INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2006; **50**:693–711
Published online 17 October 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1073

Simulation of viscous water column collapse using adapting hierarchical grids

Deborah M. Greaves*,†,‡

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

SUMMARY

An adaptive hierarchical grid-based method for predicting complex free surface flows is used to simulate collapse of a water column. Adapting quadtree grids are combined with a high-resolution interface-capturing approach and pressure-based coupling of the Navier–Stokes equations. The Navier–Stokes flow solution scheme is verified for simulation of flow in a lid-driven cavity at $Re = 1000$. Two approaches to the coupling of the Navier–Stokes equations are investigated as are alternative face velocity and hanging node interpolations. Collapse of a water column as well as collapse of a water column and its subsequent interaction with an obstacle are simulated. The calculations are made on uniform and adapting quadtree grids, and the accuracy of the quadtree calculations is shown to be the same as those made on the equivalent uniform grids. Results are in excellent agreement with experimental and other numerical data. A sharp interface is maintained at the free surface. The new adapting quadtree-based method achieves a considerable saving in the size of the computational grid and CPU time in comparison with calculations made on equivalent uniform grids. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: hierarchical grids; quadtrees; adaptive grids; volume of fluid; free surface; incompressible viscous fluid; Navier–Stokes equations

1. INTRODUCTION

Simulation of free surface waves is an important problem in ocean engineering and naval architecture. Significant advances have been made towards understanding gravity waves through the use of inviscid free surface fluid flow models [1], in which the viscosity of the fluid is ignored and the fluid motion considered to be irrotational and dominated by inertia.

*Correspondence to: Deborah M. Greaves, Department of Architecture and Civil Engineering, Claverton Down, University of Bath BA2 7AY, U.K.
†E-mail: d.m.greaves@bath.ac.uk
‡Royal Society University Research Fellow.

Contract/grant sponsor: The Royal Society

Received 4 April 2005
Revised 14 July 2005
Accepted 17 July 2005

However, there are situations in which combined effect of the free surface and the fluid viscosity is important, such as interaction of a free surface wave with a structure. The moving air–water interface together with the nonlinear governing equations and boundary conditions make modelling of a viscous fluid free surface flow an extremely challenging problem. The accuracy of the simulation depends on calculating the correct position of the moving air–water interface, and this becomes especially difficult when the wave overturns and merges with the water surface or when the interface breaks up into spray.

There are two main methodologies for predicting the position of the moving free surface: interface tracking and interface capturing. Interface-tracking methods include moving mesh and front tracking [2], which have the disadvantage of being unable to predict beyond the limit of wave-breaking and particle-tracking schemes [3, 4], which tend to be expensive and not practical in three dimensions. On the other hand, interface-capturing methods can be used for modelling large-scale deformations of the interface including wave breakup 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 [5] and from the volume fraction field in the volume of fluid (VoF) method [6].

In this work, a VoF methodology together with the high-resolution CICSAM [7] interface advection scheme is implemented on adapting quadtree grids [8]. A similar method, applied to sloshing of a viscous liquid in a rectangular tank, is described by Wang *et al.* [9]. An advantage of using adapting quadtree or octree grids is that they provide refinement and adaptation locally without requiring remeshing of the entire domain. Quadtree and octree adapting grids were used by Jeong and Yang [10, 11] to simulate the early stages of collapse of a water column using a finite element formulation with square and cubic elements. Popinet [12] describes use of quadtree and octree grids in an adaptive solver for the Euler equations using a VoF approach, with multigrid iterations for the pressure equation.

Details of the VoF method and the quadtree grid adaptation, for a solution method based on SIMPLE [13] (semi implicit method for pressure-linked equations) with power law approximations for the convective terms, are described by Greaves [14]. Some early results are included in Reference [14] of interface advection flows and simulation of collapse of a dam, which demonstrate that the same accuracy is achieved on an adapting quadtree grid as on the equivalent uniform grid. For calculations involving a free surface, the method was found to require a large band of refinement around the interface and although there was a saving in computer storage, the adaptive simulation was actually more expensive in CPU time. The large CPU requirement was identified to be due in part to slow convergence of the SIMPLE Navier–Stokes equation solution procedure. Greaves [14] suggested that the size of the refinement band at the interface may be reduced by improving interpolation procedures on the quadtree grids, and that by optimizing the solution procedure the overall CPU could be reduced.

This paper presents an improved VoF methodology with dynamic quadtree adaptation, which demonstrates savings in both computer storage and CPU time compared with equivalent uniform grids and no loss of accuracy. First, the standard benchmark test of flow in a driven cavity is used to assess the accuracy of three different face velocity interpolation schemes. Next, an adapting quadtree grid is tested against the equivalent uniform grid for this case with improved hanging node interpolations and a Gauss theorem approximation for calculating gradients implemented on the quadtree grid. In order to speed up the solution, a PISO [15]

(pressure implicit with splitting of operators) scheme is developed for quadtree grids and applied to simulation of water column collapse. The results are compared with experimental data and data previously calculated using a SIMPLE formulation [14]. The improvements made in gradient calculations and interpolations on the quadtree grid are found to reduce the required width of the high-resolution interface band. Finally, the impressive capability of the method is demonstrated when applied to simulation of water column collapse followed by interaction of the wave front with an obstacle. This case exhibits extremely complicated free surface behaviour, including jet formation, wave breaking, bubble entrapment and break up of the interface into spray.

2. SOLUTION OF THE NAVIER–STOKES EQUATIONS

Previous work by Greaves [14] used a SIMPLE-type solution procedure for the Navier–Stokes equations, which was found to suffer from slow convergence especially on quadtree grids. In an attempt to improve the efficiency of the method, the alternative PISO scheme is investigated here. In this Section, the main points of both the SIMPLE and PISO scheme are discussed in order to identify the similarities and differences between the two methods.

Governing equations in primitive form for a two-dimensional incompressible flow are the mass conservation equation and the Navier–Stokes momentum conservation equations

$$\frac{\partial(\rho u)}{\partial x}+\frac{\partial(\rho v)}{\partial y}=0 \tag{1}$$

$$\frac{\partial u}{\partial t}+\frac{\partial u^2}{\partial x}+\frac{\partial uv}{\partial y}=-\frac{1}{\rho}\frac{\partial p}{\partial x}+\nu\nabla^2 u+\frac{\partial}{\partial x}\left(\nu\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(\nu\frac{\partial v}{\partial x}\right) \tag{2}$$

$$\frac{\partial v}{\partial t}+\frac{\partial uv}{\partial x}+\frac{\partial v^2}{\partial y}=-g-\frac{1}{\rho}\frac{\partial p}{\partial y}+\nu\nabla^2 v+\frac{\partial}{\partial x}\left(\nu\frac{\partial u}{\partial y}\right)+\frac{\partial}{\partial y}\left(\nu\frac{\partial v}{\partial y}\right) \tag{3}$$

where x and y define an orthogonal Cartesian co-ordinate system, u and v are the corresponding velocity components, t the time, p the pressure, ρ the fluid density, g the gravitational acceleration and ν the fluid kinematic viscosity. For situations where the fluid viscosity is variable, such as the multifluid (air and water) flow simulations considered here, extra diffusion terms appear in the momentum equations: the last two terms in (2) and (3). According to Ferziger and Perić [16], these terms are small compared to the other diffusion terms, and so can be treated explicitly and included in the source term.

The governing equations are discretized on quadtree grids using finite volumes with collocated primitive variables (u, v and p are stored together at cell centres). The discrete expression for the u-velocity is obtained by integrating the u-momentum equation over the control volumes and multiplying by the cell area, $V_P=\delta x\delta y$, to give

$$a_P u_P=\sum a_{\mathrm{nb}}u_{\mathrm{nb}}+u_P^0\rho\frac{V_P}{\delta t}-\frac{\partial p}{\partial x}V_P=H(u)-\frac{\partial p}{\partial x}V_P \tag{4}$$

where the control volume (cell) in question is given the subscript P and the sum of the neighbour contributions are denoted, nb. The superscript 0 indicates that the term is from the

previous time step, δx, δy and δt are the horizontal and vertical dimensions of the cell and the time step for the calculation and $H(u) = \sum a_{\mathrm{nb}} u_{\mathrm{nb}} + u_P^0 \rho (V_P/\delta t)$. The coefficients, denoted a, combine the momentum fluxes for convection and diffusion transport. The discrete equation for the v component of velocity is similar. The continuity equation is discretized to give

$$S_{\mathrm{e}} u_{\mathrm{e}} - S_{\mathrm{w}} u_{\mathrm{w}} + S_{\mathrm{n}} v_{\mathrm{n}} - S_{\mathrm{s}} v_{\mathrm{s}} = 0 \tag{5}$$

where the lower case subscripts denote face values of the velocity and S_i is the area of face i.

In order to solve for the velocity and pressure field, it is necessary to couple and momentum and continuity equations in some way. Various approaches are described in the literature. The present work considers a SIMPLE [13] approach and a PISO [15] scheme.

2.1. SIMPLE

The SIMPLE approach is a pressure correction scheme, in which the velocity and pressure variables are considered to comprise a guess, denoted with superscript *, and a correction, denoted with superscript $'$, e.g. $u = u^* + u'$. Velocity correction formulae are derived from the momentum equations and interpolated to cell faces using Rhie and Chow [17] treatment to prevent checkerboard errors. These expressions are substituted into the discretized continuity equation (5) to produce the pressure correction equation

$$a_P p'_P = a_{\mathrm{E}} p'_{\mathrm{E}} + a_{\mathrm{W}} p'_{\mathrm{W}} + a_{\mathrm{N}} p'_{\mathrm{N}} + a_{\mathrm{S}} p'_{\mathrm{S}} + b \tag{6}$$

The source term is the same as the discretized continuity equation but calculated using the guessed velocity field, $b = S_{\mathrm{e}} u^*_{\mathrm{e}} - S_{\mathrm{w}} u^*_{\mathrm{w}} + S_{\mathrm{n}} v^*_{\mathrm{n}} - S_{\mathrm{s}} \delta x v^*_{\mathrm{s}}$. The coefficients, a_j for cell $j = P$ and its neighbours $j = \mathrm{E, W, N, S}$ are given by Greaves and Borthwick [8]. The algorithm for solution of the equations using the SIMPLE scheme is

1. solve momentum equations (2) and (3) using guessed pressure field,
2. solve pressure correction equation (6),
3. correct pressure and velocity variables.

This loop is repeated iteratively until a prescribed tolerance is achieved.

2.2. PISO

The PISO [15] algorithm was developed initially for non-iterative computation of unsteady compressible flows, but has since been successfully adapted for steady and unsteady incompressible flows.

The discretized u-momentum equation (4) is interpolated to cell faces using the Rhie and Chow [17] treatment mentioned above. The face velocity expressions are then directly substituted into the discretized continuity equation (5) to give an equation for pressure.

$$a_P p_P = a_{\mathrm{E}} p_{\mathrm{E}} + a_{\mathrm{W}} p_{\mathrm{W}} + a_{\mathrm{N}} p_{\mathrm{N}} + a_{\mathrm{S}} p_{\mathrm{S}} + b \tag{7}$$

Here, a_j are coefficients for node j [7] and the source term is

$$b = -\left[S_{\mathrm{e}} \left(\frac{H(u)}{a_P} \right)_{\mathrm{e}} - S_{\mathrm{w}} \left(\frac{H(u)}{a_P} \right)_{\mathrm{w}} + S_{\mathrm{n}} \left(\frac{H(u)}{a_P} \right)_{\mathrm{n}} - S_{\mathrm{s}} \left(\frac{H(u)}{a_P} \right)_{\mathrm{s}} \right]$$

The algorithm for solution of the equations using the PISO scheme is

1. solve momentum equations (2) and (3) using guessed pressure field,
2. solve pressure equation (7),
3. calculate volumetric fluxes for use in momentum equation coefficients,
4. correct velocity.

The loop from (2) to (4) is repeated iteratively until a prescribed tolerance is achieved before proceeding to the next time step (1).

2.3. Face velocity interpolation

The Rhie and Chow [17] face velocity interpolations mentioned above are used to calculate mass fluxes and in the pressure or pressure correction equations. Discretization of the convective terms in the momentum equations requires further face velocity interpolation, and three different approaches are considered in this work. The first is the power law scheme recommended by Patanker [13] (also used in Reference [14]), the second is deferred correction and the third is Jasak *et al.*'s [18] scheme. The deferred correction scheme, introduced by Ferziger and Perić [16] for the calculation of convective fluxes, is a means of stabilizing the calculation when a central differencing approximation may lead to instability. It combines upwind differencing (UDS) with central differencing (CDS). The face velocity is expressed as

$$u_f = u_f^{\mathrm{UDS}} + \lambda(u_f^{\mathrm{CDS}} - u_f^{\mathrm{UDS}})^{m-1} \tag{8}$$

where the term in brackets is evaluated using values from the previous iteration while the equations are solved using the UDS approximation. The blending factor, $0 \leqslant \lambda \leqslant 1$, introduces a linear weighting between the UDS and CDS approximations. If $\lambda = 0$ then pure upwinding is used and if $\lambda = 1$ then pure CDS is used.

The centred differencing scheme proposed by Jasak *et al.* [18] calculates a blending factor for each face. It is a high-resolution (HR scheme) convection–diffusion differencing scheme based on the normalized variable diagram (NVD) discussed by Leonard [19]. The face value of velocity is given by

$$u_f = \begin{cases} u_{\mathrm{D}} & \text{for } \tilde{u}_{\mathrm{D}} \leqslant 0 \text{ or } \tilde{u}_{\mathrm{D}} \geqslant 1 \\ \dfrac{1}{2}(u_{\mathrm{D}} + u_{\mathrm{A}}) & \text{for } k \leqslant \tilde{u}_{\mathrm{D}} < 1 \\ \left(1 - \dfrac{\tilde{u}_{\mathrm{D}}}{2k}\right) u_{\mathrm{D}} + \dfrac{\tilde{u}_{\mathrm{D}}}{2k}\, u_{\mathrm{A}} & \text{for } 0 < \tilde{u}_{\mathrm{D}} < k \end{cases} \tag{9}$$

where k is a prescribed constant between 0 and 0.5, for which Jasak *et al.* [18] recommend a value of 0.1. The subscripts D and A indicate donor and acceptor cells for a particular face and are determined by the direction of the flow. The decision factor, $\tilde{u}_{\mathrm{D}}$, is defined as

$$\tilde{u}_{\mathrm{D}} = 1 - \frac{u_{\mathrm{A}} - u_{\mathrm{D}}}{2(\nabla u)_{\mathrm{D}} \cdot \mathbf{d}} \tag{10}$$

where the vector **d** connects the computational point P with its neighbour N and $(\nabla u)_{\mathrm{D}}$ is calculated according to Gauss's divergence theorem from

$$(\nabla u)_{\mathrm{D}} = \frac{1}{V_{\mathrm{D}}} \sum_{f=1}^{n} A_f u_f \tag{11}$$

3. THE VOLUME OF FLUID METHOD, VoF

Both the SIMPLE and PISO schemes are combined with a VoF method for simulating the collapse of a water column. When considering the incompressible flow of two immiscible fluids, the divergence-free velocity field $\mathbf{u}(x,t)$ obeys

$$\nabla \cdot \mathbf{u} = 0 \tag{12}$$

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 fluid properties for a particular cell in the computational grid are calculated as a volume average of the properties of the two fluids in question. Thus, the dynamic viscosity and density are given by

$$\mu = C\mu_1 + (1.0 - C)\mu_2 \tag{13}$$

and

$$\rho = C\rho_1 + (1.0 - C)\rho_2 \tag{14}$$

The volume conservation of the first fluid can be expressed as

$$\frac{\partial C}{\partial t} + \nabla \cdot (\mathbf{u}C) = 0 \tag{15}$$

Solving this equation using UDS is stable but also is diffusive and may spread the interface over many cells, whereas the downwind scheme is unstable but sharpens the interface. 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. Here, the CICSAM (compressive interface-capturing scheme for arbitrary meshes) derived by Ubbink [7] is used. The cell face values of C, used in the discretized volume fraction equation, are determined from a combination of the convection boundedness criteria (CBC) value and the ultimate quickest (UQ) value, as described in detail by Greaves [14].

4. QUADTREE GRIDS

Quadtree grids and their application to the simulation of separated flow are explained by Greaves and Borthwick [8, 20] and their use in a high-resolution interface-capturing scheme is described by Greaves [14, 21]. In Reference [14], it was found that a high-resolution band around the free surface was necessary when simulating collapse of a dam to achieve the same accuracy as the equivalent uniform grid. The necessary width of the band was found to be

approximately 0.22 of the overall height of the domain. One reason for requiring such a band is to keep hanging node errors, inherent in quadtree grids, away from the free surface where accuracy is important for correct fluxing of the volume fraction. In this work, the required width of the interface band has been significantly reduced due in part to an improved hanging node treatment in the gradient calculations. The treatment of hanging nodes together with the criteria for adaptation of the quadtree grids is described here.

4.1. 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. There are two main approaches to the treatment of hanging nodes: to interpolate variables and treat as a regular grid, in which case the accuracy depends on the accuracy of interpolations but fluxes are not necessarily conserved; or to use a conservative approach using an irregular grid stencil.

For example, consider the pressure gradient terms occurring in the momentum equations calculated for cell P in Figure 1

$$(\nabla p)_P \approx \frac{1}{V_P} \sum_{f=1}^{n} A_f p_f \tag{16}$$

where subscript P represents the finite volume cell in question, the finite volume faces are denoted with f and face values lie at the centre of the cell face. A_f is the face area vector and n is the number of faces of a control volume. In Reference [14], hanging node terms

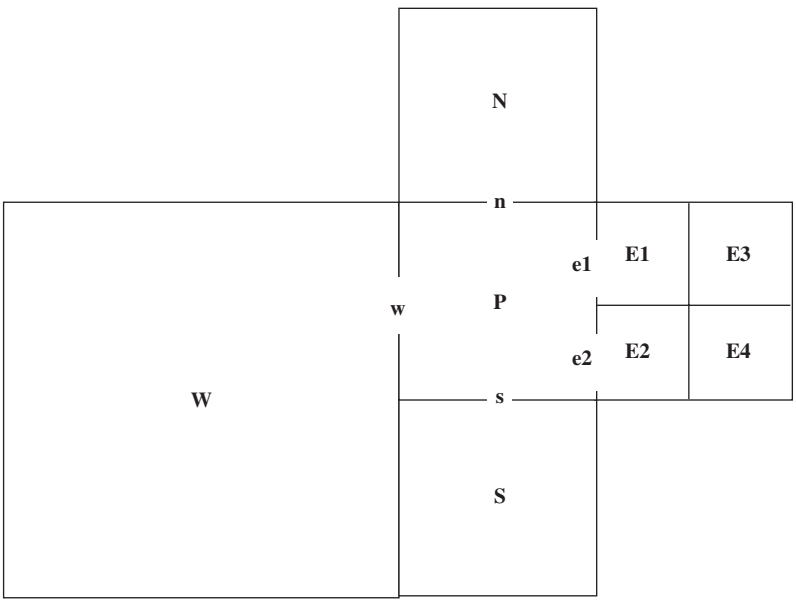


Figure 1. Hanging node treatment for fluxes.

were calculated by an interpolated regular stencil approach of zero-order accuracy, in which for the arrangement in Figure 1

$$p_{\mathrm{e}} = \frac{1}{2} p_P + \frac{1}{2}\left(\frac{p_{\mathrm{E1}} + p_{\mathrm{E2}} + p_{\mathrm{E3}} + p_{\mathrm{E4}}}{4}\right)$$

and

$$p_{\mathrm{w}} = \tfrac{1}{2} p_P + \tfrac{1}{2} p_{\mathrm{W}}$$

In the present work, a flux conserving approach of first-order accuracy is taken. Expanding (16) for the x-direction

$$\frac{\partial p}{\partial x} = A_{\mathrm{e1}}\, p_{\mathrm{e1}} n_{x\mathrm{e1}} + A_{\mathrm{e2}}\, p_{\mathrm{e2}} n_{x\mathrm{e2}} + A_{\mathrm{w}}\, p_{\mathrm{w}} n_{x\mathrm{w}} + A_{\mathrm{n}}\, p_{\mathrm{n}} n_{x\mathrm{n}} + A_{\mathrm{s}}\, p_{\mathrm{s}} n_{x\mathrm{s}} \tag{17}$$

where n_{xf} are the unit outward-pointing normal vectors for each face. The face values are

$$p_{\mathrm{e1}} = \tfrac{1}{3} p_P + \tfrac{2}{3} p_{\mathrm{E1}}, \quad p_{\mathrm{e2}} = \tfrac{1}{3} p_P + \tfrac{2}{3} p_{\mathrm{E2}}, \quad p_{\mathrm{w}} = \tfrac{2}{3} p_P + \tfrac{1}{3} p_{\mathrm{W}}, \quad p_{\mathrm{n}} = \tfrac{1}{2} p_P + \tfrac{1}{2} p_{\mathrm{N}}$$

and

$$p_{\mathrm{s}} = \tfrac{1}{2} p_P + \tfrac{1}{2} p_{\mathrm{S}}$$

Using the present scheme, the width of the band surrounding the interface and hence the total number of cells required in the adapted grid is reduced.

4.2. Grid adaptation

An advantage of quadtree grids is that they can be readily adapted by the addition and removal of panels throughout a time-dependent simulation. In this work, grid refinement is used to follow the movement of the interface, in a band surrounding the interface and at the base of the domain. The grids are also adapted to provide high resolution in areas of high vorticity in the flow.

Remeshing of the grid operates by dividing a cell into four if it lies on the interface, or is in a region of high vorticity; derefinement takes place by removing four sibling cells and replacing them with their parent if each of the four sibling cells lies away from the interface and has low vorticity. Variables are interpolated onto new cells using bi-linear interpolation from the neighbours of the divided cell. Alternatively, when 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 following adaptation of the grid is described by Greaves [14] for advection test cases. However, for the free surface flow calculations, in which a refinement band is used, the grid is always adapted away from the air–water interface. This means that newly divided cells will have the same volume fraction as their parent, and the parent of removed cells will have the same volume fraction as the cells removed.

5. RESULTS

5.1. Driven cavity

The lid-driven cavity is a well-known benchmark test, considered by Ferziger and Perić [16] amongst others, and is used here to assess the accuracy of the adapting quadtree method for solution of the Navier–Stokes equations. Both uniform and adapting quadtree grids are used to simulate a square cavity filled with a single viscous fluid driven in motion by a moving lid. The Reynolds number, based on the lid velocity and width of the cavity, is 1000 and no-slip boundary conditions are applied on all walls. The results are compared with fine mesh calculations made by Ghia *et al.* [22] using the vorticity stream function formulation. This case, using the SIMPLE Navier–Stokes solution method, is used to compare the alternative face velocity calculations: power law, deferred correction and Jasak *et al.* [18] scheme; and use of adapting quadtree versus uniform grids.

First, grid convergence is considered in steady flow simulations using the SIMPLE scheme, together with deferred correction for the face velocities with the blending factor equal to 1.0 (equivalent to CDS), on uniform 32×32, 64×64 and 128×128 grids. The horizontal velocity along the vertical axis is plotted in Figure 2(a) and the vertical velocity along the horizontal axis is plotted for comparison with Ghia *et al.*'s [22] results in Figure 2(b). It is clear that results improve as the grid size is increased and the 128×128 grid solution is very close to Ghia *et al.*'s [22] solution.

Grid convergence may be assessed by calculating the grid convergence index (GCI) recommended by Roache [23]. The GCI relates the grid convergence error to that which would be expected if grid doubling and a second-order accurate method were used. The GCI for the fine grid solution is expressed as

$$\mathrm{GCI[fine\ grid]} = 3|\varepsilon|/(r^p - 1) \tag{18}$$

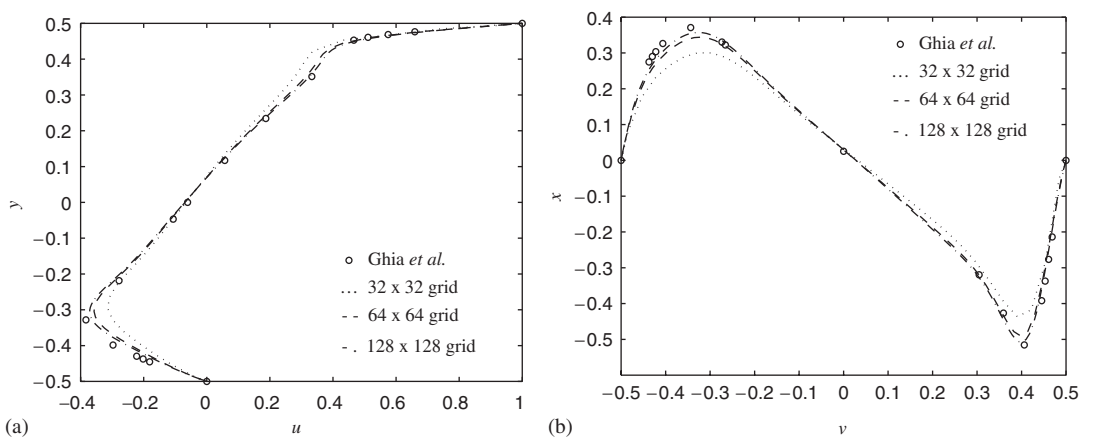


Figure 2. Comparison of grid sizes: (a) u-velocity along vertical axis; and (b) v-velocity along horizontal axis.

where the error, $\varepsilon = (f_2 - f_1)/f_1$ and f_k is the solution on the coarse ($k = 2$) and fine ($k = 1$) grids. The grid refinement ratio, $r = h_2/h_1$, where h_k is the grid spacing and p is the order or accuracy of the method. In situations where the grid is unstructured and h_k varies through the grid, such as the quadtrees used here, an effective refinement ratio is defined, $r = (N_1/N_2)^{1/D}$ using the number of elements in the coarse (N_2) and fine (N_1) grids and the dimensionality of the problem, D. The GCI values are calculated using the u-velocity solution at Ghia *et al.*'s [22] data point positions vertically along the centreline of the cavity. The GCI for the 128×128 grid to the 64×64, $\mathrm{GCI}_{7766} = 7.69\%$ and for the 64×64 grid to the 32×32 grid, $\mathrm{GCI}_{6655} = 24.68\%$. Considering Ghia *et al.*'s [22] solution to be exact, the magnitude of the fine grid error is 2.16%.

Next, similar results are given in Figure 3 to compare the different methods for calculating the face velocities used in the discretized momentum equations. Here, a 64×64 grid is used for the calculation using the SIMPLE scheme for steady flow together with each of the power law scheme, deferred correction with the blending factor equal to 1.0 (equivalent to CDS) and the Jasak *et al.* [18] scheme for face velocities. Use of the deferred correction scheme clearly improves the accuracy over the power law scheme for a given size of grid. The Jasak *et al.* [18] and deferred correction schemes show no noticeable difference.

The schemes are combined with an adaptive quadtree mesh generator of maximum division level 6 and minimum level 3 and the driven cavity simulation repeated with the Jasak *et al.* [18] scheme for face velocities. The size of the maximum division level cells is the same as for the 64×64 grid. The steady simulation is calculated for 50 time steps with grid adaptation at each time step to provide refinement in areas of high vorticity (as recommended by Jeong and Yang [10, 11]). The adapted grid and velocity vectors are shown in Figure 4 and the velocity components along the vertical and horizontal axes are plotted in Figure 5. The final adapted grid contains 2169 cells and the results agree well with the results calculated on a regular grid of the smallest cells, containing 4096 cells. Thus, use of quadtree grid adaptation leads to a considerable reduction in grid size for a solution of comparable accuracy. The GCI for this case, calculated according to Roache's method for unstructured grids [23], $\mathrm{GCI}_{6663} = 20.38\%$.

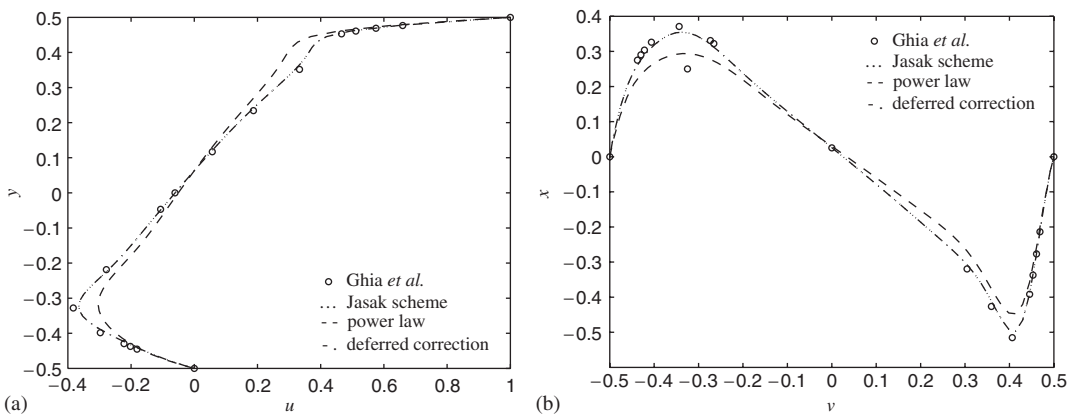


Figure 3. Comparison of velocity interpolation scheme: (a) u-velocity along vertical axis; and (b) v-velocity along horizontal axis.

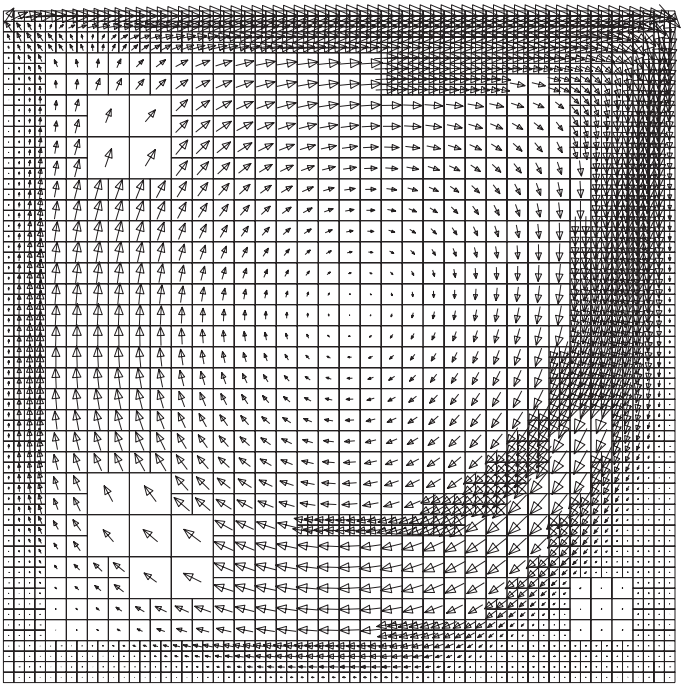


Figure 4. Adapted grid and velocity vectors.

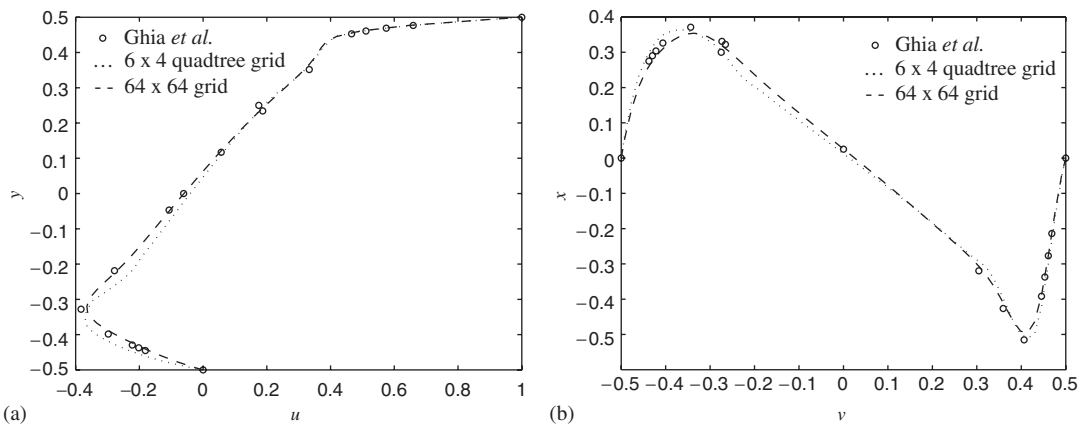


Figure 5. Adapted grid results: (a) u-velocity along vertical axis; and (b) v-velocity along horizontal axis.

5.2. *Collapse of a water column*

The collapse of a water column has been investigated numerically by various researchers, such as Ubbink [7], Jeong and Yang [10, 11], Qian *et al.* [24] and Andrillon and Alessan-

drini [25]. Experimental data have been provided by Martin and Moyce [26] and Koshizuka *et al.* [27]. Here, the quadtree adaptive high-resolution interface-capturing method is combined with both SIMPLE and PISO Navier–Stokes flow solvers for the unsteady flow simulation, and results are calculated on both uniform and quadtree grids. 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 using the SIMPLE scheme on uniform grids with time step $\mathrm{d}t = 0.0001$ s are presented by Greaves [14]. The simulations are repeated here on a series of uniform grids using the PISO formulation and found to be identical to those calculated using SIMPLE. 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 in a band surrounding the free surface. Results calculated on a 7×5 quadtree grid, with a refinement band of 14 cells at the interface, and time step $\mathrm{d}t = 0.0001$ s, are first 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. In Figure 6 the reconstructed interface and adapted quadtree grids are plotted together alongside the video images taken by Koshizuka *et al.* [27]. The results agree well with those presented by Ubbink [7] and Qian *et al.* [24] and the video images [27]. The time histories of the fronts are plotted in Figure 7 together with the results calculated on the equivalent 128×128 uniform grid. The non-dimensional height of the water column at the left wall versus the non-dimensional time is shown in Figure 7(a). The predicted height is the same for all three grids and agrees very well with the experimental data obtained by Martin and Moyce [26], which is plotted alongside the numerical data. The non-dimensional position of the leading edge is plotted against non-dimensional time in Figure 7(b). Martin and Moyce [26] presented two different sets of experimental data for this case. As observed by Ubbink [7] and others [25], the discrepancy between the experimental and numerical results here is possibly due to the difficulty in determining the exact location of the leading edge. A thin layer, similar to a jet, shoots along the base of the tank, which is difficult to capture. The time histories for the quadtree calculation are identical to those calculated on the equivalent uniform grid, thus showing that the same accuracy can be achieved on the quadtree grid as on its corresponding equivalent uniform grid.

In Reference [14], it was found that a band of 6 cells was required for the 5×3 quadtree calculation. Here, we use a Gauss' theorem approach to the pressure gradient terms and a conservative fluxing scheme with improved interpolation at hanging nodes. These improvements reduce the size of the required band around the interface to 2 cells, and thus reduce the overall number of quadtree cells (for the 7×5 grid, the refinement band is reduced in this way from 30 to 14 cells). Calculations on a 5×3 quadtree grid were made to assess the influence of the face velocity interpolation in the momentum equations on the accuracy of the solution. Each of the power law scheme, deferred correction, with a blending factor of 0.5, and the Jasak *et al.* [18] scheme were used and the results found to be identical.

Furthermore, use of adaptive quadtrees significantly reduces the size of the calculation grid and the CPU per time step without loss of accuracy (each simulation is run with time

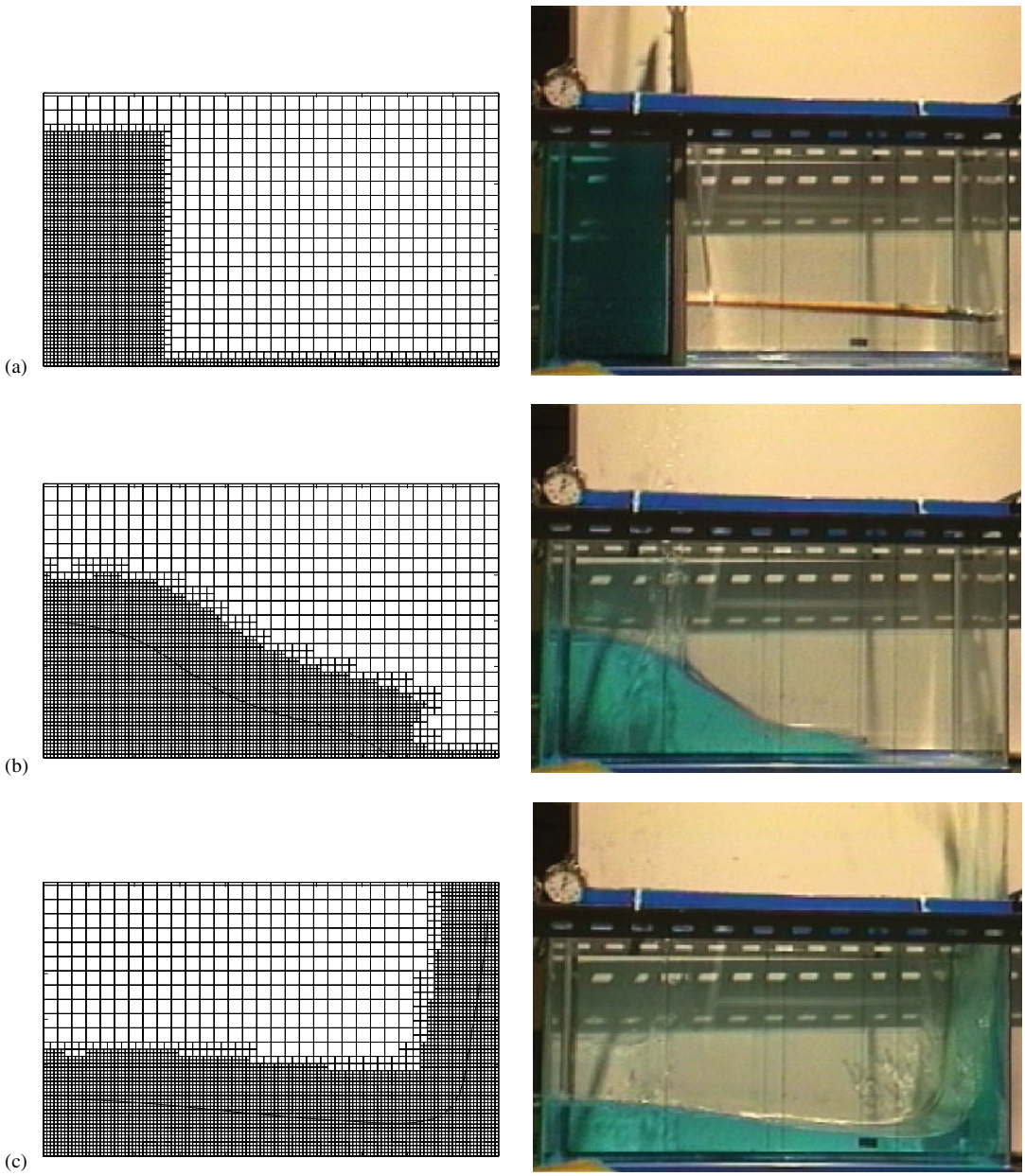
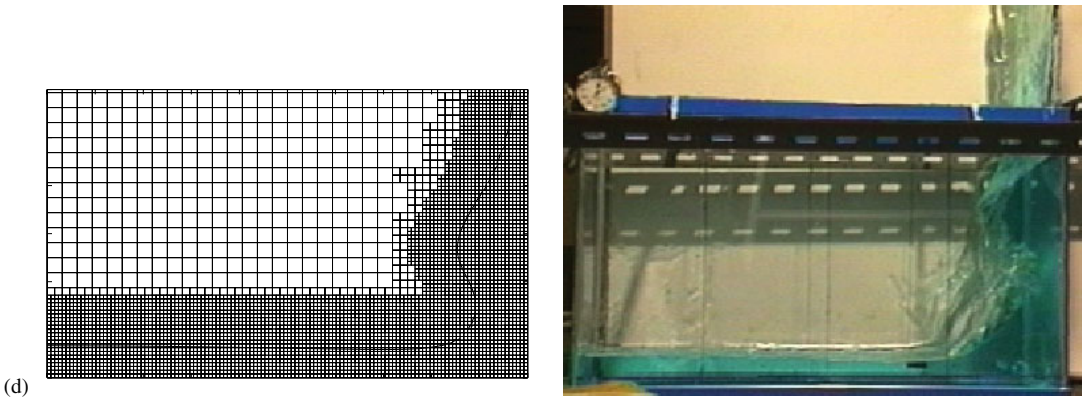
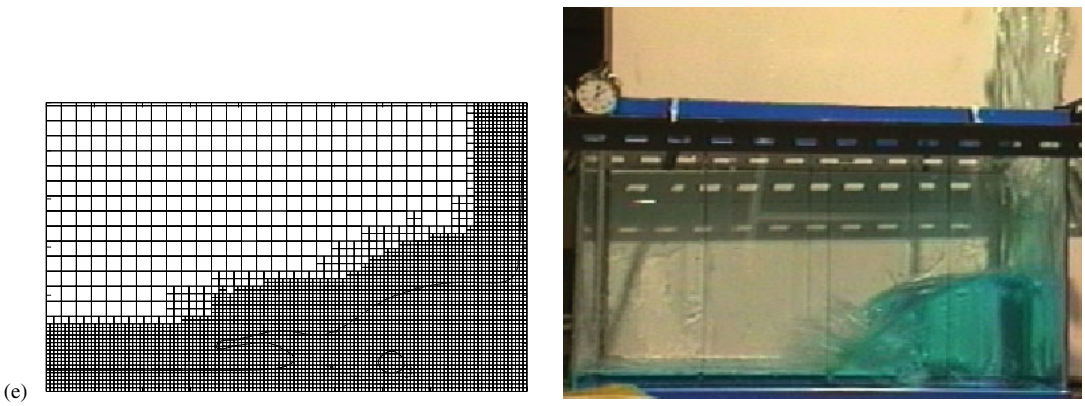


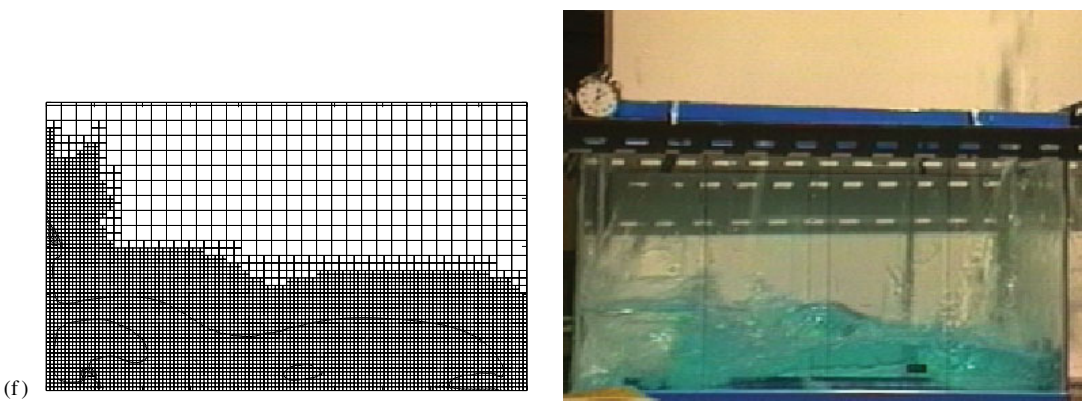
Figure 6. Interface and adapted 7×5 quadtree grid: (a) $T = 0$; (b) $T = 1.617$; (c) $T = 3.233$; (d) $T = 4.850$; (e) $T = 6.466$; and (f) $T = 8.029$.



(d)



(e)



(f)

Figure 6. *Continued.*

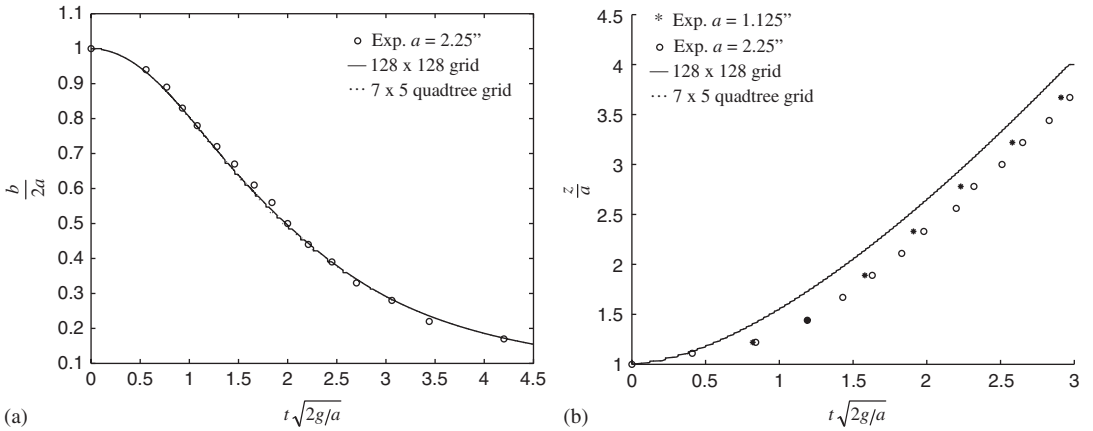


Figure 7. Comparison of adapted grid results with equivalent uniform grid: (a) height of water column; and (b) position of leading edge.

Table I. Summary of grid size and CPU for simulation of water column collapse on uniform and quadtree grids.

Grid type	Solution type			Typical no. of cells	CPU per time step (s)	Fine grid GCI (%)
	PISO	Max level	Min level			
Uniform	Matrix	5	5	1024	0.17	$GCI_{6655} = 1.22$
Quadtree	Matrix	5	3	367	0.05	$GCI_{5553} = 1.47$
Uniform	Matrix	6	6	4096	1.34	$GCI_{7766} = 0.03$
Quadtree	Matrix	6	4	1540	0.335	$GCI_{6664} = 0.07$
Uniform	Matrix	7	7	16384	10.50	
Quadtree	Matrix	7	5	5563	2.99	$GCI_{7775} = 0.01$
Quadtree	Iteration SIMPLE	7	5	5563	1.98	
Quadtree	Matrix	6	4	1540	6.11	

step $dt = 0.0001$ s). Table I summarizes the grid size and CPU per time step for the dam break calculation made on a series of quadtree and equivalent regular grids. All calculations were made on a SUNFIRE 480R with 16 Gb RAM, four 900 MHz UltraSPARC-III+ CPU's and two 36 Gb (internal) HD's running Solaris 8. The fine grid GCI values are given in Table I, calculated using (18) in which the error is based on the difference in water column height predicted by each grid and averaged over the first 2000 time steps of the simulation. Most of the simulations summarized in Table I use the PISO formulation with direct solution of the simultaneous equations at each time step. This means that the matrix of size N^2 must be stored. For the smaller grids, this works well and efficiently, but as the grid size increases the storage required becomes excessive and it is more efficient to solve the simultaneous equations by iteration. If point-by-point iteration is used, the CPU time required for the 7×5 quadtree grid is 1.98 s per time step, a saving of approximately one third compared with

matrix inversion. This saving is more significant as the grid size is increased. If the SIMPLE scheme is used, 6.11 s of CPU time per time step ($dt = 0.0001$ s) are required for the 6×4 quadtree grid calculation, approximately 18.5 times more than with PISO.

5.3. *Collapse with an obstacle*

A more extreme case, which has been considered by Andrillon and Alessandrini [25], Koshizuka *et al.* [27] and Ubbink [7] amongst others, is when an obstacle is placed in

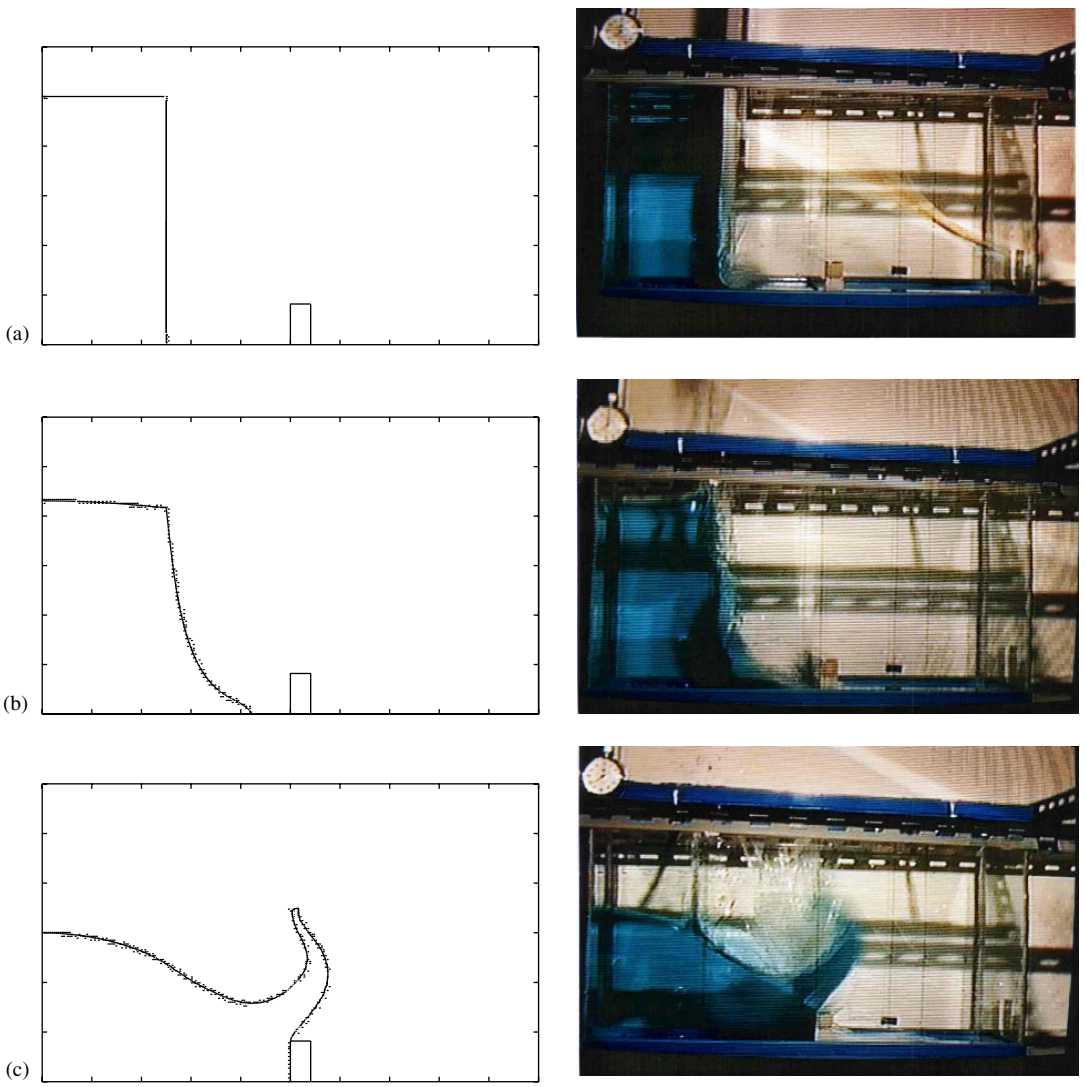


Figure 8. Interface calculated using 8×5 adapting quadtree grid (LHS) and photographs from Koshizuka *et al.* [25] (RHS) at non-dimensional time increments of 0.809: (a) $T = 0.0$; (b) $T = 0.809$; (c) $T = 1.617$; (d) $T = 2.426$; (e) $T = 3.234$; and (f) $T = 4.043$.

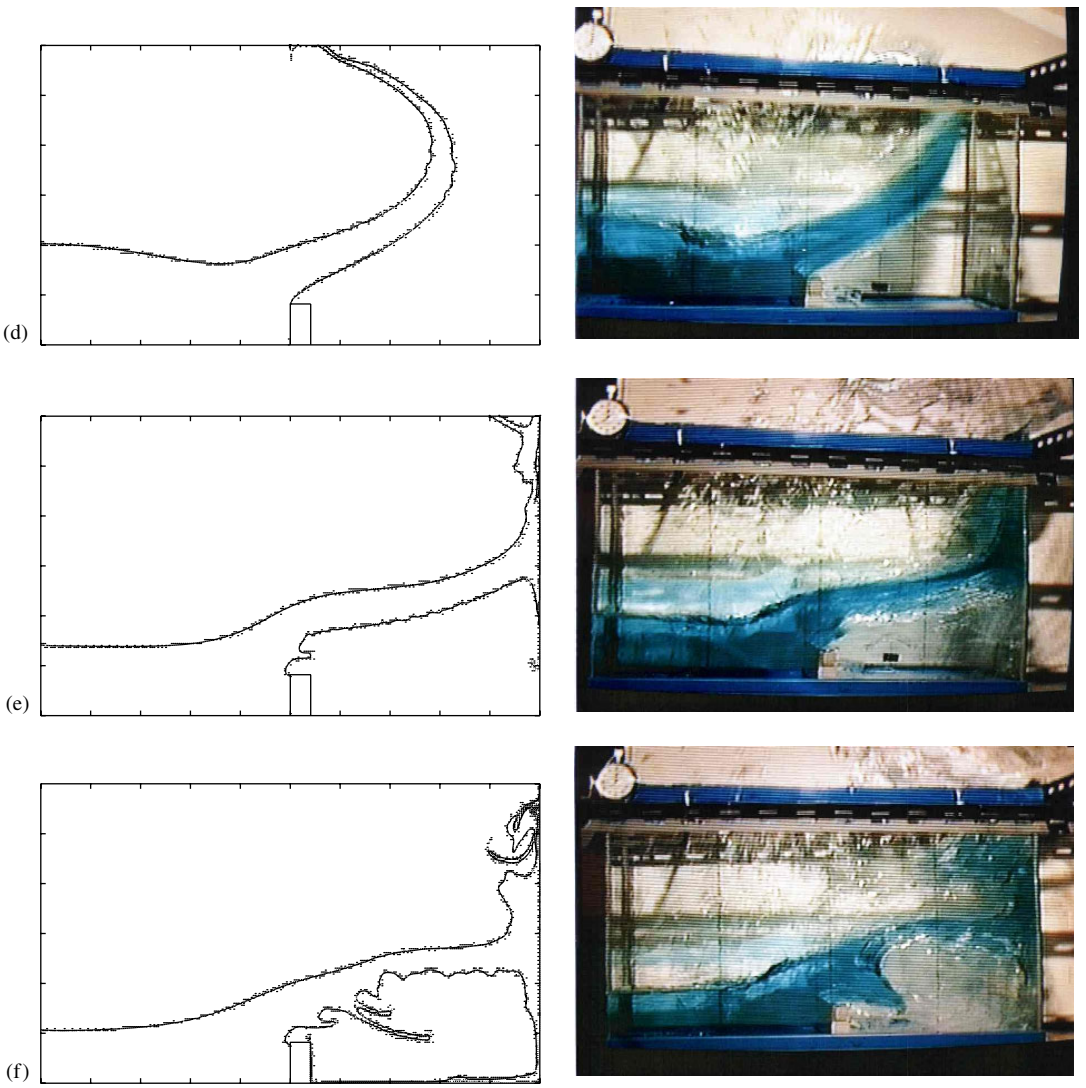


Figure 8. *Continued.*

the way of the wave front. The set-up is similar to that described in Section 5.2 with the addition of a rectangular obstacle, 0.04-unit wide $\times$ 0.08-unit high placed on the bottom of a tank, 1-unit wide $\times$ 0.6-unit high, with its leading edge at the centre of the tank. This case was simulated using an 8×5 adapting quadtree grid, having a refined interface band of 30 cells, with PISO and power law face velocity interpolation. Results showing the reconstructed interface at non-dimensional time steps, $T = t\sqrt{g/a} = 0, 0.809, 1.617, 2.426, 3.233$ and 4.043 are given in Figure 8 together with still photographs taken by Koshizuka *et al.* [27]. The free surface motion is predicted well for the first few time steps considered, but is in less

agreement at later times when the surface is highly contorted with considerable spray, breakup and overturning.

At time $T = 0.809$, the photograph shows spray at the obstacle, indicating that the jet that rushes along the bottom of the tank has reached the obstacle. However, the predicted leading edge has not yet reached the obstacle. At $T = 1.617$, the tongue of water deflected up from the obstacle is satisfactorily predicted by the numerical scheme, although details of the spray breakup are not resolved. Again, the predicted shape of the tongue of water is similar to that of the experiment at $T = 2.426$ as the tongue moves towards the right-hand wall. At $T = 3.233$, the tongue of water has made contact with the right-hand wall and begins to fall under gravity. It traps air beneath it, which resists the downward fall of the water. The numerical solution agrees well with the experiment at this point and the secondary tongue immediately above the obstacle is resolved.

At $T = 4.043$, a secondary tongue shoots into the trapped air space and towards the base of the tank. This feature is captured by the numerical scheme, but the tongue is considerably narrower and contains less water than in the photograph. The predicted interface at this point shows a larger volume of trapped air and a smaller volume of water approaching the base of the tank than in the experiment, although the shape of the primary tongue hitting the right-hand wall, overturning above and running down to the base of the tank below the point of impact is correctly reproduced. This simulation demonstrates the capability of the method to deal with large-scale wave breaking, although it is likely that use of a finer grid is necessary to resolve better the finer details of the flow.

6. 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 together with either SIMPLE or PISO for solution of the Navier–Stokes equations. Various approaches to the interpolation of face velocities used in the discrete equations are investigated; the Jasak *et al.* [18] scheme is found to work best for the steady flow lid-driven cavity case considered and the power law scheme to perform most reliably for unsteady simulation of water column collapse. For the water column collapse simulations, the grid adapts at each time step to provide a band of refinement of prescribed width around the free surface. At hanging nodes, use of a conservative fluxing scheme with first-order accurate interpolations and a Gauss' theorem approach to the pressure gradient terms reduces the necessary width of the band of refinement around the interface, when compared with Reference [14].

Results calculated for the lid-driven cavity and water column collapse are in excellent agreement with experimental and other numerical data and a sharp interface is maintained at the free surface. Use of adaptive quadtree grids achieves results of the same accuracy as the corresponding equivalent uniform grid, but with considerable savings in grid size of a factor of approximately 2.9, and in CPU of up to a factor of 4. When comparing the use of SIMPLE and PISO for unsteady simulation of water column collapse, PISO is approximately 18.5 times faster than SIMPLE. The new scheme is found to work well for highly complex free surface dynamics occurring when a falling water column interacts with an obstacle.

ACKNOWLEDGEMENTS

The author is very grateful to the Royal Society for supporting this work. The author also gratefully acknowledges several informative discussions with Ping Wang Jr.

REFERENCES

1. Wu GX, Ma QW, Eatock Taylor R. Numerical simulation of sloshing waves in a 3D tank based on a finite element method. *Applied Ocean Research* 2001; **20**:337–355.
2. Hyman JM. Numerical methods for tracking interfaces. *Physica* 1984; **12D**:396–407.
3. Monaghan JJ. Simulating free surface flows with SPH. *Journal of Computational Physics* 1994; **65**:179–214.
4. Lin H, Atluri SN. The meshless local Petrov–Galerkin (MLPG) method for solving incompressible Navier–Stokes equations. *Computer Modelling in Engineering and Sciences* 2001; **2**(2):117–142.
5. Causon DM. An efficient front tracking algorithm for multi-component fluid calculations with biomedical applications. *Zeitschrift fur Angewandte Mathematik und Mechanik* 1996; **76**(S1):371–372.
6. Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**:201–225.
7. Ubbink O. Numerical prediction of two fluid systems with sharp interfaces. *Ph.D. Thesis*, Imperial College of Science, Technology and Medicine, London, 1997.
8. 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.
9. Wang JP, Borthwick AGL, Eatock Taylor R. Finite-volume-type VOF method on dynamically adaptive quadtree grids. *International Journal for Numerical Methods in Fluids* 2004; **45**(5):485–508.
10. 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.
11. 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.
12. Popinet S. Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries. *Journal of Computational Physics* 2003; **190**:572–600.
13. Patanker SV. *Numerical Heat Transfer and Fluid Flow*. Taylor and Francis, Hemisphere Publishing Corporation: U.S.A., 1980.
14. Greaves DM. Simulation of interface and free surface flows in a viscous fluid using adapting quadtree grids. *International Journal for Numerical Methods in Fluids* 2004; **44**:1093–1117.
15. Issa RI. Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of Computational Physics* 1986; **62**(1):40–65.
16. Ferziger JH, Perić M. *Computational Methods for Fluid Dynamics*. Springer: Berlin, Heidelberg, 1996 (ISBN 3-540-59434-5).
17. 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.
18. Jasak H, Weller, HC, Issa RI, Gosman AD. High resolution NVD differencing scheme for arbitrarily unstructured meshes. *International Journal for Numerical Methods in Fluids* 1999; **31**(2):431–449.
19. Leonard BP. The ULTIMATE conservative difference scheme applied to unsteady one-dimensional advection. *Computational Methods in Applied Mechanics and Engineering* 1991; **19**:59–98.
20. Greaves DM, Borthwick AGL. Hierarchical tree-based finite element mesh generation. *International Journal for Numerical Methods in Engineering* 1999; **45**:447–471.
21. Greaves DM. A quadtree adaptive method for simulating fluid flows with moving interfaces. *Journal of Computational Physics* 2004; **194**/**1**:35–56.
22. Ghia U, Ghia KN, Shin CT. High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method. *Journal of Computational Physics* 1982; **48**:387–411.
23. Roache PJ. Perspective: a method for uniform reporting of grid refinement studies. *Journal of Fluids Engineering* (ASME) 1994; **116**:405–413.
24. Qian L, Causon DM, Ingram DM, Mingham CG. An efficient two-fluid solver for hydraulic flow problems. *Journal of Hydraulic Engineering* 2003; **129**(9):688–696.
25. Andrillon Y, Alessandrini B. A 2D+T VOF fully coupled formulation for the calculation of breaking free-surface flow. *Journal of Marine Science Technology* 2004; **8**:159–168.
26. 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* 1952; **244**:312–324.
27. Koshizuka S, Tamako H, Oka Y. A particle method for incompressible viscous flow with fluid fragmentation. *Computational Fluid Mechanics Journal* 1995; **113**:134–147.

