INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2005; **48**:1123–1147
Published online 28 April 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.981

A study of time integration schemes for the numerical modelling of free surface flows

Ahamadi Malidi[§], Steven Dufour[*,†,‡] and Donatien N'dri[§]

Département de mathématiques et de génie industriel, École Polytechnique de Montréal, C.P. 6079, succ. Centre-ville, Montréal (Québec) Canada, H3C 3A7

SUMMARY

A semi-discrete finite element methodology for the modelling of transient free surface flows in the context of Eulerian interface capturing is proposed. The focus of this study is put on the choice of an appropriate time integration strategy for the accurate modelling of the dynamics of free surfaces and of interfacial physics. It is composed of an adaptive time integration scheme for the Navier–Stokes equations, and of the implicit midpoint rule for the transport equation of the Eulerian marker variable. The adaptive scheme allows the automatic determination of a time-step size that follows the physics of the problem under study, which facilitates the accurate modelling of stiff free surface flows. It is shown that the implicit midpoint rule reduces mass loss for each fluid. Various free surface flow problems are studied to verify and validate the proposed time integration strategy. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: free surface flows; interface capturing; time integration schemes; finite element methods

1. INTRODUCTION

The numerical modelling of free surface flows is a challenging problem. Since the position of the interface separating the fluids is *a priori* unknown, specific algorithms must be used to determine its position. Several strategies are described in the literature for performing

*Correspondence to: Steven Dufour, Département de mathématiques et de génie industriel, École Polytechnique de Montréal, C.P. 6079, succ. Centre-ville, Montréal (Québec) Canada H3C 3A7.
†E-mail: steven.dufour@polymtl.ca
‡Assistant Professor.
§Research Associate.

Contract/grant sponsor: Auto21 Network of Centres of Excellence of Canada
Contract/grant sponsor: Natural Sciences and Engineering Research Council of Canada
Contract/grant sponsor: Fonds québécois de la recherche sur la nature et les technologies

Received 1 December 2004
Revised 18 March 2005
Accepted 21 March 2005

computer simulations of multifluid flows [1]. In this study, we are particularly interested in a family of methods based on a marker variable that is used to identify the region occupied by each fluid. The zero thickness interface is traded for a transition zone of the marker. This variable is then transported by the velocity field of each fluid, by solving a transport equation of the marker on a fixed grid. This Eulerian approach, which is also known as free surface capturing, is popular for modelling complex multifluid flows. Among the most popular free surface capturing methods we can mention the *volume of fluid* method [2], the *level set* method [3], and the *pseudo-concentration* method [4], to name a few. This approach is better suited for modelling the evolution of free surfaces submitted to large and complex deformations. Modelling multiple interface flows is not a complicated matter, since the interface is not treated as an explicit computational entity. Free surface splitting and reconnection are handled implicitly. However, this strategy is known to be less accurate if the numerical strategy is not well chosen. This is due to the uncertainty inherent in the use of the region of transition of the marker variable to locate the free surface, which makes it impossible to impose interfacial boundary conditions explicitly. Free surface capturing methods are also known to suffer from numerical diffusion or oscillations [5]. This results in the deformation of the region of transition of the marker variable, and therefore to mass-conservation problems and to an inaccurate modelling of interfacial physics. It is therefore important to choose the right numerical strategy in order to perform an accurate transport of the free surfaces, and to conserve mass for each fluid.

Several components in a numerical methodology influence the accuracy of a free surface capturing strategy. An inaccurate discretization may lead to an imbalance between the viscous forces and the capillary forces, which will translate into numerical instabilities [6]. It is well known that a 'good' mesh will provide an accurate discretization of the variables of the problem [6]. We also observe that the elliptic solvers introduce non-physical oscillations or diffusion when the solution of a partial differential equation contains a discontinuity or an abrupt variation [7]. Finally, the choice of an appropriate time integration scheme, to discretize the transient term of the equations, is important to help conserve mass for each fluid. To our knowledge, we do not know of a study relating the discretization in time to mass-conservation problems and interfacial instabilities when modelling free surface flows using an Eulerian approach.

Authors studying multifluid flows use various time integration schemes. Examples are: the *backward Euler* scheme for mold filling problems [8]; the *trapezoid rule* (TR) for sloshing problems [9]; and the *backward differentiation formula* (BDF) (Gear scheme) for drops dynamics problems [10] to name a few. But these studies do not mention the reasons for these choices. We are proposing to discuss various strategies for the discretization of the transient term of the partial differential equations involved in the modelling of free surface flows in an Eulerian context.

After introducing the semi-discrete finite element methodology used for modelling free surface flows using interface capturing, we first look at the time discretization of the Navier–Stokes equations, which is important for modelling stiff free surface flow problems. We then study the influence of the discretization of the transient term of the transport equation on mass conservation. We finally propose an overall time integration strategy for studying free surface flows. Various free surface flow problems are studied to verify and validate the proposed numerical strategy.

2. MODELLING OF THE PROBLEM

Consider the free surface flow problem illustrated in Figure 1. The region occupied by the first fluid is denoted Ω_1. The second fluid occupies the region Ω_2, and since the fluids are immiscible, $\Omega_1 \cap \Omega_2 = \emptyset$. The computational domain can therefore be defined as $\Omega = \Omega_1 \cup \Omega_2$. The interface separating the two fluids is denoted by $\mathcal{S}$. The unit normal to the interface is denoted $\mathbf{n}_\mathcal{S}$.

The flow of each fluid is modelled using the equations expressing conservation of mass and momentum. Therefore, the equations

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

and

$$\rho \frac{\partial \boldsymbol{u}}{\partial t} + \rho(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = \nabla \cdot \boldsymbol{\sigma} \tag{2}$$

must be solved for fluids 1 and 2, i.e. on Ω_1 and Ω_2, respectively, where

$$\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau}$$

is the Cauchy stress tensor, and the density of the fluid is denoted ρ. The extra-stress tensor $\boldsymbol{\tau}$ is related to the velocity field by the relation

$$\boldsymbol{\tau} = 2\mu\dot{\boldsymbol{\gamma}}(\boldsymbol{u})$$

where μ is the viscosity of the fluid, and the rate-of-strain tensor $\dot{\boldsymbol{\gamma}}$ is defined by

$$\dot{\boldsymbol{\gamma}}(\boldsymbol{u}) = \tfrac{1}{2}(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^{\mathrm{T}})$$

The dependent variables of the problem will therefore be denoted $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ for the velocity field of fluids 1 and 2, and the pressure of the fluids will be denoted p_1 and p_2.

Boundary conditions can either be of the essential type (Dirichlet boundary conditions): $\boldsymbol{u}_i = \boldsymbol{u}_{\partial\Omega}$ or of the natural type (Neumann boundary conditions): $\boldsymbol{\sigma}_i \cdot \mathbf{n} = \boldsymbol{t}_{\partial\Omega}$, where $i = 1, 2$ in our case. The conditions at the interface $\mathcal{S}$ are [1]: the continuity of normal velocities

$$\boldsymbol{u}_1 \cdot \mathbf{n}_\mathcal{S} = \boldsymbol{u}_2 \cdot \mathbf{n}_\mathcal{S} = \boldsymbol{u}_\mathcal{S} \cdot \mathbf{n}_\mathcal{S}$$

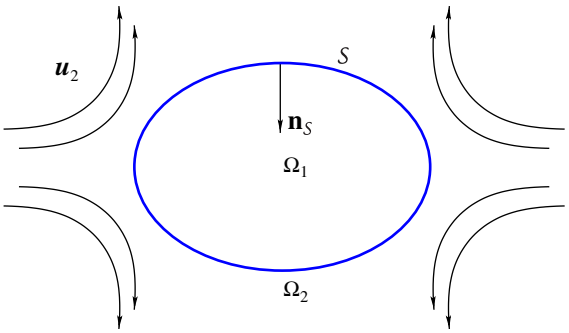

Figure 1. A free surface flow problem.

the continuity of tangential velocities

$$\boldsymbol{u}_1 \cdot \mathbf{t}_{\mathcal{S}} = \boldsymbol{u}_2 \cdot \mathbf{t}_{\mathcal{S}}$$

and the force balance at the interface

$$(\boldsymbol{\sigma}_2 - \boldsymbol{\sigma}_1) \cdot \mathbf{n}_{\mathcal{S}} = \alpha\kappa\mathbf{n}_{\mathcal{S}}$$

where $\boldsymbol{u}_{\mathcal{S}}$ is the velocity of the interface, $\alpha > 0$ is the surface tension coefficient of fluids 1 and 2, and κ is the local curvature of the interface. Finally, with no loss of generality, let $\mathbf{n}_{\mathcal{S}}$ point in fluid 1, as illustrated in Figure 1, and the radius of curvature be positive if the centre of curvature is in fluid 1. An appropriate initial divergence free velocity field and pressure distribution for each fluid

$$(\boldsymbol{u}_i(\boldsymbol{x}, t=0),\ p_i(\boldsymbol{x}, t=0)) = (\boldsymbol{u}_0(\boldsymbol{x}),\ p_0(\boldsymbol{x}))$$

completes the definition of the Navier–Stokes problem.

The proposed approach to capture free surfaces is purely Eulerian. The technique used is known as the *pseudo-concentration* method [4], where a 'colour function' defines Ω_1 and Ω_2. This marker variable is transported with the fluid flow by solving a transport equation of the pseudo-concentration F

$$\frac{\partial F}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} F = 0 \tag{3}$$

Since this equation is hyperbolic, boundary conditions on F must be prescribed at the inflow of Ω, i.e. $\Gamma^- = \{\boldsymbol{x} \in \partial\Omega \mid \boldsymbol{u} \cdot \mathbf{n} < 0\}$, where $\mathbf{n}$ is the outward normal to Ω. The initial condition on F is given by the regions occupied by each fluid:

$$F(\boldsymbol{x}, t=0) = \begin{cases} 1 & \text{if } \boldsymbol{x} \in \Omega_1 \\ 0 & \text{if } \boldsymbol{x} \in \Omega_2 \end{cases}$$

The position of the interface $\mathcal{S}$ is identified by the values of $\boldsymbol{x}$ such that $F(\boldsymbol{x}, t) = \frac{1}{2}$.

3. GENERAL NUMERICAL METHODOLOGY

The *Galerkin variational formulation* is used to solve the Navier–Stokes equations (2) on all the computational domain, using the pseudo-concentration to evaluate the physical parameters of each fluid. As an example, we obtain the viscosity on Ω using the expression

$$\mu(F) = \mu_2 + (\mu_1 - \mu_2)F(\boldsymbol{x}, t) \quad \forall \boldsymbol{x} \in \Omega \tag{4}$$

The second-order accurate *Crouzeix–Raviart* element combination is used to discretize velocity and pressure. The incompressibility constraint (1) is satisfied with the use of the iterative penalization technique of *Uzawa's algorithm* [11]. The resulting algebraic system of non-linear equations is solved using Newton's method. The *streamline upwind/Petrov–Galerkin* (SUPG) formulation

$$\int_\Omega \left(\frac{\partial F}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} F \right) \varphi \, \mathrm{d}\Omega + \sum_K \int_K \left(\frac{\partial F}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} F \right) (\tau \boldsymbol{u} \cdot \boldsymbol{\nabla} \varphi) \, \mathrm{d}K = 0$$

is used to discretize the transport equation (3). A quadratic element is used to discretize the pseudo-concentration. The stabilization parameter τ is given by $h/2\|\boldsymbol{u}\|$, where h is the local mesh size.

The *continuum surface force* model [12] is used to include the influence of surface tension in the numerical model. Surface tension is considered as a volume force defined in the smooth region of transition of the Eulerian marker variable. Formally, the surface capillary force $\boldsymbol{f}_S = \alpha\kappa\mathbf{n}_S\delta_S$ is written as a volume force $\boldsymbol{f}_V = \alpha\kappa(F)\boldsymbol{\nabla}F$, defined in the region of transition of F. The Dirac delta function δ_S, defined on the interface S, is approximated numerically in the region of transition of F using $\|\boldsymbol{\nabla}F\|$, and the curvature κ is approximated using

$$\kappa = -\boldsymbol{\nabla}\cdot\mathbf{n}_S \approx -\boldsymbol{\nabla}\cdot\left(\frac{\boldsymbol{\nabla}F}{\|\boldsymbol{\nabla}F\|}\right)$$

The capillary force $\boldsymbol{f}_V$ is included in the Navier–Stokes equations (2) to yield

$$\rho\frac{\partial\boldsymbol{u}}{\partial t} + \rho(\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{u} = \boldsymbol{\nabla}\cdot\boldsymbol{\sigma} + \boldsymbol{f}_V \tag{5}$$

4. TIME INTEGRATION SCHEMES FOR THE NAVIER–STOKES EQUATIONS

4.1. Motivation: Laplace's problem

An accurate discretization of the velocity field of the fluids is important for an accurate advection of the pseudo-concentration (cf. Equation (3)). A well-known free surface flow problem, for which an inaccurate discretization of the transient term of the Navier–Stokes equations can be problematic, is *Laplace's problem*. The problem consists of a 'two-dimensional' drop of fluid of an arbitrary shape, an ellipse for example, in another fluid at rest. The capillary force between the two fluids induces a flow which makes the drop reach a topology that minimizes energy, i.e. a circle in two dimensions. The curvature of the steady-state drop and the jump in pressure should verify *Laplace's law*,

$$p_1 - p_2 = \alpha\kappa \tag{6}$$

This problem is popular for verifying numerical surface tension models in simulation codes.

Despite its simplicity, Laplace's problem is numerically challenging. It is well documented in References [13–18] that if the numerical strategy is not well chosen, numerically induced parasitic currents will appear in the vicinity of the free surface when $\|\boldsymbol{u}\| \to 0$ (cf. Figure 2(a)). This leads to an inaccurate discretization of the jump in pressure at the free surface (cf. Figure 2(b)), and eventually to a flow that does not reach a steady-state. Several hypotheses are brought forward in the literature in order to explain this phenomenon and to find a cure.

Kothe *et al.* [14] convolve the marker variable F with various smooth kernels to obtain a mollified colour function $\tilde{F}$. This leads to a more accurate computation of the unit normal to the free surface $\mathbf{n}_S$ and of the curvature κ, which in turn reduces parasitic currents. Other authors have improved the numerical discretization they were using in order to obtain a more accurate modelling of interfacial physics. This is the case of Popinet and Zaleski [15] who use higher accuracy discretizations of the Lagrangian representation of the free surface, and of the pressure, which leads to a reduction of parasitic currents. In a similar fashion, the

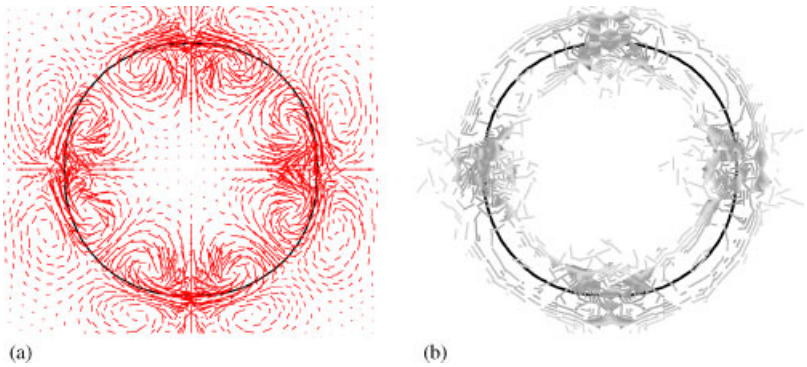

Figure 2. Numerical instabilities in modelling Laplace's problem: (a) parasitic currents; and (b) pressure contours.

PROST algorithm [17] was also devised to eliminate the spurious currents problem. In this algorithm, free surface capturing is performed with the VOF (volume of fluid) technique, using a least square fit of a quadratic surface to the colour function for each interface cell and its neighbours. The authors also implemented a higher-order advection scheme for the transport equation of the colour function. Hou *et al.* [19] rather consider Laplace's problem as a stiff problem. They devised a numerical procedure based on the boundary integral formulation to remove the stiffness due to surface tension. The numerical results found in these references show that parasitic currents are in fact reduced, but it is not clear if the steady-state is reached.

We have found that several aspects of these studies are important [20]. In particular, the fact that this problem is stiff is often overlooked, and we think, like Hou *et al.* [19], that this is the key to the accurate modelling of Laplace's problem. This led us to develop a methodology to tackle stiff free surface flow problems through the use of a time integration strategy which includes an adaptive scheme for the transient term of the Navier–Stokes equations. This strategy will allow us to modify the size of the time discretization in order to maintain the balance between the viscous forces and the capillary forces when approaching steady-state. Our hypothesis is that such a scheme will allow us to reach the steady-state of Laplace's problem with no parasitic currents at the free surface.

4.2. Adaptive time integration scheme

Adaptive time integration schemes for the Navier–Stokes equations are not new in the computational science literature (cf. Reference [21] for details). They allow the automatic computation of an 'optimal' *time-step size* Δt_n at every time step, in order to adapt the time discretization to follow the physics of the modelled problem. This strategy is a common practice for discretizing spatial variables. It is difficult to explain why it is not more popular in the CFD literature for the temporal variable. It is particularly absent from the free surface flow modelling literature.

The most popular adaptive time integration scheme is the *adaptive trapezoid rule* (ATR). Even if this scheme is devised to compute the appropriate time-step size in order to avoid the oscillations often seen with the standard TR, the ATR scheme can still exhibit wiggles

in some cases [21]. This is why we choose to use an *adaptive backward differentiation formulae* (ABDF), which is based on the constant-time-step size second-order accurate BDF, also known as Gear scheme. This choice is justified by the fact that the BDF scheme is known to damp oscillatory solutions.

The finite element discretization of velocity and pressure,

$$\boldsymbol{u}(\boldsymbol{x}) = \sum_{i=1}^{7} \boldsymbol{u}_i \varphi_i(\boldsymbol{x}) \quad \text{and} \quad p(\boldsymbol{x}) = \sum_{i=1}^{3} p_i \psi_i(\boldsymbol{x})$$

where $\varphi_i(\boldsymbol{x})$ and $\psi_i(\boldsymbol{x})$ are the finite element interpolation functions of the Crouzeix–Raviart element and the $\boldsymbol{u}_i$ and the p_i are the degrees of freedom of velocity and pressure on an element, are introduced in the elementary linearized weak forms of the conservation of mass (1) and momentum (2) equations, to yield an algebraic system of equations of the form (see Reference [22] for details)

$$\begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\vec{U}} \\ \dot{\vec{P}} \end{bmatrix} + \begin{bmatrix} A & B^{\mathrm{T}} \\ B & 0 \end{bmatrix} \begin{bmatrix} \vec{U} \\ \vec{P} \end{bmatrix} = \begin{bmatrix} \vec{F} \\ \vec{0} \end{bmatrix}$$

The vector $\vec{U}$ contains the degrees of freedom in velocity, $\vec{P}$ contains the degrees of freedom in pressure, M is the mass matrix of the transient term, A is the viscous term matrix and B is the discrete divergence matrix. But since we are using Uzawa's algorithm to enforce the incompressibility constraint, as described in Section 3, this linear system of equations reduces to

$$M\dot{\vec{U}} + A_r \vec{U} = F_r \tag{7}$$

(cf. Reference [11] for details). We therefore perform transient error control only on the degrees of freedom in velocity. The degrees of freedom in pressure will be updated by Uzawa's algorithm.

Following the development of Gresho *et al.* [21], we implemented an ABDF scheme for the discretization of the Navier–Stokes equations. Based on the second-order accurate BDF, the variable-time-step size BDF scheme can be expressed as

$$\dot{\vec{U}}_{n+1} = \frac{(2 + (\Delta t_{n-1}/\Delta t_n))\vec{U}_{n+1} - (2 + (\Delta t_{n-1}/\Delta t_n) + (\Delta t_n/\Delta t_{n-1}))\vec{U}_n + (\Delta t_n/\Delta t_{n-1})\vec{U}_{n-1}}{\Delta t_n + \Delta t_{n-1}} \tag{8}$$

Using Taylor's expansions

$$\vec{u}(t_{n\pm1}) = \vec{u}(t_n \pm \Delta t_n) \approx \vec{U}_{n\pm1} = \vec{U}_n \pm \Delta t_n \dot{\vec{U}}_n + \tfrac{1}{2}\Delta t_n^2 \ddot{\vec{U}}_n \pm \tfrac{1}{6}\Delta t_n^3 \dddot{\vec{U}}_n + \mathcal{O}(\Delta t_n^4) \tag{9}$$

in (8), we find the *local truncation error* of this variable-time-step size BDF scheme to be

$$\vec{U}_{n+1} - \vec{u}(t_{n+1}) = \frac{(\Delta t_n + \Delta t_{n-1})^2}{\Delta t_n(2\Delta t_n + \Delta t_{n-1})} \frac{\Delta t_n^3 \dddot{\vec{U}}_n}{6} + \mathcal{O}(\Delta t^4) \tag{10}$$

In order to be able to determine the next time-step size Δt_{n+1}, we need to express the local truncation error of the ABDF scheme in term of $\vec{U}_{n+1}$. We first need some kind of 'cheap'

predictor scheme with a local truncation error which is also in $\Delta t_n^3 \dddot{\vec{U}}_n$. The variable-time-step size *explicit midpoint rule* is such a candidate:

$$\vec{U}_{n+1}^{\,p} = \vec{U}_n + \left(1 + \frac{\Delta t_n}{\Delta t_{n-1}}\right) \Delta t_n \dot{\vec{U}}_n - \left(\frac{\Delta t_n}{\Delta t_{n-1}}\right)^2 (\vec{U}_n - \vec{U}_{n-1}) \tag{11}$$

with a local truncation error of

$$\vec{U}_{n+1}^{\,p} - \vec{u}(t_{n+1}) = -\left(1 + \frac{\Delta t_n}{\Delta t_{n-1}}\right) \frac{\Delta t_n^3 \dddot{\vec{U}}_n}{6} + \mathcal{O}(\Delta t^4) \tag{12}$$

which can again be obtained by using (9) in (11).

We can now express the local truncation error of the ABDF scheme in terms of only $\vec{U}_{n+1}$, $\vec{U}_{n+1}^{\,p}$ and Δt_n. Subtracting Equation (12) from Equation (10), we obtain

$$\vec{U}_{n+1} - \vec{U}_{n+1}^{\,p} \approx \left[\frac{(\Delta t_n + \Delta t_{n-1})^2}{\Delta t_n (2\Delta t_n + \Delta t_{n-1})} + \left(1 + \frac{\Delta t_n}{\Delta t_{n-1}}\right)\right] \frac{\Delta t_n^3 \dddot{\vec{U}}_n}{6}$$

which can be written as

$$\frac{\Delta t_n^3 \dddot{\vec{U}}_n}{6} \approx (\vec{U}_{n+1} - \vec{U}_{n+1}^{\,p}) \Big/ \left[\frac{(\Delta t_n + \Delta t_{n-1})^2}{\Delta t_n (2\Delta t_n + \Delta t_{n-1})} + \left(1 + \frac{\Delta t_n}{\Delta t_{n-1}}\right)\right]$$

in order to rewrite expression (10) as

$$\vec{U}_{n+1} - \vec{u}(t_{n+1}) \approx \frac{(\Delta t_n + \Delta t_{n-1})^2}{\Delta t_n (2\Delta t_n + \Delta t_{n-1})} \left\{ (\vec{U}_{n+1} - \vec{U}_{n+1}^{\,p}) \Big/ \left[\frac{(\Delta t_n + \Delta t_{n-1})^2}{\Delta t_n (2\Delta t_n + \Delta t_{n-1})} + \frac{\Delta t_n}{\Delta t_{n-1}} + 1\right] \right\}$$

After simplifying this expression, we can express the local truncation error of the ABDF scheme, which will now be denoted $\vec{d}_n$, as

$$\vec{d}_n = \frac{(1 + (\Delta t_{n-1}/\Delta t_n))(\vec{U}_{n+1} - \vec{U}_{n+1}^{\,p})}{2 + (\Delta t_{n-1}/\Delta t_n) + 2(\Delta t_n/\Delta t_{n-1})} \tag{13}$$

The new time-step size is then obtained by first observing that the local truncation error (10) of (8) gives us:

$$\frac{\vec{d}_{n+1}}{\vec{d}_n} = \frac{\vec{U}_{n+2} - \vec{u}(t_{n+2})}{U_{n+1} - \vec{u}(t_{n+1})} \approx \frac{\Delta t_{n+1}^3 \dddot{\vec{U}}_{n+1}}{\Delta t_n^3 \dddot{\vec{U}}_n} \approx \left(\frac{\Delta t_{n+1}}{\Delta t_n}\right)^3 \tag{14}$$

Since we want to control the size of the local truncation error at the next time step, i.e.

$$\|\vec{d}_{n+1}\| < \varepsilon \tag{15}$$

where ε is a *user-specified tolerance*, we substitute Equation (14) in expression (15) to obtain

$$\Delta t_{n+1} = \Delta t_n \left(\frac{\varepsilon}{\|\vec{d}_n\|} \right)^{1/3} \tag{16}$$

where $\|\vec{d}_n\|$ is computed using expression (13). The tolerance ε must not be chosen too large, which could lead to inaccurate results and possibly to an unstable solution similar to the one illustrated in Figure 2, or too small, which would make the simulation too costly. From our experience, this tolerance is application dependent and some experimentation is necessary to find an appropriate value for a given problem. We also observed that if the norm used in the computation of (16) is not well chosen, it can lead to an adaptive time integration scheme which does not perform well. We chose a norm suggested in Reference [21]

$$\|\vec{d}_n\|^2 = \frac{1}{N_u + N_v} \left[\sum_{j=1}^{N_u} \left(\frac{d^u_{n+1,j}}{|u_{n+1,j}| + u_0} \right)^2 + \sum_{j=1}^{N_v} \left(\frac{d^v_{n+1,j}}{|v_{n+1,j}| + v_0} \right)^2 \right]$$

where u and v are the x and y components of velocity, N_u and N_v are the number of degrees of freedom of each component in velocity, $d^u_{n+1,j}$ and $d^v_{n+1,j}$ are the jth components of $\vec{d}_{n+1}$ of each component in velocity and $\boldsymbol{u}_0 = (u_0, v_0)$ are reference velocities.

In summary, the adaptive time integration scheme consists of the following steps.

Given ε, Δt_0, $\vec{U}_0$, Δt_1 and $\vec{U}_1$ which can initially be computed using the 'standard' BDF scheme, for $n = 1, 2, 3, \ldots$:

- compute $\vec{U}^p_{n+1}$ using (11);
- solve (7) using (8);
- compute Δt_{n+1} using (16).

4.3. Verification: the Von Karman vortex street

A well-known benchmark for the verification of the numerical modelling of the transient Navier–Stokes equations is the problem of the *Von Karman vortex street behind a circular cylinder* [23]. It consists of a rectangular geometry containing a circular obstacle centred at the origin of the system of coordinates, as illustrated in Figure 3. The boundary condition $\boldsymbol{u} = (1, 0)$ (freestream velocity) is imposed on $\partial\Omega$, except for the outflow where we have a *free* boundary condition. The Navier–Stokes equations are non-dimensionalized as

$$\frac{1}{St} \frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \boldsymbol{\nabla})\boldsymbol{u} = \frac{1}{Re} \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}$$

where the *Reynolds number* $Re = \rho_0 u_0 L_0 / \mu_0 = 100$. We therefore choose the reference density to be $\rho_0 = 1$, the reference velocity to be $u_0 = \|(1, 0)\|_2 = 1$, the reference length to be $L_0 = 1$ and the reference viscosity to be $\mu_0 = 0.01$. The Strouhal number $St = u_0 t_0 / L_0$ also appears in the equations and is set to 1.

The paper of Engelman and Jamnia [23] includes an important quantity of qualitative data and some quantitative results. They first give the time history of various quantities at several points in the geometry. We will limit our comparison to the x and y components of velocity at $(x, y) = (4, 0)$, which are illustrated in Figure 4. A behaviour which is very similar to the

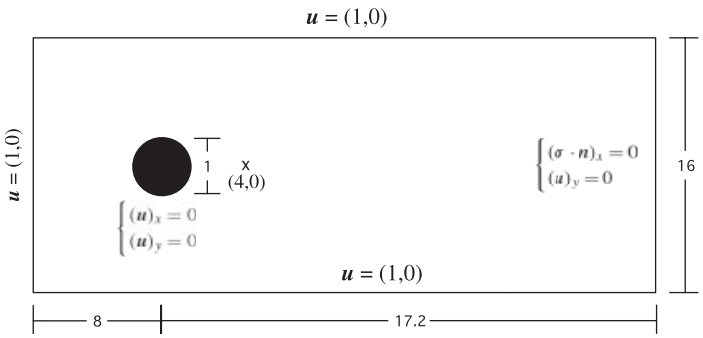

Figure 3. Von Karman vortex street problem.

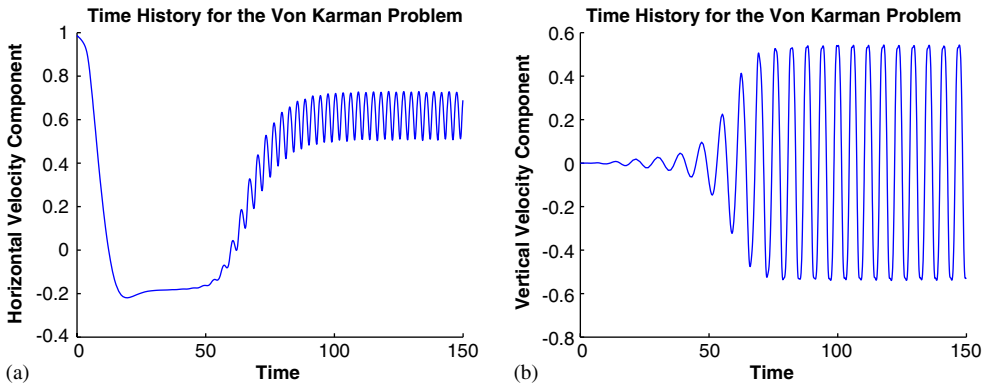

Figure 4. Time history of the velocity components at $(x, y) = (4, 0)$: (a) horizontal velocity; and (b) vertical velocity.

results of Reference [23] can be observed in the region where the periodic regime is reached. Figure 5 illustrates the time-step size Δt_n obtained from the ABDF scheme, which is larger for the transient phase of the simulation, and it gets smaller when the periodic regime is reached. This behaviour can be explained by the regularity of the solution. During the transient phase of the flow, the smoothly varying velocity field leads to larger time-step sizes. In the periodic regime, the variations of the velocity field are captured by the transient error estimator, which leads to irregular variations of smaller time-step sizes. The average time-step size, for the periodic regime, is of the order of $\Delta t_n \approx 0.096$, for a tolerance of $\varepsilon = 0.0005$. Engelman and Jamnia use an ATR scheme with the same tolerance. They obtain a time-step size of the order of $\Delta t_n \approx 0.269$ in the periodic regime. This difference can be explained by the different schemes used and by the choice of the norm for computing $\|\vec{d}_n\|$.

From a quantitative perspective, a time of reference, t_{ref}, is defined in Reference [23] as a time at which the y component of velocity goes, in the periodic regime, from a negative to a positive value at $(x, y) = (4, 0)$. Engelman and Jamnia measured the period for one shedding (during which two vortices are shed, one from the top and one from the bottom) to be

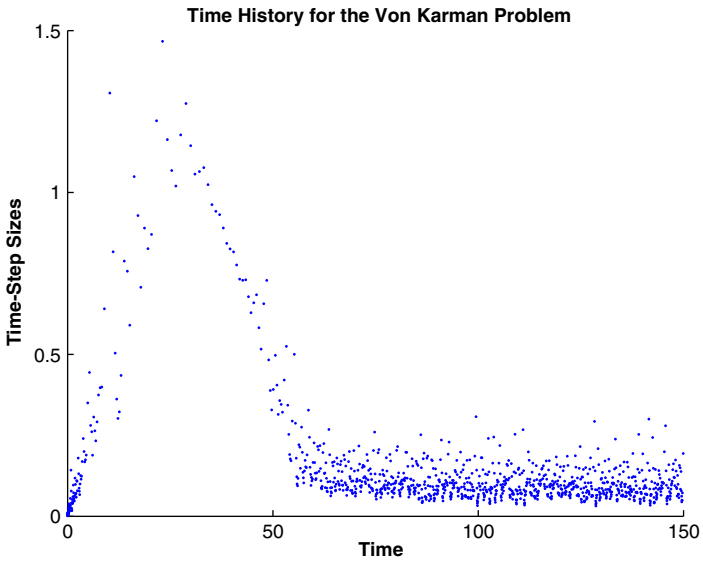

Figure 5. Time-step sizes computed by the ABDF scheme for the Von Karman problem.

$\tau = 5.80$. The wavelength of their simulations was $\lambda = 5.32$. Based on our results, we find the same period and wavelength.

Finally, Figure 6 illustrates plots of various quantities at $t = t_{\text{ref}}$, in order to compare qualitatively our results with the ones of Engelman and Jamnia. We can again see that our figures are very similar to the ones of the benchmark paper.

4.4. Laplace's problem revisited

Laplace's problem will now be studied using the ABDF time integration scheme for the Navier–Stokes equations. The discretization of the transient term of the transport equation (3), which is not a problem in this case, will be discussed in the next section.

Let us consider a '2-D' drop of fluid of area 4π in a geometry with dimensions $[-3,3] \times [-3,3]$. As an example, the initial drop can have the shape of an ellipse described by $y = \pm B\sqrt{1-(x^2/L^2)}$, such that $LB = 4$, where L and B are the longest and shortest semi-axes of the ellipse. The fluids are initially at rest, with same density and viscosity. Let us suppose that the surface tension coefficient between the fluids is $\alpha = 2$. If $L \neq B$, the capillary force will generate a flow which will deform the drop until it reaches the shape of a circle of radius 2 in this case. According to Laplace's law (6), we should observe a jump in pressure of $p_1 - p_2 = 1$.

Using the methodology described in Section 3 with the ABDF scheme, and with a structured mesh containing 140×140 elements, the numerical modelling of Laplace's problem gives us the time-step sizes illustrated in Figure 7. We observe a behaviour similar to the one illustrated in Figure 5 for the Von Karman problem, i.e. larger time-step sizes during the transient phase of the simulation, and smaller time-step sizes when the drop is reaching its steady-state shape. We, however, do not have the apparent 'noise' observed before, but we still observe small

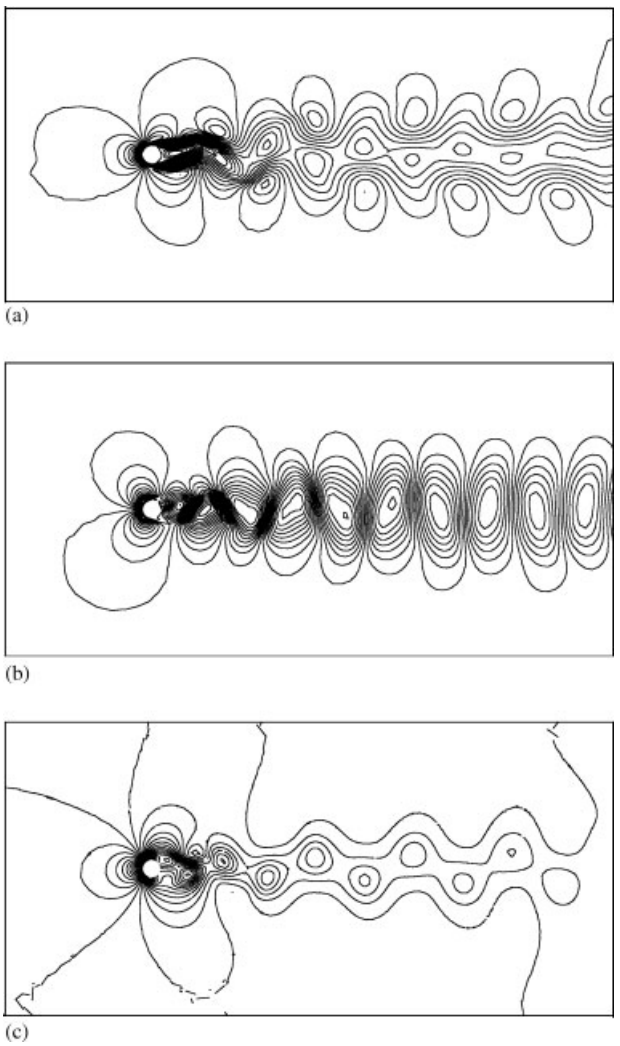

Figure 6. Plots at $t = t_{\text{ref}}$ for the Von Karman vortex street problem: (a) horizontal velocity component; (b) vertical velocity component; and (c) pressure.

variations of Δt_n. This can be explained by the fact that when the drop reaches its steady-state shape, we still observe very small deformations of the free surface that are captured by the transient error estimator. Figure 8 illustrates the time evolution of the jump in pressure. The fact that $(p_1 - p_2) \to 1$ is an indication that the steady-state is reached. Streamlines are also useful to determine if the steady-state of a free surface flow is reached. It can be seen in Figure 9(a) that the streamlines do not cross the free surface, which is another good indication that the numerical strategy allows us to reach the steady-state of this problem, with no parasitic currents. Another indication that the surface tension-driven flow simulation is performed accurately is the smooth transition of pressure, with no irregularities, as illustrated in Figure 9(b).

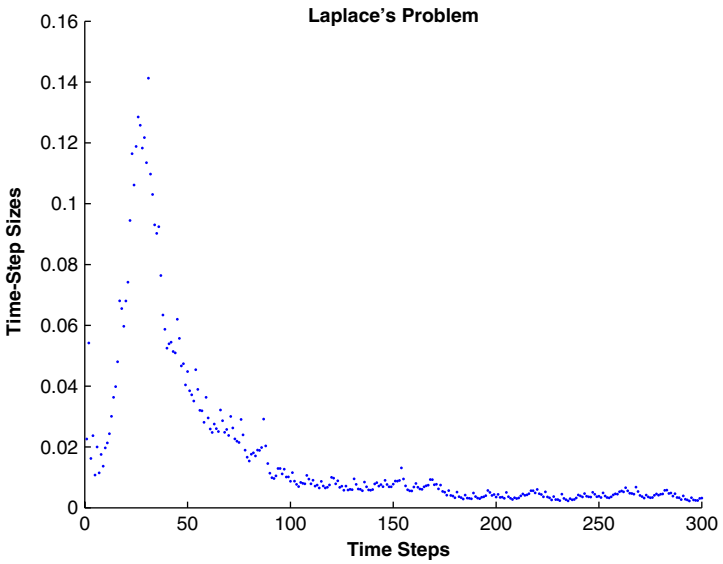

Figure 7. Time-step sizes computed by the ABDF scheme for Laplace's problem.

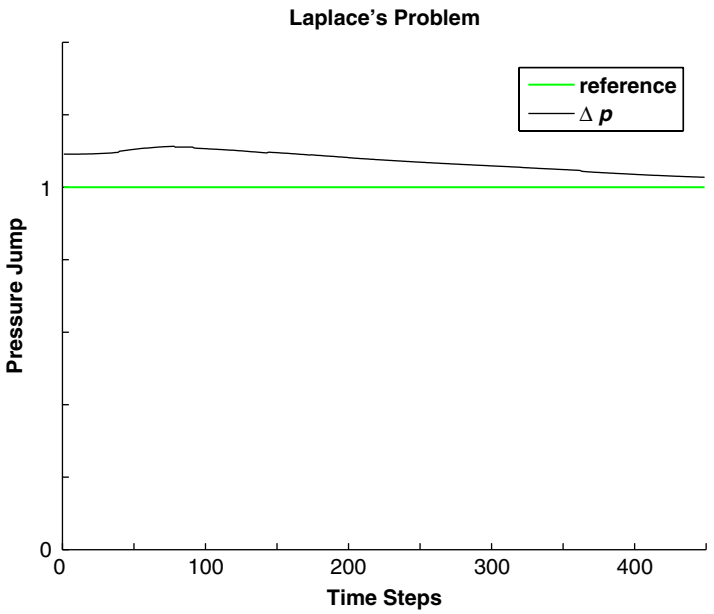

Figure 8. Time evolution of the jump in pressure for Laplace's problem.

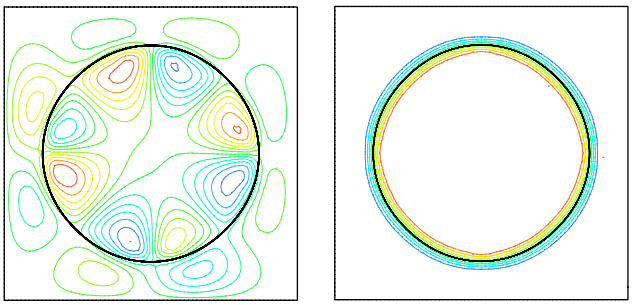

Figure 9. Steady-state of Laplace's problem: (a) streamlines; and (b) pressure contours.

5. TIME INTEGRATION SCHEMES FOR THE TRANSPORT EQUATION

5.1. *Motivation: the drop advection problem*

The choice of an appropriate time integration scheme for the transport equation (3) is important in our numerical methodology, since it directly influences mass conservation. This can be illustrated with the *drop advection problem* (cf. Figure 10). A '2-D drop', initially at rest, lies in another fluid which is submitted to the flow described by $\boldsymbol{u}(\boldsymbol{x},t)=(1,0)$. It is well known that the advection of the marker variable, which represents the initial shape of the drop, will loose its initial shape, since it is subject to numerical oscillations or diffusion. This is caused, among other reasons, by the time integration scheme used to discretize the transient term of the transport equation (3).

5.2. *Numerical comparison*

Let us consider three popular *implicit*, *second-order accurate*, *A-stable* time integration schemes applied to the transport equation (3): the TR, also known as the Crank–Nicholson scheme:

$$\frac{F^{n+1}-F^n}{\Delta t}+\frac{1}{2}(\boldsymbol{u}^n\cdot\nabla F^n+\boldsymbol{u}^{n+1}\cdot\nabla F^{n+1})=0$$

the BDF, also known as Gear scheme

$$\frac{3F^{n+1}-4F^n+F^{n-1}}{2\Delta t}+\boldsymbol{u}^{n+1}\cdot\nabla F^{n+1}=0$$

and the *implicit midpoint rule* (IMR)

$$\frac{F^{n+(1/2)}-F^n}{(\Delta t/2)}+\boldsymbol{u}^{n+(1/2)}\cdot\nabla F^{n+(1/2)}=0,\quad\text{and}\quad F^{n+1}=2F^{n+(1/2)}-F^n$$

We could also mention the first-order accurate *backward Euler* scheme:

$$\frac{F^{n+1}-F^n}{\Delta t}+\boldsymbol{u}^{n+1}\cdot\nabla F^{n+1}=0$$

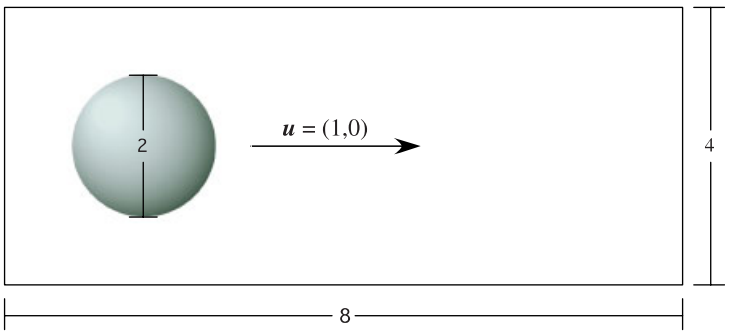

Figure 10. Drop advection problem.

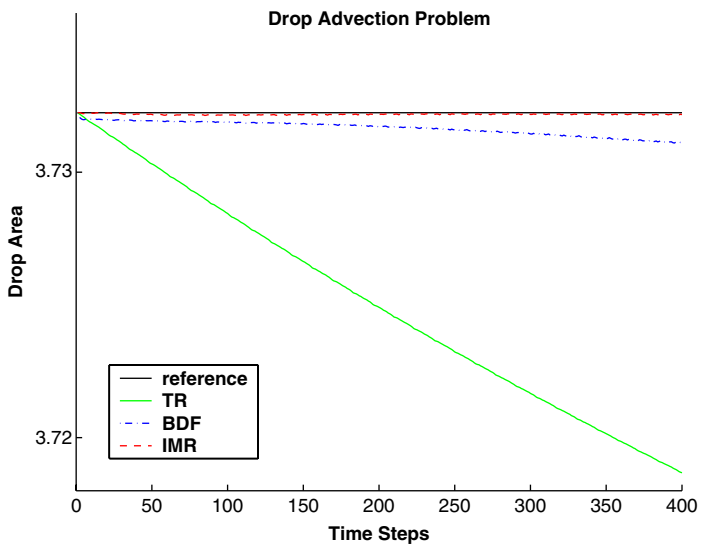

Figure 11. Mass loss induced by time integration schemes.

which is still popular in the computational science literature. It will not be included in our study. The comparison that will be performed consists in measuring the mass (the area in 2-D) of the drop for a given number of time steps. The numerical procedure for computing the mass of the drop is explained in the Appendix.

Figure 11 illustrates the performance of the TR, BDF and IMR schemes with respect to mass conservation. It may seem surprising that TR, which is one of the most popular time integration scheme used in CFD codes, performs poorly in our test. On the other hand, the IMR scheme which is not seen in many CFD papers, gives impressive results with respect to mass conservation. Quantitatively, we can observe, after only 400 non-dimensional time steps, for a fixed time-step size of $\Delta t = 0.01$ and for a mesh grid size Δx of the order of 0.07, a mass loss of the order of 0.36% for TR while we have a mass loss of 0.001% for IMR.

This can be explained by the various properties of the IMR scheme [21]. This scheme is second-order accurate, it introduces nearly no artificial numerical oscillations, and it is A-stable, yet it introduces little undesirable damping compared to other A-stable schemes, which makes this scheme interesting for an advection equation. Constant-time-step size schemes are known to give accurate results for a transport equation. Adaptive time integration schemes are suitable when the diffusion term is non-negligible.

6. OVERALL TIME INTEGRATION STRATEGY

6.1. The time integration strategy

Based on the results of the previous sections, the overall time integration strategy we propose for modelling free surface flows using an Eulerian free surface capturing approach is composed of the *ABDF scheme* for the Navier–Stokes equations coupled with the *IMR scheme* for the transport equation. Since we solve the system of partial differential equations (2) and (3) in a coupled manner, the time steps used with the IMR scheme are the ones computed by the ABDF scheme.

This strategy allows us to combine the previously described properties of both schemes, and to avoid their drawbacks. The BDF scheme for the transport equation is too diffusive, which would lead to the deformation of the pseudo-concentration and to mass loss. Moreover, an adaptive time integration scheme is not useful for an hyperbolic equation. In practice, we observe that the time-step size reaches a constant value which corresponds to the chosen tolerance, and it does not vary thereafter. The IMR scheme for the Navier–Stokes equations could lead to numerical oscillations in the discretization of the velocity components.

The proposed methodology is used to solve free surface flow problems of various nature in the following subsections. First, we will model Taylor's problem using the proposed methodology. This problem is known to reach a steady-state for specific operating conditions. The ABDF scheme should smoothly handle this problem. We will then study transient bubble interaction problems, which should be modelled accurately, with little mass loss, with the help of the IMR scheme.

6.2. Numerical validation: Taylor's problem

Let us consider an initially undeformed '2-D circular drop' of radius $a=1$, located at the centre of a computational domain of dimensions $[-5,5]\times[-5,5]$. The mesh grid size for this simulation is $\Delta x=0.2$. We are interested in the steady-state shape reached by the drop when submitted to a Cartesian linear flow. The far field flow is described by

$$\boldsymbol{u}(\boldsymbol{x})=\frac{1}{2}\dot{\gamma}\begin{pmatrix}1+\beta & 1-\beta\\ -1+\beta & -1-\beta\end{pmatrix}\begin{pmatrix}x\\ y\end{pmatrix} \tag{17}$$

where $\dot{\gamma}$ is the shear rate of the flow and β is a *flow-type parameter*. For example, the case $\beta=1$ corresponds to an *extensional flow*, while $\beta=0$ expresses a pure shear flow. The flows described by Equation (17) are generated using the *four-roll mill*, developed by Taylor [24]. The apparatus consists of four rotating cylinders, in the middle of which a drop is placed. By

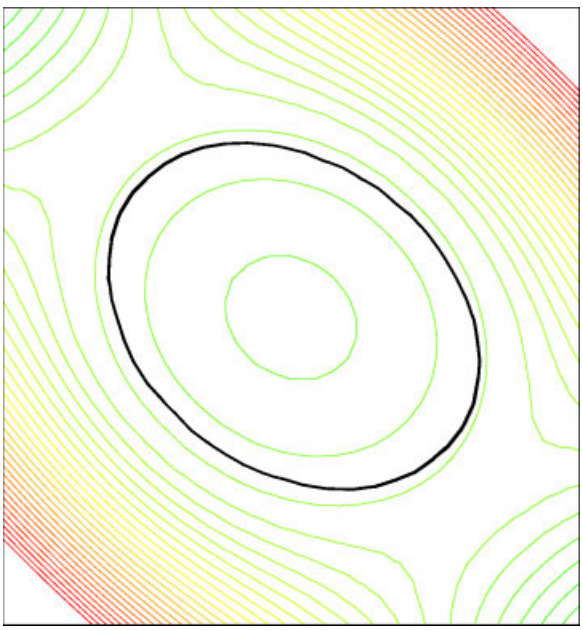

Figure 12. Steady-state shape of the drop and streamlines for Taylor's problem.

controlling the rotation rate of each cylinder, it is possible to obtain, far from the drop, the linear velocity field described by Equation (17).

As an example, let us consider the case $\beta = 0.2$ with a viscosity ratio of $\lambda = \mu_1/\mu_2 = 27.3$ (the drop is more viscous) and a density ratio of $\kappa = \rho_1/\rho_2 = 1$. For a *capillary number* of

$$Ca = \frac{\mu_2 \dot{\gamma} a}{\alpha} = 0.082$$

where the Navier–Stokes equations with the CSF model (5) are non-dimensionalized as

$$\frac{Re}{St}\,\tilde{\rho}\,\frac{\partial \tilde{\boldsymbol{u}}}{\partial \tilde{t}} + Re\tilde{\rho}(\tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{\nabla}})\tilde{\boldsymbol{u}} = \tilde{\boldsymbol{\nabla}} \cdot \tilde{\boldsymbol{\sigma}} + \frac{1}{Ca}\,\tilde{\kappa}\tilde{\boldsymbol{\nabla}}F$$

it is observed experimentally [25] that the drop reaches a steady-state shape. Quantities denoted with a tilde ('˜') represent non-dimensional values. The *Reynolds* number, *Re*, is small in this application, which leads us to neglect the acceleration term. The ratio *Re*/*St*, which includes the *Strouhal* number *St*, is set to 1. Figure 12 illustrates the shape of the drop and the streamlines, indicating that the steady-state seems to be reached. Figure 13 illustrates the time-step sizes computed by the ABDF scheme, for a tolerance of $\varepsilon = 0.0005$. We again observe larger time-step sizes for the transient phase of the simulation. The time-step sizes become smaller when the drop reaches its steady-state shape and angle, where small deformations of the free surface are again captured by the transient error estimator.

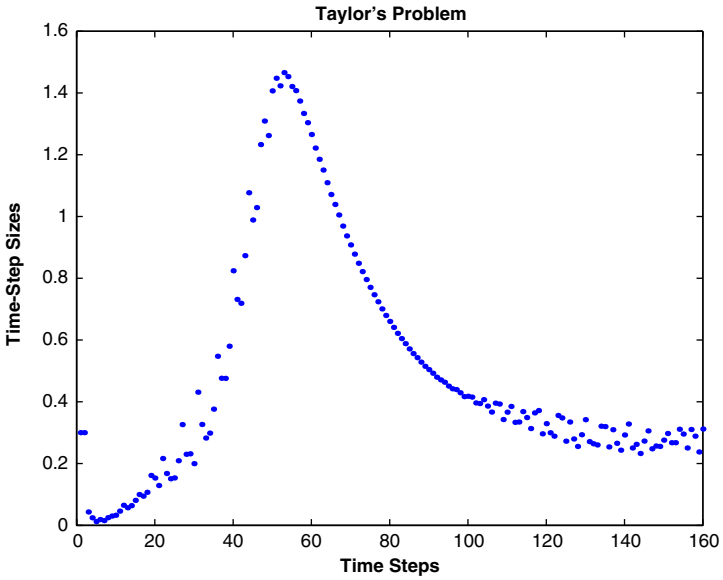

Figure 13. Time-step sizes for Taylor's problem, using the ABDF scheme ($Ca = 0.082$).

Table I. Comparison of experimental and numerical results for Taylor's problem: $\beta = 0.2$, $\lambda = 27.3$.

	Exper.		Numer.	
Ca	D	θ	D	θ
0.082	0.0635	−28.0	0.1042	−28.0
0.164	0.0786	−34.0	0.1391	−34.0
0.246	0.0788	−35.0	0.1438	−35.0
0.329	0.0798	−35.0	0.1470	−35.0

Taylor [24] defined a *deformation parameter* D as

$$D = \frac{L - B}{L + B}$$

where L and B are the longest and shortest semi-axes of the drop cross-section, in order to measure the deformation of a drop. For various values of Ca, we computed the deformation parameter D and the rotation angle θ of the drop. The results are compared to the experimental data of Bentley and Leal [25] in Table I. It can be seen that the deformation parameter is nearly constant, this being the consequence of the rotational flow. The concave *deformation curve*, illustrated in Figure 14, is typical of this situation. Our overestimation of the deformation parameter is probably caused by the fact that we perform a 2-D simulation of a 3-D problem. However, the computed angle of rotation reached by the numerical drops is accurate.

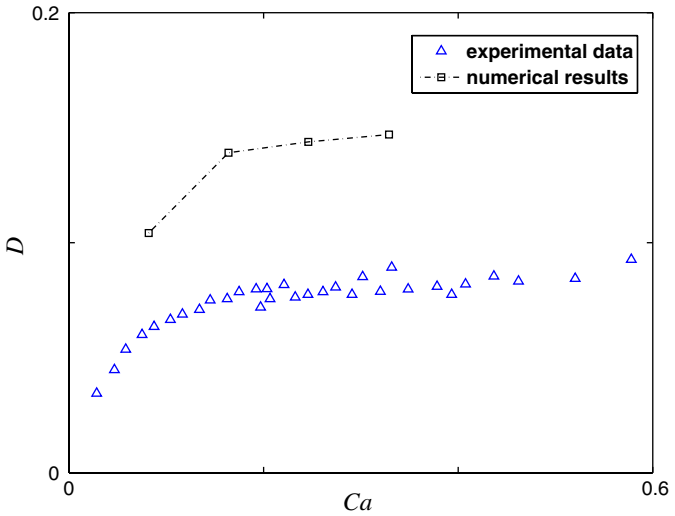

Figure 14. Deformation curve for Taylor's problem: $\beta = 0.2$, $\lambda = 27.3$.

6.3. *Numerical validation: buoyancy-driven rising bubbles*

The velocity and deformation of a single rising bubble due to buoyancy will first be studied. This test is based on the results of Sussman and Smereka [26]. Let us consider a bubble of non-dimensional radius $\tilde{a} = 1$, which is initially lying at rest in another fluid, in a geometry of dimensions $[-3, 3] \times [0, 12]$. This system is submitted to a constant body force $\boldsymbol{g} = (0, -g)$, where g is the gravitational acceleration. Since the fluids have a density ratio of $\kappa = \rho_1/\rho_2 = 0.0011$ (the fluid of the bubble is lighter), buoyancy generates a flow which makes the bubble rise. The viscosity ratio for this test is given by $\lambda = \mu_1/\mu_2 = 0.0085$.

For this problem, the Navier–Stokes equations with the CSF model (5) are non-dimensionalized as

$$\frac{1}{St}\,\tilde{\rho}\,\frac{\partial \tilde{\boldsymbol{u}}}{\partial \tilde{t}} + \tilde{\rho}(\tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{\nabla}})\tilde{\boldsymbol{u}} = \frac{1}{Re}\,\tilde{\boldsymbol{\nabla}} \cdot \tilde{\boldsymbol{\sigma}} + \frac{1}{Fr}\,\tilde{\rho}\tilde{\boldsymbol{g}} + \frac{1}{We}\,\tilde{\kappa}\tilde{\boldsymbol{\nabla}}F \tag{18}$$

where, in our case, the Reynolds number is given by

$$Re = \frac{au_0\rho_2}{\mu_2} = 9.8$$

the Froude number is defined as

$$Fr = \frac{u_0^2}{ag} = 0.76$$

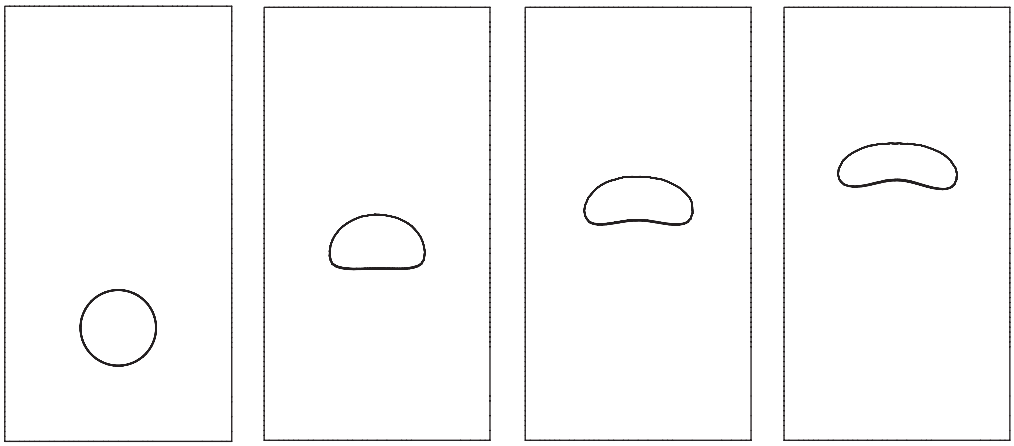

Figure 15. Buoyancy-driven rising of a single bubble.

and the Weber number is given by

$$We = \frac{a^2 g \rho_2}{\alpha} = 7.6$$

where the reference velocity u_0 is the experimentally observed [27] steady-state rise velocity of an air bubble, which is $u_0 = 21.5\,\mathrm{cm/s}$. The Strouhal number St is again set to 1. The other quantities were previously described.

Figure 15 illustrates the deformation of the rising bubble at various time steps. The simulation was performed for the complete bubble. Our methodology performs well in maintaining the symmetry of the bubble, and its vertical path. Since this simulation is performed using a Cartesian frame of reference, the computed shapes of the bubble differ slightly from the ones illustrated in Reference [26]. The computed rising speed of the bubble is underestimated when compared to the non-dimensional rising speed which should be equal to 1, this being caused again by our 2-D approximation of an axisymmetric phenomena. Finally, using our methodology, we measure a numerical mass loss of the order of 8% over 80 time steps $(t = 0, \ldots, 6.5)$.

We will now study the rise and interaction of two bubbles due to buoyancy. It is a common practice in the literature to describe bubbles dynamics problems using the *Eötvös number*, also known as the *Bond number*

$$Eo = \frac{\rho_2 g L_0^2}{\alpha}$$

and the *Archimedes number*, also known as the *Galileo number*

$$N = \frac{\rho_2^2 g L_0^3}{\mu_2^2}$$

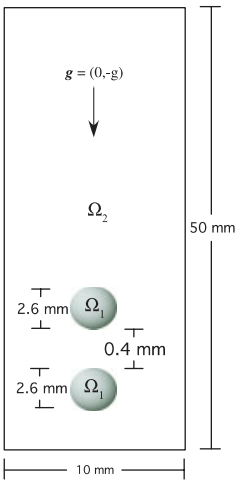

Figure 16. Bubbles rising problem.

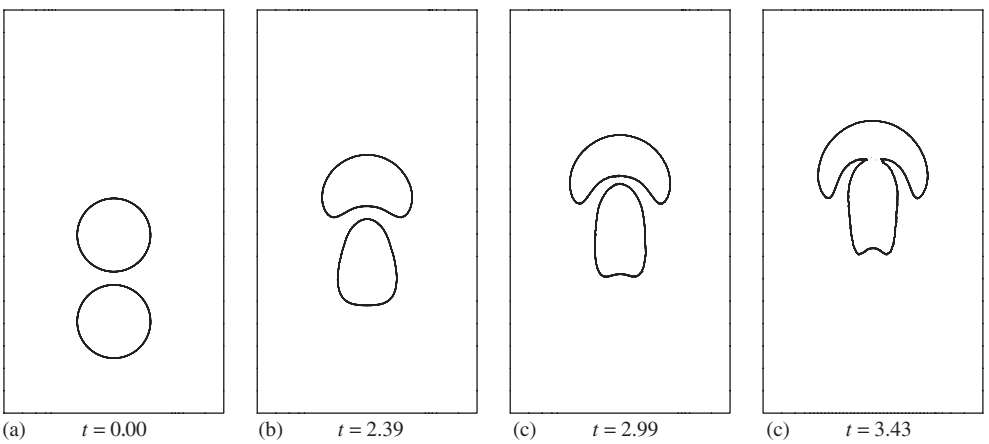

Figure 17. Buoyancy-driven rising and coalescence of two bubbles.

The *Morton number* is also seen in the literature:

$$M = \frac{g\mu_2^4}{\rho_2 \alpha^3} = \frac{Eo^3}{N^2}$$

However, it is possible to still work with the non-dimensionalized equations (18) by defining the reference velocity as $u_0 = \sqrt{gL_0}$, which gives us a relation between the non-dimensional groups (Re, Fr, We) and (N, Eo). In our case, $Re = \sqrt{N}$, $Fr = 1$ and $We = Eo$.

Our test problem, described in Reference [28], is illustrated in Figure 16. It consists of two vertically aligned bubbles with same diameter $d = 2.6\,\text{mm}$, located at a distance of $h = 0.4\,\text{mm}$

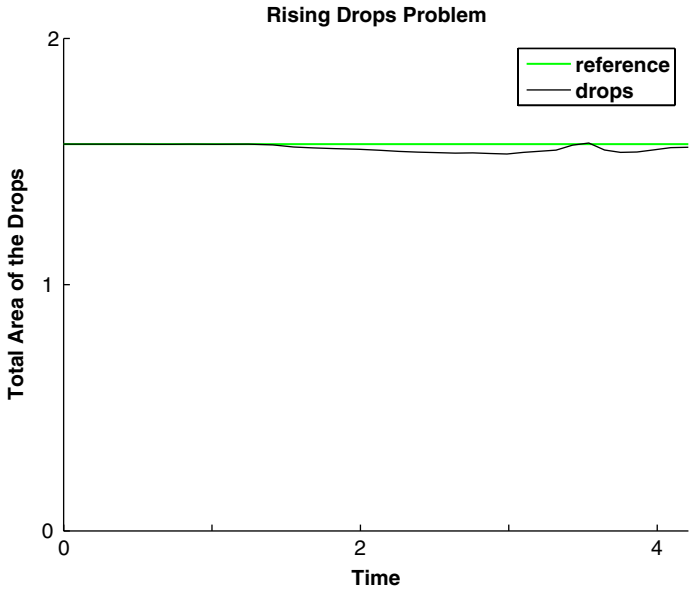

Figure 18. Rising and coalescence of two bubbles: mass conservation.

from each other. The non-dimensional groups are given by $Re = \sqrt{N} = 30$ and $Eo = We = 100$. Finally, the density ratio is given by $\kappa = \rho_2/\rho_1 = 2$ and the viscosity ratio is $\lambda = \mu_2/\mu_1 = 2$.

Figure 17 illustrates the deformation and coalescence of the two rising bubbles at various time steps. The shape of the drops are similar to what we can observe in Reference [28], the difference coming from the fact that these results come from 3-D simulations. We observe that coalescence occurs after 29 time steps ($\tilde{t} = 3.43$). Figure 18 illustrates the variation of the total mass of the bubbles (non-dimensional area) for the duration of the simulation, which is very small. It is unfortunately difficult to find quantitative results in the literature in order to perform an accurate validation for this problem.

7. CONCLUSION

A time integration strategy is proposed for modelling free surface flows in the context of Eulerian interface capturing. An adaptive second-order accurate backward differentiation formula is used to discretize the transient term of the Navier–Stokes equations, and the implicit midpoint rule is used for the transport equation of the marker variable. The adaptive scheme allows the automatic determination of a time-step size that follows the physics of the problem under study. It is shown that this mixed strategy gives accurate results for steady-state free surface flows, and that it reduces mass loss for transient multifluid flows. Such a study is novel to our knowledge. It also seems that it is the first time that the ABDF scheme was used in an applied context.

In order to keep the focus of this study on time integration strategies, several components of the numerical model necessary for performing accurate free surface flow simulations were

not discussed in this paper. Maintaining the region of transition of the pseudo-concentration sharp but smooth is very important in order to perform the accurate modelling of surface tension. Mesh adaptation in the vicinity of the free surface is also important for accurately modelling interfacial physics. The accurate numerical modelling of the capillary force is also a delicate matter. These questions were studied in References [5, 6, 16, 20] and we are still making progress developing more accurate algorithms for addressing these questions. Future work also includes the development of an axisymmetric model which should allow us to obtain numerical results that are closer to what is observed experimentally. We also plan to study applied (industrial) free surface flows involving the dynamics of drops and bubbles such as emulsions and jet break-ups.

APPENDIX A: COMPUTING THE AMOUNT OF MASS FOR EACH FLUID

Verifying mass conservation for each fluid is not a straightforward process in the context of Eulerian free surface capturing. The fact that we are working with non-structured meshes makes it more difficult to find an algorithm for computing the area of the regions Ω_1 and Ω_2.

When the interface does not go through an element, i.e. there are no points $\boldsymbol{x}$ in the element such that $F(\boldsymbol{x})=\frac{1}{2}$, the area of the triangle gives the quantity of mass of fluid 1 or 2 in that element. The other possibility is to see the interface $\mathcal{S}$ go through an element, as illustrated in Figure A1. A first approximation will be obtained by locating the two points of intersection of the free surface with the sides of the element (the points 2 and 5 in Figure A2(a)). The quantity of fluid 1 is approximated by the area of the triangle with vertices 1, 2 and 5, and the quantity of fluid 2 by the area of the quadrangle with vertices 2, 3, 4 and 5. We can then repeat this process by subdividing the element, as illustrated in Figure A2(b), and by computing the area occupied by each fluid in the subtriangles, as described above. We then repeat this subdivision process (cf. Figure A2(c)) until convergence of the computed area of the regions occupied by each fluid in that element, for a given tolerance. We observe that we compute the mass for each fluid to machine precision after 3 or 4 levels of subdivisions.

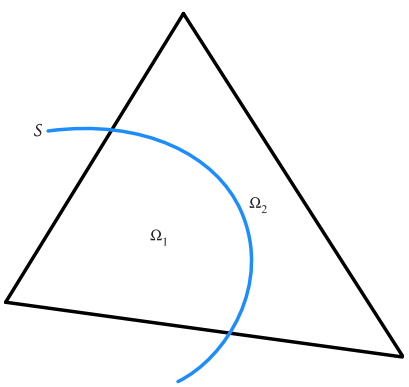

Figure A1. Elementary free surface flow problem.

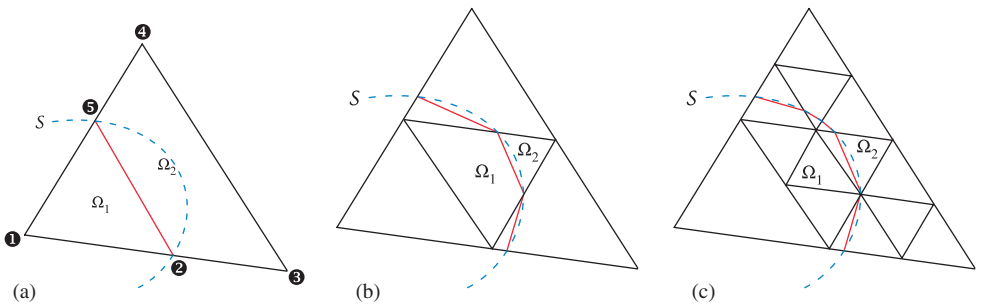

Figure A2. Mass computation for each fluid: (a) first level; (b) second level; and (c) third level.

ACKNOWLEDGEMENTS

This work was sponsored in part by the Auto21 Network of Centres of Excellence of Canada, the Natural Sciences and Engineering Research Council of Canada and the Fonds québécois de la recherche sur la nature et les technologies.

REFERENCES

1. Floryan JM, Rasmussen H. Numerical methods for viscous flows with moving boundaries. In *Applied Mechanics Review*, Metzner AWK (ed.). ASME: New York, 1989; 323–341.
2. Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**:201–225.
3. Sethian JA. *Level Set Methods. Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press: Cambridge, MA, 1999.
4. Thompson E. Use of pseudo-concentration to follow creeping viscous flows during transient analysis. *International Journal for Numerical Methods in Fluids* 1986; **6**:749–761.
5. Dufour S, Malidi A. A free surface updating methodology for marker function-based Eulerian free surface capturing techniques on unstructured meshes. *Communications in Numerical Methods in Engineering* 2004; **20**:857–867.
6. Dufour S, Pelletier D. Computations of multiphase flows with surface tension using an adaptive finite element method. *Numerical Heat Transfer, Part A* 2001; **40**:335–362.
7. Johnson C. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press: Cambridge, MA, 1987.
8. Gao DM. A three-dimensional hybrid finite element-volume tracking model for mould filling in casting processes. *International Journal for Numerical Methods in Fluids* 1999; **29**:877–895.
9. Tezduyar ET, Aliabadi S. EDICT for 3D computation of two-fluid interfaces. *Computer Methods in Applied Mechanics and Engineering* 2000; **190**:403–410.
10. Béliveau A, Fortin A, Demay Y. A two-dimensional numerical method for the deformation of drops with surface tension. *International Journal of Computational Fluid Dynamics* 1998; **10**:225–240.
11. Fortin M, Fortin A. A generalization of Uzawa's algorithm for the solution of the Navier–Stokes equations. *Communications in Applied Numerical Methods* 1985; **1**:205–208.
12. Brackbill JU, Kothe DB, Zemach C. A continuum method for modeling surface tension. *Journal of Computational Physics* 1992; **100**:335–354.
13. Rudman M. A volume-tracking method for incompressible multifluid flows with large density variations. *International Journal for Numerical Methods in Fluids* 1998; **28**:357–378.
14. Williams MW, Kothe DB, Puckett EG. Accuracy and convergence of continuum surface-tension models. In *Fluid Dynamics at Interfaces*, Shyy W, Narayanan R (eds). Cambridge University Press: Cambridge, MA, 1999; 294–305.
15. Popinet S, Zaleski S. A front-tracking algorithm for accurate representation of surface tension. *International Journal for Numerical Methods in Fluids* 1999; **30**:775–793.
16. Dufour S, Pelletier D. A study of drop dynamics using an adaptive finite element method. *14th AIAA Computational Fluid Dynamics Conference*, Norfolk, VA, *AIAA Paper 99-3318*, 1999; 12.

17. Renardy Y, Renardy M. PROST: a parabolic reconstruction of surface tension for the volume-of-fluid method. *Journal of Computational Physics* 2002; **183**:400–421.
18. Shin S, Abdel-Khalik SI, Daru V, Juric D. Accurate representation of surface tension using the level contour reconstruction method. *Journal of Computational Physics* 2005; **203**:493–516.
19. Hou TY, Lowengrub JS, Shelley MJ. Removing the stiffness from interfacial flows with surface tension. *Journal of Computational Physics* 1994; **114**:312–338.
20. Dufour S, Malidi A. An Eulerian interface capturing methodology for the numerical modeling of free surface flows with surface tension. Redaction, 2005.
21. Gresho PM, Sani RL, Engelman MS. *Incompressible Flow and the Finite Element Method*: *Advection–Diffusion and Isothermal Laminar Flow.* Wiley: New York, 1999.
22. Cuvelier C, Segal A, van Steenhoven AA. *Finite Element Methods and Navier–Stokes Equations*. D. Reidel Publishing Company: Dordrecht, 1986.
23. Engelman SE, Jamnia M-A. Transient flow past a circular cylinder: a benchmark solution. *International Journal for Numerical Methods in Fluids* 1990; **11**:985–1000.
24. Taylor GI. The formation of emulsions in definable fields of flow. *Proceedings of the Royal Society of London* 1934; **146**(A):501–523.
25. Bentley BJ, Leal LG. An experimental investigation of drop deformation and breakup in steady, two-dimensional linear flows. *Journal of Fluid Mechanics* 1986; **167**:241–283.
26. Sussman M, Smereka P. Axisymmetric free boundary problems. *Journal of Fluid Mechanics* 1997; **341**: 269–294.
27. Hnat JG, Buckmaster JD. Spherical cap bubbles and skirt formation. *Physics of Fluids* 1976; **19**:182–194.
28. de Sousa FS, Mangiavacchi N, Nonato LG, Castelo A, Tomé MF, Ferreira VG, Cuminato JA, McKee S. A front-tracking/front-capturing method for the simulation of 3D multi-fluid flows with free surfaces. *Journal of Computational Physics* 2004; **198**:469–499.

