INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2004; **44**:1093–1117 (DOI: 10.1002/fld.687)

Simulation of interface and free surface flows in a viscous fluid using adapting quadtree grids

Deborah Greaves[*,†,‡]

Department of Architecture and Civil Engineering, University of Bath, Claverton Down, Bath BA2 7AY, U.K.

SUMMARY

A new adaptive quadtree method for simulating laminar viscous fluid problems with free surfaces and interfaces is presented in this paper. The Navier–Stokes equations are solved with a SIMPLE-type scheme coupled with the Compressive Interface Capturing Scheme for Arbitrary Meshes (CICSAM) (Numerical prediction of two fluid systems with sharp interfaces, *Ph.D. Thesis*, Imperial College of Science, Technology and Medicine, London, 1997) volume of fluid (VoF) method and PLIC reconstruction of the volume fraction field during refinement and derefinement processes. The method is demonstrated for interface advection cases in translating and shearing flow fields and found to provide high interface resolution at low computational cost. The new method is also applied to simulation of the collapse of a water column and the results are in excellent agreement with other published data. The quadtree grids adapt to follow the movement of the free surface, whilst maintaining a band of the smallest cells surrounding the surface. The calculation is made on uniform and adapting quadtree grids and the accuracy of the quadtree calculation is shown to be the same as that made on the equivalent uniform grid. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: quadtree adaptive grids; volume of fluid method; free surface; interface; viscous fluid

1. INTRODUCTION

There are many engineering applications where simulation of a viscous fluid with a free surface or interface with another fluid is important. For example, wave sloshing in tanks, wave loading and run-up on ships and marine and coastal structures, bubble flow in pipes and multiphase flows. Accurate modelling of a viscous fluid free surface flow is an extremely challenging problem in CFD because of the moving air–water interface together with the non-linear governing equations and boundary conditions. In addition, the position of the free surface at a given time is not known in advance and must be calculated as part of the solution.

[*] Correspondence to: D. Greaves, Department of Architecture and Civil Engineering, University of Bath, Claverton Down, Bath BA2 7AY, U.K.

[†] E-mail: d.m.greaves@bath.ac.uk

[‡] Royal Society University Research Fellow.

Contract/grant sponsor: Royal Society

Received 3 June 2003
Revised 15 November 2003

Potential flow models, in which the fluid is considered inviscid, have been used successfully to simulate inertia driven effects [1, 2], such as wave diffraction and loading. However, they are unable to predict viscous effects, such as vortex shedding and separation, which require the Navier–Stokes equations to be solved. The Navier–Stokes equations are non-linear and are linked, through the pressure, to the mass conservation equation. Their solution is usually by an iteration technique that resolves the complex coupling between pressure and velocity, such as the pressure correction or artificial compressibility method [3].

Various techniques have emerged to predict the position of the free surface during the solution in time and fall into one of two categories. These are interface tracking methods, which include moving mesh, front tracking [4] and particle tracking schemes [5]; and interface capturing methods, which include volume of fluid (VoF) and level set techniques. Moving mesh and front tracking methods typically solve the flow equations in the liquid region only and the free surface makes a moving upper boundary on the computational domain. These methods may be accurate, but, as the free surface has to be single valued, cannot be used to calculate major deformations of the interface such as breaking waves where the interface overturns, breaks up and recombines. Particle tracking methods tend to be expensive and not practical in three dimensions. On the other hand, front capturing methods can be used for modelling large-scale deformations of the interface including wave break up and merging. They differ from front tracking in that the solution is calculated in the combined air and water fluid domains, with the fluid properties changing at the interface. The interface is then located from the zero contour of a distance function in the case of level set [6] and from the volume fraction field in the VoF method [7].

A VoF methodology is selected for use in this work; the basic method is robust and flexible and VoF schemes are widely used [7–9]. The major drawbacks of this method are its tendency to smear the interface and the high CPU cost due to the need for fine grids and small time steps. In order to overcome these problems, a new method is proposed in this work in which the high resolution CICSAM [10] interface advection scheme is implemented on adapting quadtree grids [11]. The Navier–Stokes equations are solved using a SIMPLE discretization of the convective terms together with collocated variables. Special interpolations are needed at the interface between cells of different size, i.e. at hanging nodes. Adapting quadtree grids provide extra resolution locally in areas of interest and have proved successful in the simulation of separated flows [11], where the overall size of the grid is reduced significantly for a given accuracy by providing high resolution where the flow variables are changing most rapidly.

Here, the quadtree grids provide high resolution in a band surrounding the free surface; the interface remains sharp and is tracked by adapting refinements in the quadtree grid. Results are calculated on uniform grids and adapting quadtree grids in which the smallest cells are the same size as those in the uniform grid. In this way, it is demonstrated that the same accuracy may be achieved using the adapting quadtree scheme as on the equivalent uniform grid.

2. THE VOLUME OF FLUID METHOD, VoF

When considering the incompressible flow of two immiscible fluids in two dimensions, the divergence free velocity field $\mathbf{u}(x, y, t)$ obeys

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

The location of the two fluids is specified using a volume fraction function, C, with $C = 1$ inside one fluid and $C = 0$ in the other. Cells for which C lies between 0 and 1 contain the interface. The volume conservation of the first fluid can be expressed as

$$\frac{\partial C}{\partial t} + \nabla \cdot (\mathbf{u}\mathrm{C}) = 0 \tag{2}$$

and in discrete form is given by

$$C^{n+1} = C^n + \sum_{k=1}^{K} f_k \tag{3}$$

where k is the cell face orientation and K is the number of cell faces. For a regular rectangular grid in two dimensions, $K = 4$ and e, w, n or s (the east, west, north or south face). f_k represents the flux of C across the k direction cell boundary. Thus,

$$C^{n+1} = C^n + f_{\mathrm{e}} - f_{\mathrm{w}} + f_{\mathrm{n}} - f_{\mathrm{s}} \tag{4}$$

and the cell face fluxes are

$$f_k = u_k \frac{\delta t}{\delta x} C_k \tag{5}$$

where δt and δx are the time step and mesh size, u_k is the velocity at face k and determination of the cell face value of C is critical. The key to a successful VoF scheme is to solve the volume fraction equation in a way that keeps the interface sharp.

A VoF method consists of two parts: an interface reconstruction algorithm for approximating the interface from the set of volume fractions, and a VoF transport algorithm for determining the volume fractions at the new time level from the velocity field and the reconstructed front. The original VoF scheme of Hirt and Nichols [7] uses an interface construction that approximates a curved interface as horizontal and vertical lines in each interface cell. The fluxing scheme uses a combination of upwinding and downwinding. The advantage of the upwind scheme is that it is stable, but it is diffusive and may spread the interface over many cells. The downwind scheme is unstable, but sharpens the interface and so is advantageous in interface tracking. Various VoF fluxing methods have been developed, most of which aim for a balance between the stability advantages of the upwind scheme and the front sharpening advantages of the downwind scheme.

Hirt and Nichols [7] fluxing scheme uses either upwinding or a downwinding donor–acceptor cell approach depending on the local orientation of the interface. Lafaurie *et al.* [12] describe a modified downwinding scheme to constrain the volume fraction face value to prevent more fluid being outfluxed than the cell contains and to ensure that the CFL condition is satisfied. Lafaurie *et al.* [12] also introduced a flotsam indicator field, which is used to ensure that flotsam (a cell partly containing one of the fluids entirely surrounded by cells full of the other fluid) is fluxed by upwinding rather than downwinding, which would lock it in place and prevent it being fluxed. This scheme is implemented in the SURFER program.

Traditional VoF schemes describe the interface implicitly since the volume fraction data must be inverted to find the approximate interface position. The interface may be reconstructed by simple line interface calculation (SLIC) [13] or by various piecewise linear interface calculation (PLIC) methods [14]. Gueyffier *et al.* [15] developed a VoF/PLIC method and applied it to simulation of three-dimensional droplets. From the volume fraction field, the normal vector to the interface is constructed. The interface line that divides the cell into two parts, containing the proper volume of each fluid, is calculated from the normal vector and required volume of one of the fluids. The line is advected with the local velocity over the calculation time step and the process is then reversed to determine the volume fraction at the next time step from the new position of the line.

3. CICSAM VoF

Ubbink [10] derived a compressive differencing scheme for discretization of the volume fraction equation (2). The scheme is named Compressive Interface Capturing Scheme for Arbitrary Meshes (CICSAM) and is based on the Normalized Variable Diagram (NVD) used by Leonard [16]. Ubbink's [10] scheme combines the Convection Boundedness Criteria (CBC) with the Ultimate Quickest (UQ) differencing scheme, which is a version of Leonard's [17] QUICK scheme. The normalized face value for the CICSAM differencing scheme is calculated by combining these two schemes through a weighting factor derived from the orientation of the interface relative to the fluxing direction.

In the CICSAM scheme, the cell face values of C, used in the discretized volume fraction equation, are determined from a combination of the CBC value given by

$$\tilde{C}_{f\mathrm{CBC}} = \begin{cases} \min\left\{1, \dfrac{\tilde{C}_\mathrm{D}}{c_\mathrm{D}}\right\} & \text{when } 0 \leqslant \tilde{C}_\mathrm{D} \leqslant 1 \\ \tilde{C}_\mathrm{D} & \text{when } \tilde{C}_\mathrm{D} < 0,\ \tilde{C}_\mathrm{D} > 1 \end{cases} \tag{6}$$

and the UQ value given by

$$\tilde{C}_{f\mathrm{UQ}} = \begin{cases} \min\left\{\dfrac{8c_\mathrm{D}\tilde{C}_\mathrm{D} + (1 - c_\mathrm{D})(6\tilde{C}_\mathrm{D} + 3)}{8}, \tilde{C}_{f\mathrm{CBC}}\right\} & \text{when } 0 \leqslant \tilde{C}_\mathrm{D} \leqslant 1 \\ \tilde{C}_\mathrm{D} & \text{when } \tilde{C}_\mathrm{D} < 0,\ \tilde{C}_\mathrm{D} > 1 \end{cases} \tag{7}$$

Here $\tilde{C}_\mathrm{D}$ is the normalized variable for the donor cell, calculated from

$$\tilde{C}_\mathrm{D} = \frac{C_\mathrm{D} - C_\mathrm{U}}{C_\mathrm{A} - C_\mathrm{U}} \tag{8}$$

where subscript U indicates the upwind cell, A the acceptor and D the donor cell. These are determined depending on the velocity at a given face. With reference to Figure 1, when considering the east face of cell C, if $u_\mathrm{e} > 0$ then the donor is cell C and cell E is the acceptor; however if $u_\mathrm{e} < 0$ then cell E is the donor and cell C the acceptor. The Courant number, c_D, is calculated by summing the fluxes over each cell face

$$c_\mathrm{D} = \sum_{f=1}^{n} \max\left\{\frac{-F_f\,\delta t}{V_\mathrm{D}}, 0\right\} \tag{9}$$

where n is the number of faces, F_f is the volumetric flux across a given face calculated from the product of the face velocity and face area, δt is the time step and V_D the cell volume.

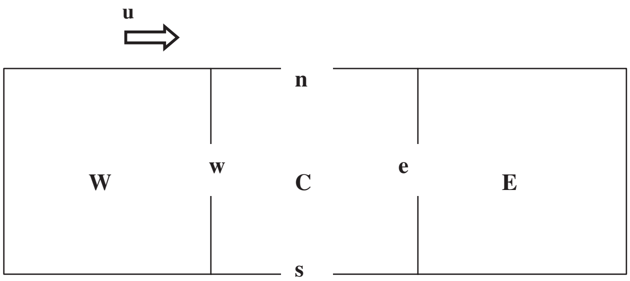


Figure 1. Diagram for x-direction fluxing.

The weighting factor used to combine the CBC and UQ contributions takes into account the orientation of the interface and the fluxing direction

$$\gamma_f = \min\left\{k_\gamma \frac{\cos(2\theta_f)+1}{2}, 1\right\} \tag{10}$$

where θ_k is the orientation angle of the interface with the fluxing direction and k_γ is a constant introduced to control the dominance of the different schemes. Ubbink [10] recommends a value of $k_\gamma = 1$. The angle, θ_k, relative to k-direction fluxing is

$$\theta_k = \arccos(n_k) \tag{11}$$

where n_k is the k-direction component of the local approximation to the interface normal,

$$\mathbf{n} = \frac{\nabla^h C}{|\nabla^h C|} \tag{12}$$

and ∇^h is a finite difference approximation to the gradient operator.

The normalized face value for the CICSAM differencing scheme is then

$$\tilde{C}_f = \gamma_f \tilde{C}_{\mathrm{fCBC}} + (1-\gamma_f)\tilde{C}_{\mathrm{fUQ}} \tag{13}$$

The weighting factor is given by

$$\beta_f = \frac{\tilde{C}_f - \tilde{C}_{\mathrm{D}}}{1 - \tilde{C}_{\mathrm{D}}} \tag{14}$$

and the face values for the new volume fraction distribution used to solve the discrete volume fraction equation are

$$C_f^* = (1-\beta_f)\frac{C_{\mathrm{D}}^t + C_{\mathrm{D}}^{t+\delta t}}{2} + \beta_f \frac{C_{\mathrm{A}}^t + C_{\mathrm{A}}^{t+\delta t}}{2} \tag{15}$$

The superscripts refer to the time level, thus the volume fraction field is advected using a Crank–Nicolson scheme. It is possible for these volume fraction values to have non-physical values, less than 0 or greater than 1. In the event of this occurring, Ubbink [10] recommends a corrector step that corrects the weighting factor β_f.

4. QUADTREE GRID GENERATION

Quadtree grids are created about a set of discrete seeding points by recursive subdivision of a unit square that at the root level surrounds the flow domain. An advantage of quadtree grids is they may be stored in a concise grid cell reference numbering system, which contains all the grid information and forms a tree data structure. The data tree can be traversed according to simple rules to obtain grid cell reference numbers of ancestor and neighbour cells, also the grids may be readily adapted by addition or subtraction of grid cells whilst maintaining the overall tree structure.

Samet [18, 19] describes the quadtree data structure and its application to spatial data problems. Although quadtree algorithms were first used in image processing, they have also been used extensively as mesh generators. Yerry and Shephard [20] applied quadtree algorithms in creating finite element meshes for structural analysis, as did Messaoud [21] for elliptic partial differential systems. Van Dommelen and Rundensteiner [22] modelled flow past a cylinder using a discrete vortex scheme, with adaptive remeshing based on quadtrees. Multigrid-quadtree meshes have been applied [23] to the solution of species transport and linearized shallow flow problems in complex domains. Quadtree finite element methods using quadrilateral and cubic elements have been used for compressible flows [24] and quadtree finite volume methods for solutions of the Euler equations [25]. Greaves and Borthwick [11, 26] demonstrated hierarchical grid generation in two dimensions for quadtree grids and in three dimensions for octree grids. Hu *et al.* [27] present an adaptive hierarchical tritree grid method with a triangular finite volume scheme for simulating laminar fluid flow.

4.1. The quadtree algorithm

The quadtree algorithm can be summarized as follows:

(1) Define the set of boundary seeding points, P_{n}, about which the grid will be generated.
(2) Define the unit square or rectangle (root cell) which surrounds the normalised domain of interest.
(3) Divide the root cell into four quadrant cells.
(4) Consider each cell; if the cell contains more than two points, continue with (5) otherwise check the next cell.
(5) Check whether the maximum division level, $M_{\max}$, has been reached. If so, the division of the cell in question is complete, so go to (4) and check the next cell. When all cells considered either have reached the maximum division, $M_{\max}$, or contain less than three points, the mesh generation is complete. Otherwise continue.
(6) Divide the cell into four cells, return to (4) and check the next cell.

Additional panels are generated to regularize the grid such that the maximum panel edge length ratio between two adjacent panels does not exceed two. This will limit the variation of neighbour arrangements encountered when solving discretized equations on the grid. Figure 2 shows the quadtree grid generated for a circular interface with radius equal to 0.125 centred at $(-0.2, 0.2)$, the origin is at the centre of the root cell. The quadtree grid has a maximum division level equal to 7 and minimum division level equal to 2. In Figure 2(a) the grid is not regularized and the large difference in size of adjacent cells would make solution of the discrete equations very difficult. In Figure 2(b) the grid is regularized to limit the edge length

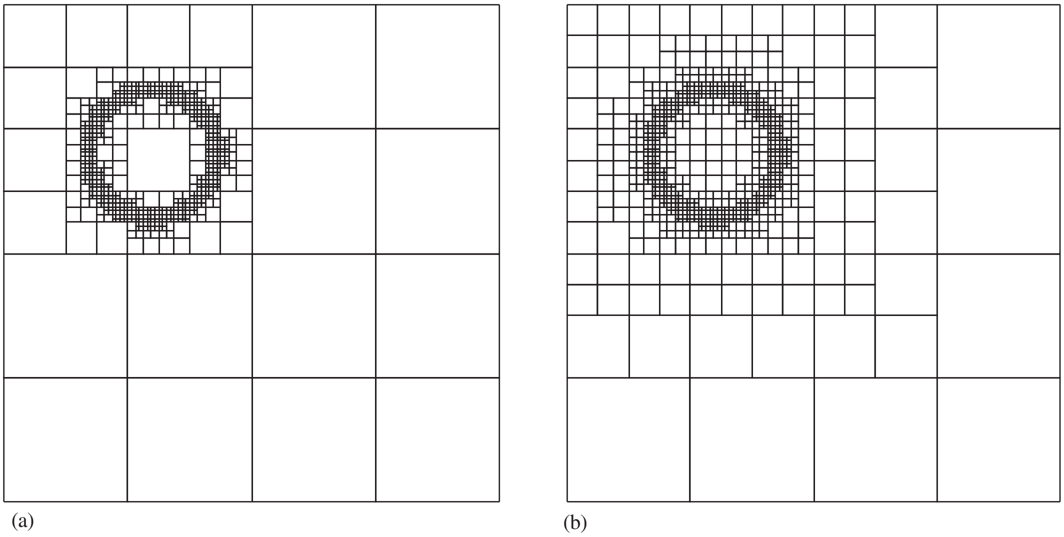


Figure 2. (a) Quadtree grid without regularization and (b) regularized quadtree grid.

ratio; whilst still maintaining a large difference in cell size throughout the grid, adjacent cells are no more than twice the size of one another. A quadtree panel in a regularized grid has eight possible face and corner neighbours of the same size, and 12 possible neighbours of one level smaller in size.

4.2. *The quadtree numbering system*

Several numbering systems are possible for storing the quadtree information, as discussed by Yiu *et al.* [28]. Here, the numbering system is essentially due to Van Dommelen and Rundensteiner [22], and enables the tree reference numbers to be stored as an array of binary digits. The reference number, N, of a given cell can be regarded as a list of successive orientations NW, SW, NE or SE corresponding to binary translations 00, 01, 10 and 11, which describe the position of the cell within its parent, and is stored in four arrays of eight digits. If the large square in Figure 3 is the root of the tree, then the four smaller squares are the children of the root cell, created at the first division. The full reference number for the NE child is given by, $N(I,1)=21000000$, $N(I,2)=00000000$ $N(I,3)=00000000$ $N(I,4)=00000000$ (note that $+1$ has been added to each significant digit in order to distinguish it from the trailing zeros).

Whenever division takes place, four cells are produced having different x and y translations from the centre of the parent cell. These digits correspond to locations NW, SW, NE or SE and are appended to the number of the parent cell being divided to produce the reference number of each child or new cell. Hence, the reference numbers of the parent and all ancestor cells are contained within the reference number of a given cell. The cell reference number may be manipulated to extract the generation level of the cell, the reference number of the parent cell, the co-ordinates of the cell centre and the reference numbers of all possible neighbours of a cell. The data tree may then be searched to locate the cell neighbours within the grid. The

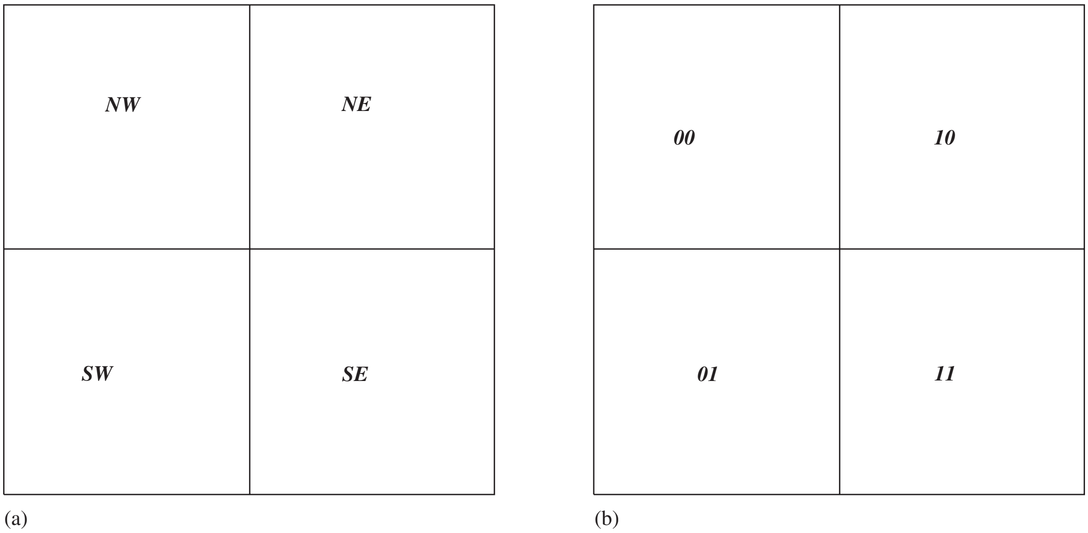


Figure 3. (a) Panel orientations and (b) corresponding binary translations.

manipulation of the numbering system to obtain all of the necessary information is described in detail by Greaves [29].

5. SOLUTION OF THE NAVIER–STOKES EQUATIONS

The governing equations in primitive form for a two-dimensional flow are the mass conservation equation

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} = 0 \tag{16}$$

and the Navier–Stokes momentum conservation equations

$$\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\nabla^2 u \tag{17}$$

$$\frac{\partial v}{\partial t} + \frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y} = -g - \frac{1}{\rho}\frac{\partial p}{\partial y} + \nu\nabla^2 v \tag{18}$$

where x and y define an orthogonal Cartesian co-ordinate system, u and v are the corresponding velocity components, t is time, p is pressure, ρ is the fluid density, g is the gravitational acceleration and ν is the fluid kinematic viscosity.

The governing equations are discretized using finite volumes with collocated primitive variables (u, v and p all defined at cell centres) and solved by a pressure correction method based on that described by Perić *et al.* [30]. The solution method is similar to SIMPLE [31], but applied on quadtree grids with interpolations to deal with the non-staggered arrangement of variables, and is described in detail by Greaves and Borthwick [11]. The discretized momentum equations given by

$$a_{\mathrm{P}}u_{\mathrm{P}} = \sum a_{\mathrm{nb}}u_{\mathrm{nb}} + a_{\mathrm{P0}}u_{\mathrm{P0}} + \frac{(p_{\mathrm{W}} - p_{\mathrm{E}})\,\mathrm{d}y}{2} \tag{19}$$

and

$$a_{\mathrm{P}}v_{\mathrm{P}} = \sum a_{\mathrm{nb}}v_{\mathrm{nb}} + a_{\mathrm{P0}}v_{\mathrm{P0}} + \frac{(p_{\mathrm{S}} - p_{\mathrm{N}})\,\mathrm{d}x}{2} \tag{20}$$

are summed over all cell faces, where subscript P indicates the cell in question, nb are the neighbour cells, W is the west, E the east, S is the south and N the north neighbour. Subscript 0 indicates values at the previous time step and a_k is a coefficient combining the convective and diffusive fluxes.

Here, the pressure gradient term has been discretized using first-order central differences. A similar approach for the velocity gradient terms in the mass conservation equation would result in a solution independent of the pressure and checkerboard oscillations in the velocity field. The face interpolation proposed by Rhie and Chow [32] is adopted here to prevent these oscillations occurring. The face value of the velocity is predicted by isolating the contribution of the pressure from the discretized momentum equation when interpolating it to the face. The contribution of the pressure gradient at the face is then added by calculating it directly from the pressure values at cell neighbours on either side of the face. Hence, the east face value is calculated from

$$u_{\mathrm{e}} = \left(\overline{\frac{\sum a_{\mathrm{nb}}u_{\mathrm{nb}} + a_{\mathrm{P0}}u_{\mathrm{P0}}}{a_{\mathrm{P}}}}\right)_{\mathrm{e}} + \left(\overline{\frac{1}{a_{\mathrm{P}}}}\right)_{\mathrm{e}} (p_{\mathrm{P}} - p_{\mathrm{E}})\,\mathrm{d}y \tag{21}$$

where the overbar indicates linear interpolation between cell centre values at cell P and E. This method for calculating the face velocity is used both in the pressure correction equation and in the discretized volume fraction equation.

5.1. Hanging node treatment

Hanging nodes are inherent in quadtree grids and occur at the centre of a cell face where cells of different size meet. In order to conserve fluxes, contributions from all cells neighbouring a given face are used to calculate the fluxes across the face. For example, Equation (4) applied to cell P in Figure 4 is

$$C^{n+1} = C^{n} + f_{\mathrm{e1}} + f_{\mathrm{e2}} - f_{\mathrm{w}} + f_{\mathrm{n}} - f_{\mathrm{s}} \tag{22}$$

where f_k is the flux of C across the k-direction boundary. Neighbouring values of pressure and velocity for the discretized momentum and pressure correction equations, however, are calculated by averaging the neighbour values where the neighbour cells are smaller than the cell in question, or by bi-linear interpolation where the neighbour cell is larger than the cell in question. With reference to Figure 5, the pressure value to the east of cell P used in the discretized equations is given by

$$p^{*} = \frac{(6p_{\mathrm{E}} + 2p_{\mathrm{NE}} + 4p_{\mathrm{P}})}{12} \tag{23}$$

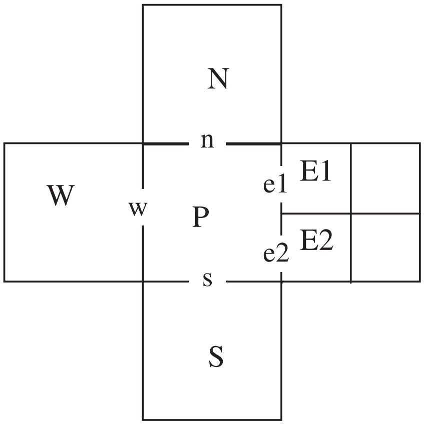


Figure 4. Hanging node treatment for fluxes.

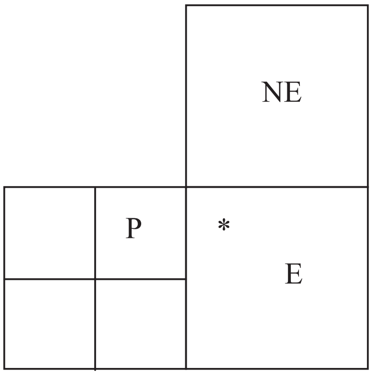


Figure 5. Interpolation at grid size interface.

The problem of hanging node interpolation across cells in different fluids does not arise because a band of refinement is provided around the interface thus ensuring that hanging nodes are kept away from the interface.

5.2. Adaptive grid scheme

An advantage of quadtree grids is that they can be readily adapted by the addition and removal of cells throughout a time-dependent simulation. In this work, grid refinement is used to follow the movement of the interface and in addition, for the collapsing water column calculations, refinement is provided at the base of the domain and in a band surrounding the interface.

Remeshing of the grid operates by dividing a cell into four if it lies on the interface (or in a band surrounding the interface). Derefinement also takes place by removing four sibling cells and replacing them with their parent. This only occurs where each of the four sibling cells lies away from the interface. Velocity and pressure variables are interpolated onto new cells using bi-linear interpolation from the neighbours of the divided cell. Alternatively, when

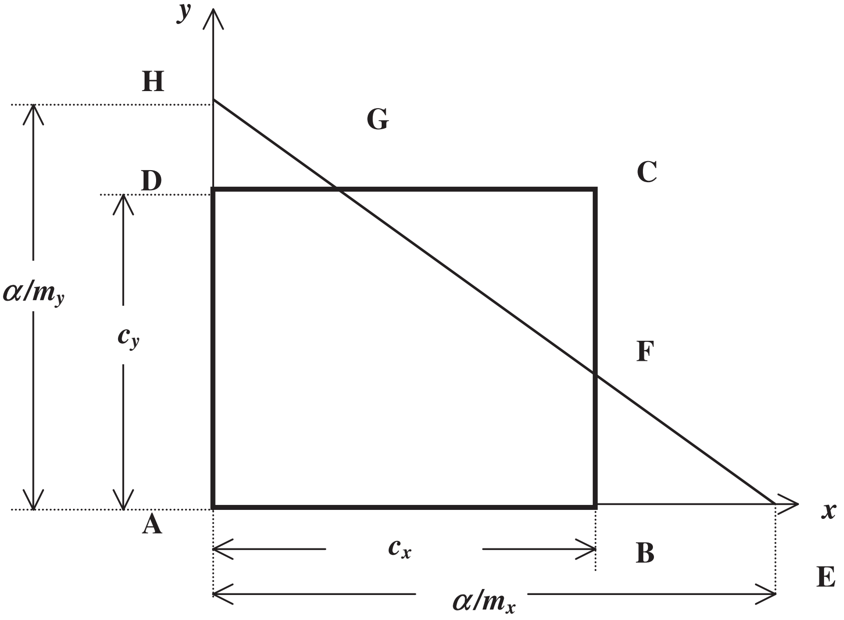


Figure 6. Cell ABCD is cut by the straight line EH, having normal **m** and parameter α, and contains fluid 1 in region ABFGD and fluid 2 in region FCG.

four sibling cells are removed and replaced with their parent, the variables assigned to the parent are the average of the four sibling values.

Interpolation and extrapolation of the volume fraction, C, is more complicated. The PLIC [15] reconstruction of the interface in the divided cell is transformed to the co-ordinates of each newly created cell and the volume fraction determined from the equation of the interface line segment in each new cell. In the PLIC reconstruction of the interface, a straight line is defined in each interface cell that divides the cell into two parts, each of which contains the correct volume of one of the two fluids. The equation for a straight line with normal **m** is

$$m_x x + m_y y = \alpha \tag{24}$$

where m_x and m_y are components of **m** and α is a parameter related to the smallest distance between the line and the origin. The interface normal is similar to the normal vector determined using Equation (12), but in this case the vector is not normalized and is calculated from

$$\mathbf{m} = \nabla^h C \tag{25}$$

The points at which the line intersects the x- and y-axis are at α/m_x and α/m_y, respectively, points E and H in Figure 6. The area below the line and contained within the cell ABCD is

$$\text{Area} = \frac{\alpha^2}{2m_x m_y}\left[1 - H(\alpha - m_x c_x)\left(\frac{\alpha - m_x c_x}{\alpha}\right)^2 - H(\alpha - m_y c_y)\left(\frac{\alpha - m_y c_y}{\alpha}\right)^2\right] \tag{26}$$

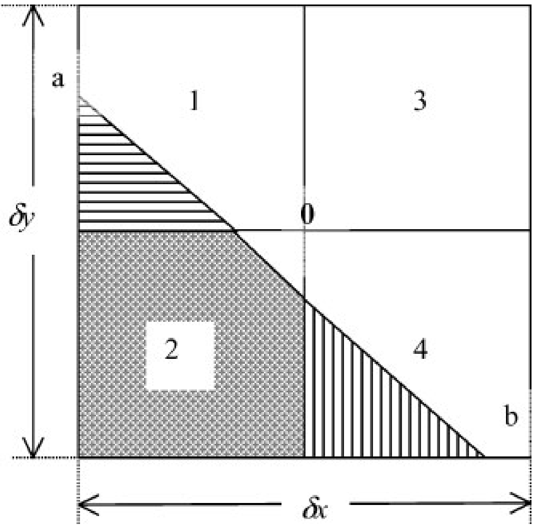


Figure 7. The shaded areas show the volume fraction of fluid in new cells 1, 2 and 4 after refinement of cell 0.

where c_x and c_y are the cell dimensions and

$$H(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x > 0 \end{cases}$$

Figure 7 shows the interface line segment ab in original cell 0, which after cell refinement lies in new cells 1, 2 and 4 only. The cell height and width in Equation (26) become $c_x = \delta x/2$, $c_y = \delta y/2$ where δx and δy are the dimensions of the parent cell and the parameter α is transformed as follows for each of the new cells:

$$\begin{aligned} &\text{Cell 2}: \quad \alpha^* = \alpha \\ &\text{Cell 3}: \quad \alpha^* = \alpha - m_y\delta y/2 - m_x\delta x/2 \\ &\text{Cell 4}: \quad \alpha^* = \alpha - m_x\delta x/2 \end{aligned} \tag{27}$$

The grid is only derefined away from the interface, so the volume fraction for each of the removed siblings and their parent will be either 0 or 1.

6. RESULTS

6.1. Interface tracking in a prescribed velocity field

In order to assess the capabilities of the interface tracking part of the numerical scheme, tests are first carried out for square and circular interfaces being advected through a prescribed velocity field. In these cases, only the volume fraction equation and not the Navier–Stokes equations are solved.

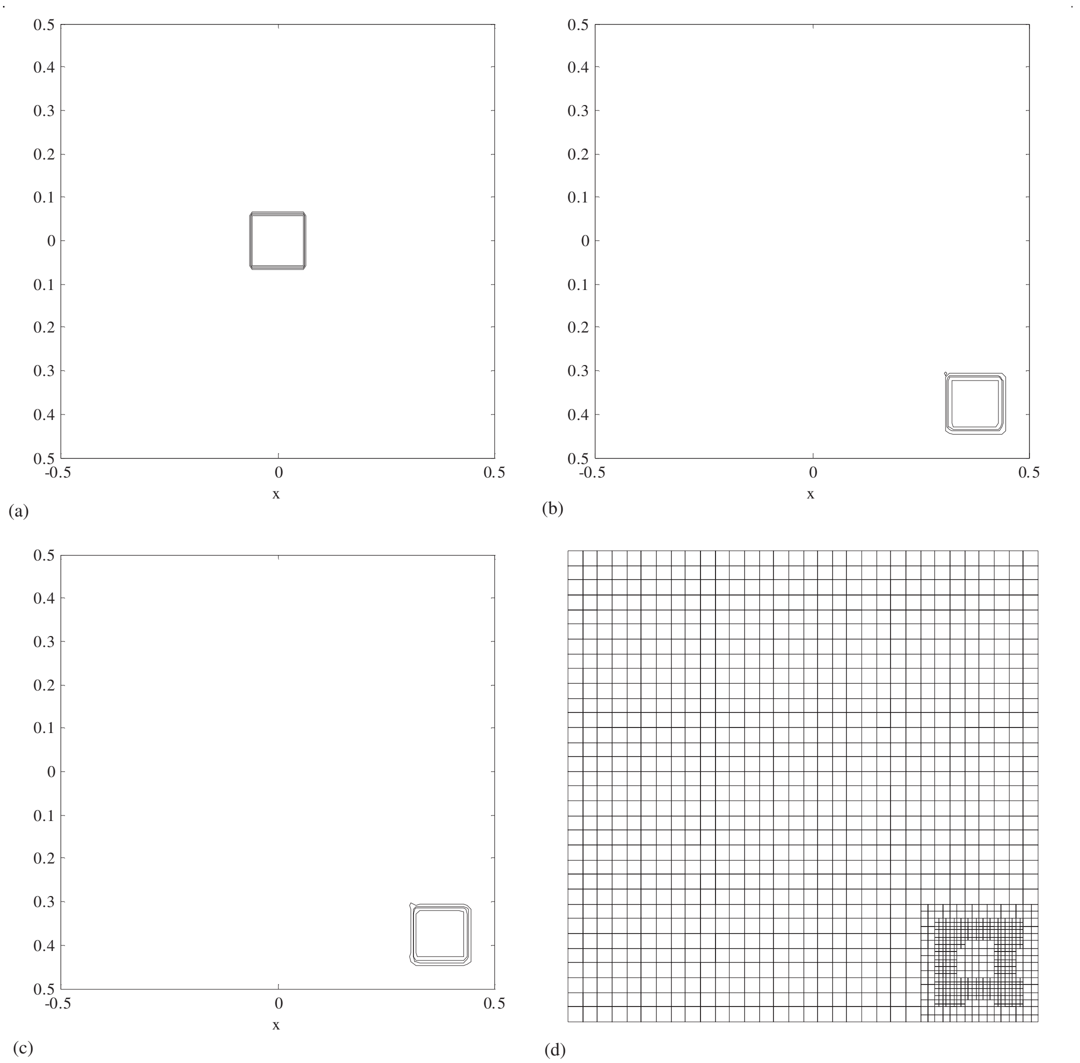


Figure 8. Advection of square interface on adaptive quadtree grids. (a) Initial interface contours, $t = 0$ s; (b) uniform grid, $t = 0.375$ s; (c) quadtree grid, $t = 0.375$ s and (d) adapted quadtree grid.

6.1.1. Comparison of uniform and adapting quadtree schemes Uniform translation of a square interface is first investigated on uniform and quadtree adapted grids in order to assess the benefits of quadtree grid adaptation. Here, the quadtree grids have a maximum division level equal to 7 and minimum division level equal to 5 and are compared with a uniform grid of the same smallest cell size, containing 128×128 cells. The grids are adapted such that cells containing the interface are refined to the maximum level and cells that no longer contain the interface are derefined to the minimum level.

A unit square domain is used and the initial contours for the square interface, size 0.125×0.125, positioned at the centre of the grid, are shown in Figure 8(a). The interface is translated

Table I.

		Error	Grid size	CPU (s)
Translation of square	CICSAM uniform grid	$1.189\mathrm{e}^{-5}$	16 884	368.76
	CICSAM adapted grid	$1.178\mathrm{e}^{-5}$	1500	77.18

in a constant velocity field, $u = 1$, $v = -1$ and the time step is determined such that the Courant number is equal to 0.125. The value of the Courant number is based on the smallest cell size and must be no greater than one to maintain stability of the numerical scheme. Several values of Courant number were investigated and 0.125 found to achieve a good balance between speed of calculation and sharpness of interface. The results are presented by plotting volume fraction contours at $C = 0.05$, 0.4, 0.6 and 0.95. With an accurate interface advection scheme the interface should be translated intact towards the bottom right-hand corner of the domain. Figures 8(b) and 8(c) show the interface at time $t = 0.375$ s calculated on the regular and quadtree grids and Figure 8(d) shows the adapted quadtree grid at time $t = 0.375$ s.

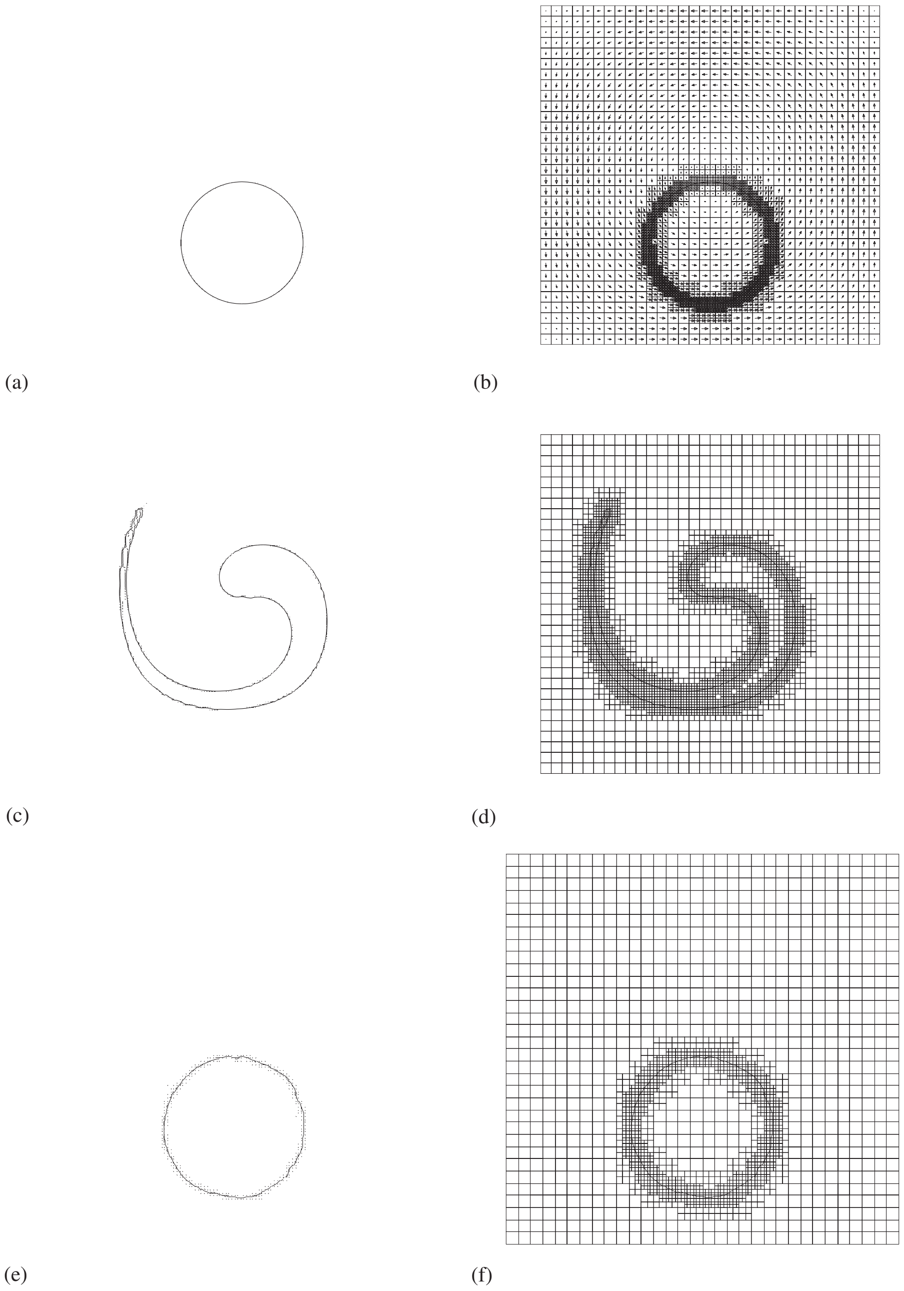
Table I lists the error, which for a grid of n cells is calculated as

$$\text{error} = \left| \frac{\sum_{i=1}^{n} C_i \delta x\, \delta y - \sum_{i=1}^{n} C_i^{\text{initial}} \delta x\, \delta y}{\sum_{i=1}^{n} C_i^{\text{initial}} \delta x\, \delta y} \right| \tag{28}$$

The error calculated for the adaptive quadtree grid scheme is slightly less than that calculated for the uniform grid scheme. The adapted grid contains typically 1500 cells compared with the uniform grid, which contains 16 384 cells; the CPU time for each calculation is also listed in Table I. Using the CICSAM scheme, the quadtree adapted grid size is more than ten times less than the uniform grid for similar accuracy and the CPU time used is nearly five times less than for the uniform grid calculation. Clearly, despite the extra CPU cost of grid adaptation, use of the quadtree adapted grid makes a considerable saving in both computer space and time.

6.1.2. Shear flow In most real interfacial flow cases, the interface is moving under the influence of fluid shear and the interface deforms considerably. Thus, here the ability of the advection scheme to operate in a shearing flow field is tested. In this case, also investigated by Ubbink and Issa [33] and Rudman [34], the velocity is prescribed to be $u = u_{\max} \sin\theta \cos\theta$, $v = -\, v_{\max} \cos\theta \sin\theta$, where $u_{\max} = 1.0$ and $v_{\max} = -1.0$. An adapting quadtree grid is used with maximum division 7 and minimum division 5, and the circle of diameter 0.36 is initially positioned at $x = 0.0$, $y = -0.2$. The interface is first advected forward up to $t = 5$ s and then the velocities are reversed for the same length of time in order to return the volume fraction field to the initial condition. A perfect advection scheme should result in the same initial volume fraction field.

The results of this test are presented in Figure 9. In this case the reconstructed interface is plotted rather than the volume fraction contours. In Figures 9(a) and 9(b) the initial interface and initial quadtree grids are shown. The grid also has the velocity vectors (scaled by 0.02) superimposed. Figures 9(c) and 9(d) show the interface and adapted grid at $t = 5.0$ s and Figures 9(e) and 9(f) show the resulting interface and grid after the velocity has been reversed



(a) (b) (c) (d) (e) (f)

Figure 9. Circular interface in shear flow. (a) Interface at $t = 0.0$ s; (b) quadtree grid at $t = 0.0$ s; (c) interface at $t = 5.0$s; (d) adapted grid at $t = 5.0$s; (e) interface after reversing from $t = 5.0$s to $= 0.0$s and (f) adapted grid after reversing from $t = 5.0$ s to 0.0 s.

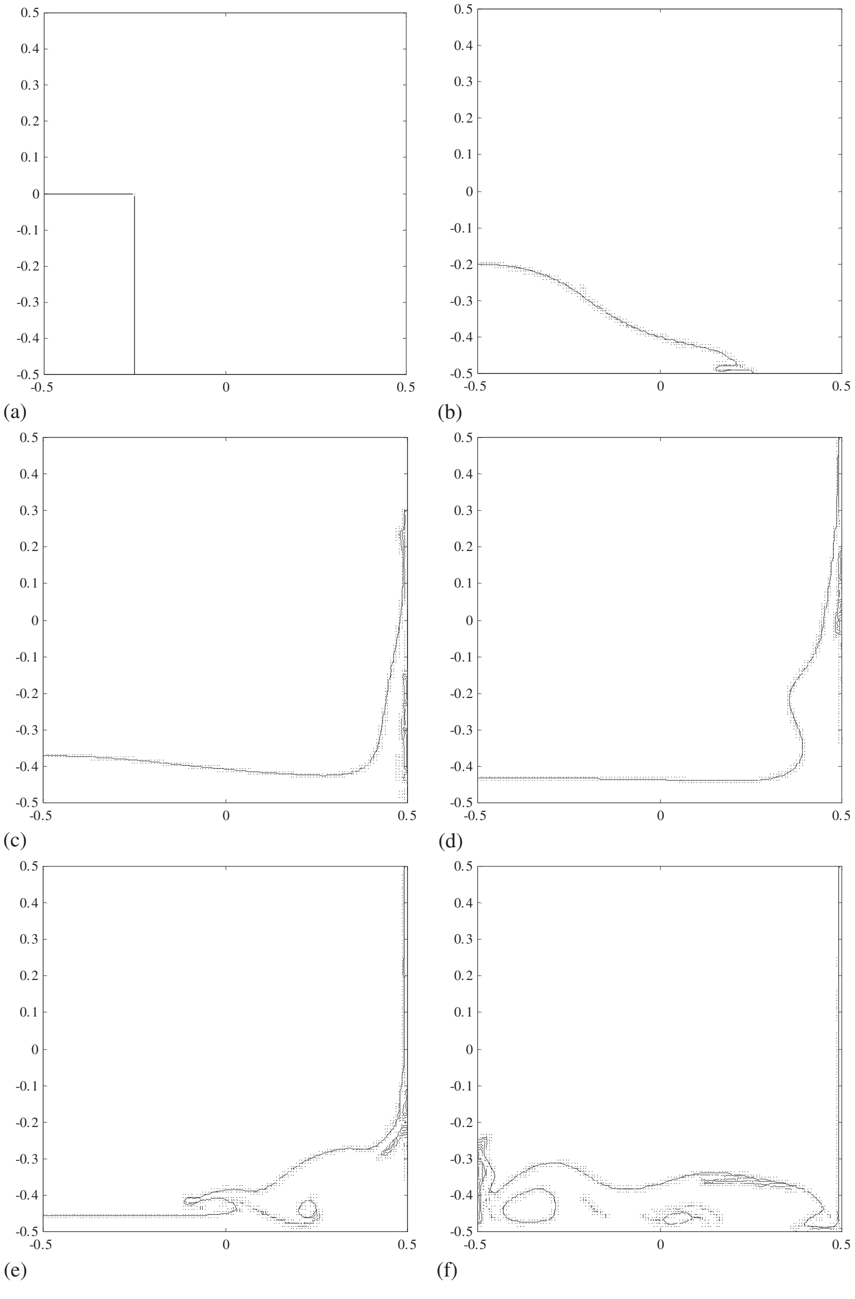


Figure 10. 128 × 128 uniform grid. (a) $T = 0$; (b) $T = 1.617$; (c) $T = 3.233$; (d) $T = 4.850$; (e) $T = 6.466$ and (f) $T = 8.029$.

and the interface advected for 5 seconds in the opposite direction. After reversing from $t = 5.0$s, the resulting circular interface is very close to the initial circular interface, despite being highly distorted at $t = 5.0$ s.

6.2. Collapse of a water column

Simulation of the collapse of a water column has been investigated numerically by various researchers, such as Ubbink [10], Jeong and Yang [35, 36] and Qian *et al.* [37] and experimental data exists from Martin and Moyce [38]. Here, the quadtree adaptive CICSAM interface tracking method is combined with the Navier–Stokes flow solver and the simulation calculated for a series of grid sizes, both uniform and quadtree adaptive. The initial conditions for this case are shown in Figure 10(a). A unit square tank contains a column of water 0.25 m wide and 0.5 m high held in place at $t = 0$ s. The restraint is then removed instantaneously and the resulting motion of the water column as it collapses under gravity is simulated. The water has dynamic viscosity $\mu_1 = 1 \times 10^{-3}$ kg/m/s and the air $\mu_2 = 1.7 \times 10^{-5}$ kg/m/s, the density of water is $\rho_1 = 1000\,\mathrm{kg/m^3}$ and for air $\rho_2 = 1\,\mathrm{kg/m^3}$, and the acceleration due to gravity is taken to be $g = 9.8$ m/s. Initially the velocity everywhere is zero; no-slip boundary conditions are applied on all walls; a free boundary condition for velocity is applied at the top of the tank and pressure at the top of the tank is fixed at zero.

Results calculated on a uniform 128×128 grid with time step $\mathrm{d}t = 0.0001$ s, are presented at non-dimensional time steps, $T = 0$, 1.617, 3.233, 4.850, 6.466 and 8.083, where $T = t\sqrt{g/a}$ and a is the width of the water column. The time step required for temporal grid convergence depends on the grid cell size (the smallest cell size in quadtree grids), however, the calculations presented here all used the time step required for the finest grid, $\mathrm{d}t = 0.0001$ s, for simplicity. In Figure 10 the reconstructed interface is plotted and in Figure 11 the velocity vectors and interface are plotted together. Velocity vectors are plotted at every fourth grid point in order not to crowd the figure. The results agree well with those presented by Ubbink [10] and Qian *et al.* [37] and the video images (not shown here) taken by Koshizuka *et al.* [39].

Results were also calculated on uniform grid sizes 32×32 and 64×64. The results of all three calculations are summarised in Figure 12. The non-dimensional height of the water column at the left wall versus the non-dimensional time is shown in Figure 12(a). The predicted height is the same for all three grid sizes and agrees very well with the experimental data obtained by Martin and Moyce [38], which is plotted alongside the numerical data. The non-dimensional position of the leading edge is plotted against non-dimensional time in Figure 12(b). Martin and Moyce [38] presented two different sets of experimental data for this case and as observed by Ubbink [10] the numerical results show that the leading edge moves faster as the grid resolution is increased. The discrepancy between the experimental and numerical results here is possibly due to the difficulty in determining the exact location of the leading edge in the experiment. A thin layer, similar to a jet, shoots along the base of the tank, which is difficult to identify precisely in the photographs. However, it is reasonable to conclude that a calculation grid with a refined mesh at the base will be better able to predict the progress of the leading edge.

The collapse of the water column was also calculated on quadtree grids, which are refined at the base of the tank and adapt during the solution to provide high resolution at the free surface. If the cell lies within a band surrounding the interface the cell is divided. Conversely,

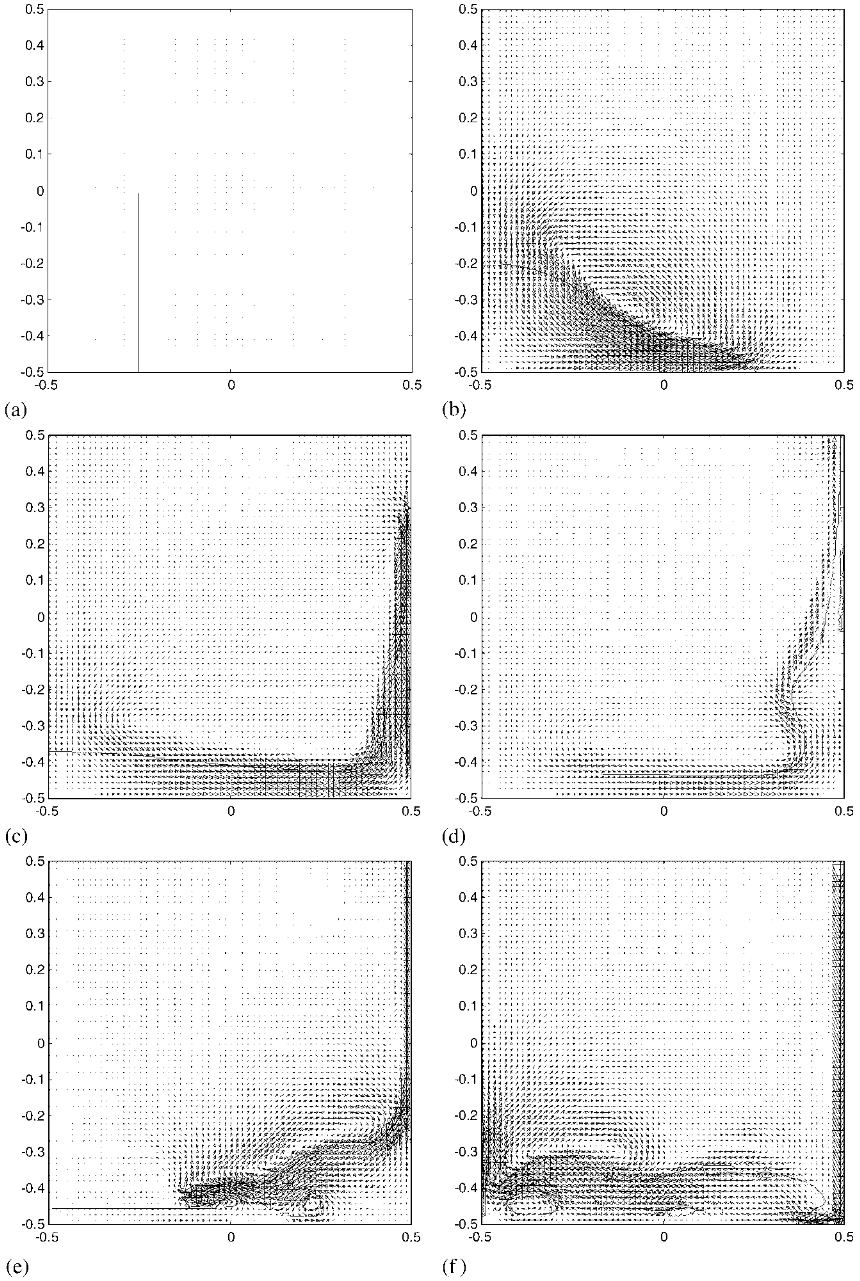


Figure 11. 128 × 128 uniform grid, interface and velocity vectors. (a) $T = 0$; (b) $T = 1.617$; (c) $T = 3.233$; (d) $T = 4.850$; (e) $T = 6.466$ and (f) $T = 8.029$.

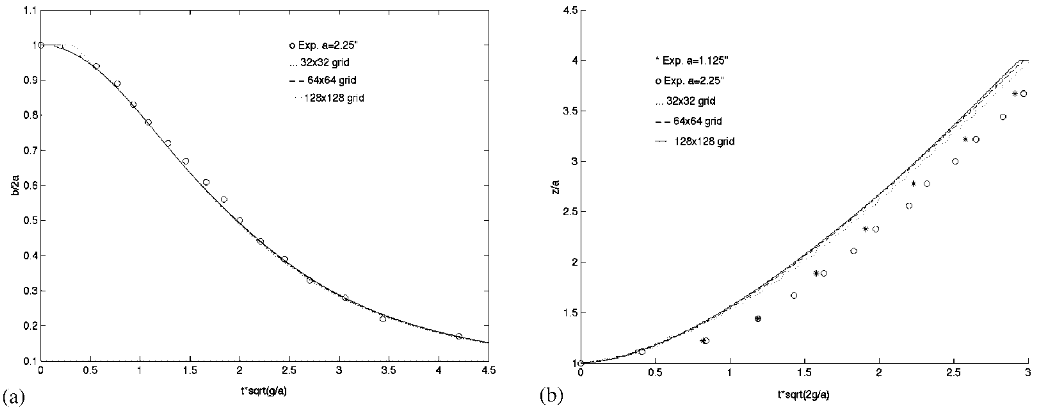


Figure 12. (a) Height of water column and (b) position of leading edge.

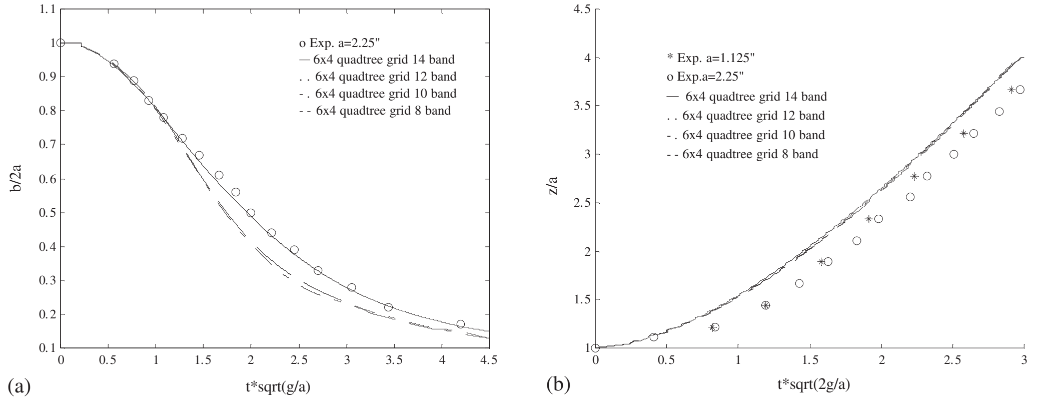


Figure 13. Comparison and band sizes for 6×4 quadtree grid. (a) Height of water column and (b) position of leading edge.

if four sibling cells lie outside of the band, the four sibling cells are removed and replaced with their parent quadtree cell. The size of the band surrounding the interface is found to be critical in obtaining an accurate quadtree solution, as also noted by Wang [40]. The largest inaccuracies in the calculation occur at the interface between cell sizes in the quadtree grid (at hanging nodes), and would accumulate at the free surface. Investigations were carried out to assess what size band of the finest cells around the interface is necessary to keep the error sufficiently distant from the free surface. A range of band widths were experimented with and the results for bands of 8, 10, 12 and 14 cells are shown for a 6×4 quadtree grid in Figure 13.

It can be seen that for the 12 and 14 band the result is very similar, but for the 8 and 10 bands, the height of the water column as it falls is not predicted correctly. The calculation was also carried out on adapting quadtree grids of size 5×3 and 7×5. Us-

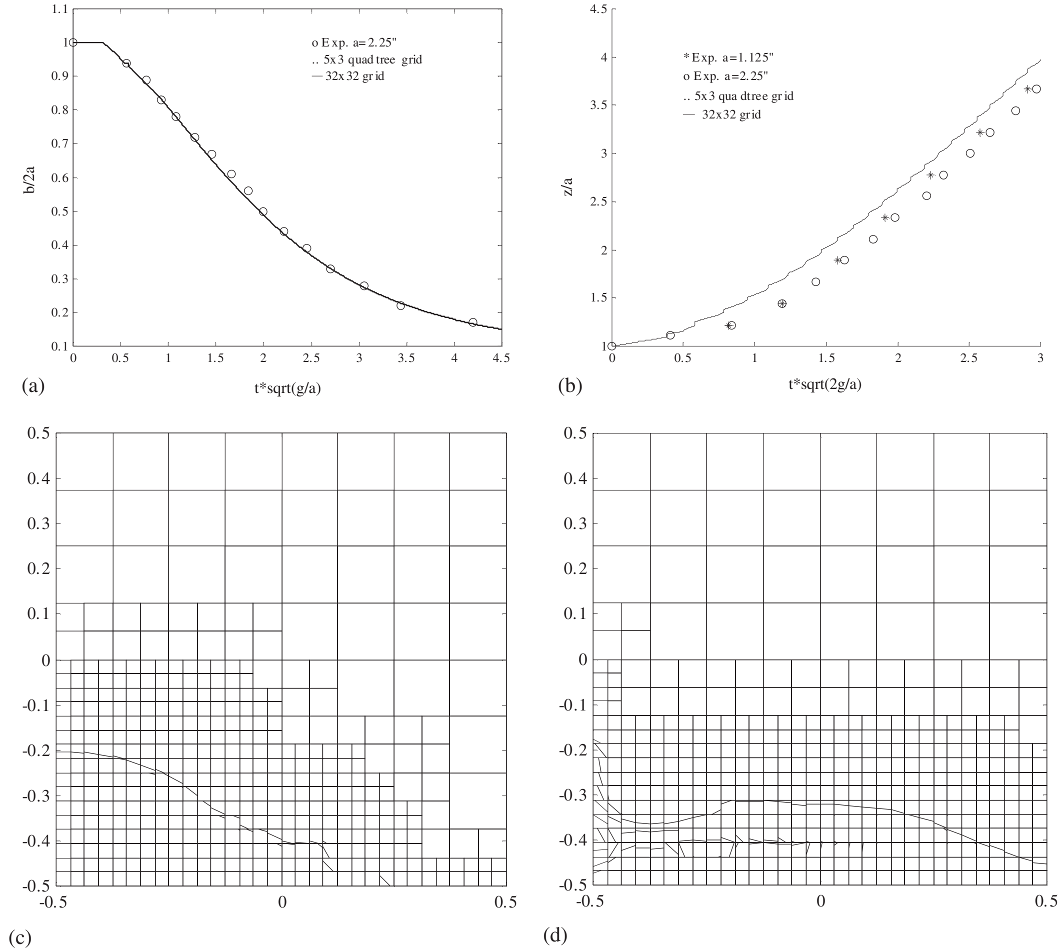


Figure 14. Results for 5×3 quadtree grid. (a) Height of water column; (b) position of leading edge; (c) $T = 1.617$ and (d) $T = 8.029$.

ing a band of 14 cells for the 7 × 5 quadtree grid however, resulted in a large deviation in the progress of the fronts when compared with the uniform grid calculation. It was found that the physical distance, rather than the number of cells, between the free surface and the first hanging node was critical. Thus for the 5 × 3 quadtree grid a band of 6 cells is necessary (distance = 0.188); for 6 × 4 14 (distance = 0.219) and for 7 × 5 a band of 28 (distance = 0.219).

The reconstructed interfaces and adapted quadtree grids obtained at $T = 1.617$ and 8.029 are shown in Figures 14–16 for each of the quadtree grids. The results for the 7 × 5 grid agree well with those calculated on the equivalent uniform grid and presented in Figure 10. The time histories of the fronts are also plotted in Figures 14–16 together with the results

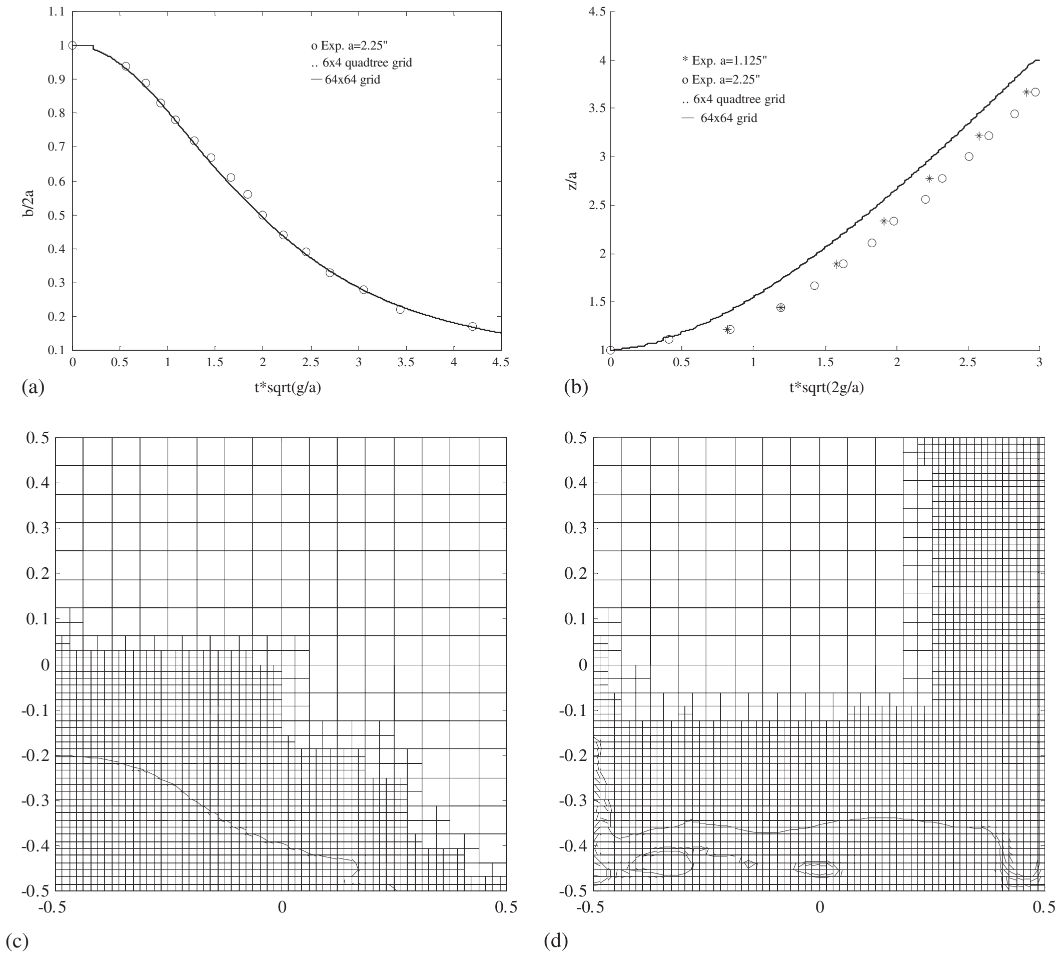


Figure 15. Results for 6×4 quadtree grid. (a) Height of water column; (b) position of leading edge; (c) $T = 1.617$ and (d) $T = 8.029$.

calculated on equivalent uniform grids with cell sizes equal to that of the smallest quadtree cell. Results comparing the progress of the height and leading edge of the collapsing water column for all three quadtree grids together with the 128×128 uniform grid are summarised in Figure 17.

The time histories for each quadtree calculation are almost identical to those calculated on the equivalent uniform grid and in Figure 17 the results for quadtree grids of increasing resolution show convergence to the finest uniform grid calculation. The results show that the same accuracy can be achieved on the quadtree grids as on their equivalent uniform grid. Furthermore, using adaptive quadtrees significantly reduces the size of the calculation grid. For the 7×5 quadtree grid the number of calculation cells is typically 7156 compared with 16 384 in the equivalent uniform grid.

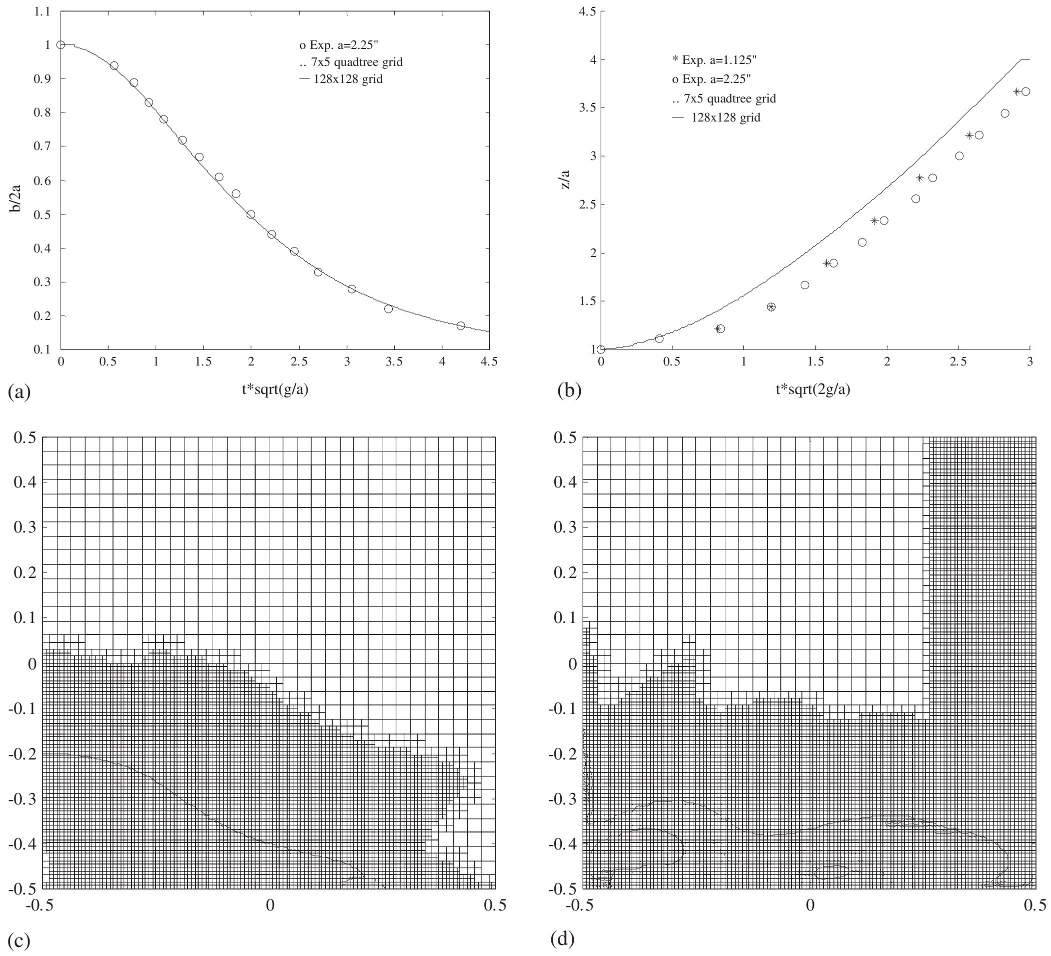


Figure 16. Results for 7×5 quadtree grid. (a) Height of water column; (b) position of leading edge; (c) $T = 1.617$ and (d) $T = 8.029$.

There is no such saving in CPU time, however, for the Navier Stokes calculation because of the CPU cost in grid adaptation and interpolation routines. Convergence of the Navier–Stokes solver on adapting quadtree grids is slow. This is partly because of the lack of order in the computational grid which necessitates point-by-point iteration and partly due to the errors introduced at each time step by interpolations on the adapting grids. Optimization of the Navier–Stokes solution method on quadtree grids and of the interpolation routines will be addressed in the next stages of this work. The results presented here demonstrate the encouraging potential of the new method in two-dimensional flow simulations. It may also be extended in a straightforward manner to three dimensions and, with the inclusion of a suitable closure model, to simulation of turbulent flows where hierarchical grids are ideally suited to resolving the fine detail of structures in the flow.

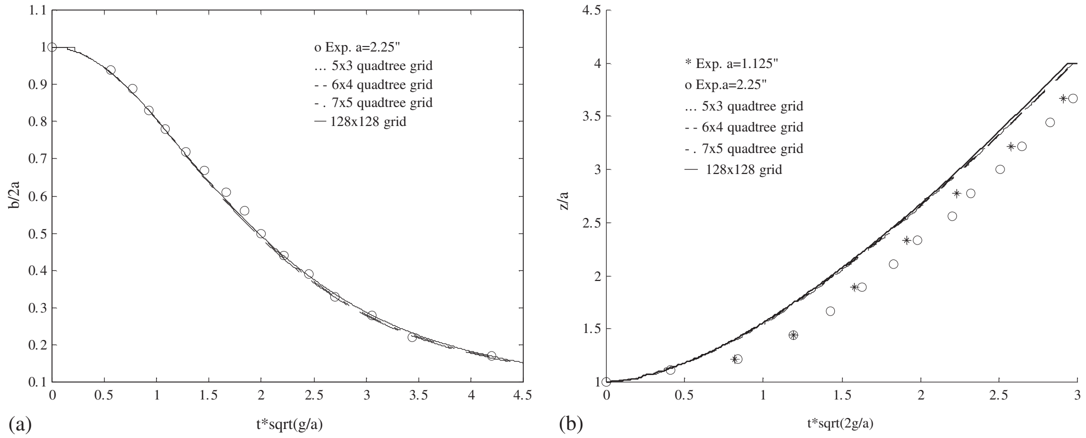


Figure 17. Comparison of quadtree grid results. (a) Height of water column and (b) position of leading edge.

7. CONCLUSIONS

The adaptive quadtree volume of fluid method developed here shows great potential in the simulation of complex free surface flows. The new method uses CICSAM differencing for advection of the interface and PLIC reconstruction for interpolation of the volume fraction as the grid adapts. The examples of interface advection in prescribed velocity fields demonstrate that the adapting quadtree grids provide high resolution at the interface at low computational cost.

When combined with a Navier–Stokes solver and used to simulate the collapse of a water column, it is found that a refined band surrounding the interface of width $0.22l$, where l is the height of the domain, is necessary to achieve the same accuracy as an equivalent uniform grid. Use of quadtree grids is particularly suitable in this case because of the extra refinement needed to track the progress of the water jet along the base of the tank. The main areas for improvement in this scheme are in the slow convergence of the Navier–Stokes solver and the use of point by point iteration on the quadtree grids. However, the adaptive quadtree results are in excellent agreement with experimental and other numerical data, a sharp interface is maintained at the free surface and a considerable saving is made in the size of the computational grid.

ACKNOWLEDGEMENTS

The author is very grateful to the Royal Society for generously supporting this work.

REFERENCES

1. Ma QW, Wu GX, Eatock Taylor R. Finite element simulation of fully nonlinear interaction between vertical cylinders and steep waves—part 1: methodology and numerical procedure. *International Journal for Numerical Methods in Fluids* 2001; **36**:265–285.

2. Ma QW, Wu GX, Eatock Taylor R. Finite element simulation of fully nonlinear interaction between vertical cylinders and steep waves—part 2: numerical results and validation. *International Journal for Numerical Methods in Fluids* 2001; **36**:287–308.
3. Ferziger JH, Perić M. *Computational Methods for Fluid Dynamics*. Springer: Berlin, 1996, ISBN 3-540-59434-5.
4. Hyman JM. Numerical methods for tracking interfaces. *Physica* 1984; **12D**:396–407.
5. Monaghan JJ. Simulating free surface flows with SPH. *Journal of Computational Physics* 1994; **65**:179–214.
6. Causon DM. An efficient front tracking algorithm for multi-component fluid calculations with biomedical applications. *Zeitscrift fur Angewandte Mathematik und Mechanik* 1996; **76**(S1):371–372.
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. Heinrich P. Nonlinear numerical model of landslide-generated water waves. *International Journal of Engineering Fluid Mechanics* 1991; **4**(4):403–416.
9. Tomiyama A, Sou A, Minagawa H, Sakaguchi T. Numerical analysis of a single bubble by VoF method. *JSME International Journal Series B* 1993; **36**:51–56.
10. Ubbink O. Numerical prediction of two fluid systems with sharp interfaces. *Ph.D. Thesis*, Imperial College of Science, Technology and Medicine, London, 1997.
11. Greaves DM, Borthwick AGL. On the use of adaptive hierarchical meshes for numerical simulation of separated flows. *International Journal for Numerical Methods in Fluids* 1998; **26**:303–322, ISSN 0271-2091.
12. Lafaurie B, Nardone C, Scardovelli R, Zaleski S, Zanetti G. Modelling merging and fragmentation in multiphase flows with SURFER. *Journal of Computational Physics* 1994; **113**:134–147.
13. Noh WF, Woodward P. SLIC (simple line interface calculations). *Lecture Notes in Physics* 1976; **59**:330–340.
14. Ashgriz N, Poo JY. FLAIR: flux line-segment model for advection and interface reconstruction. *Journal of Computational Physics* 1991; **93**:449–468.
15. Gueyffier D, Li J, Nadim A, Scardovelli R, Zaleski S. Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *Journal of Computational Physics* 1999; **152**:423–456.
16. Leonard BP. The ULTIMATE conservative difference scheme applied to steady one-dimensional advection. *Computer Methods in Applied Mechanics and Engineering* 1991; **88**:17–74.
17. Leonard BP. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computer Methods in Applied Mechanics and Engineering* 1979; **19**:59–98.
18. Samet H. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley: Reading, MA, 1990.
19. Samet H. *Applications of Spatial Data Structures*. Addison-Wesley: Reading, MA, 1990.
20. Yerry MA, Shephard MS. A modified quadtree approach to finite element mesh generation. *IEEE Computer Graphics and Applications* 1983; **3**(1):39–46.
21. Messaoud B. Parallel and adaptive algorithms for elliptic partial differential systems. *Ph.D*. Rensselaer Polytechnic Institute, Troy, New York, 1992.
22. van Dommelen L, Rundensteiner EA. Fast, adaptive summation of point forces in the two-dimensional Poisson equation. *Journal of Computational Physics* 1989; **83**:126–147.
23. Józsa J, Gáspár C. Fast, adaptive approximation of wind-induced horizontal flow patterns in shallow lakes using quadtree-based multigrid method. C.M.W.R., Denver, 1992.
24. Young DP, Melvin RG, Bieterman MB, Johnson FT, Samant SS, Bussoletti JE. A locally refined rectangular grid finite element method: application to computational fluid dynamics and computational physics. *Journal of Computational Physics* 1991; **92**:1–66.
25. de Zeeuw D, Powell KG. An adaptively refined Cartesian mesh solver for the Euler equations. *Journal of Computational Physics* 1993; **104**:56–68.
26. Greaves DM, Borthwick AGL. Hierarchical tree-based finite element mesh generation. *International Journal for Numerical Methods in Engineering* 1999; **45**:447–471, ISSN 0029-5981.
27. Hu ZZ, Greaves DM, Wu GX. Numerical simulation of fluid flows using an unstructured finite volume method with adaptive tri-tree grids. *International Journal for Numerical Methods in Fluids* 2002; **39**:403–440.
28. Yiu KFC, Greaves DM, Cruz S, Saalehi A, Borthwick AGL. Quadtree grid generation: information handling, boundary fitting and CFD applications. *International Journal of Computers and Fluids* 1995; **25**(8):759–769.
29. Greaves DM. Numerical modelling of laminar separated flows and inviscid steep waves using adaptive hierarchical meshes. *D.Phil. Thesis*, Oxford University, 1995.
30. Perić M, Kessler R, Scheurer G. Comparison of finite volume numerical methods with staggered and colocated grids. *Computers and Fluids* 1998; **16**(4):389–403.
31. Patanker SV. *Numerical Heat Transfer and Fluid Flow*. Taylor and Francis, Hemisphere Publishing Corporation: U.S.A., 1980.
32. Rhie CM, Chow WL. A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation. *AIAA Journal* 1983; **21**:1525–1532
33. Ubbink O, Issa RI. A method for capturing sharp fluid interfaces on arbitrary meshes. *Journal of Computational Physics* 1999; **153**:26–50.

34. Rudman M. Volume-tracking methods for interfacial flow calculations. *International Journal for Numerical Methods in Fluids* 1997; **24**:671–691.
35. Jeong JH, Yang DY. Finite element analysis of transient fluid flow with free surface using VOF (volume of fluid) method and adaptive grid. *International Journal for Numerical Methods in Fluids* 1998; **26**:1127–1154.
36. Jeong JH, Yang DY. Three-dimensional finite element analysis of transient fluid flow with free surface using marker surface method and adaptive grid refinement. *International Journal for Numerical Methods in Fluids* 1999; **29**:657–684.
37. Qian L, Causon DM, Ingram DM, Mingham CG. Cartesian cut cell two-fluid solver for hydraulic flow problems. *Journal of Hydraulic Engineering* 2003; **129**(9):688–696.
38. 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* 1952; **A244**:312–324.
39. Koshizuka S, Tamako H, Oka Y. A particle method for incompressible viscous flow with fluid fragmentation. *Computational Fluid Dynamics Journal* 1995; **4**(1):29–46.
40. Wang JP. Viscous free surface flow modelling using interface capturing methods on adaptive grids. *D.Phil. Thesis*, Oxford University, 2002.