BRITE III project report : Algorithm Development – Multiphase Flow

G.Tabor & H.Rusche

Report # III-3 (Draft)

1 Introduction

At the end of the BRITE-II project the main deliverable was a mathematical formulation for 2-phase flow implemented in the commercial code STAR-CD. At the time problems with the code were recognised, relating to its convergence behaviour and stability. Subsequent work has highlighted this problem. One of the aims of our current work is to try to aleviate these problems. The function of this report is to try to pinpoint the proximate and ultimate causes of the problems, and to indicate the various possible solutions, both mathematical and numerical. Some of these possible solutions are already being tried, whilst others (more complex or time-consuming) may be tried if necessary later in the project.

The first section outlines the 2-phase flows that have been simulated using STAR-CD, and discusses the sorts of problems that have been encountered. This represents the sum of experiences from CD (Mark Liu), ICI (Archie Eaglesham) and Unilever (Jan Vaessen). A summary of the two-phase flow equations as derived via conditional averaging, and of the modelling used in the BRITE-II project, is included. The next section discusses proximate and ultimate causes for the problems, and outlines the different strategies for overcoming them. Then the various approaches currently under evaluation are discussed, in particular the superficial velocity approach (section 6) and the flux-velocity approach of Weller (section 7). The test bed for examining these algorithms has been the FOAM code [1]. Following the normal FOAM nomenclature, references to `briteIIFoam` are to the time-dependent formulation (i.e. PISO-based) formulation of the algorithm developed in the previous project, together with some later improvements made by Hill [2]. `superficialFoam` refers to the implementation of the superficial velocity approach, and `bubbleFoam` to the algorithm of Weller. Finally a summary of our findings and conclusions is given.

2 Use of STAR-CD

These comments on the code are synopses of communications from the researchers using the code at ICI and Unilever.

2.1 ICI – Dr. Eaglesham

The principal use of the code at ICI is for simulating gas-liquid stirred vessels. The numerics are highly unstable even for low dipersed phase fractions : it is very difficult to obtain converged solutions, and often cases that have been converging for hundreds or thousands of iterations suddenly diverge. The code is especially problematic where there are large gradients in the volume fraction anywhere in the system. This is the most severe problem with the code and effectively prevents it being used for real ICI business problems. STAR v.3.0 seems somewhat more stable than STAR v2.25 from this point of view (probably as a result of the introduction of Partial Elimination). In addition, ICI would like to be able to estimate the power consumption. In single phase cases this can

be ascertained very easilly by integrating the drag forces on the impeller. In gas-liquid systems the power decreases with increasing gas phase fraction, due to the gas collecting behind the blades. This streamlines the blades, reducing turbulence and therefore drag. The BRITE-II code completely fails to predict the form of this decrease and significantly overestimates the power for all non-zero values of gas phase fraction, although it does predict the gas cavities. This may be related to the streamlining, or to the turbulence modelling, or to the neglect of gas compressibility.

One further problem relates to the bubble size predictions. In the cases investigated turbulent inertial breakup is the dominant mechanism. The results from the BRITE-II code are approximately correct, but cannot be made mesh-independent, presumably because of the strong mesh-dependence of the turbulent energy dissipation predictions.

2.2 Unilever

Unilever's experience with the code broadly mirrors that of ICI. They have also investigated the S_γ formulation extensively, and have found that this exacerbates the instability of the algorithm.

2.3 CD – Dr. Liu

Work at CD by Dr. Liu, and information gathered by him from other users of the code, has identified the following issues within the code :

1. Explicit nature of the coupling of the continuity equations. The continuity equations are solved in two steps, as a pressure equation followed by a dispersed phase volume fraction equation. After the pressure correction and corresponding flux and velocity corrections, the sum of the phase volume fluxes can be conserved. However solving the dispersed phase volume fraction equation gives a new volume fraction which destroys conservation of the mass fluxes for the two phases. The introduction of these non-conservative fluxes into the momentum equations causes convergence problems.

2. The insensitivity of the mass residual based on the flux. In using the pressure equation to correct the cell surface flux, the system is unable to determine if the pressure correction is due to mass imbalances in one or both phases. In other words, even if one phase is already converged, the pressure correction can still drive it away from convergence because of the pressure correction produced due to the mass imbalance from the other phase.

3. Bounding of the phase fraction. Currently the volumetric phase fraction is bounded to be $\alpha < 1$ by adjusting the underrelaxation factors. If there is any region in the domain where $\alpha \sim 1$, this constraint can set the underrelaxation factor for the entire domain, causing the volume fraction system to cease to solve.

4. The algorithm is non-symmetric under exchange of phase. In many engineering applications the 'primary' phase may be different in different regions of the flow. A truly robust algorithm should give results irrespective of which phase is being considered as the primary phase, although this may be an unrealistic aim. In addition, the turbulence modelling is strongly biassed towards the continuous phase – in regions where the dispersed phase fraction is high (90% or higher in industrial applications) the underlying assumptions may not be appropriate. This last is more of a modelling issue, and so will be considered under another task (sub-task 1.3) but it is also relevant to the algorithm issue.

5. Asymptotic limit of the volume fraction at 0 and 1. A robust algorithm should approach the limits of the phase fraction in such a way that the single-phase result should be recovered.

6. For the BRITE-II benchmark case of sudden expansion in a pipe (air-water case) convergence can be achieved by heavilly underrelaxing the dispersed phase. The converged results have abnormal dispersed phase velocities close to the walls where the phase fraction is very low ($\mathcal{O}(10^{-4})$). With reference to this case, the maximum phase fraction is only $\alpha \sim 0.25$, and it can thus be considered to be a fairly simple case in industrial terms. An algorithm that cannot deal with this problem in an effective way can hardly be considered robust enough for industrial cases.

3 2-phase Equations

Here we outline the derivation of equations of motion for the two phases. The commonly used Conditional Averaging methodology is used [3], in which an indicator function is introduced which takes the value 1 in one phase and 0 in the other[4]. The Navier-Stokes equations (NSE) are multiplied by this function and then ensemble-averaged : the indicator function projects out one of the phases, and so the resulting equation is for that phase alone. The process introduces additional terms which can be written in terms of a surface average operation

$$\widehat{Q} = \frac{1}{\Sigma} \lim_{\delta V \to 0} \overline{\frac{1}{\delta V} \int_S Q dS} \tag{1}$$

which therefore represents the effect of the interface on the dynamics of the phase under consideration. These terms will commonly require modelling. Transport equations can also be formulated for the ensemble averaged indicator function, which has the interpretation of the probability of finding the phase at that point, and for quantities relating to the small-scale geometry of the interface.

The Navier-Stokes Equations for a compressible fluid are

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla.\rho\boldsymbol{v} &= 0 \\ \frac{\partial \rho\boldsymbol{v}}{\partial t} + \nabla.(\rho\boldsymbol{v}\otimes\boldsymbol{v}) &= -\nabla p + \nabla.\boldsymbol{S} \\ \frac{\partial \rho e}{\partial t} + \nabla.\rho e\boldsymbol{v} &= -p\nabla.\boldsymbol{v} + \boldsymbol{S}.\boldsymbol{D} + \nabla.\kappa\nabla e \end{aligned} \tag{2}$$

where

$$\boldsymbol{S} = \lambda\nabla.\boldsymbol{v}\boldsymbol{I} + 2\mu\boldsymbol{D}, \qquad \boldsymbol{D} = \frac{1}{2}(\nabla\boldsymbol{v} + \nabla\boldsymbol{v}^T) \tag{3}$$

3.1 Continuity Equation

The Mass Continuity equation becomes

$$\frac{\partial \alpha_k\overline{\rho_k}}{\partial t} + \nabla.(\alpha_k\overline{\rho_k\boldsymbol{u_k}}) = -\widetilde{\rho_{k_i}S_k}\Sigma \tag{4}$$

For flows with passive (i.e. non-reacting) interfaces, the surface propagation speed is zero, and hence

$$\frac{\partial \alpha_k\overline{\rho_k}}{\partial t} + \nabla.(\alpha_k\overline{\rho_k\boldsymbol{u_k}}) = 0 \tag{5}$$

3.2 Momentum equation

Conditionally averaging the momentum equation gives

$$\begin{aligned} \frac{\partial}{\partial t}\alpha_k\rho_k\overline{\boldsymbol{u}}_k + \nabla.(\alpha_k\rho_k\overline{\boldsymbol{u}}_k\otimes\overline{\boldsymbol{u}}_k) &= -\nabla\alpha_k\overline{p_k} + \nabla.\alpha_k(\overline{\tau_k} + \sigma_k^t) \\ &\quad +\alpha_k\rho_k\boldsymbol{g} + \widetilde{p_{k_i}\boldsymbol{n_k}}\Sigma - \widetilde{\boldsymbol{\tau_{k_i}}.\boldsymbol{n_k}}\Sigma \end{aligned} \tag{6}$$

Noting that $\widetilde{\boldsymbol{n_k}}\Sigma = \nabla\alpha_k$, we can decompose the interfacial pressure p_{k_i} into mean and fluctuating components

$$\widetilde{p_{k_i}\boldsymbol{n_k}}\Sigma = \widetilde{p_{k_i}}\Sigma\nabla\alpha_k + \widetilde{p''_{k_i}\boldsymbol{n_k}} \tag{7}$$

and so

$$\frac{\partial}{\partial t}\alpha_k\rho_k\overline{\boldsymbol{u}}_k + \nabla.(\alpha_k\rho_k\overline{\boldsymbol{u}}_k\otimes\overline{\boldsymbol{u}}_k) = -\nabla\alpha_k\overline{p_k} + \nabla.\alpha_k(\overline{\tau_k} + \sigma_k^t) + \alpha_k\rho_k\boldsymbol{g} + \boldsymbol{M}_k \tag{8}$$

where we have grouped the interfacial terms together as

$$\boldsymbol{M}_k = (\widetilde{p_{k_i}} - \overline{p_k})\nabla\alpha_k + \widetilde{p''_{k_i}\boldsymbol{n_k}}\Sigma - \widetilde{\boldsymbol{\tau_{k_i}}.\boldsymbol{n_k}}\Sigma \tag{9}$$

Neglecting surface tension forces for the moment we have $\boldsymbol{M}_c = -\boldsymbol{M}_d$. We can split $\boldsymbol{M}_d$ as follows :

$$\boldsymbol{M}_d = \alpha_k \rho_k \left[\boldsymbol{F}_d + \boldsymbol{F}_{vm} + \boldsymbol{F}_l + \boldsymbol{F}_b + \boldsymbol{F}_p\right] \tag{10}$$

representing the drag, virtual mass, lift, Basset and interfacial pressure forces. The principle terms are discussed below :

Drag force :

$$\boldsymbol{F}_d = \frac{1}{2}\rho_c A C_d |\overline{\boldsymbol{u}}_r| \overline{\boldsymbol{u}}_r \tag{11}$$

The drag coefficient here depends on the characteristics of the flow surrounding the dispersed phase element, and is a function of the droplet Reynolds number

$$\mathcal{R}e = \frac{\rho_d |\overline{\boldsymbol{u}}_r| d}{\mu_c} \tag{12}$$

It is the major coupling between the equations which is important for the algorithm development, and this comes into this term through the $|\overline{\boldsymbol{u}}_r|\overline{\boldsymbol{u}}_r$ term above, not through the coefficient. For this reason, a simple formulation for the coefficient has been used in the initial algorithm development and testing. However there are several different formulas available, appropriate for different regimes.

Virtual mass forces :

$$\vec{F}_{vm} = C_{vm}\alpha_d \rho_c \left(\frac{d_c}{dt}\overline{\boldsymbol{u}}_c - \frac{d_d}{dt}\overline{\boldsymbol{u}}_d\right) \tag{13}$$

where $\frac{d_k}{dt}\overline{\boldsymbol{u}}_k$ is the phasic material derivative. The time derivatives can be neglected here, giving

$$\vec{F}_{vm} = C_{vm}\alpha_d \rho_c \left(\overline{\boldsymbol{u}}_c . \nabla \overline{\boldsymbol{u}}_c - \overline{\boldsymbol{u}}_d . \nabla \overline{\boldsymbol{u}}_d\right) \tag{14}$$

Lift force :

$$\boldsymbol{F}_l = C_l \alpha_d \rho_c (\overline{\boldsymbol{u}}_c - \overline{\boldsymbol{u}}_d) \times \nabla \times \overline{\boldsymbol{u}}_c \tag{15}$$

with the value of C_l varying according to the exact situation. In the past the lift coefficient for bubbles has been altered in an *ad hoc* manner in order to bring computational results in line with experiment, with $-0.5 < C_l < 0.5$, and only the value $C_l = 0.5$ having rigorous justification. It seems likely that the variation is a result of the deformation of the bubbles altering their aerodynamic properties. However this is irrelevant for the numerical development.

Dispersal term :

There is also a term in the equations representing the difference in pressures between the bulk and the surface. This is

$$\boldsymbol{F}_p = (\widetilde{p_{k_i}} - \overline{p_k}) \nabla \alpha_k \tag{16}$$

which drives the motion of the dispersed phase (in addition to the form drag). In the BRITE-II project this was modelled as

$$\boldsymbol{F}_p = -0.37\rho_c C_d |\overline{\boldsymbol{u}}_r|^2 \nabla \alpha_k \tag{17}$$

As a model, this has a number of deficiencies. It introduces a term of the form

$$-0.37\rho_c C_d |\overline{\boldsymbol{u_r}}|^2 \overline{\boldsymbol{u_r}}.\nabla \alpha_c \tag{18}$$

in the turbulence k equation (see below), which can drive the system to negative values of k. Moreover, the term being modelled is a dispersive one, and should be related to the turbulence, not the mean relative velocity $\overline{\boldsymbol{u_r}}$. A better model for this term would be

$$\boldsymbol{F}_p = -0.37\rho_c C_d k \nabla \alpha_k \tag{19}$$

similar to that used by Lahey and Drew[5].

The Basset force is the history effect on the particle surface. It is given as

$$\boldsymbol{F}_b = \frac{3}{2} d^2 \sqrt{\pi \rho_c \mu_c} \int_{t_0}^{t} \frac{d}{dt}(|\boldsymbol{u}_d - \boldsymbol{u}_c|_{t=t'}) \frac{dt'}{\sqrt{t-t'}} \tag{20}$$

However this is usually negligable compared to other terms, and is neglected in this case.

Hence the form for $\boldsymbol{M}_k$ used is

$$\begin{aligned} \boldsymbol{M_k} \;=\; & \frac{3}{4}\frac{\alpha_d \rho_c C_d}{d} |\overline{\boldsymbol{u}}_r| \overline{\boldsymbol{u}}_r \\ & + C_{vm} \alpha_d \rho_c \left(\overline{\boldsymbol{u}}_c.\nabla \overline{\boldsymbol{u}}_c - \overline{\boldsymbol{u}}_d.\nabla \overline{\boldsymbol{u}}_d\right) \\ & + C_l \alpha_d \rho_c (\overline{\boldsymbol{u}}_c - \overline{\boldsymbol{u}}_d) \times \nabla \times \overline{\boldsymbol{u}}_c \\ & - 0.37\rho_c C_d |\overline{\boldsymbol{u}}_r|^2 \nabla \alpha_k \end{aligned} \tag{21}$$

3.3 Bulk Cencentration Transport Equation

The conservation equation for concentration of a solute C in phase k reads

$$\frac{\partial C_k}{\partial t} + \nabla.\boldsymbol{u}_k C_k = \nabla.D_k \nabla C_k + S_{ck} \tag{22}$$

Conditionally averaging this gives

$$\begin{aligned} \frac{\partial \alpha_k \overline{C_k}}{\partial t} + \nabla.\alpha_k \overline{\boldsymbol{u}}_k \overline{C_k} \;=\; & \nabla.\alpha_k D_k \nabla \overline{C_k} - \nabla.\alpha_k \overline{\boldsymbol{u_k'} C_k'} \\ & + \widetilde{\overline{C_{k_i}(\boldsymbol{u}_{k_i} - \boldsymbol{u}_i)}}\Sigma - \widetilde{\overline{D_k \nabla C_{k_i}}.\boldsymbol{n}_k}\Sigma \end{aligned} \tag{23}$$

3.4 Turbulence Modelling

Many of the cases of interest involve turbulent flows, for which turbulent modelling needs to be introduced. The details of this modelling is the subject of other tasks in the project, but the modelling can affect the behaviour of the algorithm and may contribute to the stability issues. Hence we use the $k-\epsilon$ and C_t modelling used in the BRITE-II project for this purpose. The two-phase $k-\epsilon$ equations for the continuous phase are

$$\begin{aligned}
\frac{\partial \rho_c k}{\partial t} + \nabla.\rho_c\alpha_c k\overline{\boldsymbol{u}}_c &= \nabla.\left[\alpha\left(\mu_c + \frac{\mu_c^t}{\sigma_k}\right)\nabla k\right] \\
&\quad +\alpha_c G - \rho_c\alpha_c\epsilon + 2A_D(C_t-1)k - A_D\frac{\mu_c}{\alpha_d\alpha_c\rho_c^2\sigma}\overline{\boldsymbol{u_r}}.\nabla\alpha_c \\
\frac{\partial \alpha_c\rho_c\epsilon_c}{\partial t} + \nabla.\alpha_c\rho_c\overline{\boldsymbol{u}}_c\epsilon &= \nabla.\left[\alpha_c\left((\mu_c + \frac{\mu_c^t}{\sigma_k}\right)\nabla\epsilon\right] + C_1\alpha_c\frac{\epsilon}{k}G - C_2\alpha_c\rho_c\frac{\epsilon^2}{k} + 2C_2A_D(C_t-1)\epsilon
\end{aligned}$$

with

$$G = \mu_c^t\left[\nabla\overline{\boldsymbol{u}}_c + \nabla\overline{\boldsymbol{u}}_c^T - \frac{2}{3}(\nabla.\overline{\boldsymbol{u}}_c)\boldsymbol{I}\right] : \nabla\overline{\boldsymbol{u}}_c - \frac{2}{3}\rho_c k\boldsymbol{I} : \nabla\overline{\boldsymbol{u}}_c \tag{24}$$

(c.f. [11]). These are analogous to the single-phase $k-\epsilon$ equations, with the addition of C_t terms and interface terms. C_t represents the ratio of continuous to dispersed phase turbulence and accounts for the dispersed phase turbulence. The other terms come from the interface effects and represent the additional effect of the dispersed phase on the continuous phase. These terms may cause added numerical problems and are switched off for many of the test cases discussed below.

4 Problems with the Equations

There are three main problems with multiphase flow calculations : bounding the phase fraction to remain between 0 and 1, the behaviour of the system of equations as one phase goes to zero, and the coupling between the various equations (principally via the force terms). Bounding α to remain between 0 and 1 is of extreme importance. Not only are phase fractions which are negative or above 1 unphysical, they may also produce numerical problems such as division by zero. There is also the requirement to ensure that the phase fractions sum to 1 under all conditions.

The second problem arises where $\alpha \to 0$ (for example, see point 6, section (2.3)). For instance, equation 8 reduces to the identity $0 = 0$ at this limit, which makes solving it for a finite value of $\boldsymbol{u}_k$ rather difficult. It is possible to reformulate this equation rather simply for the superficial velocity $\boldsymbol{U}_{sk} = \alpha_k\boldsymbol{u}_k$, which has the useful property that $\boldsymbol{U}_{sk} \to 0$ as $\alpha_k \to 0$. However in this case the flux that must be evaluated for the pressure equation is $\boldsymbol{U}_{sk}/\alpha_k = \boldsymbol{u}_k$, the limit of which is not known. Even if $\alpha = 0$ is being exluded from the solution domain, then problems still arise in numerically evaluating terms which involve division by α : they can only be evaluated where their correct analytical limit is known

beforehand. Some terms may limit to an easilly-evaluatable value, in which case this will be fairly straightforward. Terms which limit to quantities related to the phase velocities (as in the example given above) cause somewhat more problems. Since ultimately this is a quantity that we are trying to evaluate, further modelling will need to be introduced in order to evaluate this limit. It may be possible to find an arangement of the equations for which this is made easier. For instance a formulation which needs the relative velocity $\boldsymbol{u}_r$ as a limit would be preferable, since this could be represented as a force balance on the bubbles or droplets. In general the $\alpha \to 0$ limit represents the problem of specifying a value for a property of a phase which is non-existant at that point. If the property is extensive, this is not a problem (it will be zero). However intensive properties such as velocity do not have this feature and so cause problems.

Finally, the details of the coupling between the equations can cause problems. This is particularly true of the individual phase momentum equations, which are coupled by complex terms representing the interfacial forces between the phases. Of these terms the drag is dominant, and if the bubble size is small enough this may lead to complete coupling between the phase momenta, i.e. cocurrent flow in parts of the domain. Hence in gas-liquid systems where buoyancy is dominant, the algorithmic problem is one of evaluating small differences between large buoyancy and drag terms. In cases where buoyancy is not important (such as liquid-liquid systems) other terms are also present and potentially cause problems : in particular the lift term which involves curls and cross products and thus represents a coupling between all vector components. It seems unlikely that we will be able to state definitively which effects are dominant, or even if any are dominant globally over the whole domain, although experience may indicate that the coupling in some terms is more important than in others. In addition, current sequential solution techniques inevitably favour one phase over the other, which may not be appropriate.

These three issues represent the main problems with the two-phase algorithm, but there are some additional issues which may be problematic. A full derivation of the equations of motion provides 2 independent bulk pressures, one in each phase which move the individual phases around, and an interface pressure which moves the interface around. The assumption is usually made that a common average pressure can be used. Unfortunately this can lead to the equation set being ill-posed [6, 7], in which case two of the characteristics are complex. This will lead to instability in the numerical solution, which has to be dealt with is some way. It is possible to introduce regularisation terms into the solution scheme, probably as part of the interfacial modelling, which suppress this instability. In particular, a system with a single pressure driving both phases can potentially violate continuity : the pressure can drive both phases out of a cell, for instance, where one phase should be entering it in order to preserve continuity. This could be ameliorated by suitable bounding of the phase fraction, but it is likely that this is what is causing some of the convergence problems in the current formulation (see, for example, point 2, section (2.3)). Alternatively, a mixture formulation could be found in which equations for the fluid as a whole are being solved. In this case there will be no possibility of a continuity error occurring, but the error will show up elsewhere (probably in the phase fraction), hopefully in a form where it will be more tractable. The third

alternative is to introduce the full set of pressures. A formulation for such a case has been implemented by a group at Los Alamos [8, 9, 10]. This formulation (discussed below) would be worth investigating.

4.1 Possible Approaches to the problems

There are 3 approaches to overcoming the problems outlined above.

- Mathematical. The formulation as posed in STAR is based on a set of equations that are poorly behaved. One option is to reformulate the problem to remove this or to direct the errors into properties that are less crucial for the success of the code (for instance, producing a velocity which is unbounded is not a problem. Producing a phase fraction which is unbounded, is).

- Modelling. The current formulation in STAR relies heavilly on careful modelling to overcome the inadequacies of the equations. It is possible that improved modelling of for example dispersive or diverging terms in the code could produce the desired result.

- Numerics. It is also conceivable that implementing more robust numerical algorithms would be appropriate. For instance, the introduction of partial elimination between source terms in the two individual phase momentum equations improved the convergence behaviour of the current STAR-CD code.

In fact, some elements of all of these approaches will be necessary to construct a working scheme. The next section (5) of this report will discuss possible rearangements of the equations. We then outline the two main alternative algorithms that have been investigated, based on solving for a superficial velocity (section 6) and on an elimination of α from the momentum equation (section 7). Numerical comparisons between these two schemes are presented for specific test cases. The final section discusses future directions for the investigation (some of which have been implemented and tested, and are the subject of a further report), as well as alternative approaches, for example block solution of the equations.

5 Alternative Formulations of the Equations

As indicated, one avenue to explore is that of rearanging the equations to improve their mathematical behaviour. For instance, combining the momentum equations to solve for a mixture velocity and a relative velocity might have advantages, in that the mixture velocity is well defined at all points in space, and is directly related to the mean pressure in the system. Moreover, much of the interfacial coupling between the phases is governed by the relative velocity $\boldsymbol{u}_r$, which ties in nicely to the concept of solving a transport equation for this quantity. This section sets out these possible rearangements. The following section (section (6)) describes a representation for the equations based on solving for the

superficial velocity $\boldsymbol{U}_s = \alpha \boldsymbol{u}_a$ in each phase. This also improves the mathematical nature of the equations, since this quantity is well-defined at all points in space (in regions where $\alpha = 0$, i.e. that phase does not exist, $\boldsymbol{U}_s = 0$). Section 7 discusses a formulation using the opposite approach, removing the phase fraction from the equations as much as possible.

5.1 Continuity equations

We can add and subtract the two individual phase equations. If we define

$$\begin{aligned} \alpha &= \alpha_a \rho_a, & \beta &= \alpha_b \rho_b \\ \overline{\boldsymbol{u}}_+ &= \frac{\alpha \overline{\boldsymbol{u}}_a + \beta \overline{\boldsymbol{u}}_b}{\alpha + \beta}, & \overline{\boldsymbol{u}}_- &= \frac{\alpha \overline{\boldsymbol{u}}_a - \beta \overline{\boldsymbol{u}}_b}{\alpha - \beta} \end{aligned}$$

then the resulting equations are

$$\begin{aligned} \frac{\partial}{\partial t}(\alpha + \beta) + \nabla.((\alpha + \beta)\overline{\boldsymbol{u}}_+) &= 0 \\ \frac{\partial}{\partial t}(\alpha - \beta) + \nabla.((\alpha - \beta)\overline{\boldsymbol{u}}_-) &= 0. \end{aligned}$$

We can define $\rho_+ = \alpha + \beta$, $\rho_- = \alpha - \beta$. Inverting these equations gives

$$\begin{aligned} \overline{\boldsymbol{u}}_a &= \frac{1}{2\alpha}\left(\rho_+ \overline{\boldsymbol{u}}_+ + \rho_- \overline{\boldsymbol{u}}_-\right) \\ \overline{\boldsymbol{u}}_b &= \frac{1}{2\beta}\left(\rho_+ \overline{\boldsymbol{u}}_+ - \rho_- \overline{\boldsymbol{u}}_-\right) \\ \overline{\boldsymbol{u}}_a - \overline{\boldsymbol{u}}_b &= \frac{\alpha^2 - \beta^2}{2\alpha\beta}(\overline{\boldsymbol{u}}_- - \overline{\boldsymbol{u}}_+) \end{aligned}$$

We can also write the continuity equation in terms of $\overline{\boldsymbol{u}}_r$. Subtracting,

$$\frac{\partial}{\partial t}(\alpha - \beta) + \nabla.\alpha \overline{\boldsymbol{u}}_a - \beta \overline{\boldsymbol{u}}_b = 0 \tag{25}$$

We have

$$\begin{aligned} \overline{\boldsymbol{u}}_a &= \overline{\boldsymbol{u}}_+ + \frac{\beta}{\alpha + \beta}\overline{\boldsymbol{u}}_r \\ \overline{\boldsymbol{u}}_b &= \overline{\boldsymbol{u}}_+ - \frac{\alpha}{\alpha + \beta}\overline{\boldsymbol{u}}_r \end{aligned}$$

and so

$$\alpha \overline{\boldsymbol{u}}_a - \beta \overline{\boldsymbol{u}}_b = (\alpha - \beta)\overline{\boldsymbol{u}}_+ + \frac{2\alpha\beta}{\alpha + \beta}\overline{\boldsymbol{u}}_r \tag{26}$$

and so

$$\frac{\partial \rho_-}{\partial t} + \nabla.\rho_- \overline{\boldsymbol{u}}_+ + \nabla.\left[\frac{2\alpha\beta}{\alpha + \beta}\overline{\boldsymbol{u}}_r\right] = 0 \tag{27}$$

5.2 $\widetilde{\alpha}$ equation

If we start from the conditionally averaged continuity equation

$$\frac{\partial}{\partial t}\alpha_a\rho_a + \nabla.(\alpha_a\rho_a\overline{\boldsymbol{u}}_a) = 0 \tag{28}$$

we can define a density-weighted indicator variable $\widetilde{\alpha}$ as

$$\widetilde{\alpha} = \frac{\alpha_a\rho_a}{\rho_+} \tag{29}$$

so this equation becomes

$$\frac{\partial}{\partial t}\rho_+\widetilde{\alpha} + \nabla.\rho_+\widetilde{\alpha}\overline{\boldsymbol{u}}_a = 0 \tag{30}$$

We can write

$$\overline{\boldsymbol{u}}_a = \overline{\boldsymbol{u}}_+ + (1-\widetilde{\alpha})\overline{\boldsymbol{u}}_r \tag{31}$$

and thus

$$\frac{\partial}{\partial t}\rho_+\widetilde{\alpha} + \nabla.\rho_+\widetilde{\alpha}\overline{\boldsymbol{u}}_+ + \nabla.\rho_+\widetilde{\alpha}(1-\widetilde{\alpha})\overline{\boldsymbol{u}}_r = 0 \tag{32}$$

(Note that

$$\overline{\boldsymbol{u}}_b = \overline{\boldsymbol{u}}_+ - \widetilde{\alpha}\overline{\boldsymbol{u}}_r \quad) \tag{33}$$

The problem now reduces to one of finding models for $\overline{\boldsymbol{u}}_r$. To stabilise this numerically we need to introduce some diffusion term. There are two routes for this. One is to introduce an entirely ficticious gradient transport term of the form

$$\nabla.\mathcal{D}\nabla\widetilde{\alpha} \tag{34}$$

which can be added (implicitly) and subtracted (explicitly) from the equation : at convergence, this will have had no net effect. Alternatively we can split the relative velocity into diffusional and non-difusional contributions, and introduce the models for the diffusional parts into the $\widetilde{\alpha}$ equation.

We can introduce density weighted variables throughout. Denoting

$$\widetilde{q} = \frac{\rho_a q}{\rho_+} \tag{35}$$

we can express

$$\frac{\alpha\beta}{\alpha+\beta} = \rho_+\widetilde{\alpha}(1-\widetilde{\alpha}) \tag{36}$$

5.3 Combined Momentum equation

Adding the two momentum equations gives

$$\begin{aligned}\frac{\partial}{\partial t}(\alpha\overline{\boldsymbol{u}}_a+\beta\overline{\boldsymbol{u}}_b)+\nabla.\left[\alpha\overline{\boldsymbol{u}}_a\otimes\overline{\boldsymbol{u}}_a+\beta\overline{\boldsymbol{u}}_b\otimes\overline{\boldsymbol{u}}_b\right] &= -\nabla(\alpha_a\overline{p}_a+\alpha_b\overline{p}_b)\\ &\quad+\nabla.\left[\alpha_a(\boldsymbol{\tau}_a+\boldsymbol{\sigma}_a^t)+\alpha_b(\boldsymbol{\tau}_b+\boldsymbol{\sigma}_b^t)\right]\\ &\quad+(\alpha+\beta)\boldsymbol{g}+(\text{surface tension forces})\end{aligned}$$

since the interface terms cancel. The first term on the lhs is the time derivative of $\rho_+\overline{\boldsymbol{u}}_+$. The second term requires some more manipulation. Let us consider

$$\begin{aligned}(\alpha+\beta)(\overline{\boldsymbol{u}}_+\otimes\overline{\boldsymbol{u}}_+) &= \frac{1}{\alpha+\beta}\left[(\alpha\overline{\boldsymbol{u}}_a+\beta\overline{\boldsymbol{u}}_b)\otimes(\alpha\overline{\boldsymbol{u}}_a+\beta\overline{\boldsymbol{u}}_b)\right]\\ &= \frac{1}{\alpha+\beta}\left[\alpha^2\overline{\boldsymbol{u}}_a\otimes\overline{\boldsymbol{u}}_a+\beta^2\overline{\boldsymbol{u}}_b\otimes\overline{\boldsymbol{u}}_b+\alpha\beta\left[\overline{\boldsymbol{u}}_a\otimes\overline{\boldsymbol{u}}_b+\overline{\boldsymbol{u}}_b\otimes\overline{\boldsymbol{u}}_a\right]\right]\end{aligned}$$

On the other hand,

$$\frac{(\alpha\overline{\boldsymbol{u}}_a\otimes\overline{\boldsymbol{u}}_a+\beta\overline{\boldsymbol{u}}_b\otimes\overline{\boldsymbol{u}}_b)(\alpha+\beta)}{\alpha+\beta}=\frac{1}{\alpha+\beta}\left(\alpha^2\overline{\boldsymbol{u}}_a\otimes\overline{\boldsymbol{u}}_a+\beta^2\overline{\boldsymbol{u}}_b\otimes\overline{\boldsymbol{u}}_b+\alpha\beta(\overline{\boldsymbol{u}}_a\otimes\overline{\boldsymbol{u}}_a+\beta\overline{\boldsymbol{u}}_b\otimes\overline{\boldsymbol{u}}_b)\right)$$

Subtracting these expressions, we have

$$\begin{aligned}(\alpha+\beta)(\overline{\boldsymbol{u}}_+\otimes\overline{\boldsymbol{u}}_+) &- (\alpha\overline{\boldsymbol{u}}_a\otimes\overline{\boldsymbol{u}}_a+\beta\overline{\boldsymbol{u}}_b\otimes\overline{\boldsymbol{u}}_b)\\ &= \frac{\alpha\beta}{\alpha+\beta}\left[\overline{\boldsymbol{u}}_a\otimes\overline{\boldsymbol{u}}_b+\overline{\boldsymbol{u}}_b\otimes\overline{\boldsymbol{u}}_a-\overline{\boldsymbol{u}}_a\otimes\overline{\boldsymbol{u}}_a-\overline{\boldsymbol{u}}_b\otimes\overline{\boldsymbol{u}}_b\right]\\ &= \frac{\alpha\beta}{\alpha+\beta}(\overline{\boldsymbol{u}}_a-\overline{\boldsymbol{u}}_b)\otimes(\overline{\boldsymbol{u}}_b-\overline{\boldsymbol{u}}_a)\\ &= -\frac{(\alpha^2-\beta^2)^2}{4\alpha\beta(\alpha+\beta)}\left[(\overline{\boldsymbol{u}}_+-\overline{\boldsymbol{u}}_-)\otimes(\overline{\boldsymbol{u}}_+-\overline{\boldsymbol{u}}_-)\right]\end{aligned}$$

Hence the joint momentum equation takes the form

$$\begin{aligned}\frac{\partial\rho_+\overline{\boldsymbol{u}}_+}{\partial t}+\nabla.(\rho_+\overline{\boldsymbol{u}}_+\otimes\overline{\boldsymbol{u}}_+) &= -\nabla.\left[\frac{\rho_+\rho_-^2}{4\alpha\beta}(\overline{\boldsymbol{u}}_+-\overline{\boldsymbol{u}}_-)\otimes(\overline{\boldsymbol{u}}_+-\overline{\boldsymbol{u}}_-)\right]\\ &\quad-\nabla(\alpha_a\overline{p}_a+\alpha_b\overline{p}_b)+\nabla.\left[\alpha_a(\boldsymbol{\tau}_a+\boldsymbol{\sigma}_a^t)+\alpha_b(\boldsymbol{\tau}_b+\boldsymbol{\sigma}_b^t)\right]+\rho_+\boldsymbol{g}\end{aligned}$$

with the surface tension forces taken to be negligable.

5.4 Difference Momentum equation

There are probably several equations that could be generated representing the difference between the momentum equations. The simplest would be to subtract the two equations.

Now the interfacial terms no longer cancel, so we have

$$\begin{aligned}\frac{\partial}{\partial t}(\alpha\overline{\boldsymbol{u}}_a - \beta\overline{\boldsymbol{u}}_b) + \nabla.\left[\alpha\overline{\boldsymbol{u}}_a \otimes \overline{\boldsymbol{u}}_a - \beta\overline{\boldsymbol{u}}_b \otimes \overline{\boldsymbol{u}}_b\right] &= -\nabla(\alpha_a\overline{p}_a - \alpha_b\overline{p}_b) \\ &\quad +\nabla.\left[\alpha_a(\boldsymbol{\tau}_a + \boldsymbol{\sigma}_a^t) - \alpha_b(\boldsymbol{\tau}_b + \boldsymbol{\sigma}_b^t)\right] \\ &\quad +(\alpha-\beta)\boldsymbol{g} + 2\boldsymbol{M}_a\end{aligned}$$

As before we can write

$$\begin{aligned}(\alpha-\beta)(\overline{\boldsymbol{u}}_- \otimes \overline{\boldsymbol{u}}_-) &- (\alpha\overline{\boldsymbol{u}}_a \otimes \overline{\boldsymbol{u}}_a - \beta\overline{\boldsymbol{u}}_b \otimes \overline{\boldsymbol{u}}_b) \\ &= \frac{\alpha\beta}{\alpha-\beta}(\overline{\boldsymbol{u}}_a - \overline{\boldsymbol{u}}_b) \otimes (\overline{\boldsymbol{u}}_a - \overline{\boldsymbol{u}}_b) \\ &= \frac{(\alpha^2-\beta^2)^2}{4\alpha\beta(\alpha-\beta)}\left[(\overline{\boldsymbol{u}}_+ - \overline{\boldsymbol{u}}_-) \otimes (\overline{\boldsymbol{u}}_+ - \overline{\boldsymbol{u}}_-)\right]\end{aligned}$$

giving a momentum equation of the form

$$\begin{aligned}\frac{\partial\rho_-\overline{\boldsymbol{u}}_-}{\partial t} + \nabla.(\rho_-\overline{\boldsymbol{u}}_- \otimes \overline{\boldsymbol{u}}_-) &= \nabla.\left[\frac{\rho_+^2\rho_-}{4\alpha\beta}(\overline{\boldsymbol{u}}_+ - \overline{\boldsymbol{u}}_-) \otimes (\overline{\boldsymbol{u}}_+ - \overline{\boldsymbol{u}}_-)\right] \\ &\quad -\nabla(\alpha_a\overline{p}_a - \alpha_b\overline{p}_b) + \nabla.\left[\alpha_a(\boldsymbol{\tau}_a + \boldsymbol{\sigma}_a^t) - \alpha_b(\boldsymbol{\tau}_b + \boldsymbol{\sigma}_b^t)\right] \\ &\quad +\rho_-\boldsymbol{g} + 2\boldsymbol{M}_a\end{aligned}$$

5.5 Slip velocity momentum equation

Another possible transport equation would be for the slip velocity $\overline{\boldsymbol{u}}_a - \overline{\boldsymbol{u}}_b$. Starting again from the α momentum equation we have

$$\begin{aligned}\frac{\partial\alpha\overline{\boldsymbol{u}}_a}{\partial t} + \nabla.(\alpha\overline{\boldsymbol{u}}_a \otimes \overline{\boldsymbol{u}}_a) &= -\nabla\alpha_a p_a + \nabla.\alpha_a(\boldsymbol{\tau}_a + \boldsymbol{\sigma_a}) \\ &\quad +\alpha\boldsymbol{g} + \boldsymbol{M}\end{aligned}$$

Splitting the derivative terms on the l.h.s., and remembering that

$$\begin{aligned}\nabla.(\alpha\overline{\boldsymbol{u}}_a \otimes \overline{\boldsymbol{u}}_a) &= \nabla.(\overline{\boldsymbol{u}}_a \otimes \alpha\overline{\boldsymbol{u}}_a) \\ &= (\alpha\overline{\boldsymbol{u}}_a.\nabla)\overline{\boldsymbol{u}}_a + \overline{\boldsymbol{u}}_a\nabla.(\alpha\overline{\boldsymbol{u}}_a)\end{aligned}$$

Thus

$$\frac{\partial\overline{\boldsymbol{u}}_a}{\partial t} + (\overline{\boldsymbol{u}}_a.\nabla)\overline{\boldsymbol{u}}_a = \frac{1}{\alpha}\left[-\nabla\alpha_a p_a + \nabla.\alpha_a(\boldsymbol{\tau}_a + \boldsymbol{\sigma_a}) + \alpha\boldsymbol{g} + \boldsymbol{M}\right]$$

by subtracting the continuity equation. We can write down the b-phase equation by inspection

$$\frac{\partial\overline{\boldsymbol{u}}_b}{\partial t} + (\overline{\boldsymbol{u}}_b.\nabla)\overline{\boldsymbol{u}}_b = \frac{1}{\beta}\left[-\nabla\alpha_b p_b + \nabla.\alpha_b(\boldsymbol{\tau}_b + \boldsymbol{\sigma_b}) + \beta\boldsymbol{g} + \boldsymbol{M}\right]$$

Subtracting the two and writing $\overline{\boldsymbol{u}}_r = \overline{\boldsymbol{u}}_a - \overline{\boldsymbol{u}}_b$ we obtain

$$\begin{aligned}\frac{\partial \overline{\boldsymbol{u}}_r}{\partial t} + \nabla.(\overline{\boldsymbol{u}}_r \otimes \overline{\boldsymbol{u}}_r) &= \nabla.\left[(\overline{\boldsymbol{u}}_a - \overline{\boldsymbol{u}}_b) \otimes (\overline{\boldsymbol{u}}_a - \overline{\boldsymbol{u}}_b)\right] - \left[(\overline{\boldsymbol{u}}_a.\nabla)\overline{\boldsymbol{u}}_a - (\overline{\boldsymbol{u}}_b.\nabla)\overline{\boldsymbol{u}}_b\right] \\ &\quad - \left(\frac{1}{\alpha}\nabla\alpha_a p_a - \frac{1}{\beta}\nabla\alpha_b p_b\right) + \left(\frac{1}{\alpha}\nabla.\alpha_a(\tau_a + \sigma_a) - \frac{1}{\beta}\nabla.\alpha_b(\tau_b + \sigma_b)\right) \\ &\quad + \left(\frac{1}{\alpha} - \frac{1}{\beta}\right)\boldsymbol{M}\end{aligned}$$

Alternatively, we can go back a couple of steps, removing $\nabla.(\overline{\boldsymbol{u}}_r \otimes \overline{\boldsymbol{u}}_r)$ from both sides, and expanding $[(\overline{\boldsymbol{u}}_a.\nabla)\overline{\boldsymbol{u}}_a - (\overline{\boldsymbol{u}}_b.\nabla)\overline{\boldsymbol{u}}_b]$ in terms of

$$\begin{aligned}\overline{\boldsymbol{u}}_a &= \overline{\boldsymbol{u}}_+ + \frac{\beta}{\alpha+\beta}\overline{\boldsymbol{u}}_r \\ \overline{\boldsymbol{u}}_b &= \overline{\boldsymbol{u}}_+ - \frac{\alpha}{\alpha+\beta}\overline{\boldsymbol{u}}_r\end{aligned}$$

In this case

$$\begin{aligned}(\overline{\boldsymbol{u}}_a.\nabla)\overline{\boldsymbol{u}}_a - (\overline{\boldsymbol{u}}_b.\nabla)\overline{\boldsymbol{u}}_b &= \overline{\boldsymbol{u}}_r.\nabla\overline{\boldsymbol{u}}_+ + \overline{\boldsymbol{u}}_+.\nabla\overline{\boldsymbol{u}}_r + \frac{\beta-\alpha}{\alpha+\beta}\overline{\boldsymbol{u}}_r.\nabla\overline{\boldsymbol{u}}_r \\ &\quad + \left(\frac{\beta-\alpha}{\alpha+\beta}\right)\overline{\boldsymbol{u}}_r \otimes \overline{\boldsymbol{u}}_r.\nabla\left(\frac{\beta-\alpha}{\alpha-\beta}\right)\end{aligned}$$

and thus

$$\begin{aligned}\frac{\partial \overline{\boldsymbol{u}}_r}{\partial t} + \overline{\boldsymbol{u}}_r.\nabla\overline{\boldsymbol{u}}_+ &= -\left[\overline{\boldsymbol{u}}_+.\nabla\overline{\boldsymbol{u}}_r + \frac{\beta-\alpha}{\alpha+\beta}\overline{\boldsymbol{u}}_r.\nabla\overline{\boldsymbol{u}}_r + \left(\frac{\beta-\alpha}{\alpha+\beta}\right)\overline{\boldsymbol{u}}_r \otimes \overline{\boldsymbol{u}}_r.\nabla\left(\frac{\beta-\alpha}{\alpha-\beta}\right)\right] \\ &\quad - \left(\frac{1}{\alpha}\nabla\alpha_a p_a - \frac{1}{\beta}\nabla\alpha_b p_b\right) + \left(\frac{1}{\alpha}\nabla.\alpha_a(\tau_a + \sigma_a) - \frac{1}{\beta}\nabla.\alpha_b(\tau_b + \sigma_b)\right) \\ &\quad + \left(\frac{1}{\alpha} - \frac{1}{\beta}\right)\boldsymbol{M}\end{aligned}$$

The advantage of solving for the slip velocity like this is that it makes it possible to make the drag term implicit (since it is a function of $\boldsymbol{u}_r$). Other ways of accomplishing this are discussed in sections 6.3 and 9.1.

5.6 Elimination of α

The final route to consider here is to attempt to eliminate α from the momentum equation altogether. Consider the rhs. of equation (8) :

$$\begin{aligned}\frac{\partial}{\partial t}\alpha_k\rho_k\overline{\boldsymbol{u}}_k + \nabla.(\alpha_k\rho_k\overline{\boldsymbol{u}}_k \otimes \overline{\boldsymbol{u}}_k) &= \alpha_k\left[\frac{\partial \rho_k\overline{\boldsymbol{u}}_k}{\partial t} + \nabla.(\rho_k\overline{\boldsymbol{u}}_k \otimes \overline{\boldsymbol{u}}_k)\right] \\ &\quad + \rho_k\overline{\boldsymbol{u}}_k \otimes \left[\frac{\partial \alpha_k}{\partial t} + \overline{\boldsymbol{u}}_k.\nabla\alpha_k\right]\end{aligned}$$

The second term here is zero. Rearanging the equation gives

$$\frac{\partial \rho_k \overline{\boldsymbol{u}}_k}{\partial t} + \nabla.(\rho_k \overline{\boldsymbol{u}}_k \otimes \overline{\boldsymbol{u}}_k) = \frac{1}{\alpha_k}\left[-\nabla \alpha_k \overline{p_k} + \nabla.\alpha_k(\overline{\tau_k} + \sigma_k^t)\right] + \rho_k \boldsymbol{g} + \frac{\boldsymbol{M}_k}{\alpha_k} \quad (37)$$

The term in brackets $[\ldots]$ is potentially troublesome. The α-dependence can be factored out to produce two terms

$$\left[-\nabla \overline{p_k} + \nabla.(\overline{\tau_k} + \sigma_k^t)\right] + \left[-p_k \boldsymbol{I} + (\overline{\tau_k} + \sigma_k^t)\right].\frac{\nabla \alpha_k}{\alpha_k} \quad (38)$$

Since $\nabla\alpha/\alpha$ is divergent as $\alpha \to 0$, this constrains $[-p_k \boldsymbol{I} + (\overline{\tau_k} + \sigma_k^t)] \to 0$ at least as fast as α. There is some debate as to the exact form of the stress tensor for multiphase flow, specifically as to whether the α dependence should be inside the derivative. Either way, the $\nabla\alpha/\alpha$ term is not thought to present problems if handled correctly and should limit to zero. This formulation for the momentum equation is the one used in the Weller algorithm presented in section 7.

6 Superficial Velocity Approach

The essence of this scheme is to solve for the superficial velocity $\boldsymbol{U}_s = \alpha \boldsymbol{u}_a$, which is well defined throughout the whole solution domain, taking the value 0 where the phase a is not present. A similar superficial velocity $\boldsymbol{V}_s$ can be defined for the second phase. In addition, introducing the superficial velocity improves the formulation of the α-equation by producing a formulation which is more symmetrical (based on the fluxes of both superficial velocities, see section (6.1)).

6.1 Continuity equations

If we assume $\rho = constant$, we can factor it out of the continuity equation, and multiply through by α_a, giving

$$\alpha_a \frac{\partial \alpha_a}{\partial t} + \alpha_a \nabla.\alpha_a \overline{\boldsymbol{u}}_a = 0 \quad (39)$$

and similarly for the second phase

$$\alpha_b \frac{\partial \alpha_b}{\partial t} + \alpha_b \nabla.\alpha_b \overline{\boldsymbol{u}}_b = 0 \quad (40)$$

Subtracting these and using the fact that $1 = \alpha_a + \alpha_b$,

$$\alpha_a \frac{\partial \alpha_a}{\partial t} - (1-\alpha_a)\frac{\partial}{\partial t}(-\alpha_a) + \alpha_a \nabla.\alpha_a \overline{\boldsymbol{u}}_a - (1-\alpha_a)\nabla.\alpha_b \overline{\boldsymbol{u}}_b = 0 \quad (41)$$

Cancelling terms and rearanging

$$\frac{\partial \alpha_a}{\partial t} = -\alpha_a \nabla.(\alpha_a \overline{\boldsymbol{u}}_a + \alpha_b \overline{\boldsymbol{u}}_b) + \nabla.\alpha_b \overline{\boldsymbol{u}}_b \quad (42)$$

Adding $\nabla.\alpha_a\overline{\boldsymbol{u}}_a$ to both sides gives

$$\begin{aligned}\frac{\partial\alpha_a}{\partial t}+\nabla.\alpha_a\overline{\boldsymbol{u}}_a &= \nabla.(\alpha_a\overline{\boldsymbol{u}}_a+\alpha_b\overline{\boldsymbol{u}}_b)-\alpha_a\nabla.(\alpha_a\overline{\boldsymbol{u}}_a+\alpha_b\overline{\boldsymbol{u}}_b)\\ &= \alpha_b\nabla.(\alpha_a\overline{\boldsymbol{u}}_a+\alpha_b\overline{\boldsymbol{u}}_b)\end{aligned}$$

But the l.h.s.of this is just the α_a equation, so we have the result that

$$\nabla.(\alpha_a\overline{\boldsymbol{u}}_a+\alpha_b\overline{\boldsymbol{u}}_b)=0 \tag{43}$$

This basically states that the flow is incompressible. Since many of the flows we need to deal with have compressible parts to them, this is not too useful, and the information can probably be included in other ways.

On the other hand, if we expand the divergence term first we get terms in

$$\alpha_a\nabla.\alpha_a\overline{\boldsymbol{u}}_a=\alpha_a\overline{\boldsymbol{u}}_a\nabla\alpha_a+\alpha_a^2\nabla.\overline{\boldsymbol{u}}_a \tag{44}$$

and similarly for α_b by rotation. Subtracting, and then using incompressibility, we have

$$\frac{\partial\alpha_a}{\partial t}+\nabla.(\alpha_a\overline{\boldsymbol{u}}_a+\alpha_b\overline{\boldsymbol{u}}_b)\alpha_a=\alpha_b^2\nabla.\overline{\boldsymbol{u}}_b-\alpha_a^2\nabla.\overline{\boldsymbol{u}}_a \tag{45}$$

This requires a certain ammount more manipulation to get into the required form. In the incompressible case we can include the fluxes inside the divergence term. We also want to express the rhs. in terms of the superficial velocities. Writing

$$\begin{aligned}\alpha_b^2\nabla.\overline{\boldsymbol{u}}_b &= \alpha_b\nabla.\alpha_b\overline{\boldsymbol{u}}_b-\alpha_b\overline{\boldsymbol{u}}_b.\nabla\alpha_b\\ &= \alpha_b\nabla.\boldsymbol{V}_s-\boldsymbol{V}_s.\nabla\alpha_b\\ &= (1-\alpha_a)\nabla.\boldsymbol{V}_s+\boldsymbol{V}_s.\nabla\alpha_a\end{aligned}$$

and similarly for the a-phase, we have

$$\frac{\partial\alpha_a}{\partial t}+\nabla.(\boldsymbol{U}_s+\boldsymbol{V}_s)\alpha_a=(1-\alpha_a)\nabla.\boldsymbol{V}_s-\alpha_a\nabla.\boldsymbol{U}_s+(\boldsymbol{V}_s+\boldsymbol{U}_s).\nabla\alpha_a \tag{46}$$

But (again since we are considering the incompressible case) $\nabla.(\boldsymbol{U}_s+\boldsymbol{V}_s)=0$, so this becomes

$$\frac{\partial\alpha_a}{\partial t}+\nabla.(\boldsymbol{U}_s+\boldsymbol{V}_s)\alpha_a=-\nabla.\boldsymbol{U}_s+(\boldsymbol{V}_s+\boldsymbol{U}_s).\nabla\alpha_a \tag{47}$$

This is really no more than the original a-phase continuity equation with $\nabla.(\boldsymbol{U}_s+\boldsymbol{V}_s)\alpha_a$ added to both sides : however it does mean that the equation is no longer source-dominated.

6.2 Momentum Equation and Algorithm Formulation

The momentum equation now takes the form

$$\frac{\partial \boldsymbol{U}_s}{\partial t} + \nabla.\frac{\boldsymbol{U}_s}{\alpha} \otimes \boldsymbol{U}_s = -\frac{\alpha}{\rho_a}\nabla p + \nabla.\alpha\boldsymbol{\tau} + \boldsymbol{f} \tag{48}$$

which we can express as

$$a_p \boldsymbol{U}_p = \mathcal{H}_a(\boldsymbol{U}) - \frac{\alpha}{\rho_a}\nabla p \tag{49}$$

We can do a similar thing with the continuous phase superficial velocity $\boldsymbol{V}$

$$b_p \boldsymbol{V}_p = \mathcal{H}_b(\boldsymbol{V}) - \frac{\beta}{\rho_b}\nabla p \tag{50}$$

rearanging and adding the equations,

$$\boldsymbol{U}_p + \boldsymbol{V}_p = \frac{\mathcal{H}_a(\boldsymbol{U})}{a_p} + \frac{\mathcal{H}_b(\boldsymbol{V})}{b_p} - \left(\frac{\alpha}{a_p\rho_a} + \frac{\beta}{b_p\rho_b}\right)\nabla p \tag{51}$$

We can now evaluate the divergence of each side. But $\nabla.(\boldsymbol{U} + \boldsymbol{V}) = 0$, so

$$\nabla.\left[\left(\frac{\alpha}{a_p\rho_a} + \frac{\beta}{b_p\rho_b}\right)\nabla p\right] = \nabla.\left(\frac{\mathcal{H}_a(\boldsymbol{U})}{a_p} + \frac{\mathcal{H}_b(\boldsymbol{V})}{b_p}\right) \tag{52}$$

This is essentially an equation for the mixture pressure-velocity system, and so should be well-posed. In particular, a_p should be finite at all times (unlike the equivalent case for bubbleFoam where $a_p = 0$ for $\alpha = 0$), and so as $\alpha \to 0$ the source term contribution for the pressure equation from the a phase $\to 0$ as well. Hence evaluation of the rhs of eqn.(52) is not a problem. The algorithm takes the form

1. Solve eq.(48), given initial guesses of flux $\phi = \boldsymbol{U}/\alpha$ and p, for $\boldsymbol{U}$ (and solve the equivalent b-phase momentum equation as well).
2. Solve eq.(52) for p
3. Evaluate new values of $\boldsymbol{U}$ and $\boldsymbol{V}$ from eqs (50) and (49), and thus the fluxes.

6.3 Partial Elimination in the Superficial Velocity Approach

The aim of partial elimination [11][12] is to improve the numerical behaviour of the momentum equation by making the drag term implicit. The drag term depends on the relative velocity between the bubble and its surroundings (as do all the interface terms, to some extent) so it introduces some coupling between the A and B phase momentum

equations. The aim is to eliminate this coupling in the drag, at the expense of introducing coupling in other terms. Solving for the superficial velocity, $\alpha \boldsymbol{u}$ introduces further problems into this, however. This is because the drag term has the form

$$\boldsymbol{F}_D = \frac{3}{4}\frac{\alpha \rho_b C_d}{d}|\boldsymbol{u}_r|\boldsymbol{u}_r \tag{53}$$

where phase $_a$ is the dispersed phase and $_b$ is the continuous one, and $\boldsymbol{u}_r = \boldsymbol{u}_b - \boldsymbol{u}_a$. We need to formulate this in terms of the superficial velocities $\boldsymbol{U}_a = \alpha \boldsymbol{u}_a$ and $\boldsymbol{U}_b = \beta \boldsymbol{u}_b$. Rearanging this gives

$$\boldsymbol{F}_D = f_D\left(\alpha \boldsymbol{u}_b - \boldsymbol{U}_a\right) = f_D\left(\xi \boldsymbol{U}_b - \boldsymbol{U}_a\right) \tag{54}$$

where

$$f_D = \frac{3}{4}\frac{\rho_b C_d}{d}|\boldsymbol{u}_r|, \tag{55}$$

and $\xi = \alpha/\beta$. This parameter is singular where $\beta \to 0$, but in this high phase fraction limit the simplistic drag law used here probably breaks down anyway.

We can now follow the standard route for the partial elimination algorithm, with some variation. We write the discretised momentum equations for an intermediate velocity $\boldsymbol{U}_a^\sharp$

$$\begin{aligned}
\left(a_a + f_D\right)\boldsymbol{U}_a^\sharp &= \mathcal{H}\left(\boldsymbol{U}_a^n\right) + f_D \xi \boldsymbol{U}_b^{n+1} \\
\left(a_b + \xi f_D\right)\boldsymbol{U}_b^\sharp &= \mathcal{H}\left(\boldsymbol{U}_b^n\right) + f_D \boldsymbol{U}_a^{n+1}
\end{aligned}$$

Rearanging these equations we have

$$\begin{aligned}
\boldsymbol{U}_a^\sharp &= \frac{\mathcal{H}\left(\boldsymbol{U}_a^n\right) + f_D \xi \boldsymbol{U}_b^{n+1}}{a_a + f_D} \\
\boldsymbol{U}_b^\sharp &= \frac{\mathcal{H}\left(\boldsymbol{U}_b^n\right) + f_D \boldsymbol{U}_a^{n+1}}{a_b + \xi f_D}
\end{aligned} \tag{56}$$

We can now write the entire timestep in differenced form as

$$\begin{aligned}
\left(a_a + f_D\right)\boldsymbol{U}_a^{n+1} &= \mathcal{H}\left(\boldsymbol{U}_a^{n+1}\right) + f_D \xi \boldsymbol{U}_b^\sharp \\
&= \mathcal{H}\left(\boldsymbol{U}_a^{n+1}\right) + f_D \xi \frac{\mathcal{H}\left(\boldsymbol{U}_b^n\right) + f_D \boldsymbol{U}_a^{n+1}}{a_b + \xi f_D} \\
\Rightarrow \left(a_a + f_D - \frac{f_D^2 \xi}{a_b + f_D \xi}\right)\boldsymbol{U}_a^{n+1} &= \left(a_a + \frac{a_b f_D}{a_b + \xi f_D}\right)\boldsymbol{U}_a^{n+1} \\
&= \mathcal{H}\left(\boldsymbol{U}_a^{n+1}\right) + \left(\frac{f_D \xi}{a_b + \xi f_D}\right)\mathcal{H}\left(\boldsymbol{U}_b^n\right) \\
\Rightarrow \left(a_a + a_b \mathcal{B}\right)\boldsymbol{U}_a^{n+1} &= \mathcal{H}\left(\boldsymbol{U}_a^{n+1}\right) + \xi \mathcal{B}\mathcal{H}\left(\boldsymbol{U}_b^n\right)
\end{aligned} \tag{57}$$

where

$$\mathcal{B} = \frac{f_D}{a_b + \xi f_D} \tag{58}$$

Similarly for the other equation

$$(a_b + a_a \xi \mathcal{A})\, \boldsymbol{U}_b^{n+1} = \mathcal{H}\left(\boldsymbol{U}_b^{n+1}\right) + \mathcal{A}\mathcal{H}\left(\boldsymbol{U}_a^{n}\right) \tag{59}$$

$$\mathcal{A} = \frac{f_D}{a_a + f_D} \tag{60}$$

7 Weller Algorithm for Multiphase Flow

The second approach being investigated as part of the project is based on work done by one of our collegues at IC, Mr.Henry Weller. In addition to being the principle author of the FOAM code, he is a consultant to CD, and at our request worked with us to develop and test a more robust algorithm for multiphase flow. A short description of the algorithm is given here, and a more complete description will be provided in the form of a report to CD on his work[13]. Some test cases demonstrating the robustness of the algorithm follow.

7.1 The Equations.

The α equation takes the form

$$\frac{\partial \alpha}{\partial t} + \nabla.(\boldsymbol{\phi}\alpha) + \nabla.(\beta \boldsymbol{\phi}_r \alpha) = \nabla.\left[C_\alpha \nu_t \nabla \alpha\right] \tag{61}$$

where the fluxes are

$$\boldsymbol{\phi} = \alpha_F \boldsymbol{\phi}_a + \beta_F \boldsymbol{\phi}_b \tag{62}$$

$$\boldsymbol{\phi}_r = \boldsymbol{\phi}_a - \boldsymbol{\phi}_b \tag{63}$$

and $_F$ refers to values of the variable evaluated on the cell face. We also have velocity fluxes

$$\boldsymbol{\phi}_{\boldsymbol{U}_a} = \boldsymbol{\phi}_a + \nu_{eff_a} \frac{\nabla \alpha}{\alpha}$$

$$\boldsymbol{\phi}_{\boldsymbol{U}_b} = \boldsymbol{\phi}_b + \nu_{eff_b} \frac{\nabla \beta}{\beta}$$

with ν_{eff_a} as the total (laminar plus turbulent) viscosity in phase a. Coefficients for the virtual mass, drag, lift terms are all as normal. We define additional terms

$$\mathcal{VM}_a = 1 + \frac{C_{VM}\rho_b \beta}{\rho_a}$$

$$\mathcal{VM}_b = 1 + \frac{C_{VM}\rho_a \alpha}{\rho_b}$$

With these we can write the momentum equations as

$$\begin{aligned}\frac{\partial \boldsymbol{U}_a}{\partial t} \quad + \quad & \nabla.(\boldsymbol{\phi}_{\boldsymbol{U}_a}\boldsymbol{U}_a) - (\nabla.\phi_{\boldsymbol{U}_a})\boldsymbol{U}_a - \nabla.\nu_{eff_a}\nabla\boldsymbol{U}_a \\ & -\nabla.\left[\nu_{eff_a}(\nabla\boldsymbol{U}_a - \frac{2}{3}\boldsymbol{I}\nabla.\boldsymbol{U}_a)\right] - \nu_{eff_a}\left[\frac{\nabla\alpha}{\alpha}.\left(\nabla\boldsymbol{U}_a^t - \frac{2}{3}\boldsymbol{I}\nabla.\boldsymbol{U}_a\right)\right] \\ & = \frac{\boldsymbol{g}}{\mathcal{VM}_a} - \frac{\beta C_D}{\rho_a \mathcal{VM}_a}\boldsymbol{U}_a - \frac{\beta}{\rho_a\mathcal{VM}_a}\left(C_F - C_{VM}\rho_b\frac{D\boldsymbol{U}_b}{Dt}\right) \end{aligned} \tag{64}$$

Operators $\mathcal{A}_a$ and $\mathcal{H}_a$ are defined in the usual way on this equation, and similarly for the b equation :

$$\begin{aligned}\frac{\partial \boldsymbol{U}_b}{\partial t} \quad + \quad & \nabla.(\boldsymbol{\phi}_{\boldsymbol{U}_b}\boldsymbol{U}_b) - (\nabla.\phi_{\boldsymbol{U}_b})\boldsymbol{U}_b - \nabla.\nu_{eff_b}\nabla\boldsymbol{U}_b \\ & -\nabla.\left[\nu_{eff_b}(\nabla\boldsymbol{U}_b - \frac{2}{3}\boldsymbol{I}\nabla.\boldsymbol{U}_b)\right] - \nu_{eff_b}\left[\frac{\nabla\beta}{\beta}.\left(\nabla\boldsymbol{U}_b^t - \frac{2}{3}\boldsymbol{I}\nabla.\boldsymbol{U}_b\right)\right] \\ & = \frac{\boldsymbol{g}}{\mathcal{VM}_b} - \frac{\alpha C_D}{\rho_b \mathcal{VM}_b}\boldsymbol{U}_b - \frac{\alpha}{\rho_b\mathcal{VM}_b}\left(C_F - C_{VM}\rho_a\frac{D\boldsymbol{U}_a}{Dt}\right) \end{aligned} \tag{65}$$

Here

$$\begin{aligned}\frac{D\boldsymbol{U}_a}{Dt} &= \frac{\partial \boldsymbol{U}_a}{\partial t} + \nabla.(\boldsymbol{\phi}_a\boldsymbol{U}_a) - (\nabla.\boldsymbol{\phi}_a)\boldsymbol{U}_a \\ \frac{D\boldsymbol{U}_b}{Dt} &= \frac{\partial \boldsymbol{U}_b}{\partial t} + \nabla.(\boldsymbol{\phi}_b\boldsymbol{U}_b) - (\nabla.\boldsymbol{\phi}_b)\boldsymbol{U}_b \end{aligned}$$

We make an initial estimate of the fluxes as

$$\begin{aligned}\boldsymbol{\phi}_a^* &= \frac{\mathcal{H}_a\delta A}{\mathcal{A}_a} + \frac{\beta C_D}{\rho_a\mathcal{VM}_a\mathcal{A}_a}\boldsymbol{\phi}_b \\ \boldsymbol{\phi}_b^* &= \frac{\mathcal{H}_b\delta A}{\mathcal{A}_b} + \frac{\alpha C_D}{\rho_b\mathcal{VM}_b\mathcal{A}_b}\boldsymbol{\phi}_a \end{aligned}$$

In terms of these quantities the pressure equation takes the form

$$\nabla.\left[\left(\frac{\alpha_F}{\rho_a\mathcal{A}_a\mathcal{VM}_a} + \frac{\beta_F}{\rho_b\mathcal{A}_b\mathcal{VM}_b}\right)\nabla p\right] = \nabla.\left(\alpha_F\boldsymbol{\phi}_a^* + \beta_F\boldsymbol{\phi}_b^*\right) \tag{66}$$

7.2 Pressure update algorithm.

The pressure update algorithm is complex.

1. An estimate of the individual phase fluxes is made in terms of their old values

$$\begin{aligned}\boldsymbol{\phi}_a^* &= \frac{\mathcal{H}_a\delta A}{\mathcal{A}_a} + \frac{\beta C_D}{\rho_a\mathcal{VM}_a\mathcal{A}_a}\boldsymbol{\phi}_b \\ \boldsymbol{\phi}_b^* &= \frac{\mathcal{H}_b\delta A}{\mathcal{A}_b} + \frac{\alpha C_D}{\rho_b\mathcal{VM}_b\mathcal{A}_b}\boldsymbol{\phi}_a \end{aligned}$$

   (note the crossover of the flux terms).

2. The face-valued phase fractions α_F, β_F are evaluated from their cell-centred values.
3. Equation (66) is constructed and solved for p.
4. The individual phase fluxes are updated as

$$\begin{aligned} \boldsymbol{\phi}_a &= \boldsymbol{\phi}_a^* - \frac{1}{\rho_a \mathcal{A}_a \mathcal{VM}_a} \|\delta A\| \nabla p \\ \boldsymbol{\phi}_b &= \boldsymbol{\phi}_b^* - \frac{1}{\rho_b \mathcal{A}_b \mathcal{VM}_b} \|\delta A\| \nabla p \end{aligned}$$

5. Finally, the overall flux is evaluated using (62) :

$$\boldsymbol{\phi} = \alpha_F \boldsymbol{\phi}_a + \beta_F \boldsymbol{\phi}_b$$

This will be referred to as *pressure update*.

7.3 Overall algorithm

We are now in a position to state the overall algorithm.

1. Solve the α-equation, eq. (61).
2. Calculate the physical coefficients C_D, C_{VM} etc.
3. Construct (but do not *solve*) the momentum equations (64), (65).
4. Execute the following loop $N = 4$ times :
   (a) Solve the α-equation
   (b) *Pressure update*
5. Evaluate the cell-centred velocities by integrating the appropriate fluxes over the cell. Evaluate the convective derivatives $D\boldsymbol{U}_a/Dt$, $D\boldsymbol{U}_b/Dt$.
6. Solve $k - \epsilon$ system of equations.

7.4 Demonstration case : Phase Separation

As an extreme demonstration case, phase separation was simulated. The case used was a 2-dimensional rectangular domain entirely bounded by walls, and the initial conditions for the simulation were that the domain was entirely filled with an air-water mixture at 50% phase fraction. Initially the domain was at 45° to the vertical, and the separation effect due to gravity was simulated. The evolution of the system is shown in the top row of figure 1. Red represents $\alpha = 1.0$, i.e. air, and blue $\alpha = 0.0$, i.e. water. The two phases separate out cleanly, and a stable interface is formed between the regions. On the bottom

row, the domain is turned upright (actually, the direction of gravity is redefined) and the interface settles quickly into its new orientation. The stability demonstrated in this test case partly addresses Dr. Liu's point 5, and shows that the algorithm is capable of solving simple single-phase problems to give reasonable solutions.

7.5 Comparison with INPT test case

Several experimental test cases were investigated as part of the BRITE-II project[14][15], and simulations performed to validate the multiphase algorithm developed during that project[2]. These cases should be recalculated using the Weller algorithm in order to validate it. Preliminary results for one specific case (case 1.2, using the nomenclature adopted before) are given here, in order to demonstrate the algorithm's potential. The case consists of a shear flow in a duct of about 2 m height, bubbles of $1.5 - 2.5$ mm diameter are introduced into the flow at a phase fraction of 1.9%, and profiles of α and the axial component of the phase velocities are measured at various distances downstream from the splitter plate. Measured inlet quantities were the phase fraction and continuous phase velocity profiles, together with the axial component of the r.m.s. continuous phase fluctuating velocity. This is insufficient to define the inlet, so assumptions were made analogous to those made by Hill to complete the inlet. The bubbles were assumed to have reached terminal rise velocity by the inlet, and so have a similar profile to that of the continuous phase, and a specific anisotropy for the turbulence was assumed. The case was simulated on a $15,000$ cell mesh.

The results are shown in figure 2, for α and continuous phase velocity at 6, 20 and 40 cm downstream of the inlet. The results of two separate runs are shown. In the first, the dispersive coefficient α_α is set to zero. The resulting profiles match the experimental data very well at the initial measurement station (much better than Hill managed using the previous code), but the central α peak remains too strong further downstream, due to a lack of turbulent diffusive effects on α. In the second run, a value for this coefficient of $\alpha_\alpha = 0.25$ is used, which ameliorates this somewhat. Some fine-tuning of the modelling would probably improve the results, in particular to find a valid value for α_α, but this is irrelevant to the algorithm development.

7.6 Bubble Column

The other demonstration case that has been calculated is that of a bubble column with gas being injected at the base. In the past in multiphase flow calculations attempts have been made to create some kind of 'top-surface' boundary, which does not allow the liquid to leave the domain but does allow the gas to escape. However, given the capabilities of bubbleFoam, it turned out to be easiest to calculate this case by including the region above the liquid in the domain, allowing the code to calculate the position of the gas-liquid interface wherever necessary, and using a simple outlet boundary at the top of the domain to allow gas to escape. The domain chosen corresponds to a bubble column of 1 m height and 10 cm diameter, with water as the continuous phase, and a dispersed phase of air

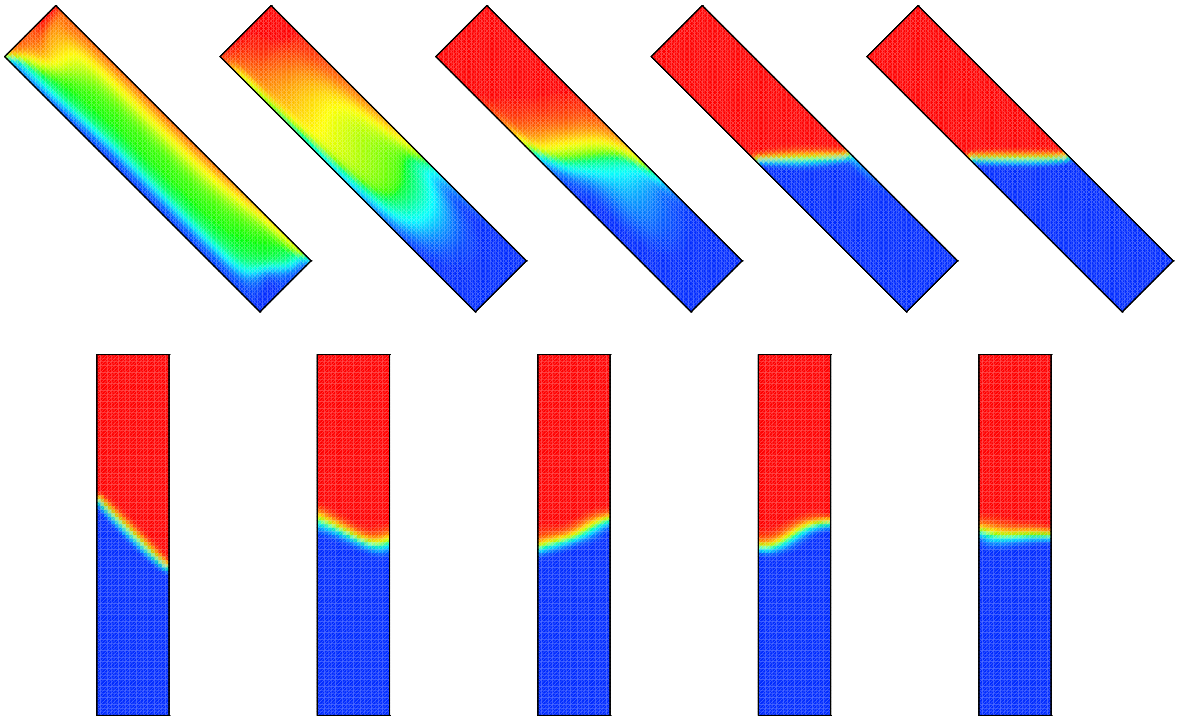

Figure 1: Phase separation. A wall-bounded domain, initially at 45°, is filled with an air-water mixture at a volume fraction of $\alpha = 0.5$. The water is allowed to separate out under the influence of gravity, as shown in the top figures – red represents $\alpha = 1.0$, blue represents $\alpha = 0.0$. The domain is then turned 'upright', and the interface levels out (lower figures).

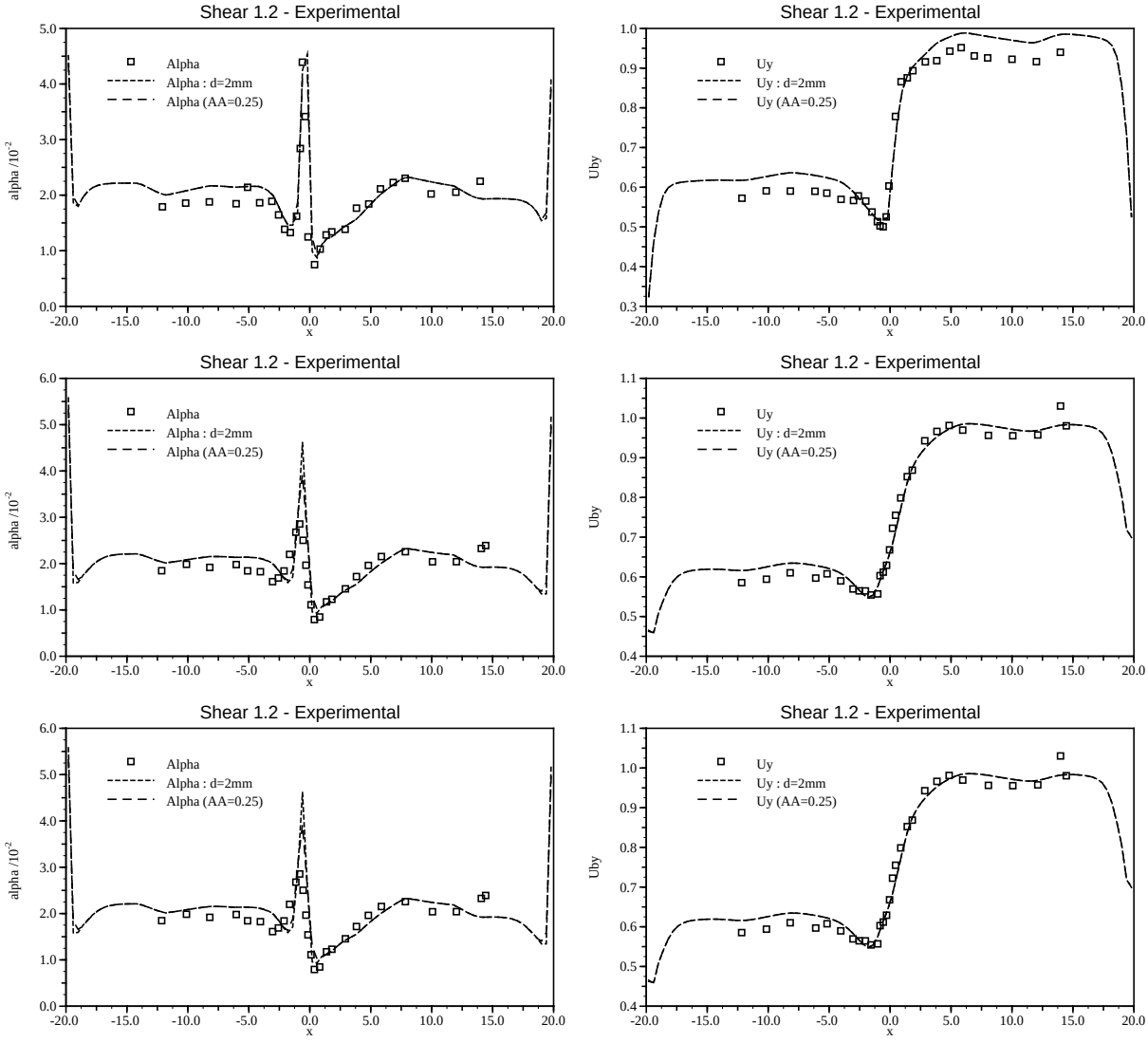

Figure 2: INPT comparison : profiles of α (left column) and axial component of the continuous phase velocity (right column) at 6, 20 and 40 cm downstream of the inlet.

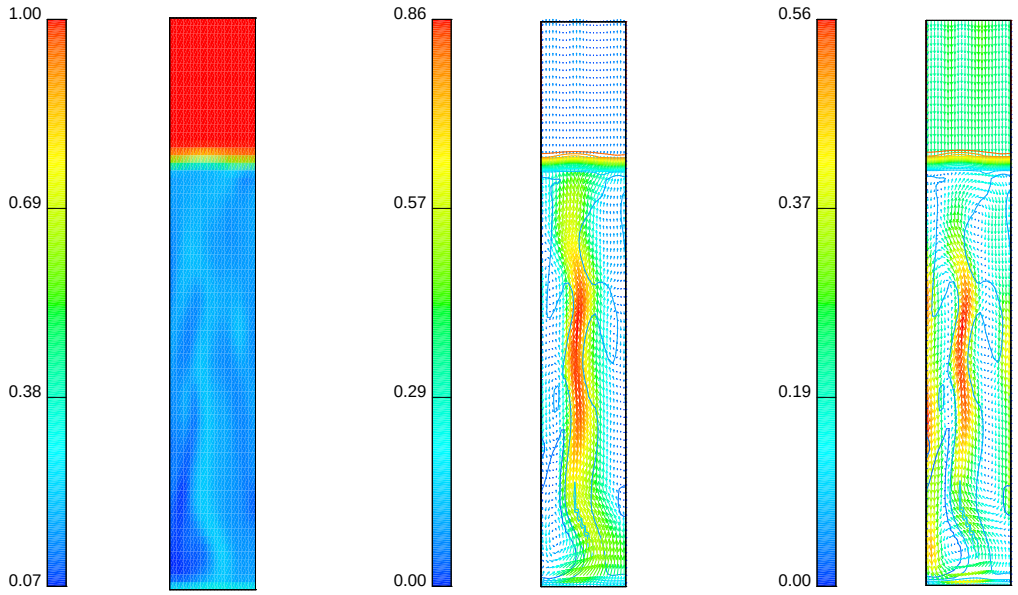

Figure 3: Bubble column being sparged by air. From left to right the figures show : α, dispersed phase velocity $\boldsymbol{u}_a$, continuous phase velocity $\boldsymbol{u}_b$. Contours of the phase fraction are also superimposed on the velocity diagrams

being sparged in at the bottom at a superficial velocity of 0.05 ms^{-1} and void fraction of 0.5. The overall behaviour was found to be transient, and an animation of the behaviour has been prepared. The results at one typical time are shown in figure 3, which shows the phase fraction α, and the two phase velocities $\boldsymbol{u}_a$ and $\boldsymbol{u}_b$.

8 Stability and Convergence Comparisons between the Algorithms.

The two algorithms have been implemented as described, and comparative tests run on BRITE-II test cases to evaluate their relative performance. Both time-dependent (PISO) and time-independent (SIMPLE) variants of the algorithms have been constructed.

** THis section needs to change **

This algorithm has been implemented and tested in FOAM. One specific test case is discussed here : a shear flow in a wall-bounded duct, calculated on a 6400 cell mesh. Figure 4 shows the residual histories for the previous algorithm (code briteIIFoam) and the superficial velocity alogrithm (code superficialFoam). Initial residuals of α, p, and the axial velocities (superficial velocities in the case of superficialFoam) at each timestep are plotted against timestep. The convergence history of superficialFoam is similar to that

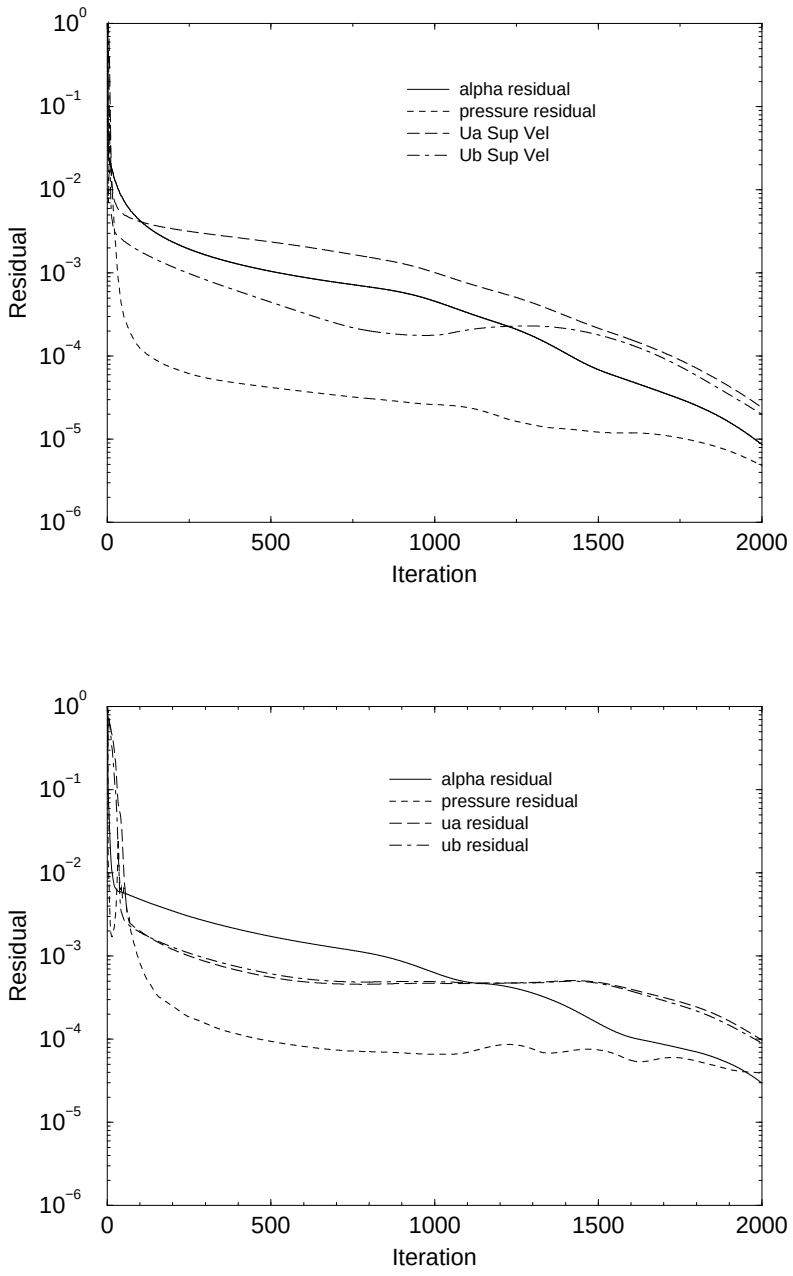

Figure 4: Residual convergence for superficialFoam and briteIIFoam. The case is a shear flow in a duct with walls, calculated on a 6400 cell mesh. For superficialFoam (top figure), which implements the superficial velocity alogorithm, residuals of α, p and the axial component of the two superficial velocities is plotted against iteration number. For briteIIFoam, which implements the algorithm developed in the previous BRITE project, residuals of the true velocity are shown, as well as α and p.

of the previous algorithm, although it does show something of an improvement towards the end. For this particular case, the superficial velocity approach seems to improve the convergence rate as compared to the previous algorithm, although not by a great factor. However, calculations on other cases demonstrate that the algorithm is no more stable than the previous version. Problems arise where the phase fraction $\alpha \rightarrow 0$, for instance in the BRITE-II shear flow (see section 7.5) and sudden expansion cases. The problem is similar to that raised by Dr.Liu with reference to the original code (section 2.3), and becomes accentuated when the mesh is refined. This is due to the fact that upwind differencing is being used throughout the algorithm, which stabilises the scheme by introducing numerical diffusion ; the diffusion is reduced as the mesh is refined. The typical mode of failure is illustrated in figure 5, which focusses on a small section of the mesh close to the splitter plate for the INPT shear flow experiment. During the evolution of the calculation towards the steady state, the phase fraction on one side of the shear layer drops to zero, whilst at the same time large random real velocities are generated in the vicinity, since the real velocity $\boldsymbol{u}$ is being determined as the ratio of two small numbers, a numerically unstable procedure. The algorithm also fails on extreme cases, such as the phase separation case discussed in section 7.4 below.

A SIMPLE version of the algorithm has been constructed and is being tested. This shows much the same behaviour. It is somewhat more stable than the PISO version, but displays the same convergence behaviour as the SIMPLE version of briteIIFoam. It also fails on extreme cases such as phase separation.

The problem with such cases lies in deriving the true velocity $\boldsymbol{u} = \boldsymbol{U}/\alpha$ in the limit as $\alpha \rightarrow 0$. The true velocity is important in the algorithm in two ways, in evaluating the physics (which is determined by the relative velocity of the bubbles w.r.t. the continuous phase), and as the flux $\boldsymbol{\phi}$ of $\boldsymbol{U}$ in the momentum equation. Consider first the flux. The limit of $\boldsymbol{U}/\alpha$ is $\boldsymbol{u}$, which is an unknown (determining this is in fact the whole point of the algorithm) but is *definitely non-zero*. In the limit $\alpha = 0$ this may be irrelevant, since $\boldsymbol{\phi}$ only appears in the divergence term and is multiplied by $\boldsymbol{U} = 0$ at this point. However as α approaches 0, problems arise. If $\vec{u}$ is evaluated directly, then the algorithm will be subject to instabilities. If the calculation is stabilised by evaluating $\boldsymbol{u} = \boldsymbol{U}/(\alpha + \varepsilon)$, the flux transporting $\boldsymbol{U}$ will be systematically underpredicted. In the INPT case above, $\varepsilon = 0.01$ will stabilise the calculation, but this is a value of ε similar to the α values being calculated.

There are a couple of possible ways around this. One is somehow to evaluate $\boldsymbol{u}$ separately in regions where the phase fraction is low. This might be possible by assuming a force balance for the dispersed phase and then taking a phase-fraction-weighted average of the two estimates for ϕ. This would be somewhat cumbersome. The alternative is to solve for $\boldsymbol{U}_{\surd} = \sqrt{\alpha}\boldsymbol{u}$. This has much the same limiting behaviour as the superficial velocity, but now the flux and the dependent variable will be the same, so there is no need to evaluate the flux separately. However formulating the other equations in the system, in particular the equation for α (or more likely $\sqrt{\alpha}$) will be difficult – possibly non-conservative.

Even where the flux is not a problem, evaluating the superficial velocity introduces

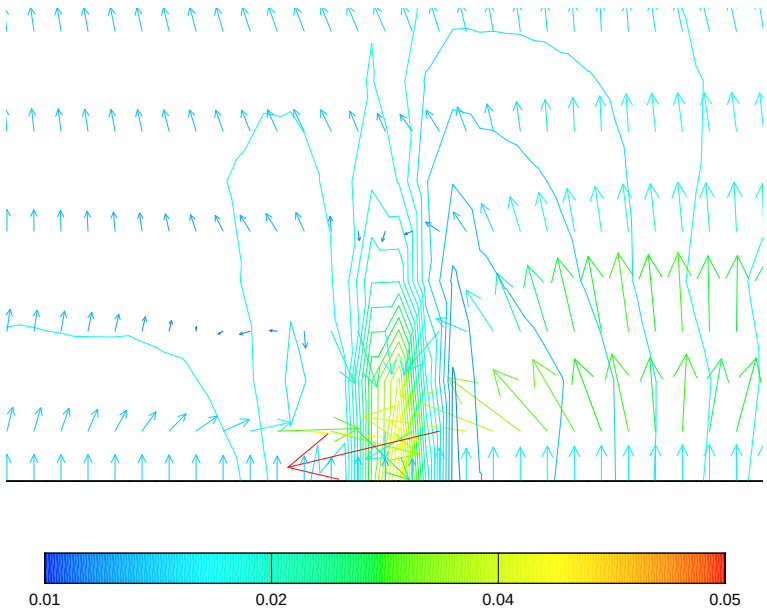

Figure 5: Illustration of the failure mode of superficialFoam. The case being calculated is the INPT ducted shear flow case discussed in more detail in section 7.5. Velocity vectors are shown superimposed on isolines of phase fraction : in the vicinity of the zero of phase fraction large random vectors are generated.

an added level of complexity. Since it is the real velocities (in fact largely the relative velocity) that determine the physics, and these are derived quantities, there is always the possibility of instability occurring in the algorithm. In particular the derivation of the partial elimination algorithm for the superficial velocity approach demonstrates the added complexity of this approach. The advantages of the algorithm seem unfortunately to be outweighed by these disadvantages.

9 Further Work.

The Weller algorithm has demonstrated important advances in stability on BRITE-II project test cases, and it can be used to solve cases such as phase separation which are well beyond the capabilities of any of the other algorithms (and beyond the remit of the project as well). The relevant issue as regards stability is the reformulation of the momentum equation : the bounding of the α equation is important for physical reasons, whilst the pressure update algorithm – the use of cell-integration techniques to make a colocated mesh code behave like a staggered mesh code – becomes critical where there are very steep density gradients (such as where there are phase interfaces). The next step in developing the algorithm is to evaluate the costs and benefits of implementing these various elements in STAR. In particular, there are other ways of ensuring the boundedness of α alternative to the way outlined here, which need to be evaluated, and the costs and

benefits of the cell integration need to be adequately weighed up. Some of these issues will be dealt with in a future report.

9.1 Block Solution Approaches

Another approach which has been (briefly) considered is that of block solution. All the approaches currently under investigation involve segregated solution of the equations. A segregated solution technique must favour one phase equation over another, by solving it first. However the two momentum equations are strongly coupled, especially through the drag term. If this term is implemented explicitly, then the coupling can lead to oscillation as $\boldsymbol{v}_a$ is driven towards $\boldsymbol{v}_b$ and vice versa. The Partial Elimination method attempts to treat the drag terms in a more implicit way by eliminating completely one of the phase velocities from each of the phase momentum equations, at the expense of introducing coupling in other terms. This is accomplished by splitting up the linearised drag term, so that one of the phase velocities can be included implicitly, and then rearanging to eliminate the other phase velocity. An extension of this would be to introduce block solution of the momentum equations, which enables the drag term to be treated implicity. (The only other way of achieving this is to solve an equation for $\boldsymbol{u}_r$, as discussed above).

There are several ways in which block solution could be used. The least computationally expensive would be to use a 2-component block solver to solve simultaneously individual components of the individual momentum equations. Thus $u_{a,x}$ and $u_{b,x}$ would be solved as a system. This is a direct extension of the partial elimination technique, in that if the drag law is linear in $\boldsymbol{u}_a - \boldsymbol{u}_b$ then this can make the coupling essentially implicit in each equation component. However it is not clear how an algorithm with such a strong component-wise coupling will behave for a generic 3-d flow. In particular, the interfacial force coupling between the phase momenta equations is not this straightforward, with the lift term introducing curls and cross-products between the phase velocities, and thus coupling between components, not just between the phases. Although the drag force is probably the largest individual contribution to the coupling, it is not obviously overwhelming. Lathouwers and van den Akker have shown this approach to work[16], using a 2-component block solver to solve the velocity components, and also to solve the pressure-phase fraction system simultaneously (which may help to achieve boundedness and convergence on α). However it should be noted that they have simplified the equation set grossly, by assuming that the only forces acting in the system are due to a (linearised) drag law, and ignoring lift, virtual mass, and history effects.

A more complex use of block solution would be to solve the 6-component momentum system as a unit. This would enable drag and lift to be treated implicitly, but would be computationally expensive, as would solving the whole system of equations as one unit. The major factor to be considered in terms of implementing block solvers in general would be the effort involved in writing one. The standard approach for a block solver (say for two scalars A and B) is to create a vector $\{A_i, B_i\}$ of the values, leading to a large matrix which is still sparse, but which has a much more complex structure than is normal in fluid dynamics. In particular, cross terms in the equations lead to new off-diagonal

entries. In fact the matrix for this case has a 4-block structure : top left block representing $A-A$ coupling, bottom right $B-B$ coupling, and the other two blocks representing the cross-coupling (see [16], figure 1, for an illustration of this). Not only is the structure of the equation much more complex, there is no guarantee that it is diagonally dominant. Straightforward adaption of a preexisting code is not really practical ; the solvers have to be completely rewritten, as does the addressing. This does not leave much of most CFD codes left, so it is probably no more difficult to start from scratch and write a simple (Cartesian, structured-mesh, possibly 2-d) code to investigate the applicability of the proposed method. Implementing block solution in FOAM presents another set of challenges, in particular the problem of finding a formulation that fits with the design methodology of tensor fields and equation objects. This would be possible, involving the introduction of a new class of `fvBlockMatrix` object which understands the coupling between the dependent variables, but again, it would involve a lot of work. However one benefit of this approach would be that it would permit a different approach to the block matrix formulation to be taken. If instead of formulating the variable vector as $\{A_i, B_i\}$, we write it as a vector of dyads $\{(A_i, B_i)_i\}$, then the matrix retains its previous form, but with entries being 2×2 (or in general $n \times n$) submatrices. Inversion of the submatrices could be made direct (for the 2×2 case, it could be written out explicitly) within the overall ICCG algorithm. This could well turn out to be a more effective way of solving the block system than the traditional one, since the matrix retains its diagonal dominance and is readilly solveable, and all the solution problems will be in the inversion of the submatrix, which is direct anyway. Moreover this could be implemented in a fairly straightforward way by templating the FOAM solvers.

10 Conclusions

The development of a more stable multiphase flow algorithm is the single most important task for the BRITE-III project. Its absence would render the rest of the project irrelevant, since there would be little point in developing new modelling for the physics if the resulting equations could not be solved. Moreover from an industrial point of view, a robust algorithm would be of enormous use. Such an algorithm must be able to cope with extremes of phase fraction, since even if the global phase fraction is at a reasonable level ($\alpha \sim 0.25$) physical effects will result in phase redistribution. The algorithm developed in the previous project is simply not adequate for the task.

Two algorithms have been investigated in detail here. The superficial velocity approach showed initial signs of promise. By solving for $\boldsymbol{U}_s = \alpha \boldsymbol{u}$ the problem with the behaviour at $\alpha \to 0$ is removed from the pressure equation and potentially can be solved separately. Moreover one is solving for a well-defined quantity at all points in space, since $\boldsymbol{U}_s = 0$ where $\alpha = 0$. However the algorithm is complicated by the fact that $\boldsymbol{u}$ is now a derived quantity, and its extensive use in the physics makes the algorithm much more convoluted than the previous one. Any evaluation of $\boldsymbol{U}_s/\alpha$ is complicated by the fact that we do not know the limit of this quantity when $\alpha \to 0$. If anything this algorithm is more sensitive

to instabilities where the phase fraction $\alpha \to 0$.

The second algorithm is that of Weller. It represents a significant departure from the previous methodology. Rearangement of the momentum equations allows the α-dependence to be concentrated in terms of the form $\frac{\nabla\alpha}{\alpha}\nabla \boldsymbol{u} \to 0$ as $\alpha \to 0$, which can thus be dealt with in the usual manner. The other aspect of the algorithm is the direct solution for the flux rather than for the velocity, i.e. the algorithm behaves as if it were on a staggered mesh rather than a colocated one. This does appear to be important for the stabilisation of the drag term. The current formulation solves for the individual phase velocities : however the methodology could be adapted for a mixture formulation. This could prove valuable when considering turbulence modelling, which is clearly the next issue to be resolved in the project.

References

[1] Weller, H. G., Tabor, G., Jasak, H., and Fureby, C.: "A Tensorial Approach to Computational Continuum Mechanics using Object Orientated Techniques", *Computers in Physics*, 12(6):??, 1998.

[2] Hill, D. P.: *The Computer Simulation of Dispersed Two-Phase Flow*, PhD thesis, University of London, Imperial College, 1998.

[3] Drew, D. A.: "Mathematical Modelling of Two-Phase Flow", *Ann.Rev.Fluid Mech.*, 15:261 – 291, 1983.

[4] Dopazo, C.: "On conditional averages for intermittent turbulent flows", *J. Fluid Mech.*, 81:433 – 438, 1977.

[5] Lahey, R. P.: "A CFD Analysis of Multidimensional Two-Phase Flow and Heat Transfer using a Four Field Model", Lecture given at IC.

[6] Stewart, H. B. and Wendroff, B.: "Two-Phase Flow : Models and Methods", *J.Comp.Phys*, 56:363 – 409, 1984.

[7] Gidaspow, D., Lyczkowski, R. W., Solbrig, C. W., Hughes, E. D., and Mortensen, G. A.: "Characteristics of Unsteady One-Dimensional Two-Phase Flow", *Trans.Amer.Nuc.Soc*, 17:249 – 250, 1973.

[8] Kashiwa, B. A. and Rauenzahn, R. M.: "A Multimaterial Formalism", In *Numerical Methods in Multiphase Flows*, volume FED 185, pages 149 – 157. ASME, 1994.

[9] Kashiwa, B. A., Padial, N. T., Rauenzahn, R. M., and VanderHeyden, W. B.: "A Cell-Centered ICE method for Multiphase Flow Simulations", In *Numerical Methods in Multiphase Flows*, volume FED 185, pages 159 – 167. ASME, 1994.

[10] Kumar, S., VanderHeyden, W. B., Devanathan, N., Padial, N. T., Duduković, M. P., and Kashiwa, B. A.: "Numerical Simulation and Experimental Verification of the Gas-Liquid Flow in Bubble Columns", In *Industrial Mixing Fundamentals with Applications*, volume 91#305, pages 11 – 19. AIChE., 1995.

[11] Wang, D. M., Issa, R. I., and Gosman, A. D.: "Numerical Prediction of Dispersed Bubbly Flow in a Sudden Enlargement", In *International Conference on Two-Phase Flow, Lake Dahoe, Nevada*, Proceedings of ASME. ASME, 1994.

[12] de Oliviera, P.: *Computer Modelling of Multidimensional Multiphase Flow and Applications to T-junctions*, PhD thesis, University of London, 1992.

[13] Weller, H. G.: "Report on Multiphase CFD using a Pressure-Implicit Flux Algorithm", Technical report, In preparation for Computational Dynamics., 1999.

[14] Roig, V., Suzanne, C., and Masbernat, L.: "Experimental Investigation of a Turbulent Bubbly Mixing Layer", *Int.J.Multiphase Flow*, 24(1):35 – 54, 1998.

[15] Roig, V.: *Zone de Melange D'ecoulements Diphasiques à Bulles*, PhD thesis, INPT, 1993.

[16] Lathouwers, D. and van den Akker, H. E. A.: "A Numerical Method for the Solution of Two-Fluid Model Equations", In *Fluids Engineering Division Conference*, volume 236, pages 121 – 126. ASME, 1996.