INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2009; **60**:149–175
Published online 6 August 2008 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1877

A dual-time central difference finite volume scheme for interface capturing on unstructured meshes

H. Gough, Ann L. Gaitonde*, † and D. P. Jones

Department of Aerospace Engineering, University of Bristol, University Walk, Bristol BS8 1TR, U.K.

SUMMARY

This paper describes a central difference scheme for the prediction of flows with an interface. The interface is captured rather than tracked and the key to the current approach is a correction to the hydrostatic pressure. The correction enables the scheme to evaluate pressures at cell faces in a consistent manner so that the source term in the equations is correctly balanced at the interface and on non-equispaced meshes. This prevents the development of large errors in the solution, which can lead to the divergence of the numerical scheme. The current approach allows interface flows to be calculated by a simple modification of existing central difference codes. Results for a number of test cases are presented, with comparisons made with both experimental data and other numerical solutions. Copyright © 2008 John Wiley & Sons, Ltd.

Received 4 March 2008; Revised 29 May 2008; Accepted 2 June 2008

KEY WORDS: interface capturing; dual-time; central difference; unsteady; finite volume; CFD

1. INTRODUCTION

Numerical simulations of flows with a moving interface separating two immiscible fluids are complex. The numerical simulation of two fluid flows, particularly water and air flows, is central to many different types of applications, including the flow through rivers and aqueducts, and is crucial in the design and construction of naval hydrodynamics, examples of which include ships, submerged bodies, oil platforms and buoys. Various numerical techniques have been developed to predict the motion of an interface between two fluids. The main approaches that have been developed can be characterized as surface fitting, surface tracking and surface capturing [1]. Surface-fitting methods involve adapting the topology of the computational mesh to match the

*Correspondence to: Ann L. Gaitonde, Department of Aerospace Engineering, University of Bristol, University Walk, Bristol BS8 1TR, U.K.
†E-mail: ann.gaitonde@bristol.ac.uk

Contract/grant sponsor: EPSRC
Contract/grant sponsor: BAe Systems

interface position [2–5]. The use of surface fitting maintains a high interface resolution; however, such methods are limited to surfaces that do not undergo large deformations. Large deformations in the interface would lead to significant distortion of the computational mesh, which degrade solution convergence and quality. Previous results [6] have shown that even simple interface geometries can produce large mesh deformations. Another disadvantage of surface-fitting methods is the need to recompute the metrics of the computational mesh after each mesh adaption, which can be computationally costly depending upon the implementation used. Both surface-tracking and surface-capturing schemes use meshes that do not move to conform to the free surface; hence, the grid-related issues of surface-fitting schemes are avoided. Surface-tracking methods explicitly track the interface location through marker points or functions. The reconstruction of the free surface location, which is required for such methods, can be complex, especially in three dimensions. The surface-capturing approach treats the interface as a contact discontinuity in the density field that is captured by the numerical scheme along with the velocity and pressure fields. This approach is analogous to shock capturing in single-phase compressible flow. The need for complex surface-tracking and reconstruction procedures is eliminated [7–9] and complicated interface motion can be predicted. The use of such schemes on arbitrarily unstructured meshes is no more complicated than on its structured counterpart. Furthermore, this approach can readily handle situations of fluid break-up, entrapment, reconnection and also situations in which the interface develops as part of the flow solution provided the computational mesh resolution is sufficiently fine for the case in question. Existing schemes of this type utilize an upwind flux formulation based on the *Riemann* problem at each cell face [1, 10–12]. A central difference formulation with added dissipation has not previously been reported for interface capturing and is described in this paper. This represents a direct extension of the widely used scheme developed for compressible flow [13, 14] and previously extended to single-phase incompressible flows via an artificial compressibility approach [15].

2. GOVERNING EQUATIONS FOR INTERFACE CAPTURE

The fundamental assumptions of the approach to interface capturing used in this paper are that each of the fluids is assumed to be incompressible and isothermal, so that a solution can be obtained for the pressure and velocity without regard to energy transport.

In a single-phase flow of an incompressible isothermal fluid, the non-dimensional form of the governing equations of motion is

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

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U}\mathbf{U}) = -\nabla p + \frac{1}{Re} \nabla \cdot [\nabla \mathbf{U} + (\nabla \mathbf{U})^{\mathrm{T}}] + \mathbf{S} \tag{2}$$

where $\mathbf{S}$ contains any source terms, the *Reynolds* number $Re = U_0 L_0 / \nu_0$ and U_0, L_0 and ν_0 are reference values of speed, length and kinematic viscosity, respectively. However, for a two-phase flow, the density will change in space and time at and near the interface. To allow for this situation, the density ρ is reintroduced into the incompressible momentum equation. In addition, a separate equation for the density is solved to capture the interface position. The density is assumed to remain constant along particle paths with the resulting transport equation written in conservative form using the incompressible continuity equation [16]. The new system of equations is given in

DOI: 10.1002/fld

non-dimensional form by

$$\nabla \cdot \mathbf{U} = 0 \tag{3}$$

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot (\rho \mathbf{U}\mathbf{U}) = -\nabla p + \frac{\mu}{Re} \nabla \cdot [\nabla \mathbf{U} + (\nabla \mathbf{U})^{\mathrm{T}}] + \mathbf{S} \tag{4}$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0 \tag{5}$$

where the reference density for non-dimensionalization is taken to be that of the denser fluid. The source term is given by

$$\mathbf{S} = \frac{\rho \hat{\mathbf{g}}}{Fr^2} + \mathbf{f}_\sigma \tag{6}$$

where the two terms on the right-hand side represent the buoyancy forces due to gravity and surface tension, respectively. The *Froude* number $Fr = U_0/\sqrt{|\mathbf{g}|L_0}$ results from non-dimensionalization and represents the ratio of the fluid flow speed to the speed of small-amplitude surface waves [17]. The surface tension term is obtained using the *continuum surface force* (CSF) model, which represents the surface tension effects as a continuous volumetric force acting in the direct proximity of the interface, see [16, 18]. The stress tensor and continuity equations remain the same as for a single-phase incompressible isothermal fluid, since the fluids are still both incompressible and isothermal. An expression for the mixture viscosity is required and is given by

$$\mu = \alpha + (1 - \alpha)\mu_r \tag{7}$$

where μ_r is the viscosity ratio and the volume fraction α is given by

$$\alpha = \frac{\rho - \rho_r}{1 - \rho_r} \tag{8}$$

and ρ_r is the density ratio of the fluids. The density of the heavier fluid has been used as the reference density; hence, the non-dimensional fluid density for the heavier fluid is equal to one and the non-dimensional density of the lighter fluid equals the density ratio ρ_r. Similarly the reference viscosity is that of the heavier fluid; hence, the non-dimensional fluid viscosity for the heavier fluid equals one and the non-dimensional viscosity of the lighter fluid is given by μ_r, the viscosity ratio of the fluids.

As a fixed mesh finite volume scheme is to be used, the equations are cast into an integral form as

$$\mathbf{I}_0 \frac{\partial}{\partial t} \int_V \mathbf{W}\, \mathrm{d}V + \oint_{\partial V} \mathbf{F} \cdot \mathrm{d}\mathbf{S} = \int_V \mathbf{S}\, \mathrm{d}V \tag{9}$$

These are solved using a dual-time-stepping approach, which introduces an additional fictitious derivative with respect to a pseudo-time to the equations:

$$\mathbf{\Gamma}_\beta \frac{\partial}{\partial \tau} \int_V \mathbf{W}\, \mathrm{d}V + \mathbf{I}_0 \frac{\partial}{\partial t} \int_V \mathbf{W}\, \mathrm{d}V + \oint_{\partial V} \mathbf{F} \cdot \mathrm{d}\mathbf{S} = \int_V \mathbf{S}\, \mathrm{d}V \tag{10}$$

A solution of this modified system that is steady state in pseudo-time is a time-accurate solution of the original system of equations (9).

DOI: 10.1002/fld

The vector $\mathbf{W}$ is given by

$$\mathbf{W}=(p,\rho u,\rho v,\rho w,\rho)^{\mathrm{T}} \tag{11}$$

The flux vector can be decomposed into convective and diffusive components:

$$\mathbf{F}=\mathbf{F}^{\mathrm{C}}-\mathbf{F}^{\mathrm{D}} \tag{12}$$

where

$$\begin{aligned}\mathbf{F}^{\mathrm{C}}&=(\mathbf{U},\rho u\mathbf{U}+p\hat{\mathbf{i}},\rho v\mathbf{U}+p\hat{\mathbf{j}},\rho w\mathbf{U}+p\hat{\mathbf{k}},\rho\mathbf{U})^{\mathrm{T}}\\ \mathbf{F}^{\mathrm{D}}&=\left(0,\frac{\mu}{Re}\left(\frac{\partial\mathbf{U}}{\partial x}+\nabla u\right),\frac{\mu}{Re}\left(\frac{\partial\mathbf{U}}{\partial y}+\nabla v\right),\frac{\mu}{Re}\left(\frac{\partial\mathbf{U}}{\partial z}+\nabla w\right),0\right)^{\mathrm{T}}\end{aligned} \tag{13}$$

The matrices $\mathbf{I}_0$ and $\boldsymbol{\Gamma}_{\boldsymbol{\beta}}$ are given by

$$\mathbf{I_0}=\begin{pmatrix}0&0&0&0&0\\ \vdots&1&\ddots&\vdots&\vdots\\ \vdots&\ddots&1&\vdots&\vdots\\ \cdots&\cdots&\cdots&1&\vdots\\ \cdots&\cdots&\cdots&\cdots&1\end{pmatrix},\quad \boldsymbol{\Gamma_\beta}=\begin{pmatrix}1/\beta^2&0&0&0&0\\ \vdots&1&\ddots&\vdots&\vdots\\ \vdots&\ddots&1&\vdots&\vdots\\ \cdots&\cdots&\cdots&1&\vdots\\ \cdots&\cdots&\cdots&\cdots&1\end{pmatrix} \tag{14}$$

where β is the artificial compressibility parameter [19] used to obtain a hyperbolic system of equations and the source vector is given by

$$\mathbf{S}=\left(0\left(\rho\frac{\hat{g}_x}{Fr^2}+f_{\sigma_x}\right)\hat{\mathbf{i}},\left(\rho\frac{\hat{g}_y}{Fr^2}+f_{\sigma_y}\right)\hat{\mathbf{j}},\left(\rho\frac{\hat{g}_z}{Fr^2}+f_{\sigma_z}\right)\hat{\mathbf{k}},0\right)^{\mathrm{T}} \tag{15}$$

3. CENTRAL DIFFERENCE SCHEME

Equations (10) are solved using a standard central difference scheme [14] together with a new method to modify the hydrostatic pressure on cell faces. If the standard scheme is used without the pressure modification, then the source term will not be correctly balanced in the equations. This can lead to divergence of the numerical scheme.

3.1. Spatial discretization

A finite volume discretization of Equation (10) is formulated by integrating over an arbitrary *fixed* cell, P, of an unstructured mesh

$$\Delta V_P\Gamma_\beta\frac{\mathrm{d}\mathbf{W}_P}{\mathrm{d}\tau}+\Delta V_P\mathbf{I}_0\frac{\mathrm{d}\mathbf{W}_P}{\mathrm{d}t}+\mathbf{C}_P=\mathbf{S}_P\Delta V_P \tag{16}$$

where $\mathbf{W}_P$ is the cell-average value of the flow variables located at the cell-centroid, $\mathbf{S}_P$ is the source term evaluated using cell-average values, ΔV_P is the cell area and $\mathbf{C}_P$ is the approximation

DOI: 10.1002/fld

to the flux integral. In evaluating the flux terms, the values of the flow variables, $\mathbf{W}$, together with velocity and temperature gradients on the cell face are needed. The face values of the flow variables are approximated by taking the average of the flow variables of the cell-centres bounding the face [20]:

$$\mathbf{W}_f = \tfrac{1}{2}(\mathbf{W}_L + \mathbf{W}_R) \tag{17}$$

and the gradients are first obtained at cell-centres using Green's theorem before being averaged as above to give face values. The hydrostatic component of pressure is modified on the face to ensure that the source term is correctly balanced, see Section 3.4. In order to prevent odd–even decoupling, numerical dissipation, either scalar or matrix, is explicitly added to the solution as detailed in Section 3.5.

The convective, viscous, numerical dissipative fluxes and the source term can all be combined into a single residual $\mathbf{R}_P^*$ so that Equation (16) can now be written in semi-discrete form as follows:

$$\Delta V_P \boldsymbol{\Gamma}_{\boldsymbol{\beta}} \frac{\mathrm{d}\mathbf{Q}_P}{\mathrm{d}\tau} + \Delta V_P \mathbf{I}_{\mathbf{0}} \frac{\mathrm{d}\mathbf{Q}_P}{\mathrm{d}t} + \mathbf{R}_P^* = 0 \tag{18}$$

3.2. Time integration

An implicit real-time algorithm is obtained by approximating Equation (18) at real-time level $n+1$ and putting the real-time derivative into discrete form using a second-order backwards difference formula [21]:

$$\Delta V_P \boldsymbol{\Gamma}_{\boldsymbol{\beta}} \frac{\mathrm{d}\mathbf{Q}_P^{n+1}}{\mathrm{d}\tau} + \Delta V_P \mathbf{I}_{\mathbf{0}} \frac{3\mathbf{Q}_P^{n+1} - 4\mathbf{Q}_P^{n} + \mathbf{Q}_P^{n-1}}{2\Delta t} + \mathbf{R}^{*n+1}_P = 0 \tag{19}$$

By combining the discrete real-time discretization with the residual term, $\mathbf{R}_P^{*\,n+1}$, Equation (19) can be written in a more compact form:

$$\Delta V_P \boldsymbol{\Gamma}_{\boldsymbol{\beta}} \frac{\mathrm{d}\mathbf{Q}_P^{n+1}}{\mathrm{d}\tau} + \mathbf{R}_P^{n+1} = 0 \tag{20}$$

The required time-accurate solution is found by marching (20) to steady state in pseudo-time at each real time step using an explicit four stage Runge–Kutta scheme [14, 15]. Local pseudo-time-stepping and multigrid are used for convergence acceleration.

3.3. Local pseudo-time-stepping

The local pseudo-time-step in cell P is calculated from both convective and diffusive components:

$$\frac{1}{\Delta\tau_P} = \frac{1}{\Delta\tau_P^{\mathrm{C}}} + \frac{1}{\Delta\tau_P^{\mathrm{D}}} \tag{21}$$

where the convective time step, for a fixed mesh, is given by

$$\Delta\tau_P^{\mathrm{C}} = \frac{2\Delta V_P \ \mathrm{Co}}{\sum_f (|\overline{\lambda}_f||\Delta S_f|)} \tag{22}$$

where Co is the *Courant* number, $\overline{\lambda}_f$ is the cell spectral radius in the face normal direction and ΔS_f is the cell face area. This differs from the structured mesh approach to the local time step [22],

DOI: 10.1002/fld

which uses a spectral radius scaled in each of the coordinate directions. However, this information is not available on a fully unstructured mesh. The diffusive component of the local time step is given by

$$\Delta\tau_P^{\mathrm{D}} = \frac{\mathrm{Co}(\Delta V_P)^2}{2k/Re(\sum_f |\Delta S_f|^2)} \tag{23}$$

where k is an adjustable constant that usually takes a value of 2. For unsteady calculations where the real time step, Δt, is small compared with the local pseudo-time-step, as calculated above, it has been shown that the discretization scheme can become unstable [13]. To overcome this a further limit is applied to the pseudo-time-step [13] to prevent it becoming too large in comparison with the real time step. For the current scheme this leads to

$$\Delta\tau_P = \min(\Delta\tau_P, \tfrac{2}{3}\Delta t) \tag{24}$$

3.4. Hydrostatic pressure discretization

Central difference schemes calculate the value of a variable at a cell face, for surface flux evaluations, using a simple average of the cell values bounding the face. When this is applied to obtain the face pressure, both the kinematic and hydrostatic pressure components will be averaged. The source term on the right-hand side of Equation (16) should be balanced by the hydrostatic pressure flux calculated using the central difference scheme. Away from the interface, on an equispaced computational mesh, the hydrostatic pressure flux will correctly balance the gravitational source term. This is not true when the central difference scheme is applied using a computational mesh that is not equispaced. The situation is further complicated at the interface itself where the steep change in the density profile can introduce large errors into the computation especially for low *Froude* number problems where the source term becomes large and correspondingly the hydrostatic pressure gradient increases. An upwind scheme uses gradient extrapolation to calculate the flow variables at a cell face. Therefore, in regions away from the interface the correct hydrostatic pressure field will be calculated at the cell faces even on a mesh that is not equispaced. In the vicinity of the interface, however, even for an upwind scheme the problem of correctly balancing the hydrostatic pressure field with the source term is a known issue [23]. If the position of the interface were known, then it would be possible to average the kinematic pressure component and directly include the correct hydrostatic pressure. However, this is not possible with interface capturing schemes since the interface position is never explicitly known. Therefore, a novel approach has been developed to calculate the hydrostatic pressure consistently at the cell faces for a central difference scheme without the position of the interface being known. The new scheme produces values for pressure at cell faces throughout the computational domain, including the interface itself, which are consistent and balance the source term correctly. The hydrostatic pressure on the face is first approximated by the cell-centred scheme as a simple average:

$$(p_H)_f = \frac{(p_H)_{\mathrm{L}} + (p_H)_{\mathrm{R}}}{2} \tag{25}$$

Figure 1 shows the situation for a mesh that is not equispaced and for the purposes of drawing the figure, gravity is assumed to act in the negative z direction. Two cells with different densities are considered bounding a face f. Then the cell-centre hydrostatic pressure values in Equation (25), which are set correctly at the start of the computation, are assumed to remain correct for all

DOI: 10.1002/fld

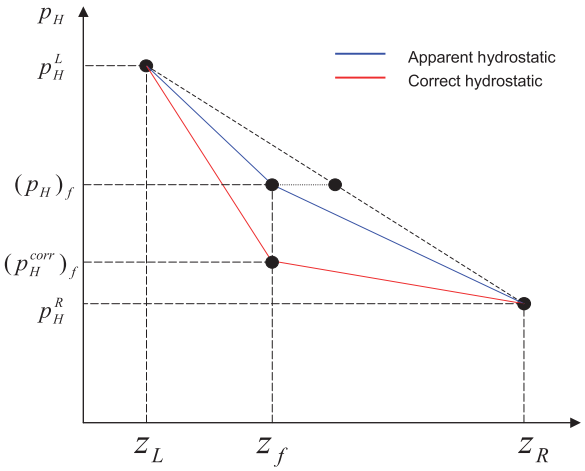

Figure 1. Hydrostatic pressure gradients at a cell face—shown for case when cell left is at a greater depth than cell right.

time. They can then be rewritten in terms of the correct face hydrostatic pressure $(p_H^{\text{corr}})_f$ and the hydrostatic gradient in each of the cells adjacent to the face, which are given by

$$\nabla p_H^L = \frac{\rho_{\text{L}}\hat{\mathbf{g}}}{Fr^2}$$

$$\nabla p_H^R = \frac{\rho_{\text{R}}\hat{\mathbf{g}}}{Fr^2}$$

Equation (25) becomes

$$(p_H)_f = \frac{(p_H^{\text{corr}})_f + \nabla p_H^L \cdot (\mathbf{x}_{\text{L}} - \mathbf{x}_f) + (p_H^{\text{corr}})_f + \nabla p_H^R \cdot (\mathbf{x}_{\text{R}} - \mathbf{x}_f)}{2}$$

$$= (p_H^{\text{corr}})_f + \hat{\mathbf{g}} \cdot \left(\frac{\rho_{\text{L}}(\mathbf{x}_{\text{L}} - \mathbf{x}_f) + \rho_{\text{R}}(\mathbf{x}_{\text{R}} - \mathbf{x}_f)}{2Fr^2} \right)$$

where $\mathbf{x}_f$ is the position vector of the cell face centre and $\mathbf{x}_{\text{L}}$ and $\mathbf{x}_{\text{R}}$ are the position vectors of the cell-centres of the left and right cells bordering the face.

On rearranging, this gives

$$(p_H^{\text{corr}})_f = (p_H)_f - \varepsilon \tag{26}$$

where the error term ε is given by

$$\varepsilon = \hat{\mathbf{g}} \cdot \left(\frac{\rho_{\text{L}}(\mathbf{x}_{\text{L}} - \mathbf{x}_f) + \rho_{\text{R}}(\mathbf{x}_{\text{R}} - \mathbf{x}_f)}{2Fr^2} \right) \tag{27}$$

Using the corrected value for the hydrostatic pressure, the cell face value of the total pressure, p, can be obtained from

$$(p^{\text{corr}})_f = \frac{p_{\text{L}} + p_{\text{R}}}{2} - \varepsilon \tag{28}$$

DOI: 10.1002/fld

Note that if the cell densities and the cell dimensions in the direction of the gravity field are equal

$$\rho_{\mathrm{L}} = \rho_{\mathrm{R}}$$

$$\hat{\mathbf{g}} \cdot (\mathbf{x}_f - \mathbf{x}_{\mathrm{L}}) = -\hat{\mathbf{g}} \cdot (\mathbf{x}_f - \mathbf{x}_{\mathrm{R}})$$

then the error term will be zero. This approach for the hydrostatic pressure discretization ensures that the field is correctly reproduced throughout the flow domain including at the interface. A different hydrostatic gradient correction approach has been recently developed to address this issue for an upwind *Roe* flux scheme by Qian *et al.* [23].

3.5. Numerical dissipation

Numerical dissipation is added to central difference schemes to prevent numerical oscillations in the solution. Both scalar [14] and matrix [24] dissipation models have been implemented in this study.

3.5.1. Scalar dissipation model, JST. The scalar dissipation model used in this research is a modified version of that first described by Jameson *et al.* [14] for compressible flows. The original dissipation model consisted of a fourth-order term that stabilizes the central difference scheme without smearing the solution and a second-order term that is activated in regions of strong gradients; typically for compressible flows this would be regions of 'shock' where the pressure gradient changes rapidly. For simple incompressible flows no regions of large gradients equivalent to shock waves in compressible flows exist; therefore, only the fourth-order dissipation would be necessary. However, for a flow with an interface, regions of large gradients will exist and it is necessary to use both the second- and fourth-order dissipation terms in the current scheme. The dissipation 'shock detector' or switch for compressible flow is based on pressure. In this work the switch has the same form as for compressible flow, but the detector is in this case based on density.

The scalar dissipation model is easy to formulate and quick to compute. However, each dissipative flux equation is scaled by the same factor, the spectral radius of the Jacobian matrix. This can degrade the numerical results particularly for flows that include regions of flow within boundary layers and for flows that contain large solution gradients where the second-order dissipation is necessary, e.g. when the flow features an interface. This issue is partly addressed by the *matrix* dissipation model.

3.5.2. Matrix dissipation model. The *matrix* dissipation model [24] allows different magnitudes of dissipation for each of the discrete equations. *Matrix* dissipation was developed by considering the differences between upwind and central difference schemes and formulating the *matrix* dissipation to mimic a first-order upwind scheme in the vicinity of 'shocks' or large solution gradients. The vector form of the dissipative flux is, in this case, scaled by the absolute value of the Jacobian matrix. The switch is modified compared with the scalar case. Some implementations of matrix dissipation have used a switch based on the Van Leer flux limiter that produces a total variation diminishing (TVD) scheme [25, 26]. However, in this study a weaker form of the switching function has been used as this gives improved convergence [16, 27] compared with the *Van Leer* limiter.

3.5.3. Numerical dissipation correction. The numerical dissipation schemes are based on the differences of the flow variables in $\mathbf{W}$. The pressure variable, p, contains the combined kinematic p_k and hydrostatic p_H components. If differencing of p itself is used it will introduce unnecessary dissipation. For example, consider a container filled with water at rest, then the correct

DOI: 10.1002/fld

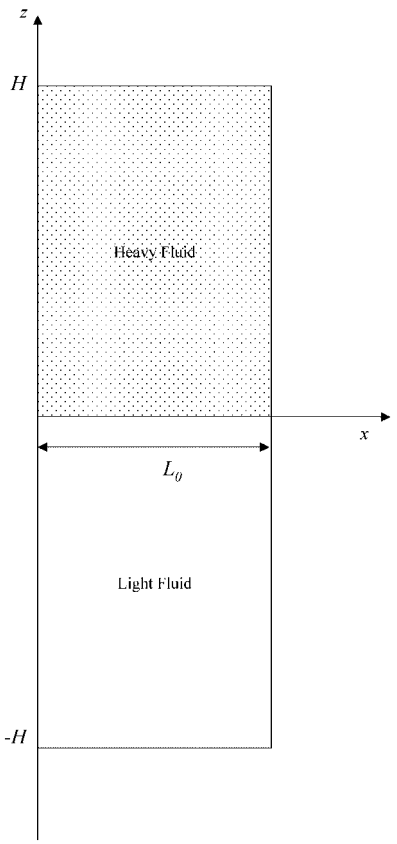

Figure 2. Set-up for Rayleigh–Taylor instability in dimensional variables.

steady solution will have a varying pressure due to the hydrostatic pressure gradient. This solution should not have any dissipation introduced into the continuity equation as the variations in the pressure field are a correct feature of the flow. Thus, the dissipation should only use the differences of the kinematic pressure. The correct differences in the kinematic pressure can be made using a similar approach to that in Section 3.4. The difference in pressure at the face will be given by

$$\Delta p_f = \Delta(p_H)_f + \Delta(p_k)_f \tag{29}$$

where $\Delta p_f = p_{\mathrm{R}} - p_{\mathrm{L}}$ and similarly for Δp_H and Δp_k. For the numerical dissipation scheme the difference of the kinematic pressure at the face is required. This can be obtained by rearranging Equation (29) as

$$\Delta(p_k)_f = \Delta p_f - \Delta(p_H)_f \tag{30}$$

and noting that if

$$\Delta(p_H)_f = (p_H)_{\mathrm{R}} - (p_H)_{\mathrm{L}}$$

DOI: 10.1002/fld

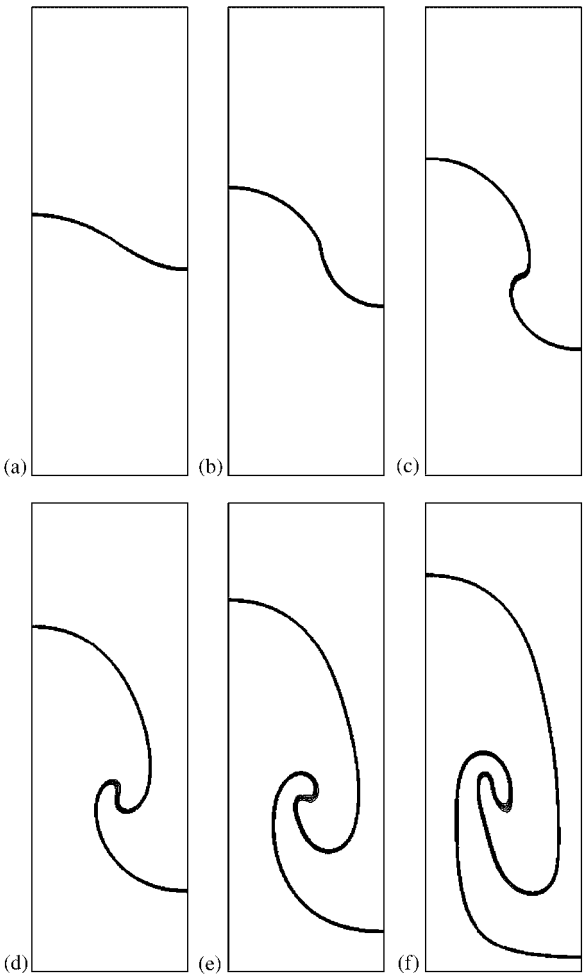

Figure 3. Contours for the Rayleigh–Taylor instability *matrix* scheme: (a) $t=0.8$; (b) $t=1.6$; (c) $t=2.4$; (d) $t=3.2$; (e) $t=4.0$; and (f) $t=4.8$.

then expressing $(p_H)_{\mathrm{R}}$ and $(p_H)_{\mathrm{L}}$ in terms of the correct hydrostatic pressure on the face gives

$$\Delta(p_H)_f = \frac{\hat{\mathbf{g}}\cdot[\rho_{\mathrm{L}}(\mathbf{x}_f - \mathbf{x}_{\mathrm{L}}) - \rho_{\mathrm{R}}\cdot(\mathbf{x}_f - \mathbf{x}_{\mathrm{R}})]}{Fr^2} \tag{31}$$

Therefore, the final required difference term for the numerical dissipation will be given by the following expression:

$$\Delta(p_k)_f = (p_{\mathrm{R}} - p_{\mathrm{L}}) - \frac{\hat{\mathbf{g}}\cdot[\rho_{\mathrm{L}}(\mathbf{x}_f - \mathbf{x}_{\mathrm{L}}) - \rho_{\mathrm{R}}\cdot(\mathbf{x}_f - \mathbf{x}_{\mathrm{R}})]}{Fr^2} \tag{32}$$

DOI: 10.1002/fld

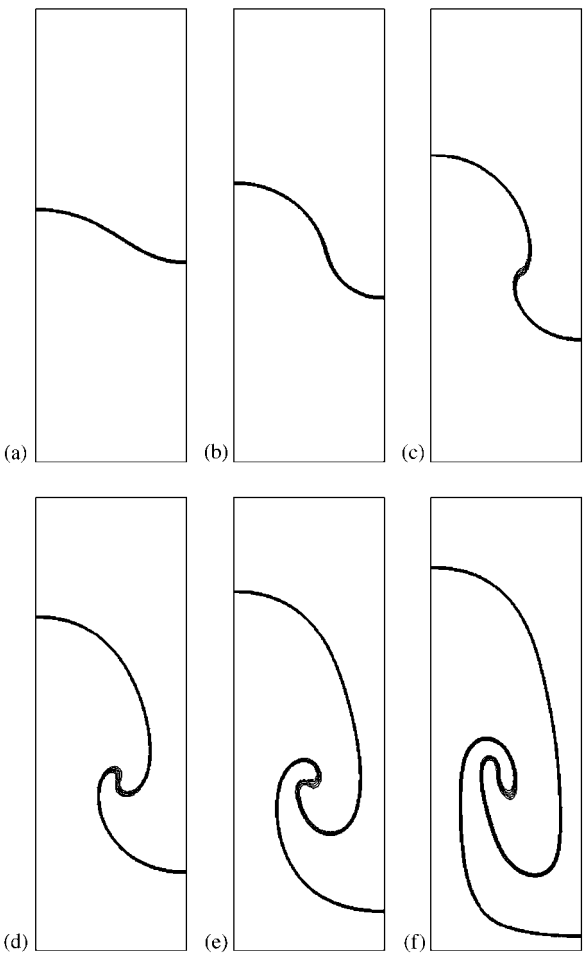

Figure 4. Contours for the Rayleigh–Taylor instability JST scheme: (a) $t=0.8$; (b) $t=1.6$; (c) $t=2.4$; (d) $t=3.2$; (e) $t=4.0$; and (f) $t=4.8$.

The expression for kinematic pressure differences given by Equation (32) is used in the numerical dissipation given for the continuity equation.

4. BOUNDARY CONDITIONS AND INITIALIZATION

The boundary conditions are implemented using 'ghost' or 'halo' cells at the boundary.

4.1. Impermeable no-slip walls

The density is extrapolated from the interior value to the wall using a zero gradient and the pressure is extrapolated with a fixed gradient. The velocity of the fluid at the wall is set equal to that of the wall.

DOI: 10.1002/fld

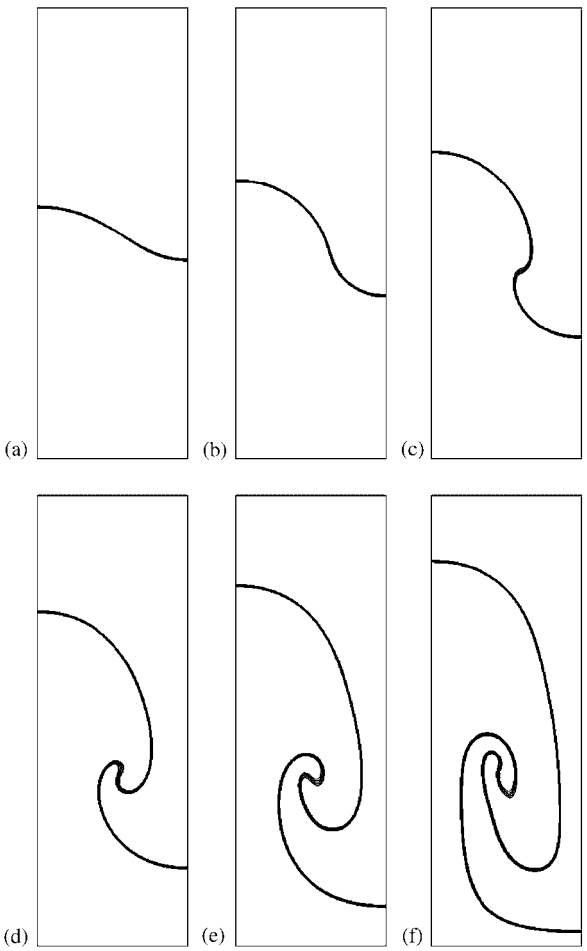

Figure 5. Contours for the Rayleigh–Taylor instability upwind scheme: (a) $t=0.8$; (b) $t=1.6$; (c) $t=2.4$; (d) $t=3.2$; (e) $t=4.0$; and (f) $t=4.8$.

4.2. Inlet/outlet

The method of characteristics has been used at inlet/outlet boundaries. The characteristic variables of the interface capturing equations are extrapolated either from the flow interior or from the exterior values depending upon the sign of the corresponding eigenvalue. In order to avoid wave reflection at the outflow, wave damping is used [28–31] to dissipate all the wave energy of the outgoing waves. In the surface-tracking approach the damping in the outflow can be achieved by damping the surface elevation. For the surface-capturing approach this is not directly possible because the interface location is not known; hence, a different approach is needed. One possibility would be to use a spatially static viscosity ramp [32, 33] such that the fluid thickens near the outflow, which would have the desired effect of dissipating the energy of the outgoing waves.

DOI: 10.1002/fld

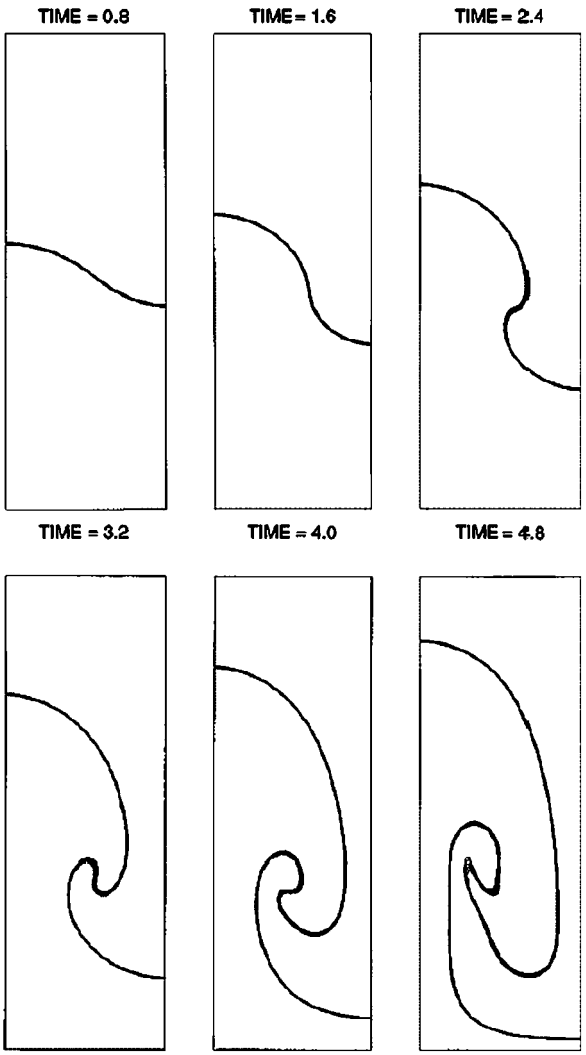

Figure 6. Contours for the Rayleigh–Taylor instability from Kelecy and Pletcher [1].

However, for calculations with explicit pseudo-time-stepping,[‡] the increase in viscosity yields a considerable limitation on the allowable local time step. This viscous limitation on the time step becomes rapidly more restrictive than the convective limitation. Consequently, the viscous time step restriction deteriorates the overall efficiency of the numerical method. A second approach that has been utilized here is to damp the fluid motion in the direction of the gravity vector [31] using a damping factor that is switched on in a damping zone around the outflow boundary.

[‡]The maximum local time step depends on both inviscid and viscous contributions.

DOI: 10.1002/fld

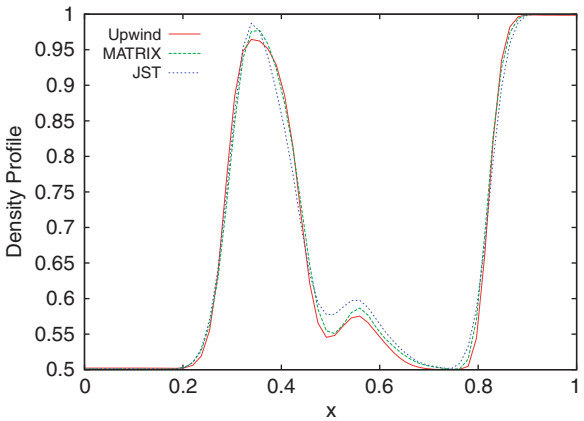

Figure 7. Comparison of density profiles at $z=-0.3$, $t=4$ for *Rayleigh–Taylor* instability.

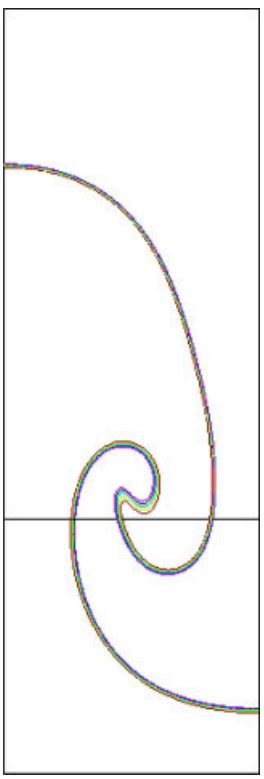

Figure 8. Cut location at $z=-0.3$, $t=4$ for *Rayleigh–Taylor* instability density profile comparison.

DOI: 10.1002/fld

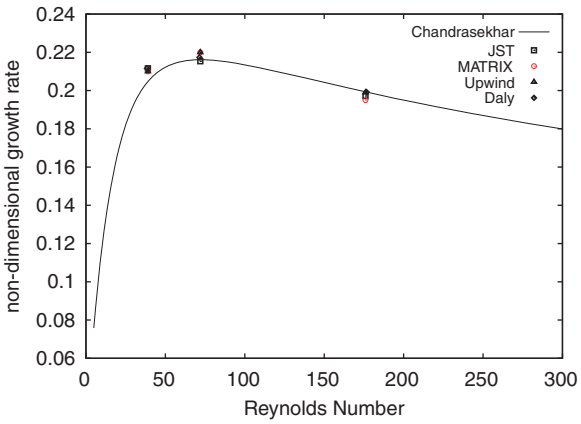

Figure 9. Non-dimensional growth rate for *Rayleigh–Taylor* instability at $Re = 39, 72$ and 176.

4.3. Initialization

The initialization of the flow variables is of particular importance when dealing with the interface capturing scheme. In particular, the pressure field has to be initialized to the correct hydrostatic distribution in order to satisfy the continuity and momentum equations. The initial interface location is known at the start of the simulation; hence, the hydrostatic pressure distribution can be initialized to the correct values easily. The correct initialization is dependent on the particular test case and is explained in the description of the results for each test case.

5. RESULTS

For the new interface capturing scheme, results have been produced for various test cases in order to verify the accuracy of the predicted results, in particular the interface position. The interface is represented by the contour of the average value of the densities of the two fluids, which is given in non-dimensional variables by

$$\rho_{\text{ave}} = \frac{1+\rho_r}{2} \tag{33}$$

For the purpose of comparison, results are also presented using an upwind scheme implemented in the same computational environment [23, 34, 35]. Note that in all cases the value of the artificial compressibility parameter is taken to be 1 and the CFL number is 1. The environment in which this scheme was implemented allows for any general unstructured mesh. In this study hybrid meshes are used with a background Cartesian mesh with enriched pockets around features of interest such as solid and free surface [36].

5.1. Rayleigh–Taylor instability

The Rayleigh–Taylor instability is a well-known fluid dynamics problem [37, 38] for which an analytical solution can be produced for the initial stages using linear perturbation analysis [39]. The problem consists of a heavy fluid sitting on top of a lighter fluid, with the interface subject

DOI: 10.1002/fld

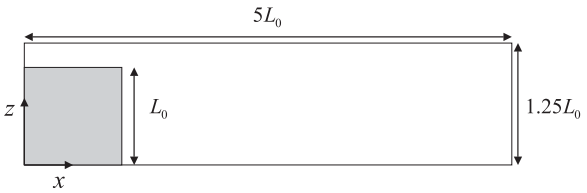

Figure 10. Set-up for the broken dam problem in dimensional variables.

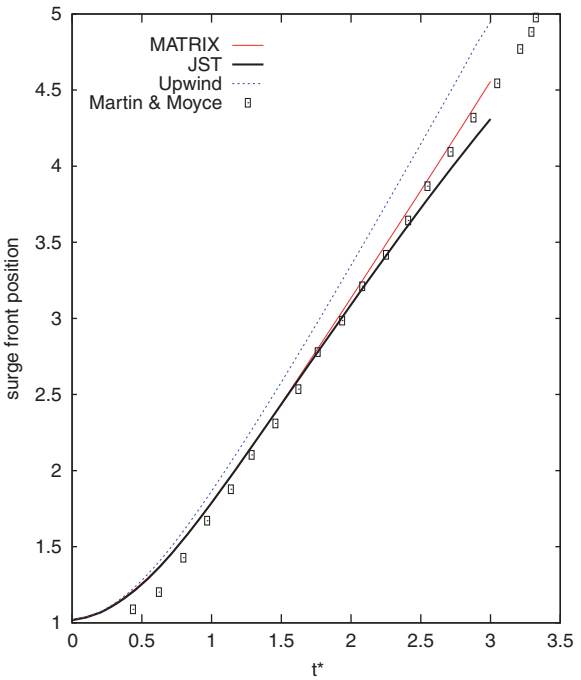

Figure 11. Surge front position versus non-dimensional time for the coarse mesh.

to a perturbation. Buoyancy forces cause the amplitude of the perturbation to grow with time. Previous computational studies have investigated the problem [1, 10, 11, 40] because it provides an interesting test case for interface schemes due to the mixing of the two fluids. The specific test case considered here has a density ratio $\rho_r = 0.5$ and the kinematic viscosities of the two fluids are set to be equal so that $\mu_r = 0.5$. A single wavelength perturbation is introduced at the fluid interface using the following non-dimensional velocity field [40]:

$$u = \begin{cases} \chi \sin(\pi x) e^{-\pi|z|}, & z > 0 \\ -\chi \sin(\pi x) e^{-\pi|z|}, & z < 0 \end{cases}$$

$$w = -\chi \cos(\pi x) e^{-\pi|z|}$$

DOI: 10.1002/fld

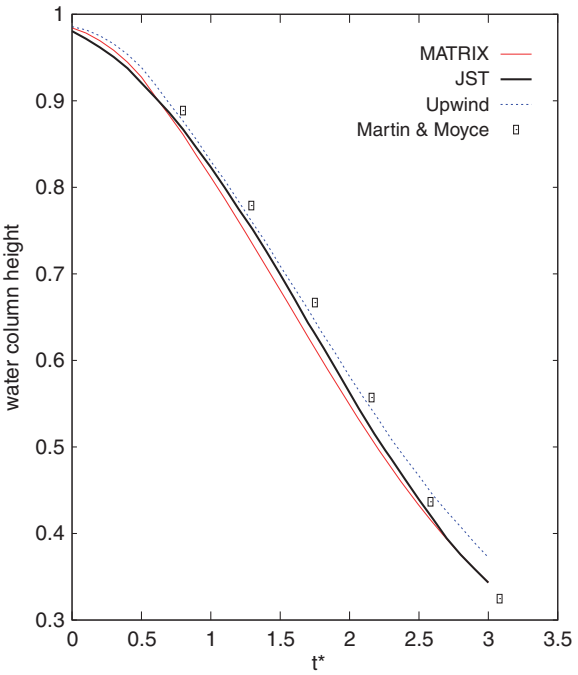

Figure 12. Dam break column height versus non-dimensional time for the coarse mesh.

where χ is a constant that governs the amplitude of the initial displacement. For this test case the value was set to $\chi = 0.25$, as data are available for comparison [11]. Figure 2 shows the initial set-up. The computational mesh has dimensions of 30×90. The reference length for non-dimensionalization L_0 is taken to be the width of the computational domain and the depth of each fluid is $H = 1.5L_0$. The *Reynolds* number was set to $Re = 283$. In keeping with other studies, the boundary conditions applied are no-slip walls on the top and bottom surfaces of the computational domain and symmetry conditions on the sides of the domain.

Figures 3 and 4 show the results obtained from the central difference scheme with the new hydrostatic pressure correction using *matrix* and *JST* dissipation, respectively. Figure 5 shows results obtained using a standard upwind scheme implemented in the same computational environment as the central difference scheme by the authors. The plots show density contours around $\rho_{\text{ave}} = 0.75$, with five contour lines plotted in the range $0.7 \leqslant \rho \leqslant 0.8$. It can be seen from the figures that the initial velocity perturbation causes the fluid to rise on the left-hand side and sink on the right-hand side. This motion causes a mixing of the two fluids resulting in the familiar mushroom shape. Figure 6 shows the results obtained using an upwind scheme by Kelecy and Pletcher [1] for comparison. There are some small differences in the shape of the interface between both central difference scheme results and the upwind results of *Kelecy and Pletcher*. However, the central difference results compare well with the results from the upwind scheme implemented by the authors. This suggests that the difference from the results of *Kelecy and Pletcher* is mainly due to differences in the computational environment rather than the use of central differencing in place of upwinding.

DOI: 10.1002/fld

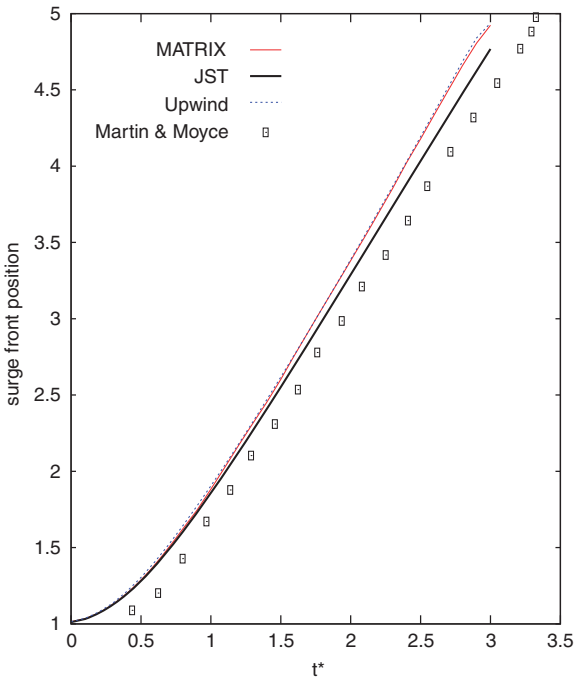

Figure 13. Surge front position versus non-dimensional time for the fine mesh.

In order to further study the density spread, Figure 7 displays the density profile at the position $z=-0.3$ for time $t=4$. The location of the slice is more clearly depicted in Figure 8, which shows the location of the cut plane. At this time the mushroom shape of the *Rayleigh–Taylor* instability is starting to form. Figure 7 shows that the cut profile: passes from the lighter fluid to the heavier one at $x=0.28$; then transforms back to the lighter fluid at $x=0.42$; is influenced by the proximity of the region of heavier fluid at $x=0.58$, which causes a small secondary spike; and passes into the heavier region $x=0.81$. The profiles for all the schemes have a similar shape, but there are differences particularly at the peaks and troughs between all three schemes. For this test case the *matrix* scheme took 40% more computational time than the *JST* scheme and the upwind scheme took approximately 50% more computational time than the *JST* scheme.

Chandrasekhar [39] derived an analytical expression for the linear growth rate n^* of the *Rayleigh–Taylor* instability as a function of the *Reynolds* number. In order to verify that the current interface capturing methodology can accurately predict the growth rates, several cases were run at different *Reynolds* numbers. The amplitude of the displacement was then used to calculate the linear growth rate as outlined in [41]. The amplitude of the displacement is taken to be the average of the absolute value of the displacement of the left and right of the instability. After the stable modes of the disturbance in the initial transient have decayed away, the natural logarithm of the amplitude of the displacement becomes linear with time. The slope of the linear region gives the growth rate n^* of the corresponding *Reynolds* number. The slope is calculated based on the time interval $t=0.2$–0.5. The linear growth rate for the *Reynolds* numbers 39, 72 and 176 has been computed such that a comparison with other research can be made. The predicted growth rate values in Figure 9 show

DOI: 10.1002/fld

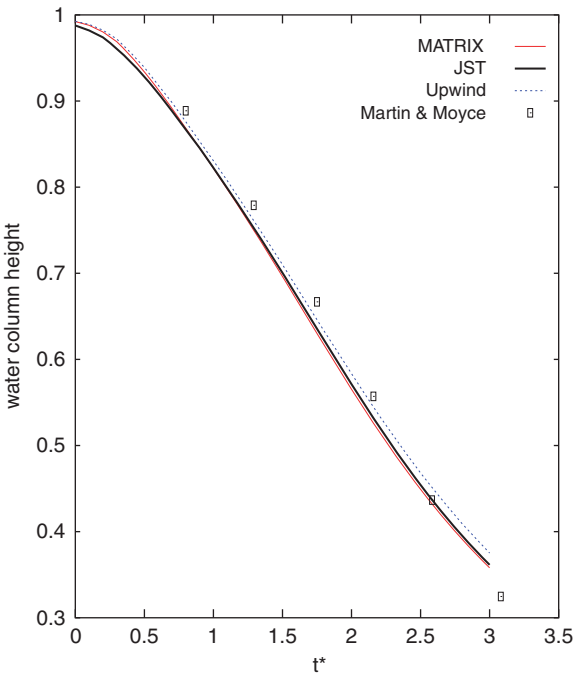

Figure 14. Dam break column height versus non-dimensional time for the fine mesh.

good agreement with the analytical curve based on the linear analysis of *Chandrasekhar* and also with the results obtained in another numerical study by Daly [40, 41].

6. BROKEN DAM

The broken dam or water column collapse has been extensively investigated by various researchers both numerically [42–45] and also experimentally [46, 47]. It is therefore often used as a benchmark test case. The initial configuration and computational domain is depicted in Figure 10 at $t=0$ when the column of water is confined to the left of a tank. For $t>0$ the restraint is removed and the water is allowed to collapse under the influence of gravity. The reference length used in non-dimensionalization is the height of the water column L_0, the reference density is taken to be that of the water and the reference velocity $U_0=\sqrt{|g|L_0}$. The density ratio $\rho_r=0.001$ and the viscosity ratio $\mu_r=0.01$. The *Reynolds* number is then given by

$$Re=\frac{\rho_0|g|^{1/2}L_0^{3/2}}{\mu_0} \tag{34}$$

The pressure distribution is initialized to a hydrostatic distribution relative to the top of the water column. The computational domain of this simulation represents a closed container with impermeable walls. This set-up differs from the experimental one that utilized an open container,

DOI: 10.1002/fld

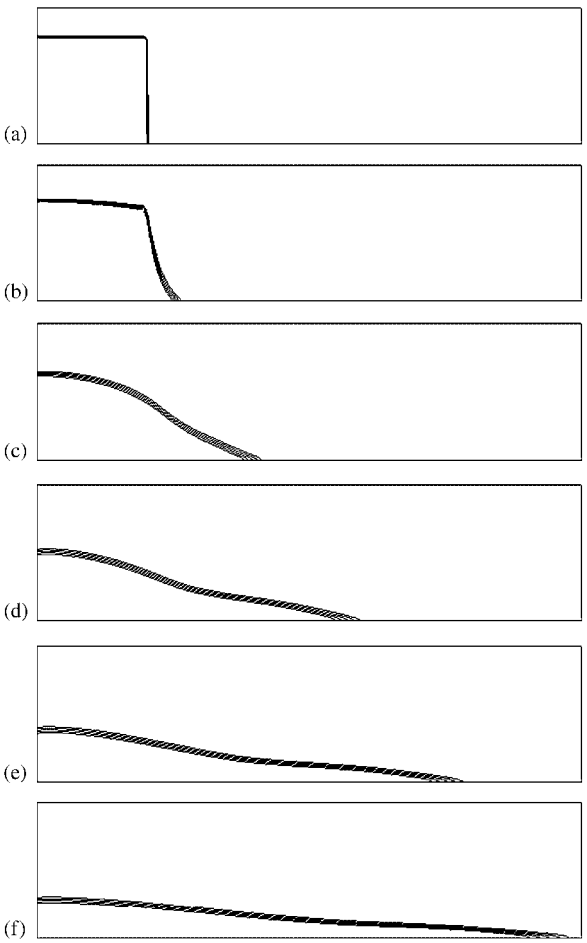

Figure 15. Contours of ρ for the breaking dam problem: (a) $t=0.0$; (b) $t=0.6$; (c) $t=1.2$; (d) $t=1.8$; (e) $t=2.4$; and (f) $t=3.0$.

but is the usual set-up for numerical simulations. All of the computational boundaries are set to be slip walls for consistency with other numerical studies [43, 44].

For the current test case $L_0=0.05715$ m, $Re=43\,000$ and $\rho_{\text{ave}}=0.5005$. Two different computational meshes were constructed for the simulation: a coarse mesh of dimensions 80×20 and a fine mesh of dimensions 160×40.

Figure 11 plots the location of the surge front against the non-dimensional time for the coarse computational mesh and is compared with experimental data from Martin and Moyce [46]. The *matrix* and *JST* solutions are closer to the experimental surge front position than the upwind results, which has over-predicted the surge front position. Near the end of the simulation, $t>2.50$, the *JST* dissipation starts to under-predict the surge front position. The column height is plotted against the non-dimensional time for the coarse mesh in Figure 12. The graph shows that there are some differences between all the schemes and the experimental data. Solutions at this mesh density can

DOI: 10.1002/fld

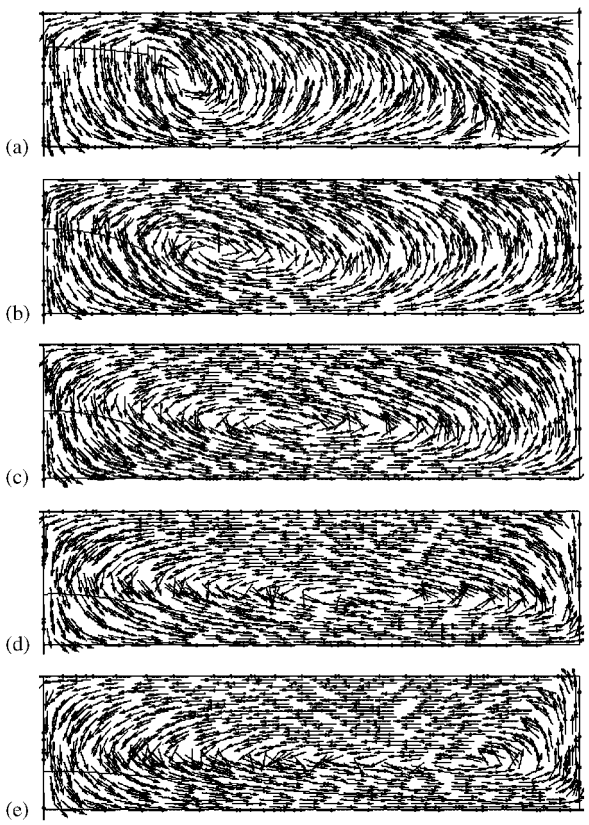

Figure 16. Vector plots for the breaking dam problem: (a) $t=0.6$; (b) $t=1.2$; (c) $t=1.8$; (d) $t=2.4$; and (e) $t=3.0$.

seem to show an exaggerated level of agreement with the experimental data. In fact, these solutions were not fully converged with respect to the mesh size and obtaining a converged solution led to results further from the experimental data.

Figure 13 shows the surge front position on the fine computational mesh. The results of the *matrix* dissipation more closely match those obtained using the upwind formulation. This is in keeping with the way the *matrix* dissipation scheme has been constructed to closely mimic an upwind formulation. However, the central differences schemes and the upwind scheme all over-predict the surge front location; that is, the simulated surge front moves faster than the experimental one. This phenomenon has been reported in other research [43, 44, 47] and is attributed to the difficulty in determining the exact location of the leading edge. A thin layer, similar to a jet, shoots along the bottom of the tank, which is difficult to capture, followed by the bulk of the flow. Furthermore, the discrepancy between the simulated and experimental results might also be due to the removal of the barrier holding the liquid in the left-hand corner at the start of the experimental simulation. In the numerical experiment this barrier is removed instantaneously, whereas in the physical world the removal of the barrier would have a time associated with it.

DOI: 10.1002/fld

The predicted results for the column height on the fine mesh, Figure 14, agree well with the experimental data.

Figure 15 shows density contours and Figure 16 shows vector plots calculated using *matrix* dissipation on the fine computational mesh. In Figure 15, 10 density contours are plotted within

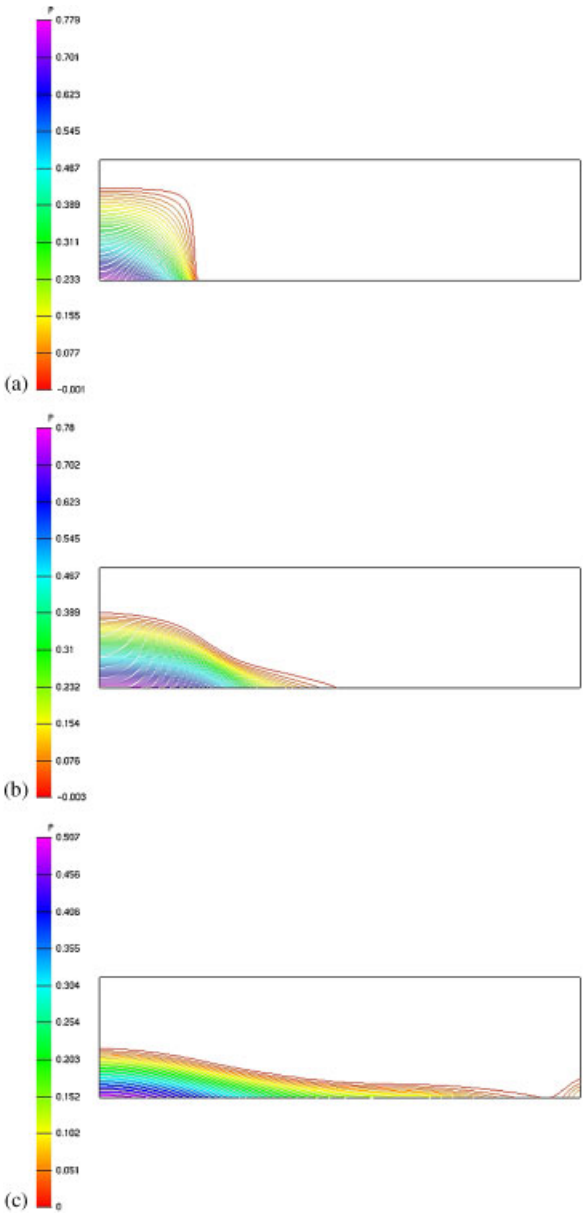

Figure 17. Pressure contours for the breaking dam problem: (a) $t=0.0$; (b) $t=1.2$; and (c) $t=2.4$.

DOI: 10.1002/fld

the range $0.3 \leqslant \rho \leqslant 0.6$. The interface has been captured with a small spread. For this test case and the others reviewed in this research, the spread of the standard deviation of the interface was over approximately 10 cells and this was consistent as the computational mesh was refined. Figure 16 shows an interesting feature of the dam break, which is the vortex that develops just above the interface and remains there for the duration of the simulation as a result of the forces upon the air next to the water.

Figure 17 shows the pressure contours obtained at times $t = 0.0, 1.2$ and 2.4 for the *matrix* dissipation. At time $t = 2.4$ contours arise in the lower right-hand corner. These are due to the fact that, consistent with other numerical studies, the computational domain is a closed container. The thin layer that crosses the domain ahead of the bulk of the flow hits the side of the container giving rise to the pressure disturbance.

7. THREE-DIMENSIONAL WATER COLUMN COLLAPSE

The preceding test cases have been simulated as two-dimensional by creating a thin slice of the flow to create a three-dimensional mesh with appropriate boundary conditions on the symmetry planes. In order to test the current implementation in a fully three-dimensional regime, a challenging test case is considered here. A series of three-dimensional dam break experiments was conducted by Martin and Moyce [46]. These consisted of a cylinder of water at $t = 0.0$, which was subsequently allowed to collapse under the influence of gravity. The experimental data measured the axial surge front position as a function of time. The initial configuration is axisymmetric and the initial radial slice is the same as shown in Figure 10. The reference length was taken to be the base radius of the initial water column and the reference density was that of water, resulting in a *Reynolds* number of $Re = 43\,000$ and density and viscosity ratios of $\rho_r = 0.001$, $\mu_r = 0.01$, respectively. All of the boundaries were set to be viscous walls. The computational mesh used was of dimensions $120 \times 120 \times 40$ in the radial, circumferential and vertical directions, respectively.

Figure 18 shows the predicted surge front position plotted against time for *JST*, *matrix* and upwind schemes together with the results obtained from *Martin* and *Moyce*. This shows that the

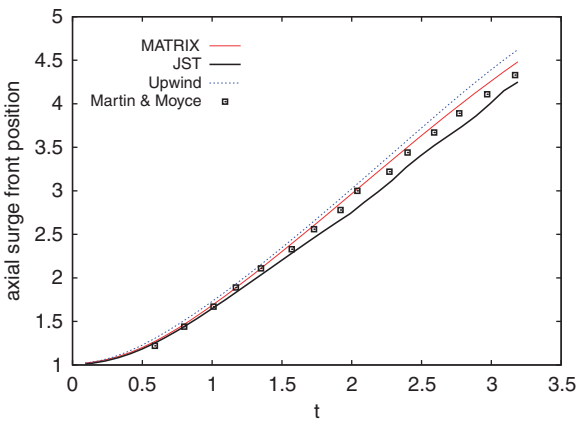

Figure 18. Axial surge front position for three-dimensional dam break.

DOI: 10.1002/fld

JST scheme slightly under-predicts the surge front position. The upwind and *matrix* schemes give similar results until the latter stages when the upwind scheme over-predicts the surge front position The difference between the *JST* and *matrix* numerical dissipation schemes is more pronounced for this three-dimensional test case because of the three-dimensional motion of the liquid, which creates diffusion in both the normal and stream-wise directions relative to the interface. Each equation in the *JST* dissipation is scaled by the spectral radius, whereas the *matrix* dissipation scales the respective equations corresponding to the characteristic wave speed of the flow. The slight oscillations in the *JST* curve due to the increased diffusion of the scheme makes it more difficult to identify the location of the surge front. Figure 19 shows the iso-surfaces of the average density ρ_{ave} computed using the *matrix* dissipation scheme. The progression of the water column collapse can be seen.

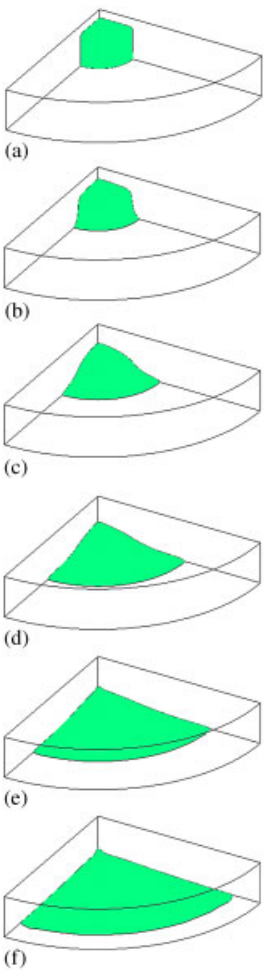

Figure 19. Iso-contours of ρ_{ave} for the three-dimensional breaking dam problem: (a) $t=0.0$; (b) $t=0.6$; (c) $t=1.2$; (d) $t=1.8$; (e) $t=2.4$; and (f) $t=3.0$.

DOI: 10.1002/fld

8. CONCLUSIONS

This paper has described a central difference scheme for the interface capturing scheme. Although unstructured meshes have been used in this study, the methodology could equally be applied to structured mesh codes. Results for a number of test cases have been simulated using the *JST* and *matrix* dissipation. Upwind results have also been calculated for comparison purposes. For these test cases the new hydrostatic pressure discretization ensures that the source term is balanced by the hydrostatic pressure flux term. The results for the internal flows using either *JST* and *matrix* scheme show that this formulation can accurately capture the main physical characteristics of the flow for the test cases considered, but at a lower computational cost compared with the upwind scheme. The *JST* scheme has been shown to introduce a larger amount of interfacial spread but still performs well. This work shows that sufficiently small mesh spacing is required within the region of the interface for these flows to be accurately simulated with the central difference schemes. Insufficient mesh spacing can cause diffusion in the interface that degrades the results. With a mesh spacing $O(10^{-3})$, the current implementation has been shown to predict the flow with reasonable accuracy. From the results of the interface capturing test cases, the standard deviation of the interface was found to be approximately 10 cells on average. Further, mesh spacing reduction would improve the predicted results but this would create very large computational meshes and perhaps another approach, e.g. mesh adaption, is needed in order to produce an implementation that can be used as an efficient design tool.

ACKNOWLEDGEMENTS

This work was funded by EPSRC and BAe Systems. This support is gratefully acknowledged.

REFERENCES

1. Kelecy FJ, Pletcher RH. The development of a free surface capturing approach for multidimensional free surface flows in closed containers. *Journal of Computational Physics* 1997; **138**:939–980.
2. Ferziger JH, Perić M. *Computational Methods for Fluid Dynamics*. Springer: Berlin, 1996.
3. Takizawa A, Koshizuka S, Kondo S. Generalization of physical component boundary fitted coordinate (PCBFC) method for the analysis of free-surface flow. *International Journal for Numerical Methods in Fluids* 1992; **15**:1213–1237.
4. Li YS, Zhan JM. An efficient three-dimensional semi-implicit finite-element scheme for simulation of free surface flows. *International Journal for Numerical Methods in Fluids* 1993; **5**(8):1904–1913.
5. Farmer J, Martinelli L, Jameson A. Fast multigrid method for solving incompressible hydrodynamic problems with free surfaces. *AIAA Journal* 1994; **32**:1175–1182.
6. Ramaswamy B, Kawahara M. Lagrangian finite element analysis applied to viscous free surface fluid flow. *International Journal for Numerical Methods in Fluids* 1987; **7**(9):953–984.
7. Hirt CW, Nichols BD. Volume of fluid VOF method for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**:201–225.
8. Youngs DL. Time-dependent multi-material flow with large fluid distortion. *Numerical Methods for Fluid Dynamics*. Academic Press: London, 1982; 273–285.
9. Unverdi SO, Tryggvason G. A front tracking method for viscous incompressible flows. *Journal of Computational Physics* 1992; **100**:25–37.
10. Pan D, Chang C-H. The capturing of free surfaces in incompressible multi-fluid flows. *International Journal for Numerical Methods in Fluids* 2000; **33**:203–222.
11. Zhao Y, Hui Tan H, Zhang B. A high-resolution characteristics based dual time-stepping VOF method for free surface flow simulation on unstructured grids. *Journal of Computational Physics* 2002; **183**:233–272.

DOI: 10.1002/fld

12. Qian L, Causon M, Ingram DM, Mingham CG. Cartesian cut cell two-fluid solver for hydraulic flow problems. *Journal of Hydraulic Engineering* 2003; **129**(9):688–696.
13. Arnone A, Liou M-S, Povinelli LA. Multigrid time-accurate integration of the Navier–Stokes equations. *AIAA 93-3361*, 1993.
14. Jameson A, Schmidt W, Turkel E. Numerical solution of the Euler equations by finite volume methods using Runge–Kutta time stepping schemes. *AIAA Paper 81-1259*, 1981.
15. Gaitonde AL. A dual-time method for two-dimensional unsteady incompressible flow calculations. *International Journal for Numerical Methods in Engineering* 1998; **41**:1153–1166.
16. Gough HJ. Incompressible interface capturing methods for application to free surfaces and cavitation. *Ph.D. Thesis*, University of Bristol, 2007.
17. Acheson DJ. *Elementary Fluid Dynamics*. Oxford Applied Mathematics and Computing Science Series. Oxford University Press: Oxford, 2003.
18. Brackbill JU, Kothe DB, Zemach C. A continuum method for modelling surface tension. *Journal of Computational Physics* 1992; **100**:335–354.
19. Chorin AJ. A numerical method for solving incompressible viscous flow. *Journal of Computational Physics* 1967; **2**:12–26.
20. Jameson A, Mavriplis D. Finite volume solution of the two-dimensional Euler equations on a regular triangular mesh. *AIAA Paper 85-0435*, 1985.
21. Jameson A. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. *AIAA Paper 91-1596*, 1991.
22. Tejera-Cuesta P. A parallel dual-time multigrid method for incompressible flows over deforming geometries. *Ph.D. Thesis*, University of Bristol, 2000.
23. Qian L, Causon DM, Mingham CG, Ingram DM. A free-surface capturing method for two fluid flows with moving bodies. *Proceedings of the Royal Society A* 2006; **462**:21–42.
24. Swanson RC, Turkel E. On central-difference and upwind schemes. *Journal of Computational Physics* 1992; **101**:292–306.
25. Sweby PK. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM Journal on Numerical Analysis* 1984; **21**:995–1011.
26. Jorgensson P, Turkel E. Central difference TVD schemes for time dependent and steady state problems. *Journal of Computational Physics* 1993; **107**:297–308.
27. Swanson RC, Radespiel R, Turkel E. On some numerical dissipation schemes. *AIAA 97-1945*, 1993.
28. Hino T. An unstructured grid method for incompressible viscous flows with a free surface. *AIAA-1997-0862*, 1997.
29. Löhner R, Yang C, Oñate E, Idelssohn S. An unstructured grid-based, parallel free surface solver. *AIAA-1997-1830*, 1997.
30. Hino T. A finite-volume method with unstructured grid for free surface flow simulations. *Proceedings of 6th International Symposium on Numerical Ship Hydrodynamics*, Iowa City, 1993.
31. Park J-C, Kim M-H, Miyata H. Fully non-linear free surface simulations by a 3D viscous numerical wave tank. *International Journal for Numerical Methods in Fluids* 1999; **29**:685–703.
32. Wasistho B, Geurts BJ, Kuerten GM. Simulation techniques for spatially evolving instabilities in compressible flow over a flat plat. *Computers and Fluids* 1997; **26**(7):713–739.
33. Liu S, Liu C. Fourth order finite difference and multigrid methods for modelling instabilities in flat plate boundary layers. *Computers and Fluids* 1994; **23**:955–982.
34. Roe PL. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics* 1981; **43**:357.
35. Swanson RC, Turkel E. On central difference and upwind schemes. *Technical Report 182061*, ICASE, 1990.
36. Leatham M, Stokes S, Shaw JA, Cooper J, Appa J, Blaylock TA. Automatic mesh generation for rapid-response Navier–Stokes calculations. *AIAA2000-2247*, 2000.
37. Rayleigh L. Investigation of the character of the equilibrium of an incompressible heavy fluid of variable density. *Scientific Papers* 1900; 200.
38. Taylor GI. The instability of liquid surfaces when accelerated in a direction perpendicular to their planes. *Proceedings of the Royal Society of London, Series A* 1950; **201**:192.
39. Chandrasekhar S. *Hydrodynamic and Hydromagnetic Stability*, Chapter X. Oxford University Press: Oxford, 1961; 428–447.
40. Daly BJ. Dynamics of liquids in moving containers. *Physics of Fluids* 1967; **10**:297.

DOI: 10.1002/fld

41. Daly JB. Numerical study of two fluid Rayleigh–Taylor instability. *Physics of Fluids* 1967; **10**(2):297–307.
42. Walhorn E, Kölke A, Hübner B, Dinker D. Fluid–structure coupling within a monolithic model involving free surface flows. *Computers and Structures* 2005; **83**:2100–2111.
43. Greaves D. Simulation of viscous water column collapse using adapting hierarchical grids. *International Journal for Numerical Methods in Fluids* 2005; **50**(6):693–711.
44. Ubbink O. Numerical prediction of two fluid systems with sharp interfaces. *Ph.D. Thesis*, Imperial College, University of London, 1997.
45. Andrillon Y, Bertrand A. A 2D+T VOF fully coupled formulation for the calculation of breaking free-surface flow. *Journal of Marine Science and Technology* 2004; **8**:159–168.
46. Martin JC, Moyce WJ. An experimental study of the collapse of liquid columns on a rigid horizontal plane. *Philosophical Transactions of the Royal Society of London*, *Series A*: *Mathematical*, *Physical and Engineering Sciences* 1952; **244**:312–324.
47. Koshizuka S, Tamako H, Oka Y. A particle method for incompressible viscous flow with fluid fragmentation. *Computational Fluid Mechanics Journal* 1995; **4**:29–46.

DOI: 10.1002/fld