1} + c_{2i-1,2j}). \tag{8}$$

The standard grid values of density are given by

$$\rho_{i,j} = C_{i,j}\rho_1 + (1 - C_{i,j})\rho_2, \qquad \rho_{i+1/2,j} = C_{i+1/2,j}\rho_1 + (1 - C_{i+1/2,j})\rho_2,$$

$$\rho_{i,j+1/2} = C_{i,j+1/2}\rho_1 + (1 - C_{i,j+1/2})\rho_2. \tag{9}$$

This formulation ensures that cell-edged densities are a more accurate representation of the mean density in the x- and y-momentum control volumes and allow good estimates to be made of the average density of fluid crossing the edge of each momentum control volume. They are also an essential part of ensuring accuracy and monotonicity of the solution and enabling strict conservation of momentum to be achieved.

Comparison of advection errors between C-advection on the fine grid and advection on a standard grid indicates that the fine grid advection used here results in global errors that are always less than those arising from use of the standard grid. An example is provided in Figure 2, which shows advection of a circular fluid region of radius $\pi/5$ with the velocity field

$$U(x, y) = \cos(x)\sin(y),$$

$$V(x, y) = -\sin(x)\cos(y),$$

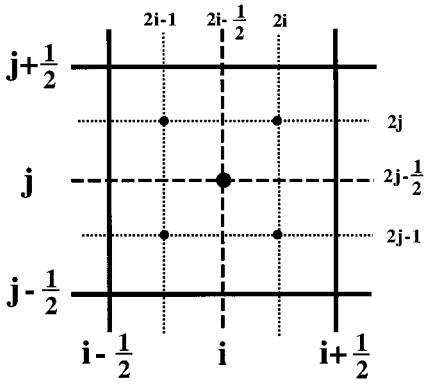

Figure 1. The coarse-grid cell (i, j) is shown with solid lines, and its four fine-grid cells are shown with the dashed line. The dotted lines represent the fine-grid cell centres and show the fine-grid cell indices.

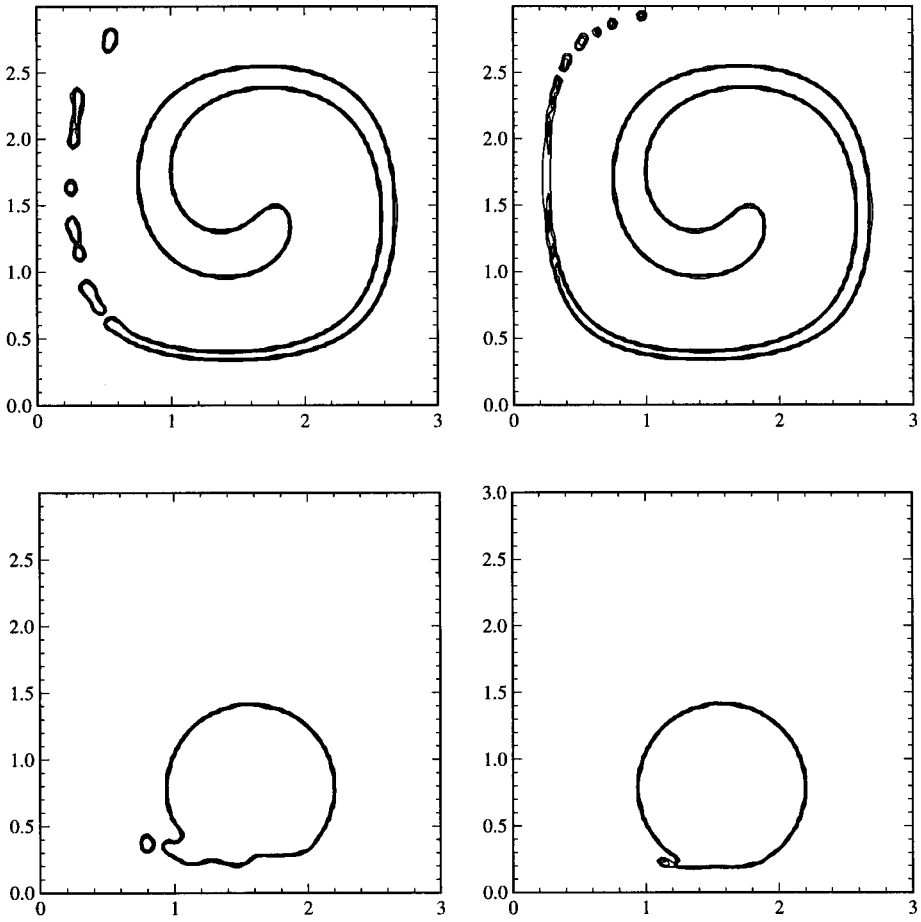

Figure 2. Comparison of the standard Youngs' method (left) with the fine-grid method discussed here (right). Advection of a circular region 2500 steps forward (top) and 2500 steps back (bottom). The L_1 error in the FGVT method is approximately four times less in this example.

on a domain with $x, y \in [0, \pi]$. The standard grid is 100×100 and the simulation is integrated forward in time for 2500 steps with a Courant number of 0.25 (top figures) before reversing the sign of the velocity field and integrating an additional 2500 steps (lower figures). A perfect advection scheme would return the initial C configuration. The percentage L_1 error in the returned solution is 7.5% for volume tracking on the standard grid and 1.8% for the method described here on the fine grid. In general, the ratio of errors between the two methods is not constant and depends on the geometric complexity of the velocity field and fluid region being advected.

The use of the fine grid everywhere in the domain is not essential to the FGVT method, and could be replaced by an adaptive grid refinement that occurred only in the vicinity of the interface. Adaptive grid refinement has not yet been implemented.

2.3. Momentum advection

Due to the difficulties associated with conserving momentum in the presence of large density jumps, flux limiting is necessary to ensure that oscillations do not develop near density

discontinuities. An additional consideration is that good estimates of flux densities must be made to ensure that momentum fluxes are consistent with mass fluxes.

Momentum advection is based on the fully multidimensional flux corrected transport algorithm of Zalesak [16] (ZFCT), with two significant differences:

1. The calculation of densities and momentum fluxes at control volume edges;
2. The min/max values used in the flux limiter.

An assumption is made in the following arguments, that although density (and hence momentum) may be discontinuous across an interface, the velocity field varies smoothly. Although this is not always true, it is reasonable in most circumstances. Because the velocity varies smoothly, traditional estimates of velocity at cell edges (upwind, upwind-biased or centred) will yield meaningful estimates of velocities at control volume edges, even in the presence of a density discontinuity. However, attempting to find high-order upwind or centred estimates of cell-edge momentum near a density discontinuity fails because the Taylor expansion of density is not valid in the neighbourhood of the discontinuity. Use of standard differencing techniques will yield unstable solutions and even ZFCT with momentum as the fluxed variable was found to be inadequate—interpolation and averaging errors accumulated and rapidly destroyed the solution.

2.3.1. Density at momentum control volume edges. Calculation of the average density of fluid crossing each edge of a momentum control volume (the momentum-flux densities) is most easily explained using the following example. Consider advection of x-momentum, $(\rho U)_{i+1/2,j}$. Values of the density are required at the x-edges of the momentum control volume $(i+1, j)$, (i, j) and the y-edges $(i+1/2, j+1/2)$, $(i+1/2, j-1/2)$. Similar quantities are needed at edges of the y-momentum control volume centred on $(i, j+1/2)$. In a standard MAC algorithm, these densities are obtained using linear or bilinear averages of nearby C-values and Equation (2). In FGVT, an estimate of the average density of the fluid crossing each side of a momentum control volume can be made using information obtained from the advection of c on the fine grid. Consider the x-momentum control volume centred on $(i+1/2, j)$ and shown in Figure 3.

The average density of the fluid passing through the x-edge at $(i+1, j)$ can be found by first determining the mean void fraction of the fluid $(\tilde{C}_{i+1,j})$ crossing this side in a time step

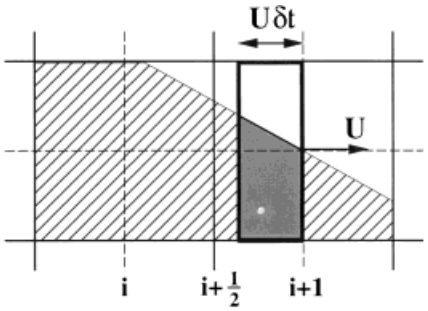

Figure 3. Estimation of the momentum-flux density at the $(i+1, j)$ edge of the $(i+1/2, j)$ momentum control volume is based on the average density crossing the $(i+1, j)$ edge in time δt. This average density is calculated from the sum of the fluid 1 fluxes across the right-hand edges of the fine-grid cells $(2i+1, 2j)$ and $(2i+1, 2j-1)$ (area of solid colour) divided by the total volume fluxed (area enclosed in the heavy rectangle). Similar estimates are made for the other three sides of the x-momentum control volume, and each side of the y-momentum control volume.

δt. The value of $\tilde{C}_{i+1,j}$ is equal to the flux of fluid 1 across the right faces of the fine-grid cells $(2i+1, 2j)$ and $(2i+1, 2j-1)$ (the area of solid colour in Figure 3) divided by the total volume fluxed through these two edges, $0.5(U_{i+1/2,j} + U_{i+3/2,j})\,\delta t\,\delta y$ (the area enclosed in the heavy rectangle in Figure 3). The flux of fluid 1 across each side of a fine-grid cell is a quantity calculated in the volume tracking routine in order to update c, and is therefore available to estimate momentum-flux densities. Once $\tilde{C}_{i+1,j}$ is found, the momentum-flux density is given by

$$\tilde{\rho}_{i+1,j} = \tilde{C}_{i+1,j}\rho_1 + (1 - \tilde{C}_{i+1,j})\rho_2.$$

Similar calculations are made to determine momentum-flux densities at (i, j), $(i+1/2, j+1/2)$ and $(i+1/2, j-1/2)$, and at the four sides of a y-momentum control volume.

The momentum-flux densities are representative of the average density of the fluid fluxed across a momentum control volume edge. When coupled with a good estimate of the (smooth) velocity of the fluid being fluxed across the control volume edge, a more accurate estimate can be made of the momentum flux than that obtained using weighted or harmonic averages of ρ. It is far more difficult to obtain estimates of the momentum-flux densities when the grid used for C-advection is the same as the momentum-pressure grid: this is the reason the void fraction grid is twice as fine.

2.3.2. Low- and high-order momentum-flux calculation. The basic scheme for ZFCT is illustrated here using a one-dimensional example for the advection of a cell centred quantity, Q_i. First, an estimate of the new time Q is made using a low-order, monotonic advection scheme,

$$Q'_i = Q^n_i - (F^L_{i+1/2} - F^L_{i-1/2}), \tag{11}$$

where F^L is the low-order flux which, for first-order upwinding (FOU), is

$$F^L_{i+1/2} = \begin{cases} U_{i+1/2}\delta t Q_i & \text{if } U_{i+1/2} \geq 0, \\ U_{i+1/2}\delta t Q_{i-1} & \text{if } U_{i+1/2} < 0. \end{cases} \tag{12}$$

Next, an anti-diffusive flux is defined using a higher-order scheme that attempts to correct the numerical diffusion resulting from the low-order scheme. An initial estimate of the anti-diffusive flux ($F^A_{i+1/2}$) is given by the difference between the high- and low-order flux approximations,

$$F^A_{i+1/2} = F^H_{i+1/2} - F^L_{i+1/2}. \tag{13}$$

Application of the entire anti-diffusive flux, Equation (13), simply results in application of the high-order (often unstable) flux. Thus, correction factors $q_{i+1/2}$ that limit the anti-diffusive fluxes are introduced to ensure monotonicity and stability. The correction factors are calculated to ensure that no new extrema are introduced into the solution after application of the anti-diffusive fluxes. The min/max values allowed for a grid cell i are based on the values of Q^n and Q' in cell i and its two neighbours, $i-1$, $i+1$. The procedure used to limit the fluxes is described in detail by Zalesak [16]. In the final step of ZFCT, the corrected anti-diffusive fluxes are applied and new values of Q are estimated as

$$Q^{n+1}_i = Q'_1 - (q_{i+1/2}F^A_{i+1/2} - q_{i-1/2}F^A_{i-1/2}). \tag{14}$$

Low-order momentum fluxes in FGVT are calculated using the momentum-flux densities $\tilde{\rho}$ and an FOU estimate of velocity, e.g.

$$(\rho U)_{i,j} = \begin{cases} \tilde{\rho}_{i,j} U_{i-1/2,j} & \text{if } U_{i,j} \geq 0, \\ \tilde{\rho}_{i,j} U_{i+1/2,j} & \text{if } U_{i,j} < 0. \end{cases} \tag{15}$$

Momentum fluxes for other edges are estimated in a similar manner. Any velocities required at locations other than those at which they are defined are calculated using either linear or bilinear interpolation.

High-order momentum fluxes in FGVT use $\tilde{\rho}$ and third-order QUICK estimates of velocity (see Reference [17]), e.g.

$$(\rho U)_{i,j} = \frac{1}{6} \begin{cases} \tilde{\rho}_{i,j}(2U_{i+1/2,j} + 5U_{i-1/2,j} - U_{i-3/2,j}) & \text{if } U_{i,j} \geq 0, \\ \tilde{\rho}_{i,j}(2U_{i-1/2,j} + 5U_{i+1/2,j} - U_{i+3/2,j}) & \text{if } U_{i,j} < 0. \end{cases} \tag{16}$$

Other upwind biased or centred schemes may be used because the subsequent flux limiting will always ensure stability.

2.3.3. Flux limiting. Once low- and high-order fluxes have been estimated, anti-diffusive fluxes are calculated (Equation 13) and limited in a similar way to the fully multidimensional procedure of Zalesak [16]. The difference in the method used here is that the allowable min/max values of momentum for a cell (i, j) are not based on values of $(\rho U)^n$ and $(\rho U)'$ in the five-neighbourhood[1], but are based on the product of the new value of density $\rho^{n+1}_{i+1/2,j}$ and the min/max values of old time and upwind velocity estimates in the five-neighbourhood (e.g. $\rho^{n+1}_{i+1/2,j}$ times min/max values of U^n and U' in the five-neighbourhood of $(i+1/2, j)$). The procedure for calculating the limiting factors q is described in detail in Reference [16] and is not discussed further here in the interests of brevity. The flux limiting procedure here ensures that no new velocity extrema are introduced into the solution, whereas the original procedure in Reference [16] ensures no new extrema are introduced into the solution of the advected quantity (ρU). The process of flux limiting is involved, but it was found that a less stringent limiting procedure resulted in rapid instability in flows in which the density ratio varied by more than approximately 10.

2.4. Viscous stresses

In the general case, the viscous terms in Equation (3) must be written in stress divergence form to allow for variable viscosity. If the rate of strain tensor is defined as

$$S^{\alpha\beta} = \frac{1}{2}\left(\frac{\partial U_\alpha}{\partial x_\beta} + \frac{\partial U_\beta}{\partial x_\alpha}\right),$$

finite difference expressions for $\nabla \cdot \tau$ at cell edges are obtained in the usual (MAC) manner. Consider the example of the x-momentum equation at $i+1/2, j$:

$$\nabla \cdot \tau_{i+1/2,j} = \frac{1}{\delta x}(\mu_{i+1,j} S^{xx}_{i+1,j} - \mu_{i,j} S^{xx}_{i,j}) + \frac{1}{\delta y}(\mu_{i+1/2,j+1/2} S^{xy}_{i+1/2,j+1/2} - \mu_{i+1/2,j-1/2} S^{xy}_{i+1/2,j-1/2}). \tag{17}$$

Components of the rate of strain tensor $(S^{\alpha\beta})$ are calculated using centred differences. Viscosities at $(i+1, j)$, (i, j), $(i+1/2, j+1/2)$ and $(i+1/2,\ j-1/2)$ must also be defined. First, fine-grid cell-centred viscosities, $\mu^f_{k,l}$, are calculated for each fine-grid cell (k, l), using

[1] For example, the five-neighbourhood of $(i+1/2, j)$ is $(i-1/2, j)$, $(i+1/2, j)$, $(i+1/2, j-1)$, $(i+1/2, j+1)$ and $(i+3/2, j)$.

$$\mu^f_{k,l} = c_{k,l}\mu_1 + (1 - c_{k,l})\mu_2.$$

The $\mu^f_{k,l}$ are then used to define viscosities at both sides of each standard grid x- and y-momentum control volume, (termed '+' and '−' viscosities). For example, at $(i+1/2, j+1/2)$

$$\mu^- = \frac{1}{2}(\mu^f_{2i,2j} + \mu^f_{2i+1/2j}), \qquad \mu^+ = \frac{1}{2}(\mu^f_{2i,2j+1} + \mu^f_{2i+1,2j+1}).$$

The cell-edge viscosity at $(i+1/2, j+1/2)$ is then defined as the harmonic average of μ^+ and μ^-:

$$\mu_{i+1/2,\,j+1/2} = \frac{2\mu^+\mu^-}{\mu^+ + \mu^-}.$$

Averaging in this way is essential to ensure that fluid stresses do not give rise to excessive accelerations in less dense, less viscous fluid regions near interfaces.

2.5. Surface force calculation

Forces resulting from interfacial tension are written as a surface volume force (which is a force per unit volume) as introduced in the CSF method of Brackbill *et al.* [18]. The surface volume force $\boldsymbol{F_S}$ is written

$$\boldsymbol{F_S} = \frac{1}{We}\kappa\delta(\boldsymbol{r}_\mathrm{I})\hat{\boldsymbol{n}}, \tag{18}$$

where We is the Weber number (Equation (5)), κ is the radius of curvature of the interface at $\boldsymbol{r}_\mathrm{I}$, $\delta(\boldsymbol{r}_\mathrm{I})$ is a one-dimensional delta function that is zero everywhere except at the interface, and $\hat{\boldsymbol{n}}$ is the unit inward-pointing normal to the interface.

The unknowns required for the surface volume force are κ, the surface unit normal $\hat{\boldsymbol{n}}$ and the surface delta function, $\delta(\boldsymbol{r}_\mathrm{I})$. As in the high-order kernel method of Aleinov and Puckett [19], convolution with a kernel (and its derivatives) is used to define normals and curvature. Although the eight-order used in Reference [19] is smooth and analytically differentiable many times, it is highly peaked, having a radial dependence like $(1 - r^2)^9$ for small r. Thus, if the smoothing length is small (e.g. twice the grid spacing) nearly all of the information for smoothing comes from the central point ($r = 0$) and virtually none from neighbouring points. In order to obtain a smooth convolved field, the smoothing length needs to be large.

In contrast, lower-order kernels such as those widely used in the smoothed particle hydrodynamics (SPH) method of Monaghan [20] have been shown to have good smoothing properties for smoothing lengths of twice the particle spacing (equivalent to twice the grid spacing on a uniform grid). The cubic B-spline [20] which has continuous first and second derivatives is used for smoothing in FGVT, and is given by

$$K(r,h) = \frac{1}{h^2}\begin{cases} \dfrac{40}{7\pi}\left(1 - 6\left(\dfrac{r}{h}\right)^2 + 6\left(\dfrac{r}{h}\right)^3\right) & \text{if } \dfrac{r}{h} < \dfrac{1}{2} \\ \dfrac{80}{7\pi}\left(1 - \dfrac{r}{h}\right)^3 & \text{if } \dfrac{r}{h} < 1 \\ 0 & \text{otherwise} \end{cases}. \tag{19}$$

Given the kernel K and smoothing length h, a smoothed C field $\tilde{C}$ is defined by

$$\tilde{C}_{i,j} = \sum_{k,l} C_{k,l} K(|\boldsymbol{r}_{i,j} - \boldsymbol{r}_{k,l}|, h)\, \delta x\, \delta y = K^* C,$$

where the kernel has been suitably normalised

$$\sum_{k,l} K(|\boldsymbol{r}_{i,j} - \boldsymbol{r}_{k,l}|, h)\, \delta x\, \delta y = 1 \qquad \forall i, j. \tag{20}$$

(Note that all surface force calculation is done on the standard C grid, not the fine c grid.) Defining the interface normal by

$$\boldsymbol{n} = (K_x^* C, K_y^* C),$$

where $K_x = \partial K / \partial x$, the unit normals and interface delta function are found from

$$\hat{\boldsymbol{n}} = \boldsymbol{n}/|\boldsymbol{n}|, \qquad \delta(\boldsymbol{r}_\mathrm{I}) = |\boldsymbol{n}|.$$

Curvatures are defined using (see Reference [18])

$$\kappa = -\nabla \cdot \hat{\boldsymbol{n}}.$$

Instead of convolving C with second derivatives of the kernel as done in Reference [19], the curvature is written in terms of $\boldsymbol{n}$,

$$\kappa = \frac{1}{|\boldsymbol{n}|}\left[\frac{n_x}{|\boldsymbol{n}|}\left(\frac{\partial |\boldsymbol{n}|}{\partial x}\right) + \frac{n_y}{|\boldsymbol{n}|}\left(\frac{\partial |\boldsymbol{n}|}{\partial y}\right) - \left(\frac{\partial n_x}{\partial x} + \frac{\partial n_y}{\partial y}\right)\right]. \tag{21}$$

Each of the derivative terms in Equation (21) is then estimated by convolving the normal components n_x, n_y and magnitudes $|\boldsymbol{n}|$ with derivatives of the smoothing kernel. Thus, C is smoothed twice to obtain estimates of the curvature. Although the effective stencil used to estimate κ is quite large, surface forces are restricted to cells in which the surface normal is non-zero. Because the surface normals are determined from data that has been smoothed only once, the thickness of the region over which surface forces are non-zero is not increased by the double smoothing.

The presence of 'parasitic' surface currents [21] that result from errors in the surface force calculation are still a problem with this implementation of surface tension forces, although preliminary studies suggest they are of smaller magnitude than the original CSF method. An example is shown in Figure 4, which compares the surface currents arising from the standard CSF method and the method discussed here for a static two-dimensional drop on a 32×32 grid. In this example $We = 10$, the drop density is 100 times that of the surrounding fluid and a smoothing length of $2.5\delta x$ is used. Before calculating surface normals and curvatures in the CSF method here, the C field was smoothed with the same kernel and smoothing length as those used in the FGVT method.

The true solution is a zero velocity field everywhere, but the generation of spurious surface currents with both methods can be seen in Figure 4. The growth in total kinetic energy in the domain as a function of time is shown in Figure 5, which further highlights the difference between the CSF and FGVT models. The approach described recently in Reference [22] to ensure an energy consistent surface tension model may alleviate this problem (for both methods).

2.6. Pressure correction solution

The pressure correction equation is

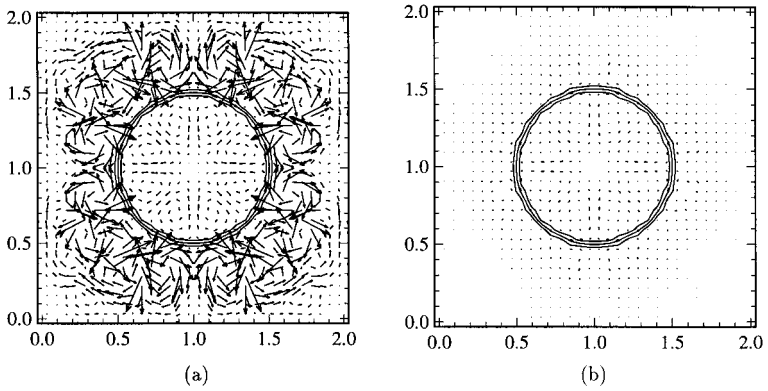

Figure 4. Parasitic surface currents arising from (a) the CSF method, and (b) the FGVT method at $t = 10$ for $We = 10$, a density ratio of 100:1 and a smoothing length of $2.5\delta x$.

$$\nabla \cdot \left(\frac{1}{\rho} \nabla \delta P \right) = \frac{1}{\delta t} \nabla \cdot \boldsymbol{U}^*. \tag{22}$$

Equation (22) is solved using a defect-correction multigrid method based on that described by Wesseling [23]. Defining the standard grid as level M and the coarsest grid as level 1, at each multigrid level $L (1 \leq L \leq M)$, Equation (22) is written in operator form as

$$[\boldsymbol{A}]^L \phi^L = S^L, \tag{23}$$

where ϕ^L is the level L defect correction. Note that at level M we solve for δP and not ϕ^M. In general, after a number of iterations at level L, the approximate solution $\tilde{\phi}^L$ will not satisfy Equation (23). The error in the solution is $(\phi^L - \tilde{\phi}^L)$, which satisfies

$$[\boldsymbol{A}]^L (\phi^L - \tilde{\phi}^L) = S^L - [\boldsymbol{A}]^L \tilde{\phi}^L. \tag{24}$$

In the restriction step, Equation (24) is interpolated or 'restricted' to the $L - 1$ grid,

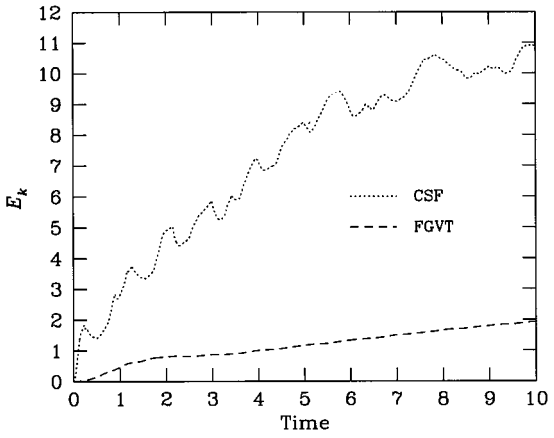

Figure 5. The growth of total kinetic energy versus time for the CSF and FGVT surface tension models for a static cylindrical drop.

$$[\boldsymbol{R}]^L[\boldsymbol{A}]^L(\phi^L - \tilde{\phi}^L) = S^{L-1}, \tag{25}$$

where $[\boldsymbol{R}]^L$ represents the operation of restriction from level L to level $L-1$. The source term in Equation (25) is

$$S^{L-1} = [\boldsymbol{R}]^L(S^L - [\boldsymbol{A}]^L\tilde{\phi}^L). \tag{26}$$

The level $L-1$ defect correction is $\phi^{L-1} = [\boldsymbol{R}]^L(\phi^L - \tilde{\phi}^L)$. In order to derive an appropriate coarse-grid operator $[\boldsymbol{A}]^{L-1}$, an approximation to $(\phi^L - \tilde{\phi}^L)$ at level L is found by interpolating or 'prolonging' ϕ^{L-1} to level L, i.e.

$$(\phi^L - \tilde{\phi}^L) \approx [\boldsymbol{P}]^{L-1}\phi^{L-1}, \tag{27}$$

where $[\boldsymbol{P}]^{L-1}$ is the operator that prolongs the level $L-1$ defect correction to level L. Substituting Equation (27) into (25) yields the equation for the level $L-1$ defect correction

$$[\boldsymbol{R}]^L[\boldsymbol{A}]^L[\boldsymbol{P}]^{L-1}\phi^{L-1} = S^{L-1}.$$

Thus, the level $L-1$ operator is defined as

$$[\boldsymbol{A}]^{L-1} = [\boldsymbol{R}]^L[\boldsymbol{A}]^L[\boldsymbol{P}]^{L-1}. \tag{28}$$

This is equivalent to the Galerkin coarse-grid operator discussed in Reference [23], although it is derived here in a less rigorous manner.

The prolongation and restriction operators must be chosen carefully for good performance of a multigrid method. Following Reference [23] for cell-centred multigrid with discontinuous coefficients, the prolongation operator used here is constant interpolation with a stencil $[\boldsymbol{P}]_S$, given by

$$[\boldsymbol{P}]_S = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

Thus, the level $L-1$ defect correction $\phi^{L-1}_{i,j}$ is prolonged to the four level L corrections $\phi^L_{2i,2j}$, $\phi^L_{2i-1,2j}$, $\phi^L_{2i,2j-1}$ and $\phi^L_{2i-1,2j-1}$, as

$$\phi^L_{2i,2j} = \phi^L_{2i,2j} + \phi^{L-1}_{i,j}, \qquad \phi^L_{2i-1,2j} = \phi^L_{2i-1,2j} + \phi^{L-1}_{i,j}, \text{ etc.}$$

Given $[\boldsymbol{P}]_S$, there are a number of possibilities for $[\boldsymbol{R}]$ and the choice used here is the one that gave the most rapid convergence. Its stencil $[\boldsymbol{R}]_S$ is written

$$[\boldsymbol{R}]_S = \frac{1}{8}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 5 & 5 & 1 \\ 1 & 5 & 5 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \tag{29}$$

For example, restriction of a quantity Q to level $L-1$ cell (i,j) is written

$$Q^{L-1}_{i,j} = \frac{1}{8}\left\{\begin{array}{l} Q^L_{2i-2,2j-2} + Q^L_{2i-1,2j-2} + Q^L_{2i,2j-2} + Q^L_{2i+1,2j-2} + \\ Q^L_{2i-2,2j-1} + 5Q^L_{2i-1,2j-1} + 5Q^L_{2i,2j-1} + Q^L_{2i+1,2j-1} + \\ Q^L_{2i-2,2j} + 5Q^L_{2i-1,2j} + 5Q^L_{2i,2j} + Q^L_{2i+1,2j} + \\ Q^L_{2i-2,2j+1} + Q^L_{2i-1,2j+1} + Q^L_{2i,2j+1} + Q^L_{2i+1,2j+1} \end{array}\right\}.$$

The restriction defined by Equation (29) is an average of $[\boldsymbol{P}]_{\mathrm{T}}$ and $[\boldsymbol{P}]_{\mathrm{L}}$, where $[\boldsymbol{P}]_{\mathrm{T}}$ is the adjoint operator of the prolongation corresponding to linear interpolation 'in triangles' (see Reference [23]) and $[\boldsymbol{P}]_{\mathrm{L}}$ is the adjoint operator of the prolongation corresponding to linear interpolation 'in lines' [23].

The fine-grid operator $[\boldsymbol{A}]^M$ is the usual five-point stencil arising from a centred discretisation of Equation (22) on a staggered grid. The coarse-grid operators $[\boldsymbol{A}]^L$, $1 \leq L \leq M-1$ are found from Equation (28) using an algorithm similar to that discussed in Reference [23]. For the restriction and prolongation operators used here, all coarse-grid operators are nine-point stencils, and do not grow to a greater size. Use of other restriction and prolongation operators, however, may lead to stencils that become progressively larger at coarser grid levels.

The appropriate boundary conditions for the pressure correction and the defect correction at each multigrid level, for all boundaries except periodic, are zero normal gradient. When forming the coarse-grid operator $[\boldsymbol{A}]^{L-1}$, careful attention must be paid to boundary conditions which must be implicitly applied in both the fine-grid operator $[\boldsymbol{A}]^L$ and the restriction operator $[\boldsymbol{R}]^L$. For example, at boundary cell $(1, j)$ reference to $\phi^L_{0,j}$ must be removed from $[\boldsymbol{A}]^L$ using $\phi^L_{0,j} = \phi^L_{1,j}$ and making the appropriate adjustments to $[\boldsymbol{A}]^L$. Equally important are the similar adjustments made to $[\boldsymbol{R}]^L$ in cells adjacent to boundaries. For example, the restriction stencils at boundary points $(1, j)$ and $(1, 1)$ are replaced with

$$[\boldsymbol{R}]_{S_{(1,j)}} = \frac{1}{8}\begin{bmatrix} 0 & 2 & 1 & 1 \\ 0 & 6 & 5 & 1 \\ 0 & 6 & 5 & 1 \\ 0 & 2 & 1 & 1 \end{bmatrix} \quad \text{and} \quad [\boldsymbol{R}]_{S_{(1,1)}} = \frac{1}{8}\begin{bmatrix} 0 & 2 & 1 & 1 \\ 0 & 6 & 5 & 1 \\ 0 & 8 & 6 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Implicitly applying boundary conditions in both $[\boldsymbol{A}]$ and $[\boldsymbol{R}]$ when forming the coarse-grid stencils leads to doubling of the convergence rate, compared with an implementation in which this is not done. This method of implicitly applying Neumann conditions is in contrast to Reference [23] in which it is suggested that zero Dirichlet conditions may be used successfully at all boundaries when determining coarse-grid operators.

The implementation here uses a V-cycle to restrict errors to coarser grids and prolong corrections to finer grids, with two red–black Gauss–Seidel iterations at each multigrid level except the coarsest (level 1) at which the solution is iterated to convergence. There is most likely some scope for improving the convergence of the method by using an adaptive algorithm instead of a V-cycle [23].

Convergence of the multigrid solver as a function of mesh size and density ratio is shown in Figure 6 for a simple Rayleigh–Taylor problem. In all cases shown in Figure 6, the simulation was run up until the time the dense plume had hit the floor of the computational domain and deflected across to the opposite wall. This corresponded to different numbers of time steps for the mesh refinement tests and different simulation times for the density ratio tests. The mean number of work units taken to reduce the L_2 error norm was determined as a function of the $\log_{10}$ of the residual. (Note that one work unit is equivalent to one fine-grid iteration.)

The results for a density ratio of 10:1 for different mesh sizes is shown in Figure 6(a). Ideally for a multigrid method, all three lines should be almost identical. In the example here, the average work for all three grids remains almost equal for reductions in the L_2 error norm up to five orders of magnitude. However there is a 20% increase in the average work required to reduce the error by ten orders of magnitude for a 64×192 grid when compared with the work

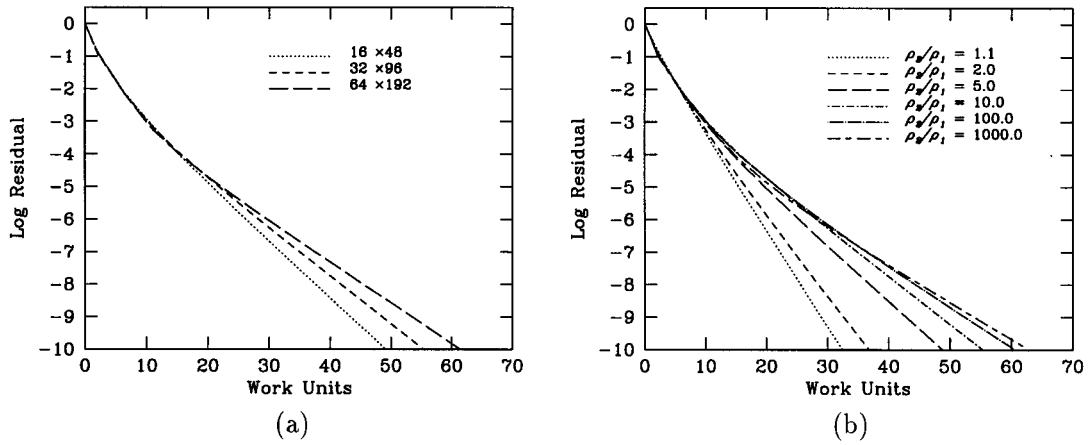

Figure 6. Average reduction in L_2 error norm as a function of work units (equivalent to one fine-grid iteration), (a) for different resolutions and a density ratio of 10:1, (b) for different density ratios and a mesh spacing of 32×96.

required for a 16×48 grid. It is believed that this result is due to the higher resolution simulations allowing the formation of smaller scale features. (Such small scale features do not tend to form with the coarser mesh simulations of the same problem because small-scale instability is damped.) As these small features are sheared by the flow, they become thinner (one or two mesh cells) and the pressure solver is observed to take a longer time to converge in these cases. This feature of the present solver is its only weakness and is believed to be due to the constant prolongation operator that tends to prolong excessively high corrections to thin, low-density regions. This point is revisited in the next section.

The nature of the relationship between density ratio and convergence is difficult to quantify, and depends on the problem being considered. However, the general result is that higher density ratios require more work to reduce the L_2 error norm by a fixed number of orders of magnitude. The results for a 32×96 mesh for different density ratios is shown in Figure 6(b). There is a clear dependence of convergence on density ratio, although the increase in work as a function of density ratio is not a simple logarithmic function. Importantly, the difference between density ratios of 1.1:1 and 1000:1 is only a factor of two.

2.7. Ad hoc adjustments

During the course of simulation, small bubbles (or droplets) may sometimes be detached from the rest of the gas (or liquid). Although these features are usually large enough to be resolved on the grid when they are formed, they are often stretched by the flow into thin filaments. When the dimension of these filaments approaches the grid spacing, they can no longer be resolved by the volume tracking or surface tension schemes. As discussed in the previous section, their presence is also detrimental to the convergence of the pressure-correction solver. As a result, they are deleted from the simulation despite the fact that this process does not conserve either mass or momentum. It is done only to ensure the continued progress of a calculation. Other alternatives to this ad hoc adjustment are being investigated.

The ad hoc adjustments sometimes required in FGVT raises the important issue of how to handle the problem of small-scale features in multifluid simulation. If a high-order flux-limited scheme were used to advect C (or c) such as in References [7] or [9], small features would

slowly diffuse away and probably cause few problems with stability or convergence. Although mass and momentum would be conserved in that case, the solutions are unlikely to be more representative of the true solution than if the small features were deleted. One way around this problem would be to use adaptive grid refinement to ensure that all features were at least a fixed number of grid cells in size. In practice, such an approach would almost certainly lead to very large numbers of small (and probably dynamically unimportant) features consuming significant computational resources.

3. APPLICATIONS

The motion of large gas bubbles plays a key role in many industrial processes and applications include gas sparging, pipeline hold-up due to slugs, boiling heat transfer under downward-facing heating surfaces and bubbling in electrolytic cells. A good understanding of the physics of bubble rise is essential, as bubble motion affects heat transfer in boiling processes and voltage drop in electrolytic applications. A number of applications are presented here that illustrate the applicability of the numerical method described above.

3.1. Bubble rise in an inclined channel

A demanding test case for a multifluid code is to simulate the rise of bubbles in an inclined channel. This flow has applications in boiling heat transfer beneath sloping heat exchangers, and in sloping anode Hall–Heroult cells used in the production of aluminium [24]. Experimental studies of the rise of gas bubbles underneath an inclined plane have been published by Maneri and Zuber [25], Che *et al*. [24] and Maxworthy [26]. The two-dimensional bubbles considered here may be thought of as a prototype for more realistic three-dimensional flows and more complex geometries such as those for bubble rise in inclined pipes, and are presented to demonstrate the applicability of the FGVT method.

In Reference [25], gas bubbles were released in a sloping channel sandwiched between two vertical glass plates and, to a good approximation, the resulting flow is two-dimensional. Bubble rise velocity was found to be dependent on bubble size, channel height and channel inclination. The experimental results of Maneri and Zuber [25] at 10° inclination are compared with numerical results of rise velocity in Figure 7, which shows dimensionless rise velocity ($U/\sqrt{g\mathscr{L}}$) versus equivalent bubble radius (the radius of the circle with equal cross-sectional area to the bubble). The agreement is seen to be good and, as observed experimentally, the bubble rise velocity decreases as the equivalent bubble radius decreases.

For an example of a 4° channel inclination and a non-dimensional bubble area of 0.5, velocity fields and bubble shapes (visualised by a volume fraction contour of 0.5) are shown in Figures 8 and 9. The parameter values are $Fr = 1$, $We = 100$ and $Re = 5\ 000$ and corresponds approximately to an air–water experiment with a channel height of 20 mm. The simulation was performed on a 384×96 grid with domain size 9×1. Figure 8 shows instantaneous velocity fields and bubble shapes for one instant in time. Clearly seen in Figure 8(b) are vortices that have been shed from the rising bubble. Interaction of these vortices with the interface ensures unsteadiness in the flow and sometimes leads to detachment of significant fractions of the bubble. The existence of trailing vortices has been inferred from (unreported) flow visualisation experiments in which bubble passage was associated with episodic entrainment of particles from a bed on the channel floor. Unsteadiness resulting from vortex shedding

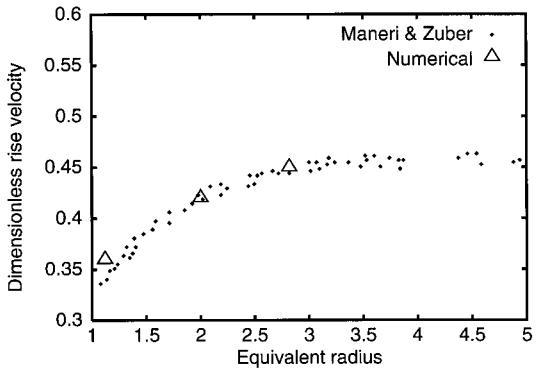

Figure 7. Dimensionless bubble rise velocity as a function of equivalent bubble radius (cm). Comparison between numerical results and the experimental results of Maneri and Zuber [25].

may be beneficial in cooling applications and may help to mix any thermal stratifications that arise in the system. In electrolytic applications it may also be beneficial, helping to mix the electrolyte and ensuring a uniform chemical composition. However, the vortex shedding may also be detrimental to the metal layer that would form on the lower cathodic surface in electrolytic applications. If metal were entrained into the electrolyte it may then back-react with dissolved gas or larger gas bubbles and would decrease the efficiency of the process.

The time-averaged velocity field and bubble shape (determined in the frame of reference moving with the leading edge of the bubble) are presented in Figure 9. The time-mean bubble profile can be split into several pieces: (i) a round leading edge that is well represented with a circular arc; (ii) an almost straight edge that is absent in small bubbles and increases in length once the bubble fills approximately half the channel width; and (iii) a tail that is approximately horizontal. The radius of curvature of the bubble leading edge increases with increasing bubble

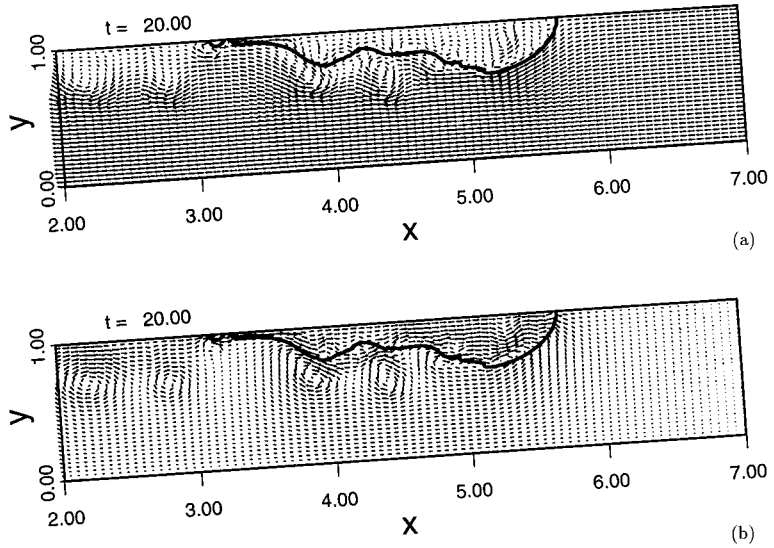

Figure 8. Instantaneous bubble profile and velocity vectors in (a) the co-ordinate frame moving with the leading edge of the bubble, and (b) the co-ordinate frame stationary with respect to the wall.

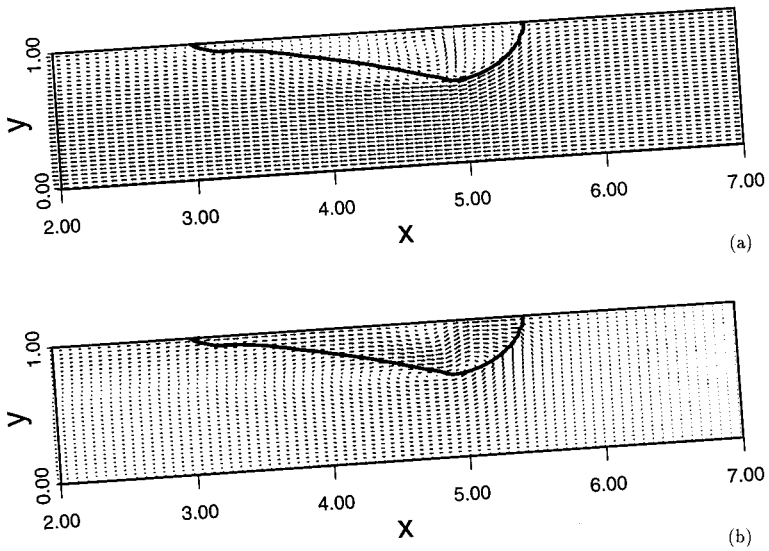

Figure 9. Time-averaged bubble profile and velocity vectors in (a) the co-ordinate frame moving with the leading edge of the bubble, and (b) the co-ordinate frame stationary with respect to the wall.

size. A fourth feature that is often present, especially for larger bubble sizes, is a small cusp that joins the circular nose to the flat tail. It is at this point that the flow generally separates from the bubble surface, leading to the formation of wake vortices and unsteadiness behind the bubble.

3.2. A bursting bubble

Although no experimental results are available for bursting of a two-dimensional bubble through a surface, application of the code to this flow demonstrates the ease with which fluid merging and break-up is handled. Results are shown in Figure 10 for a density ratio of 1000:1, $Fr = 1$, $We = 25$ and $Re = 500$. The computational grid is 128×384, the domain size is 3×9 and the initial bubble diameter is 1.0.

The initially circular bubble (Figure 10(a)) becomes crescent-shaped (Figure 10(b,c)) during the early stage of its rise, but soon develops the familiar cap-bubble shape (Figure 10(d,e)). This shape is quite unsteady, and oscillates significantly as the bubble rises. As the bubble approaches the flat surface (Figure 10(f)), it flattens and widens (Figure 10(g)). The thin film of liquid that forms above the bubble (Figure 10(h)) eventually thins to such a point that capillary instability leads to the formation of a number of liquid drops (Figure 10(i)). Finally, a broad, weak jet forms, and soon collapses. The dynamics in this case are influenced by the proximity of the boundaries as the bubble bursts.

3.2.1. Grid convergence. The simulation shown in Figure 10 was performed at resolutions of 64×192, 128×384, 192×576 and 256×768. The position of the top-most point of the bubble versus time is plotted for each resolution in Figure 11 (only until such time as the bubble bursts through the surface).

The results at each resolution are very close (and indistinguishable up until a time of $t = 5$). They indicate that even at the coarse resolution (64×192) the solution is sufficiently

well-resolved to predict bubble rise. The details of bubble-burst differ between the three simulations, primarily because the minimum film thickness above the bubble that can be resolved is limited by grid resolution. This is why the time at which each curve terminates and the curve shapes at termination differ slightly. Clearly, simulation of the mechanisms of film-draining and rupture cannot be captured by the method unless the resolution is sufficiently fine to resolve the film thickness. At present, this level of resolution is computationally difficult to achieve.

4. CONCLUSIONS

A conservative finite difference numerical method and second-order time integration algorithm has been presented that allows the simulation of immiscible multifluid problems to be

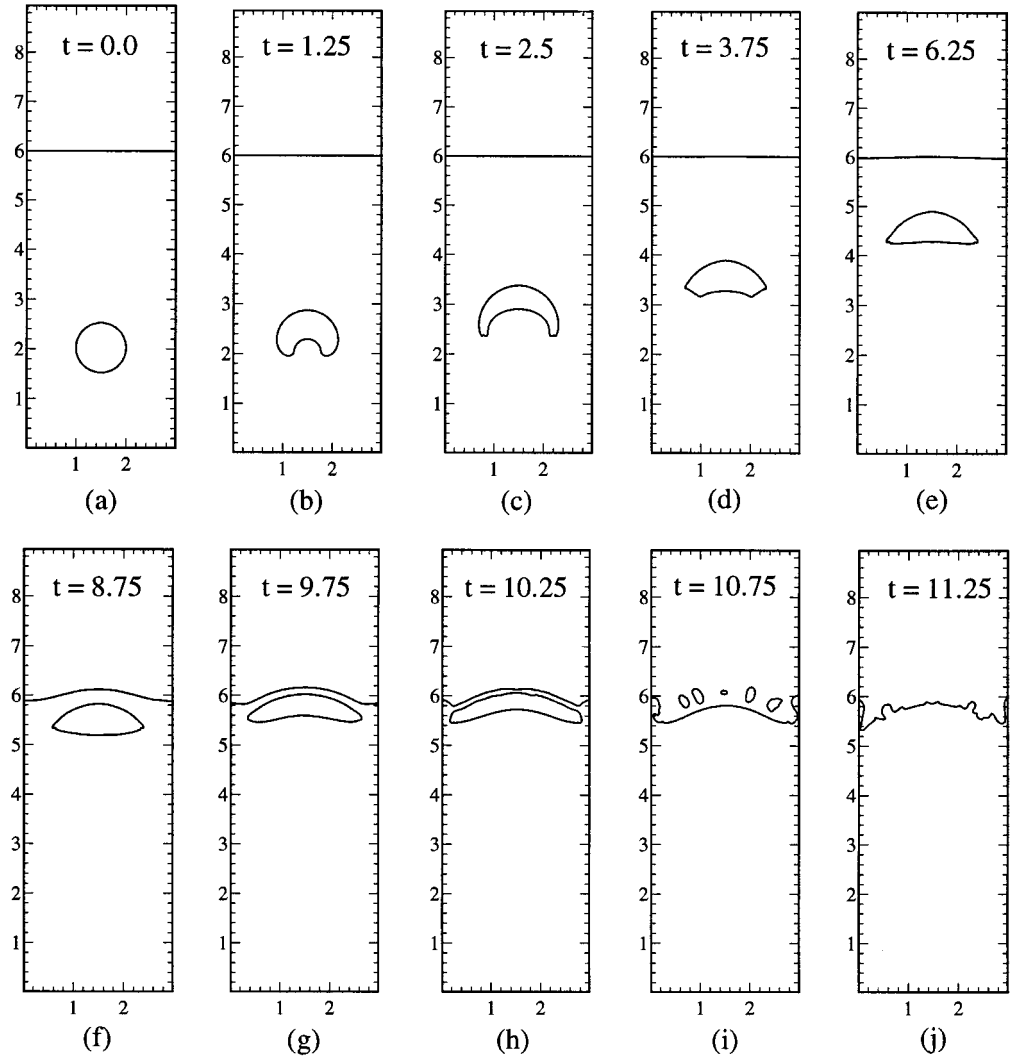

Figure 10. Rise of a 2D cylindrical bubble bursting through a free surface.

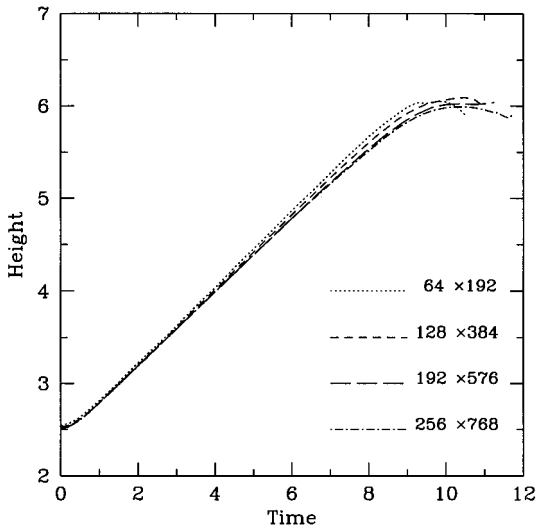

Figure 11. Comparison of bubble position versus time for different grid resolutions.

undertaken. The method is demonstrated with two flow problems: a two-dimensional bubble rising against an inclined plane, and a two-dimensional bubble bursting through an interface. The success of the method relies on four features:

1. A piecewise-linear interface reconstruction method for tracking volume fraction on a grid twice as fine as the pressure–velocity grid. Fine grid C advection allows both cell-edge and momentum-flux densities to be defined as accurately as the interface reconstruction technique allows. Accurate density information is important in ensuring consistency between mass and momentum fluxes on the staggered grid.
2. Momentum fluxes calculated from fine-grid mass fluxes and limited using a variant of the multidimensional FCT method of Zalesak [16]. The momentum-flux densities are crucial in ensuring stability, and flux limiting is performed on $\boldsymbol{U}$ (not momentum) to ensure monotonicity of the velocity field.
3. A surface tension algorithm that utilises the smoothness and compactness of low-order smoothing kernels to determine interface normals, curvatures and hence surface volume forces. This implementation allows different kernels and smoothing lengths to be utilised depending on the problem. Although spurious interface currents (as discussed in Reference [21]) are still present in the method, they grow more slowly than those arising when using the original CSF method [18].
4. A robust defect-correction multigrid solver for the pressure-correction equation that is based on the Galerkin coarse grid approximation of Wesseling [23] ensures convergence of the pressure-correction solution for arbitrary density ratios (up to 10000 have been undertaken).

The method is able to handle fluid merging and break-up in a natural way, and can handle density variations between fluid components of many orders of magnitude. The creation of fluid regions with one dimension the same scale as the grid spacing leads to convergence problems. The current approach of deleting these regions to ensure continued progress of the calculation is unsatisfactory, and needs further work. The most outstanding issue regarding

efficiency of the method is related to the capillary wave time step restriction that can easily be two (or more) orders of magnitude smaller than the Courant condition. It can lead to prohibitively long computational times. An implicit version of the code that is currently under investigation ameliorates the capillary wave time step restriction, however it worsens the effect of parasitic surface currents.

Finally, there is no reason why the finer grid used for C should be only twice as fine as the pressure-momentum grid. The same procedure could be used to make the C grid four or eight (or more) times as fine, allowing finer features in the scalar distribution to be resolved. This additional resolution would not be accompanied by an additional resolution in the momentum and pressure fields, thus the advantage of such a scheme would only apply in those situations where surface tension was unimportant and C was marking a passive scalar, not a dynamically important variable such as density. In applicable situations, the fine grid C would play a similar role to marker particles in the MAC scheme and would be advected with a velocity field of the same order as that used to advect MAC particles. One advantage over a particle representation would be an ability to estimate interfacial length, an important parameter in mixing applications. True fine grid resolution could be obtained by use of an adaptive grid refinement technique in the vicinity of the interface.

REFERENCES

1. J.E. Welch, F.H. Harlow, J.P. Shannon and B.J. Daly, 'The MAC method. A computing technique for solving viscous, incompressible, transient fluid-flow problems involving free surfaces', *LASL Report LA-3425*, Los Alamos, NM, 1965.
2. R.K.-C. Chan, R.L. Street and T. Strelkoff, 'Computer study of finite amplitude water waves', *Tech. Report 104*, Dept. Civil Engineering, Stanford University, CA, 1969.
3. B.D. Nichols and C.W. Hirt, 'Calculating three-dimensional free surface flows in the vicinity of submerged and exposed structures', *J. Comput. Phys.*, **12**, 234–246 (1973).
4. S.P. Wang and K.K. Wang, 'A net inflow method for incompressible viscous flow with moving free surfaces', *Int. J. Numer. Methods Fluids*, **18**, 669–694 (1994).
5. B.D. Nichols, C.W. Hirt and R.S. Hotchkiss, 'SOLA-VOF: A solution algorithm for transient fluid flow with multiple free boundaries', Los Alamos Scientific Laboratory Report, *LA-8355*, Los Alamos, NM, 1980.
6. S.O. Unverdi and G. Tryggvason, 'A front-tracking method for viscous, incompressible multi-fluid flow', *J. Comput. Phys.*, **100**, 25–37 (1992).
7. K.A. Pericleous, K.S. Chan and M. Cross, 'Free surface flow and heat transfer in cavities: The SEA algorithm', *Numer. Heat Transfer, Part B*, **27**, 487–507 (1995).
8. L.H. Howell, 'A multilevel adaptive projection method for unsteady incompressible flow', *6th Copper Mountain Conference on Multigrid Methods*, Copper Mountain, CO, April 4–9, 1993.
9. J.B. Bell and D.L. Marcus, 'A second-order projection method for variable density flows', *J. Comput. Phys.*, **101**, 334–348 (1992).
10. W.J. Rider, D.B. Kothe, S.J. Mosso, J.H. Cerutti and J.I. Hochstein, 'Accurate solution algorithms for incompressible multiphase flows', *AIAA Paper No. 95-0699*, 33rd Aerospace Sciences Meeting, Reno, NV, 1995.
11. J.E. Pilliod and E.G. Puckett, 'Second order volume-of-fluid interface tracking algorithms', Unpublished manuscript, to be submitted to J. *Comput. Phys.* (1996).
12. E.Y. Tau 'A second-order projection method for the incompressible Navier–Stokes equations in arbitrary domains', *J. Comput. Phys.*, **115**, 147–152 (1994).
13. A.J. Chorin, 'Numerical solution of the Navier–Stokes equations', *Math. Comput.*, **22**, 745–762 (1968).
14. D.L. Youngs, 'Time-dependent multi-material flow with large fluid distortion', in K.W. Morton and M.J. Baines (eds.), *Numerical Methods for Fluid Dynamics*, Academic Press, New York, 1982, pp. 273–285.
15. M. Rudman, 'Volume tracking methods for interfacial flow calculations', *Int. J. Numer. Methods Fluids*, **24**, 671–691 (1997).
16. S.T. Zalesak, 'Fully multi-dimensional flux corrected transport algorithms for fluid flow', *J. Comput. Phys.*, **31**, 335–362 (1979).
17. Y. Li and M. Rudman, 'Assessment of higher-order up-wind schemes incorporating FCT for convection dominated problems', *Numer. Heat Transfer, Part B*, **27**, 1–21 (1995).

18. J.U. Brackbill, D.B. Kothe and C. Zemach, 'A continuum method for modelling surface tension', *J. Comput. Phys.*, **100**, 335–354 (1992).
19. I. Aleinov and E.G. Puckett, 'Computing surface tension with high-order kernels', in K. Oshima (ed.), *Proc. 6th International Symposium on Computational Dynamics*, Lake Tahoe, CA, September 4–8, 1995.
20. J.J. Monaghan, 'Smoothed particle hydrodynamics', *Ann. Rev. Astrophys.*, **30**, 543–574 (1992).
21. B. LaFaurie, C. Nardone, R. Scardovelli and G. Zanetti, 'Modelling merging and fragmentation in multiphase flows with SURFER', *J. Comput. Phys.*, **113**, 134–147 (1994).
22. D. Jacqmin, 'A variational approach to deriving smeared-interface surface tension models', *Preprint* (1997).
23. P. Wesseling, *An Introduction to Multigrid Methods*, John Wiley and Sons, Chichester, UK, 1992.
24. D.F. Che, J.J.J. Chen and M.P. Taylor, 'Gas bubble formation and rise velocity beneath a downward facing inclined surface submerged in a liquid', *Trans. IChemE*, **69**, 25–29 (1991).
25. C.C. Maneri and N. Zuber, 'An experimental study of plane bubbles rising at inclination', *Int. J. Multiphase Flow*, **1**, 623–645 (1974).
26. T. Maxworthy, 'Bubble rise under an inclined plate', *J. Fluid Mech.*, **229**, 659–674 (1991).

