INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2004; **45**:485–508 (DOI: 10.1002/fld.712)

Finite-volume-type VOF method on dynamically adaptive quadtree grids

J. P. Wang*,†, A. G. L. Borthwick and R. Eatock Taylor

Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, U.K.

SUMMARY

This paper describes a finite-volume volume-of-fluid (VOF) method for simulating viscous free surface flows on dynamically adaptive quadtree grids. The scheme is computationally efficient in that it provides relatively fine grid resolution at the gas–liquid interface and coarse grid density in regions where flow variable gradients are small. Special interpolations are used to ensure volume flux conservation where differently sized neighbour cells occur. The numerical model is validated for advection of dyed fluid in unidirectional and rotating flows, and for two-dimensional viscous sloshing in a rectangular tank. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: finite-volume; VOF; viscous free surface; dynamically adaptive; quadtree grid

1. INTRODUCTION

Dynamically adaptive grids can facilitate high resolution simulations of complicated flow fields without making excessive demands on computer resources. Of the various grid generation techniques available, hierarchical methods are automatic, robust and efficient to implement. A hierarchical data tree holds cell information in a structured form that is easy to interrogate. Grid adaptation simply involves the addition or removal of leaf cells from the data tree according to user-specified rules. For two-dimensional simulations, commonly used hierarchical grid techniques are quadtrees for rectangular grids (see e.g. References [1–3]) and tritrees for triangular grids (see e.g. References [4, 5]).

The volume-of-fluid (VOF) method was originally proposed by Hirt and Nichols [6], who used a finite-difference donor–acceptor scheme to capture the moving free surface interface of a liquid. The VOF method has since been the subject of much development (e.g. References [7–10]) and applied widely (e.g. References [11–13]). In implementing the VOF

*Correspondence to: J. P. Wang, 4F, No. 6, Lane 307, Chun-Cheng Road, Hsin-Tien City, Taipei County (231), Taiwan, R.O.C.
†E-mail: jpwang100@hotmail.com

Contract/grant sponsor: Scholarship from the Taiwan Government, R.O.C.
Contract/grant sponsor: EPSRC, U.K.; contract/grant number: GR/N22595

Published online 8 April 2004
Received 21 June 2003
Revised 22 January 2004

method it is necessary to minimize smearing of a time-varying volume-fraction function F. Kothe and Rider [14], and Rider and Kothe [15] provide a detailed discussion of this issue. To this end, Ubbink and Issa [16] propose a high resolution finite-volume-type VOF method, the CICSAM scheme (Compressive Interface Capturing Scheme for Arbitrary Meshes), which treats the whole gas–liquid domain as a single fluid system with variable density and viscosity. There is no need to specify kinematic and dynamic free surface boundary conditions at the gas–liquid interface. Herein, a VOF (CICSAM) model is based on dynamically adaptive quadtree grids to achieve enhanced resolution at the gas–liquid interface of free surface viscous flows. Similar concepts are employed by Sussman *et al.* [17] who have used adaptive mesh refinement with a level-set interface tracking algorithm.

A major difficulty encountered with quadtree grids arises at the faces between neighbouring cells of different sizes where hanging nodes occur. In the present study, special interpolation schemes are derived, which utilize solely the values at nearest neighbour cells in the formulation. A minimal interpolation stencil is used that conserves volume fluxes at cell faces, an important property when solving the discretized governing equations [18].

Section 2 provides a brief description of the quadtree grid generator. The governing equations for viscous incompressible fluid flows are then discretized using the finite-volume method for a collocated arrangement of flow variables. The new volume flux conserved interpolation scheme is used to evaluate the discretized governing equations on quadtree grids. Results are presented for the following validation tests: (i) colour advection (see Reference [19]); and (ii) two-dimensional viscous sloshing in a rectangular tank.

2. QUADTREE GRID GENERATION

2.1. Quadtree grid generator and cell numbering system

Quadtree grid generation involves the subdivision of any square cell into four equal size child cells according to user-specified criteria, e.g. the presence of digitized seeding points that delineate the domain boundary. The quadtree grid generator is based on that of Greaves and Borthwick [20], and Rogers *et al.* [21]. The procedure is outlined below:

1. Create a root square that encloses the entire computational domain.
2. Digitize wall boundary into a set of discrete seeding points.
3. Divide the root square into four equal size child (or leaf) cells. This is called level one subdivision.
4. Check each leaf cell in turn. If the subdivision criterion is met, then divide the cell into four further child (leaf) cells.
5. Store information including parent cell, sister cells, neighbouring cells, cell-centre co-ordinates, cell boundary co-ordinates, subdivision level, etc. in an array.
6. Continue the leaf cell subdivision process until either the maximum subdivision level is reached or the subdivision criterion is no longer met.
7. Regularize the grid so that the subdivision level difference between two neighbouring cells is limited to 1.

The cell numbering system is the same as that proposed by van Dommelen and Rundensteiner [1]. Neighbour finding is undertaken using the nearest common ancestor

concept of Samet [22] with cell identification numbers manipulated according to rules given by Greaves and Borthwick [20] and Rogers *et al.* [21].

2.2. Adaptation of the quadtree grid

The dynamic adaptation procedure is as follows:

1. If a leaf cell meets a specified grid enrichment criterion, the cell is subdivided and four new leaf cells are created.
2. If each of four sister leaf cells meet a given grid coarsening criterion, all four sister cells are replaced by their parent cell.
3. Grid regularization is undertaken at the end of the adaptation process.

The adaptation process is implemented at prescribed time intervals during unsteady flow simulation.

2.3. Quadtree grid generation according to the volume-fraction values

Assuming that there are two immiscible fluids, Fluid-1 and Fluid-2, the volume-fraction value F of Fluid-1 in a control volume is defined as

$$F = \frac{\text{Volume of fluid-1}}{\text{Total volume of the control volume}} \tag{1}$$

The volume-fraction value F can also serve as a flow-related variable used by quadtree cell adaptation criterion. If a cell or any one of its neighbours has an F value in the range $0 < F < 1$, cell subdivision is required, unless the cell is already at a user prescribed maximum resolution level. Figure 1 shows a typical quadtree grid generated according to volume-fraction values for free surface sloshing in a tank, where the maximum subdivision level is 7. To account for the high flow gradients near the free surface, the cells are subdivided to the maximum resolution level for a user-prescribed number of layers either side of the free surface. An example is shown in Figure 2 for 5 layers either side of the free surface with cells set to a maximum subdivision level of 7. This strategy avoids linear interpolation having to be applied between different size neighbouring cells located near the free surface.

3. MATHEMATICAL FORMULATION

The governing equations of viscous incompressible Newtonian fluid flows are the continuity and Navier–Stokes momentum equations:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_k}{\partial x_k} = 0 \tag{2}$$

$$\frac{\partial}{\partial t}(\rho u_j) + \frac{\partial}{\partial x_k}(\rho u_j u_k) = -\frac{\partial p}{\partial x_j} + \frac{\partial}{\partial x_i}\left[\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] + \rho g_j \tag{3}$$

where ρ is the fluid density, u_k the velocity component in the k-direction, t the time, p the pressure, μ the fluid dynamic viscosity and g_j the body force component in the j-direction.

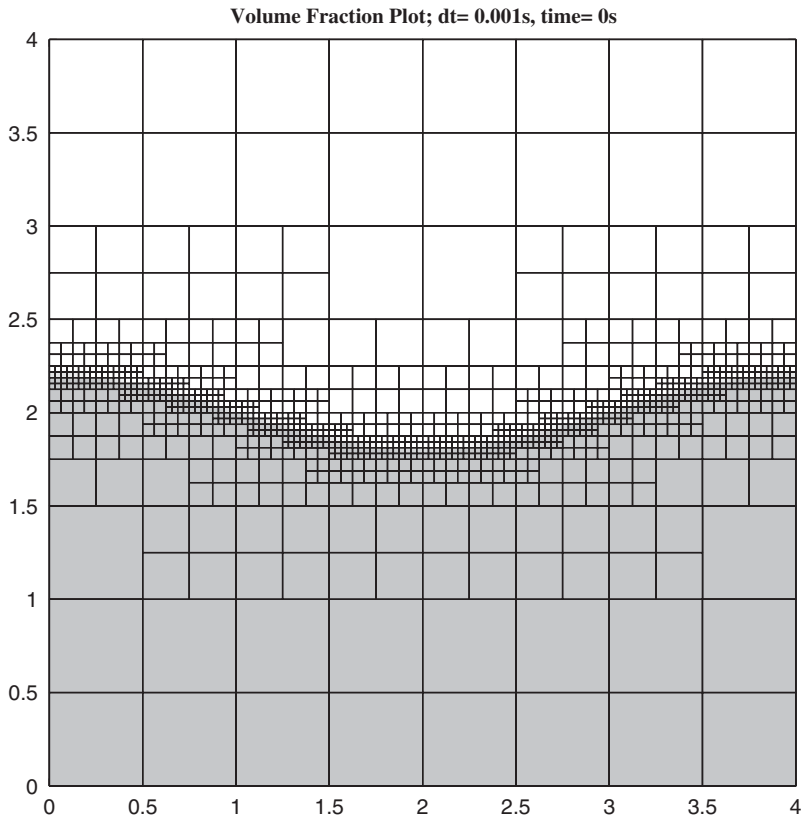

Figure 1. Quadtree grid generated according to volume-fraction F values.

The volume-fraction value F is updated by the transport equation:

$$\frac{\mathrm{D}F}{\mathrm{D}t} = \frac{\partial F}{\partial t} + u_k \frac{\partial F}{\partial x_k} = 0 \tag{4}$$

The density and dynamic viscosity of each control volume (cell) are:

$$\rho = F\rho_1 + (1 - F)\rho_2 \tag{5}$$

and

$$\mu = F\mu_1 + (1 - F)\mu_2 \tag{6}$$

where ρ_1, ρ_2, μ_1 and μ_2 are the densities and dynamic viscosities of Fluid-1 and Fluid-2, respectively.

From the above definition, the quantity ρu is discontinuous at the interface. In order to solve the whole domain as a one-fluid system (with variable density), the continuity equation

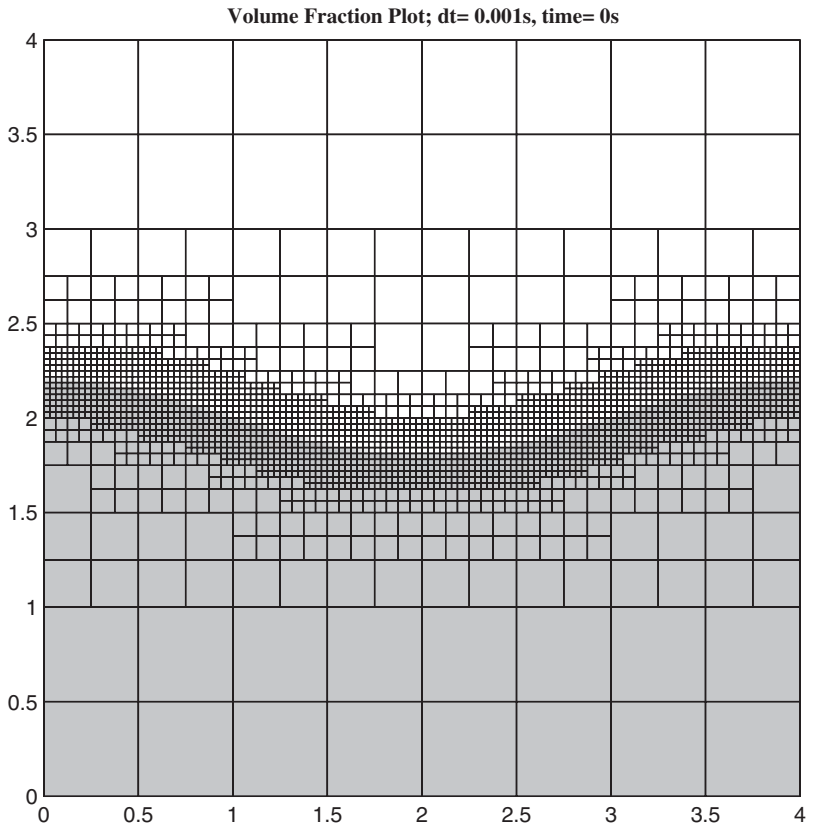

Figure 2. Quadtree grid generated using user-specified layers of cell subdivision in the vicinity of the gas–liquid interface.

is reformulated as follows [23]:

$$\frac{\partial u_k}{\partial x_k} = \frac{-1}{\rho}\left(\frac{\partial \rho}{\partial t} + u_k \frac{\partial \rho}{\partial x_k}\right) = \frac{-1}{\rho}\frac{\mathrm{D}\rho}{\mathrm{D}t} \tag{7}$$

Although Equation (7) is a non-conservative form of the continuity equation, it is suitable for numerical solution of two-fluid systems with high density ratios, e.g. liquid and gas. This is because the velocity is continuous at the interface, otherwise the interface would be torn apart [24].

It is assumed that both fluids are incompressible, and so $\mathrm{D}\rho_1/\mathrm{D}t = 0$ and $\mathrm{D}\rho_2/\mathrm{D}t = 0$. By substituting (5) into (7) and using (4), the continuity equation (7) becomes

$$\frac{\partial u_k}{\partial x_k} = \frac{-(\rho_1 - \rho_2)}{\rho}\left(\frac{\mathrm{D}F}{\mathrm{D}t}\right) = 0 \tag{8}$$

It should be noted that although the dynamic viscosity μ is constant in each fluid region, it is not constant over the whole fluid domain. Therefore, the Navier–Stokes equations have

the following form:

$$\frac{\partial}{\partial t}(\rho u_j) + \frac{\partial}{\partial x_k}(\rho u_j u_k) - \mu \frac{\partial^2 u_j}{\partial x_i \partial x_i} = -\frac{\partial p}{\partial x_j} + \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\frac{\partial \mu}{\partial x_i} + \rho g_j \tag{9}$$

Using (8), the volume-fraction transport equation (4) can be rewritten as

$$\frac{\partial F}{\partial t} + \frac{\partial (F u_k)}{\partial x_k} = 0 \tag{10}$$

In summary, for the flow problems encountered in the present study, the governing equations that need to be solved simultaneously are the continuity equation (8), the Navier–Stokes equations (9) and the volume-fraction transport equation (10), together with the constitutive relations for density (5), viscosity (6) and suitable wall boundary conditions.

4. FINITE-VOLUME DISCRETIZATION ON A UNIFORM GRID

4.1. Discretization of the volume-fraction transport equation

Integrating the volume-fraction transport equation (10) over the control volume, gives

$$\int_V \frac{\partial F}{\partial t}\,\mathrm{d}V + \int_V \frac{\partial}{\partial x_k}(F u_k)\,\mathrm{d}V = 0 \tag{11}$$

where V is the control volume. Using the finite-volume discretization with a Crank–Nicolson scheme at the cell face F values, and assuming small variation of volumetric flux $\dot{V}$, (11) may be written

$$\frac{1}{\delta t}(F_P^{t+\delta t} - F_P^t)V_P = -\sum_{f=1}^{N} \frac{1}{2}(F_f^t + F_f^{t+\delta t})\dot{V}_f \tag{12}$$

where the subscript P is the centroid of the control volume, the subscript f refers to values evaluated at the cell face-centre locations, and $\dot{V}$ is the volumetric flux at the control volume face:

$$\dot{V}_f = (u_k A_k)_f \tag{13}$$

where A_k is the k-component of control surface area. The calculation of the volumetric flux $\dot{V}$ will be described later in Section 4.3.

Equation (12) can be rearranged to

$$F_p^{t+\delta t}\frac{V_P}{\delta t} + \sum_{f=1}^{N} \frac{1}{2} F_f^{t+\delta t}\dot{V}_f = S_{F_P} \tag{14}$$

where the source term S_{F_P} contains the F values at the previous time level, and is defined as

$$S_{F_P} = F_P^t \frac{V_P}{\delta t} - \sum_{f=1}^{N} \frac{1}{2} F_f^t \dot{V}_f \tag{15}$$

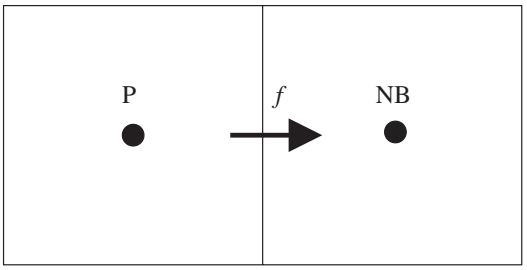

Figure 3. Cell-P and its nearest face neighbour at cell face-f.

In (14), the face volume-fraction value F_f is interpolated from the cell-centre volume-fraction values for the collocated variable arrangement. Using the CICSAM scheme proposed by Ubbink and Issa [16] the face volume-fraction value is interpolated from the cell-centre volume-fraction value of a cell and its nearest face neighbour (see Figure 3):

$$F_f = (1 - \beta_f)F_P + \beta F_{\mathrm{NB}} \tag{16}$$

where

$$\beta_f = \frac{\tilde{F}_f - \tilde{F}_D}{1 - \tilde{F}_D} \tag{17}$$

The function $\tilde{F}$ is defined as

$$\tilde{F} = \frac{F - F_U}{F_A - F_U} \tag{18}$$

where subscripts A and U refers to acceptor and upwind cell, respectively. The function $\tilde{F}_f$ is calculated using a blended Hyper-C scheme [16]. Hence, (14) can be rewritten as

$$\mathscr{A}_{F_P} F_P^{t+\delta t} + \sum_{\mathrm{nb}=1}^{N} \mathscr{A}_{F_{\mathrm{nb}}} F_{\mathrm{nb}}^{t+\delta t} = S_{F_P} \tag{19}$$

where the subscript nb refers to the nearest face neighbours, $\mathscr{A}_{F_P}$ is the coefficient of the volume-fraction equation of Cell-P, and $\mathscr{A}_{F_{\mathrm{nb}}}$ are the coefficients of the volume-fraction equation of the nearest face neighbours.

Equation (19) is the finite-volume discretization of the volume-fraction transport advection equation.

4.2. *Discretization of the Navier–Stokes equations*

The finite-volume-type discretization of the Navier–Stokes equations on a uniform grid is similar to that proposed by Ubbink and Issa [16].

4.2.1. Convection term

$$\int_V \frac{\partial}{\partial x_k}(\rho u_j u_k)\,\mathrm{d}V \approx \sum_{f=1}^{N}(\rho \dot{V} u_j)_f \tag{20}$$

The face density is approximated using (5):

$$\rho_f = F_f^* \rho_1 + (1 - F_f^*)\rho_2 \tag{21}$$

where

$$F_f^* = \frac{1}{2}(F_f^t + F_f^{t+\delta t}) \tag{22}$$

A high resolution convection–diffusion differencing scheme (HR scheme) proposed by Jasak [25] is used to approximate the face velocity components:

$$u_f = \begin{cases} u_D & \text{for } \tilde{u}_D \leqslant 0 \text{ or } \tilde{u}_D \geqslant 1 \\ \frac{1}{2}(u_D + u_A) & \text{for } k \leqslant \tilde{u}_D < 1 \\ (1 - \frac{\tilde{u}_D}{2k})u_D + \frac{\tilde{u}_D}{2k} u_A & \text{for } 0 < \tilde{u}_D < k \end{cases} \tag{23}$$

where $k = 0.1$. The subscripts D-donor and A-acceptor are determined by the direction of the flow, taking Figure 3 for example and considering Cell-P:

$$\dot{V}_f \geqslant 0 \Rightarrow D = P,\ A = \mathrm{NB} \tag{24}$$

The decision factor $\tilde{u}_D$ is defined as

$$\tilde{u}_D = 1 - \frac{u_A - u_D}{2(\partial u/\partial x_k)_D d_k} \tag{25}$$

where $(\partial u/\partial x_k)_D$ is calculated according to the general form of Gauss' theorem:

$$\left(\frac{\partial u}{\partial x_k}\right)_D \approx \frac{1}{V_D}\sum_{f=1}^{N} u_f (A_k)_f \tag{26}$$

4.2.2. Diffusion term. The diffusion term is discretized as follows:

$$\int_V \mu \frac{\partial}{\partial x_i}\frac{\partial u_j}{\partial x_i}\,\mathrm{d}V \approx \sum_{f=1}^{N} \mu_f \left(\frac{\partial u_j}{\partial x_i}\right)_f (A_i)_f \tag{27}$$

in which the term $(\partial u_j/\partial x_i)_f$ is calculated directly from

$$\frac{(u_j)_{\mathrm{NB}} - (u_j)_P}{|\mathbf{d}|} \tag{28}$$

where $\mathbf{d}$ is the distance vector pointing from P to NB.

The other part of the diffusion term is treated explicitly as a source term, since its value is quite small as compared with term (27) (see Reference [18]). It can be written as

$$\int_V \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\frac{\partial \mu}{\partial x_i}\,\mathrm{d}V \approx \sum_{f=1}^{N}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)_f \mu_f (A_i)_f \tag{29}$$

The face dynamic viscosity μ_f is calculated using linear interpolation, which possesses second-order accuracy:

$$\mu_f = \chi_P \mu_P + (1 - \chi_P)\mu_{\mathrm{NB}} \tag{30}$$

where χ_P is defined by

$$\chi_P = \frac{|\mathbf{fNB}|}{|\mathbf{d}|} \tag{31}$$

and **fNB** is the distance vector between the cell face-centre f and cell-centre NB (see Figure 3).

4.2.3. Temporal discretization. Using a first-order Euler implicit scheme, the discretized Navier–Stokes equations are

$$V_P \frac{(\rho u_j)_P^{t+\delta t}}{\delta t} + \sum_{f=1}^{N} \rho_f \dot{V}_f (u_j)_f^{t+\delta t} - \sum_{f=1}^{N} \mu_f (A_i)_f \left(\frac{\partial u_j}{\partial x_i}\right)_f^{t+\delta t} = S_{u_P} - V_P \left(\frac{\partial p}{\partial x_j}\right)_P^t \tag{32}$$

where

$$S_{u_P} = V_P \frac{(\rho u_j)_P^t}{\delta t} + g_j \rho_P V_P + \sum_{f=1}^{N} \left(\frac{\partial u_i^t}{\partial x_j} + \frac{\partial u_j^t}{\partial x_i}\right)_f \mu_f (A_i)_f \tag{33}$$

The Euler implicit scheme is satisfactory because the velocity varies much more slowly than the density over a time increment.

All the face velocities in the above equation are represented using the cell-centre velocity components of the cell and its nearest face neighbours. The resulting finite-volume-type formulations of the Navier–Stokes equations read:

$$\mathscr{A}_{u_P} (u_j)_P^{t+\delta t} + \sum_{\mathrm{nb}=1}^{N} \mathscr{A}_{u_{\mathrm{nb}}} (u_j)_{\mathrm{nb}}^{t+\delta t} = S_{u_P} - V_P \left(\frac{\partial p}{\partial x_j}\right)_P^t \tag{34}$$

where $\mathscr{A}_{u_P}$ is a coefficient related to the momentum equations of Cell-P, and $\mathscr{A}_{u_{\mathrm{nb}}}$ are the corresponding coefficients related to the momentum equations of the nearest face neighbours.

Equation (34) can be also written as

$$(u_j)_P^{t+\delta t} = \frac{H(u_j)_P}{\mathscr{A}_{u_P}} - \frac{V_P}{\mathscr{A}_{u_P}} \left(\frac{\partial p}{\partial x_j}\right)_P^t \tag{35}$$

where

$$H(u_j)_P = -\sum_{\mathrm{nb}=1}^{N} \mathscr{A}_{u_{\mathrm{nb}}} (u_j)_{\mathrm{nb}}^{t+\delta t} + S_{u_P} \tag{36}$$

This formulation is used in the pressure equation derived in the next section.

4.3. Pressure equation

The pressure equation is derived from the integral of the incompressible equation (8) over the control volume

$$\int_V \frac{\partial u_k}{\partial x_k}\,\mathrm{d}V = 0 \tag{37}$$

Gauss' theorem is applied to (37) to give

$$\sum_{f=1}^{N} (u_k A_k)_f = 0 \tag{38}$$

In the collocated variable arrangement, special care needs to be taken in the calculation of the face velocity $(u_j)_f$ to avoid decoupling of the velocity and pressure fields (see Reference [18]). Using the interpolation scheme proposed by Rhie and Chow [26], the face velocity is calculated from (35)

$$(u_j)_f = \left[\frac{H(u_j)}{\mathscr{A}_{u_P}}\right]_f - \left(\frac{V_P}{\mathscr{A}_{u_P}}\right)_f \left(\frac{\partial p}{\partial x_j}\right)_f \tag{39}$$

where the face values are calculated using linear interpolation similar to (30), that is

$$\left[\frac{H(u_j)}{\mathscr{A}_{u_P}}\right]_f = \chi_P \left[\frac{H(u_j)}{\mathscr{A}_{u_P}}\right]_P + (1-\chi_P)\left[\frac{H(u_j)}{\mathscr{A}_{u_P}}\right]_{\mathrm{NB}} \tag{40}$$

and

$$\left(\frac{V_P}{\mathscr{A}_{u_P}}\right)_f = \chi_P \left(\frac{V_P}{\mathscr{A}_{u_P}}\right)_P + (1-\chi_P)\left(\frac{V_P}{\mathscr{A}_{u_P}}\right)_{\mathrm{NB}} \tag{41}$$

The pressure gradient is calculated directly from the centre values of the cells that share the interface. Substitution of $(u_j)_f$ from (39) into (38) gives a finite-volume-type discretization of the pressure equation, which may be rearranged to give

$$\mathscr{A}_{p_P} p_P + \sum_{\mathrm{nb}=1}^{N} \mathscr{A}_{p_{\mathrm{nb}}} p_{\mathrm{nb}} = \sum_{f=1}^{N} (A_k)_f \left[\frac{H(u_k)}{\mathscr{A}_{u_P}}\right]_f \tag{42}$$

where $\mathscr{A}_{p_P}$ is the pressure equation coefficient of Cell-P, and $\mathscr{A}_{p_{\mathrm{nb}}}$ are the pressure equation coefficients of the nearest face neighbours.

Since the face velocity is given by (39), the volumetric flux can be calculated as

$$\dot{V}_f = (A_j)_f \left\{ \left[\frac{H(u_j)}{\mathscr{A}_{u_P}}\right]_f - \left(\frac{V_P}{\mathscr{A}_{u_P}}\right)_f \left(\frac{\partial p}{\partial x_j}\right)_f \right\} \tag{43}$$

This concludes the finite-volume-type discretization of the governing equations for volume-fraction values, velocity and pressure fields. The solution algorithm for the velocity and pressure fields used herein is the PISO scheme proposed by Issa [27].

4.4. Boundary conditions

Solid wall boundary conditions are specified as fixed value (Dirichlet) or fixed gradient (Neumann), and the required boundary values or gradients are used directly in the calculation of face gradients and face values. For example, consider the term in (39):

$$\left(\frac{V_P}{\mathscr{A}_{uP}}\right)_f \left(\frac{\partial p}{\partial x_j}\right)_f \tag{44}$$

If fixed value boundary conditions are specified (i.e. p_b is known), then the pressure gradient should be calculated from

$$\left(\frac{\partial p}{\partial x_j}\right)_f = \frac{(p_b - p_P)}{|\mathbf{d}_n|} \tag{45}$$

The term $-p_P/|\mathbf{d}_n|$ is incorporated into the central coefficient $\mathscr{A}_{pP}$, and $p_b/|\mathbf{d}_n|$ is included in the source term.

The present study treats the whole gas–liquid domain as a single fluid system with variable density and viscosity. Therefore, there is no need to specify kinematic and dynamic free surface boundary conditions at the gas–liquid interface [16]. Zero F gradient conditions are applied to the free surface and contact solid wall, which are also used in the CICSAM scheme [16]. This also means that the contact angle between the free surface and solid wall is not considered in the present model.

4.5. Solution procedure for flows with a gas–liquid interface

The simulation procedure for an unsteady flow with a gas–liquid interface can be summarized as follows:

1. Initialize all variables.
2. Obtain the previous time step's volume-fraction values.
3. Calculate new density and viscosity values from the updated volume-fraction values.
4. Use the PISO scheme [27] to solve the momentum and continuity equations and obtain updated velocity and pressure fields.
5. Solve the volume-fraction transport equation (19).
6. March to the next time step's calculation.

5. FINITE-VOLUME DISCRETIZATION ON A QUADTREE GRID

The discretization of the governing equations on a quadtree grid is similar to that on a uniform grid, except interpolations between different size cells are required when calculating cell face-centre values at the interfaces.

5.1. Interpolation on the quadtree grid

On a regularized quadtree grid, the subdivision level of the neighbouring cell can be one level higher, the same level or one level lower than the cell considered (see Figure 4). If the subdivision level of the neighbouring cell is one level higher than the cell considered, it

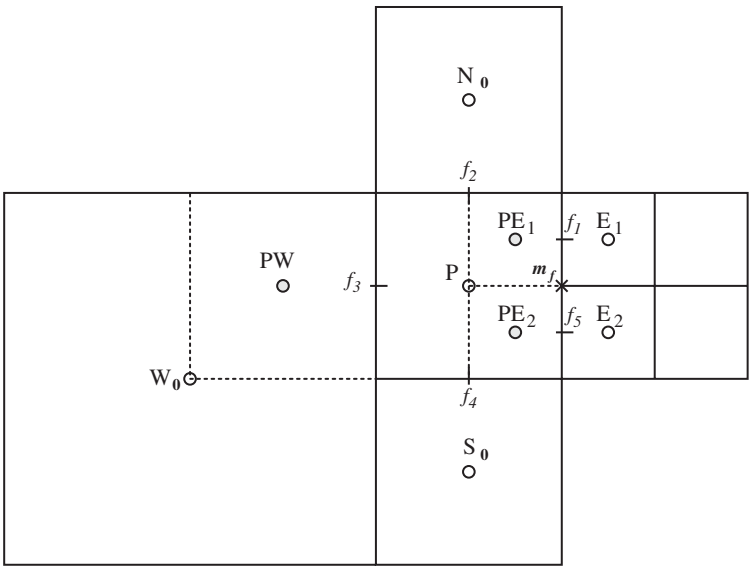

Figure 4. Schematic of the quadtree grid neighbouring cells.

produces a hanging node where the cell corner of the smaller cell lies at the face-centre of the larger cell. In such cases, we consider the larger cell to have two face neighbours when summing the flux contribution. For example, in Figure 4, Cell-P has five face neighbours (Cells-E_1, E_2, N_0, W_0 and S_0). Linear interpolation is used to estimate cell face-centre values on a regularized quadtree grid. The interpolations are categorized as follows:

1. If the neighbour cell is one level higher than the cell considered: the value at Point-m_f (Figure 4) is calculated from

$$m_f = \tfrac{1}{3} P + \tfrac{1}{3} E_1 + \tfrac{1}{3} E_2 \tag{46}$$

   where P, E_1, E_2 refer to cell-centre values of Cells-P, E_1 and E_2. The fictitious cell-centre PE_1 value is then calculated from

$$PE_1 = \tfrac{2}{3} P + \tfrac{2}{3} E_1 - \tfrac{1}{3} E_2 \tag{47}$$

   Cell-PE_1 is a fictitious cell having the same size as Cell-E_1, and its face-centre value is estimated from

$$f_1 = \tfrac{1}{2} (PE_1 + E_1) \tag{48}$$

   A similar procedure is applied to calculate the PE_2 value and the face-centre value at Face-5.
2. If the neighbour cell is of the same subdivision level as the cell considered, the face-centre value of Face-2 is

$$f_2 = \tfrac{1}{2} (P + N_0) \tag{49}$$

   where N_0 refers to the north neighbour value.

3. If the neighbour cell is one level lower than the cell considered, the PW value is approximated by

$$PW = \tfrac{2}{3} P + \tfrac{2}{3} W_0 - \tfrac{1}{3} S_0 \tag{50}$$

where PW, P, W_0 and S_0 refer to values at the cell-centres of Cells-PW, P, W_0 and S_0 respectively. Cell-PW is a fictitious cell having the same size as Cell-P. The face-centre value of Face-3 is then calculated using the central differencing scheme

$$f_3 = \tfrac{1}{2} (PW + P) \tag{51}$$

The interpolation methodology is different to that used by Greaves and Borthwick [20] and Rogers *et al.* [21], and has the advantage that it uses nearest neighbour values only. This simplifies the various cases that may arise on the quadtree grid, and avoids the smoothing effect that otherwise would occur by including contributions from farther neighbours. Moreover, it guarantees volume flux conservation at cell face-centre locations.

5.2. Interpolation of cell face-centre coefficients on the quadtree grid

Face-centre values of $H(u_j)$ and $\mathscr{A}_{u_P}$ are needed in Equations (39). It is straightforward to calculate these values using a central-differencing scheme on a uniform grid and for the same cell size case (Section 5.1 item 2) on the quadtree grid. However, if the interpolation schemes (item 1 and item 3 of Section 5.1) are applied to the calculation of the face-centre $H(u_j)$ and $\mathscr{A}_{u_P}$ directly on a quadtree grid, the results are incorrect. To explain this, a more detailed discussion is necessary.

The coefficient $\mathscr{A}_{u_P}$ contains temporal, advection and viscous terms, and is written

$$\mathscr{A}_{u_P} = \frac{(\rho V)_P}{\delta t} + \rho_f u_i A_i + \mu \frac{\partial}{\partial x_i} A_i \tag{52}$$

Consider the first term of $\mathscr{A}_{u_P}$, which is formed from the fluid density ρ, cell volume V_P, and time step δt. The time step δt is a constant throughout the computational domain. The density values ρ change from cell to cell when the volume-fraction values F are different. However, according to the grid adaptation criterion used herein, two adjacent cells with different subdivision levels must have the same F values (either $F = 0$ or 1). Therefore, two adjacent cells with different subdivision levels must have the same density values. The only variable left is the cell volume V_P. It is obvious that the cell volume values do not vary linearly across the cells when adjacent cells have different subdivision levels. Hence, it is clear that the $\mathscr{A}_{u_P}$ values at the cell face-centres cannot be obtained using a linear interpolation method when the adjacent cells have different subdivision levels.

Close examination of (39) reveals that the value needed at the cell face-centre location is $V/\mathscr{A}_{u_P}$. The term $V/\mathscr{A}_{u_P}$ has units (m^3 s/kg). Given that the time step (s) is the same throughout the computational domain and the remaining units of $V/\mathscr{A}_{u_P}$ (m^3/kg) are the same as those of $1/\rho$, it is acceptable to apply linear interpolation when calculating the face-centre coefficient $V/\mathscr{A}_{u_P}$ in situations where cell sizes are different.

The same analysis is applied to the coefficient $[H(u_j)/\mathscr{A}_{u_P}]_f$. It is incorrect to calculate $[H(u_j)]_f$ and $(\mathscr{A}_{u_P})_f$ separately using linear interpolation. Rather, the coefficient needed at the cell face-centre is obtained using $[H(u_j)/\mathscr{A}_{u_P}]_f$. Its units are (m/s), and correspond to a

speed of propagation. The linear interpolation scheme may therefore be used to evaluate the coefficient $[H(u_j)/\mathscr{A}_{u_P}]_f$ even when adjacent cell sizes are different.

In summary, only the values of the combinations $(V/\mathscr{A}_{u_P})_f$ and $[H(u_j)/\mathscr{A}_{u_P}]_f$ can be calculated using the linear interpolation schemes mentioned in Section 5.1 when the cell sizes are different.

A final concern is about the calculation of the volume-fraction function. In the CICSAM scheme, a parameter β_f is used in the calculation of the cell face volume-fraction F (see Equations (16) and (17)). In the present model, any pair of adjacent cells with different subdivision levels must have the same F values (either $F=0$ or 1), and so $F_P = F_{\mathrm{NB}}$ and β_f equals 0 or 1 according to the flux direction and the free surface orientation. Hence, $F_f = F_P = F_{\mathrm{NB}}$ and there is no difficulty in calculating the values of β_f on a quadtree grid even when neighbour cell sizes are different.

5.3. *Sparse matrix solver*

On a dynamically adaptive quadtree grid, the cell numbering system is not ordered, for convenient identification of neighbour cells. Moreover, the number of neighbour cells varies with the local grid configuration. Hence, the set of simultaneous linear equations forms a sparse matrix, which is solved using the 'Y12M' direct solver available from *netlib*, http://www.netlib.org.

The matrix of the pressure Poisson equation is indefinite, and so the solution sets differ by a constant. To demonstrate this, the pressure Poisson equation is rewritten in matrix form

$$\mathbf{A}\mathbf{x} = \mathbf{B} \tag{53}$$

where $\mathbf{A}$ is the coefficient matrix, $\mathbf{x}$ is a column vector of unknown variables and $\mathbf{B}$ is a column vector of source terms. The elements of a well-organized pressure Poisson equation matrix satisfy

$$\mathscr{A}_{p_P} + \sum_{\mathrm{nb}=1}^{N} \mathscr{A}_{p_{\mathrm{nb}}} = 0 \tag{54}$$

where $\mathscr{A}_{p_P}$ is the pressure coefficient of Cell-P, and $\mathscr{A}_{p_{\mathrm{nb}}}$ are the pressure coefficients of the corresponding neighbours. If $\mathbf{x}$ is the solution set, (53) can be written as

$$\mathscr{A}_{p_P} x_P + \sum_{\mathrm{nb}=1}^{N} \mathscr{A}_{p_{\mathrm{nb}}} x_{\mathrm{nb}} = B_p \tag{55}$$

If we now use (54) and replace the solution by $(\mathbf{x} - x_0)$, where x_0 is a non-zero arbitrary constant, then (55) has the form

$$\begin{aligned}
&\mathscr{A}_{p_P}(x_P - x_0) + \sum_{\mathrm{nb}=1}^{N} \mathscr{A}_{p_{\mathrm{nb}}}(x_{\mathrm{nb}} - x_0) \\
&= \left(\mathscr{A}_{p_P} x_P + \sum_{\mathrm{nb}=1}^{N} \mathscr{A}_{p_{\mathrm{nb}}} x_{\mathrm{nb}}\right) - \underbrace{\left(\mathscr{A}_{p_P} + \sum_{\mathrm{nb}=1}^{N} \mathscr{A}_{p_{\mathrm{nb}}}\right)}_{0} x_0 \\
&= \mathscr{A}_{p_P} x_P + \sum_{\mathrm{nb}=1}^{N} \mathscr{A}_{p_{\mathrm{nb}}} x_{\mathrm{nb}} \\
&= B_P
\end{aligned} \tag{56}$$

That is, if $\mathbf{x}$ is a solution of (53), $(\mathbf{x} - x_0)$ is also a solution, where x_0 is a non-zero constant. Although the pressure Poisson equation has an infinite number of solutions, numerical difficulties do not arise because pressure differences (i.e. gradients) are used in the momentum equations, not absolute pressure values.

Another issue concerns the necessary condition for the pressure Poisson equation to have non-trivial solutions. For an arbitrary vector $\mathbf{z}$, (53) implies

$$\mathbf{z}^{\mathrm{T}}\mathbf{A}\mathbf{x} = \mathbf{z}^{\mathrm{T}}\mathbf{B} \tag{57}$$

However, from (54), a non-zero vector $\mathbf{z}$ exists such that $\mathbf{z}^{\mathrm{T}}\mathbf{A} = 0$, and so we require $\mathbf{z}^{\mathrm{T}}\mathbf{B} = 0$ for (53) to have non-trivial solutions. From the pressure Poisson equation, the value $(A_k)_f$ $[H(u_k)/\mathscr{A}_{u_P}]_f$ calculated at the east face of Cell-P cancels out the corresponding value at the west face of Cell-NB (see Figure 3), and so the sum of the entire right-hand side $\mathbf{z}^{\mathrm{T}}\mathbf{B}$ equals zero. This therefore ensures that the discretized pressure Poisson equation has non-trivial solutions.

6. CASE STUDIES

Simulations are presented of the advection of regions of dyed fluid in unidirectional and rotating flow fields to validate the volume-fraction transport solver on dynamically adaptive quadtree grids. Results are also given for two-dimensional viscous free surface sloshing of a liquid in a rectangular tank using uniform and dynamically adapted quadtree grids.

6.1. Movement of an initially square region of dyed liquid in unidirectional flow

We first consider advection of a square patch of dyed fluid by unidirectional flow in a $1\,\mathrm{m} \times 1\,\mathrm{m}$ square domain. At time $t = 0$ s, the dye occupies a square region located at $(0.1250 \leqslant x \leqslant 0.21875\,\mathrm{m},\ 0.6875 \leqslant y \leqslant 0.78125\,\mathrm{m})$, in which $F = 1$. The remainder of the fluid domain is undyed with $F = 0$. The quadtree grid has maximum and minimum subdivision levels of 6 and 1, respectively. Initially, all cells on the quadtree grid within the region $(0.0 \leqslant x \leqslant 0.3\,\mathrm{m},\ 0.6 \leqslant y \leqslant 0.9\,\mathrm{m})$ are resolved to the finest level to account for the square patch of dye. Therefore, the total number of cells (including all leaf and parent cells) at $t = 0$ s is 1505. After coalescence, the number of the leaf cells at $t = 0$s is 190. Grid regeneration occurs every 9 time steps. Two cases are considered. In one, a uniform flow field is applied with velocity components $u = 0.02$ m/s and $v = 0$ m/s. The time step $\mathrm{d}t$ is 0.1 s, and the Courant number $(Cr) = \max\{|u|\,\mathrm{d}t/\mathrm{d}x, |v|\,\mathrm{d}t/\mathrm{d}y\} = 0.128$. Figure 5 shows the concentration contours superimposed on the quadtree grid at times $t = 0$ and 20 s, the latter obtained by solving the volume-fraction transport equation on the dynamically adaptive quadtree grid. It is clear that the region of dyed liquid retains almost its exact shape in pure translation. At $t = 20$ s the number of the leaf cells is 199. CPU times on an AMD Duron 1.2 GHz PC for the quadtree grid and for an equivalent (4096 cell) uniform grid are 0.52 and 1.13 s, respectively. As would be expected, the CPU time for the simulation on the dynamically adaptive quadtree grid system is much less than that on the uniform grid because far fewer cells are utilized. In the second case, a uniform flow with velocity components, $u = 0.02$ m/s and $v = -0.02$ m/s, is considered. The time step $\mathrm{d}t$ is 0.01 s, corresponding to a Courant number $Cr = 0.0128$. Figure 6 illustrates the advection of the dye as it translates diagonally across the square

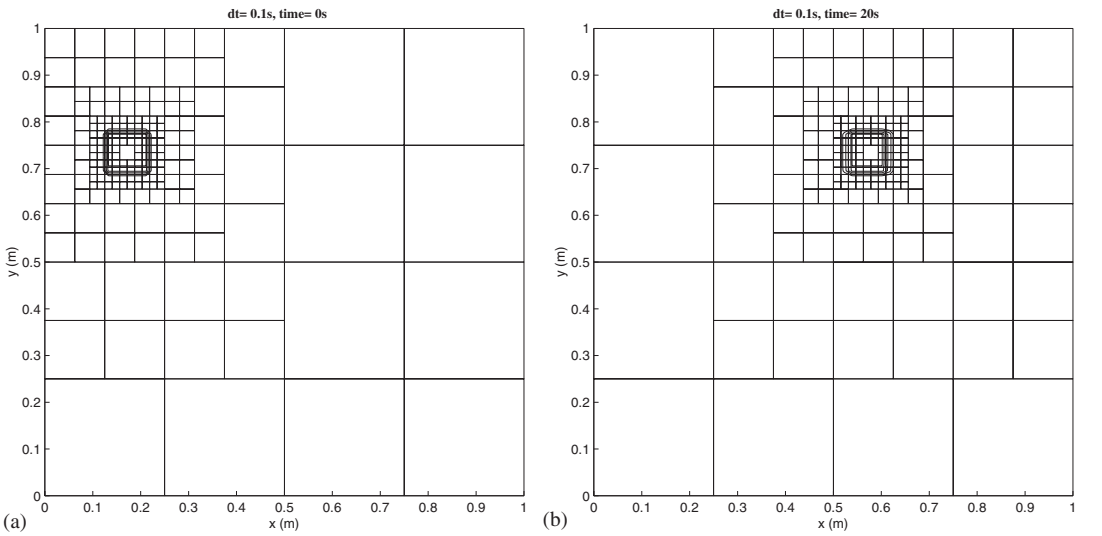

Figure 5. Movement of a region of dyed liquid in x-direction flow on a dynamically adaptive quadtree grid (Contours plotted at $F = 0.1$, 0.3, 0.5, 0.7, 0.9 and 0.99).

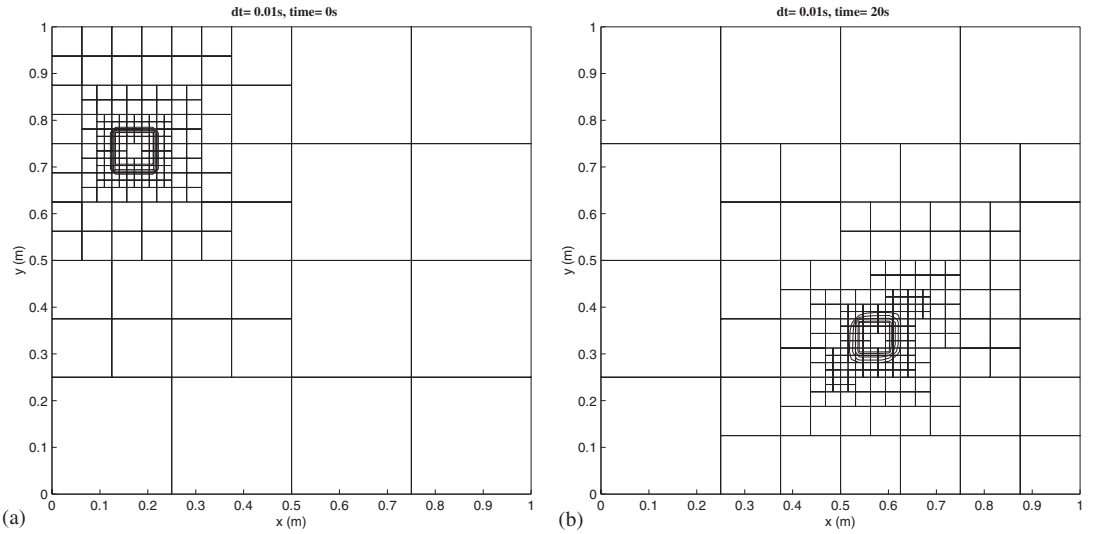

Figure 6. Diagonal movement of a region of dyed liquid in unidirectional flow on a dynamically adaptive quadtree grid (Contours plotted at $F = 0.1$, 0.3, 0.5, 0.7, 0.9 0.99).

domain. Overall, the movement of the dye patch is as expected, even though slight smearing of the corners of the patch is discernible. At $t = 20$ s the adapted quadtree grid contains 244 leaf cells. CPU times on an AMD Duron 1.2 GHz PC for the quadtree grid and for an equivalent (4096 cell) uniform grid are 3.63 and 10.46 s, respectively. This again confirms the improved computational efficiency of the model based on dynamically adaptive quadtree grids.

In both cases, the dye patch advects in exact accordance with the underlying flow kinematics, and retains an almost square shape, provided the Courant number is small. The dynamically adaptive quadtree grid follows the dye motion closely, with each grid being generated precisely according to the volume-fraction values. Finest mesh enrichment occurs at the dye front, while cell coalescence takes place in regions where the dye is no longer present.

6.2. *Movement of an initially rectangular region of dyed fluid in a rotating flow field*

We now consider rotation of a rectangular region of dyed fluid by a forced vortex. The computational domain is the same as above, except that the unidirectional flow is replaced by a circular flow with velocity components defined as $u = -(y - y_0)/25\text{m/s}$ and $v = (x - x_0)/25\text{m/s}$, where $(x_0, y_0) = (0.5, 0.5)\,\text{m}$. Thus, the angular velocity is $0.04\,\text{rad/s}$. In this test case, the dye occupies a $0.64\,\text{m} \times 0.32\,\text{m}$ area and is initially located at the centre of the computational domain. The boundary of the dyed liquid is represented by the $F = 0.5$-contour. The maximum subdivision level is 6 and the minimum subdivision level is 1. The time step $\mathrm{d}t$ is 0.1 s. The dynamically adaptive quadtree grid is regenerated every 9 time steps. The total number of cells generated initially is 2965, of which 808 are leaf cells.

Figure 7 shows the numerical predictions at $t = 0$, 19.63, 39.27 and 78.54 s, which correspond to anticlockwise angular rotations of the rectangle of 0, 45, 90 and 180°, and total leaf cell numbers of 808, 1090, 1024 and 1030, respectively. The CPU times required to rotate the dye rectangle by 180° are 6.10 and 4.56 s on the dynamically adaptive quadtree grid and on an equivalent 64×64 uniform grid, respectively. The dynamically adaptive quadtree grid is less efficient because the rectangular dye patch occupies a significant proportion of the computational domain, almost all of which is therefore of interest. Moreover, the ratio of the number of the full domain maximum subdivision leaf cells (uniform division) divided by the number of the exact leaf cells in the initial adaptive quadtree grid is $4096/808 \approx 5.07$. This low ratio implies that it is likely to be less efficient to solve the problem on a dynamically adaptive quadtree grid. Overall, the numerical predictions demonstrate that the dye patch advects correctly according to the flow kinematics. Slight smearing of the boundary contour would of course be reduced by using higher grid resolution and a smaller time step.

From the foregoing examples, it may be concluded that the finite-volume-type volume-fraction transport equation solver has been correctly solved on the dynamically adaptive quadtree grid.

6.3. *Two-dimensional viscous sloshing in a rectangular tank*

Two-dimensional sloshing of a viscous liquid in a rectangular tank is simulated to validate the viscous free surface flow solver on dynamically adaptive quadtree grids. The test parameters are as follows: tank width 4 m, tank height 4 m, mean depth of liquid $d = 2\,\text{m}$. The Reynolds number is defined as $Re = d\sqrt{gd}/\nu$, where g is the acceleration due to gravity and ν is the fluid kinematic viscosity. The non-dimensional time τ is defined as $\tau = t\sqrt{g/d}$ and the non-dimensional wave elevation ζ is defined as $\zeta = \eta/a$, where η is the vertical wave elevation measured from the liquid still water level, and a is the amplitude of the initial *cosine* wave profile. The gravity wave motions start from rest, with a hydrostatic pressure field and the velocity components set to zero everywhere. An initial quadtree grid is generated according to the volume-fraction F values of the *cosine* wave profile.

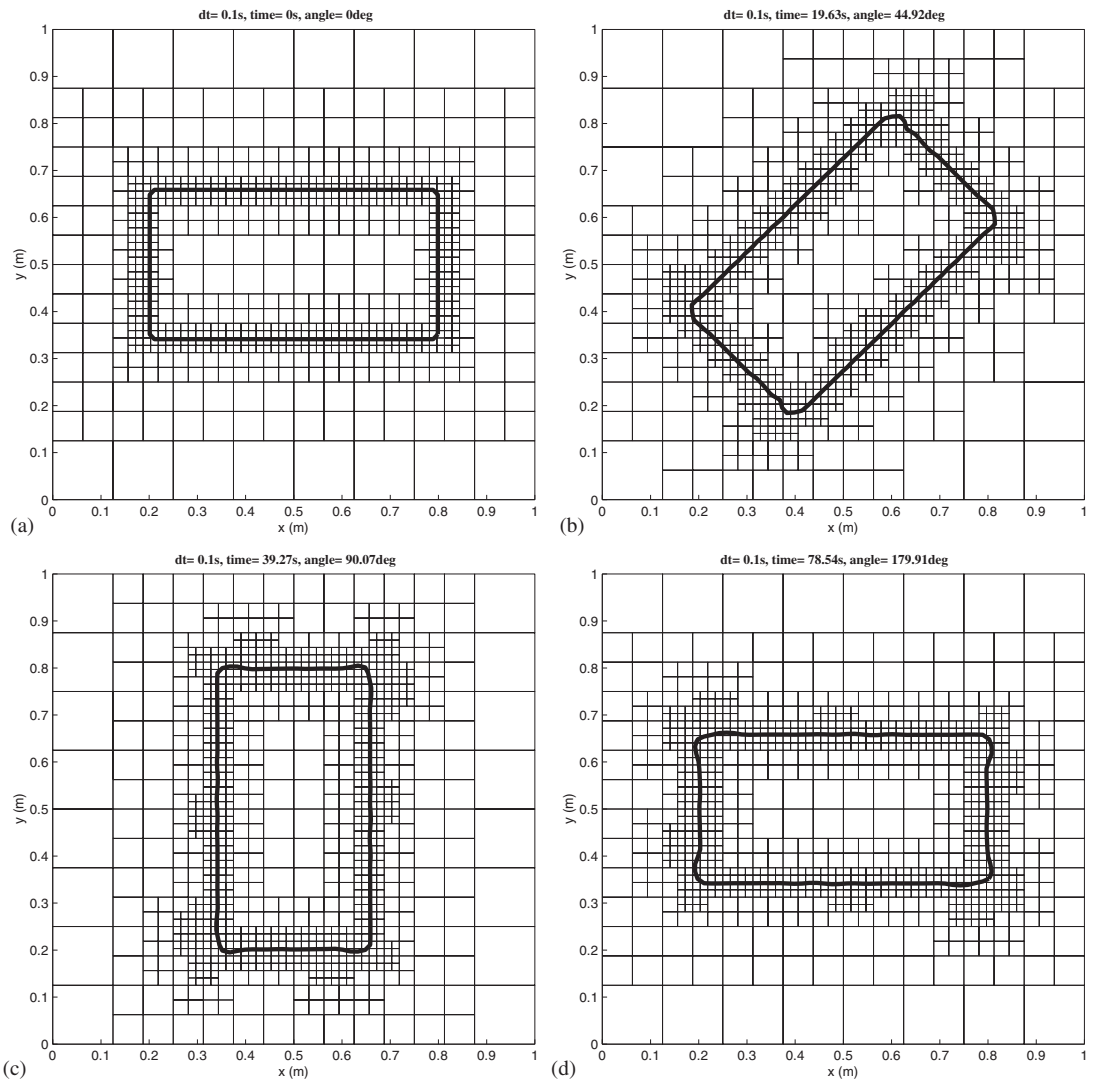

Figure 7. Rotation of a rectangular region of dyed liquid in circular flows on a dynamically adaptive quadtree grid, $\mathrm{d}t = 0.1$ s (Contours plotted at $F = 0.5$).

Figure 8 depicts the time-dependent behaviour of the wave profile for an initially large amplitude wave where $a/d = 0.1$ and $Re = 20$. The dynamic quadtree grid has maximum-6 and minimum-1 subdivision levels. Layers of 5 cells thick at maximum subdivision level are created either side of the free surface boundary. The total number of cells generated at $\tau = 0$ is 1829, of which 1108 are leaf cells. The time step $\mathrm{d}t$ is 1×10^{-4} s. The simulation is performed on a 128-Intel Pentium III processor (1.26 GHz) cluster computer system at the Oxford Supercomputing Centre. From Figure 8 it may be seen that the quadtree grid adapts properly to the motion of the free surface.

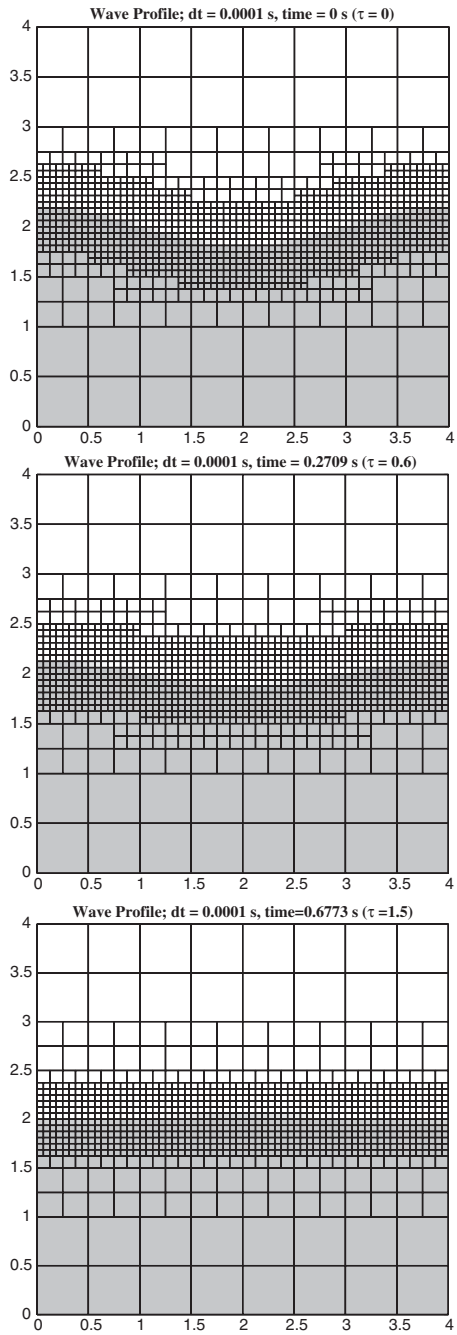

Figure 8. History of the wave profile on a Max.6 Min.1 quadtree grid.

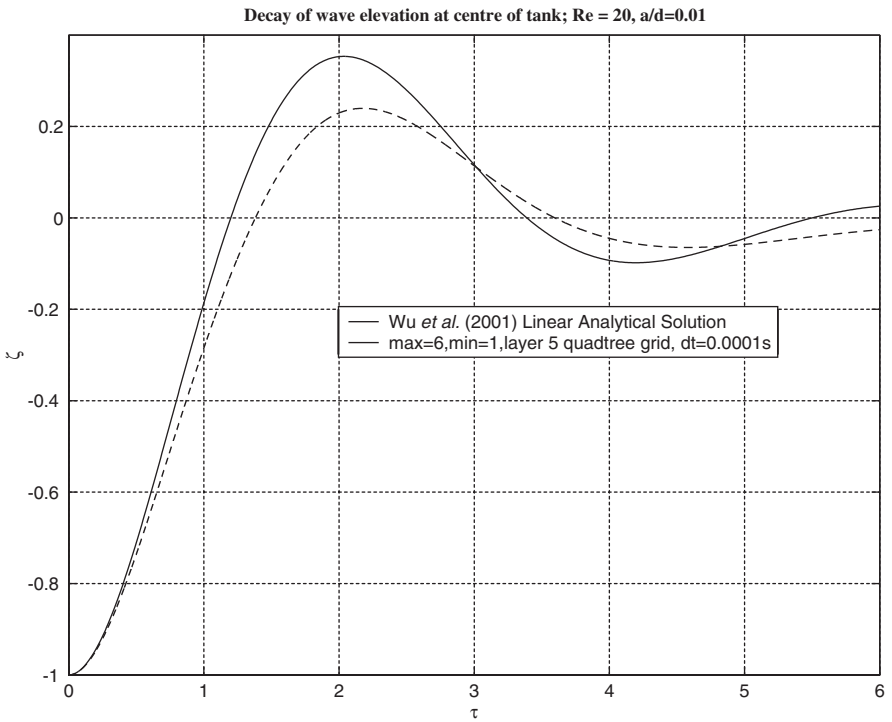

Figure 9. Comparison of the calculated results at the centre of the tank on a dynamically adaptive quadtree grid, $Re = 20$.

Small amplitude viscous sloshing in a rectangular tank has been studied by Wu *et al.* [28] who derived analytical solutions of the linearized Navier–Stokes equations. Wu *et al.* [28] assume that disturbances to the fluid are small and that the non-linear convection terms in the Navier–Stokes equations may be neglected (see Reference [29] for a discussion of the limitations introduced into the analytical model by these assumptions). To compare numerical predictions with the linear analytical solution of Wu *et al.* [28], a case involving small amplitude sloshing where the initial amplitude ratio is $a/d = 0.01$ has been simulated. Again the quadtree grids have maximum level 6, minimum level 1, and five layers of finest resolution cells either side of the liquid-gas interface. At $\tau = 0$, the quadtree grid contains 1365 cells in total, of which 928 are leaf. Free-slip boundary conditions are specified at the solid walls of the tank, in accordance with Wu *et al.* [28]. The time step dt is 1×10^{-4} s. Figure 9 plots the free surface elevation histories at the centre of the tank obtained using the numerical model and the analytical solution of Wu *et al.* [28]. It should be noted that those results obtained by Wu *et al.* [28] are derived from linearized model, which are not exact solutions. Therefore, the discrepancy in result between the present model and that of Wu *et al.* [28] is expected. Both sets of results have similar profiles, although the numerically predicted free surface oscillations are clearly more damped than the corresponding linear analytical solution of Wu *et al.* [28]. The CPU requirement is 0.09 s per time step. Figure 10 presents the

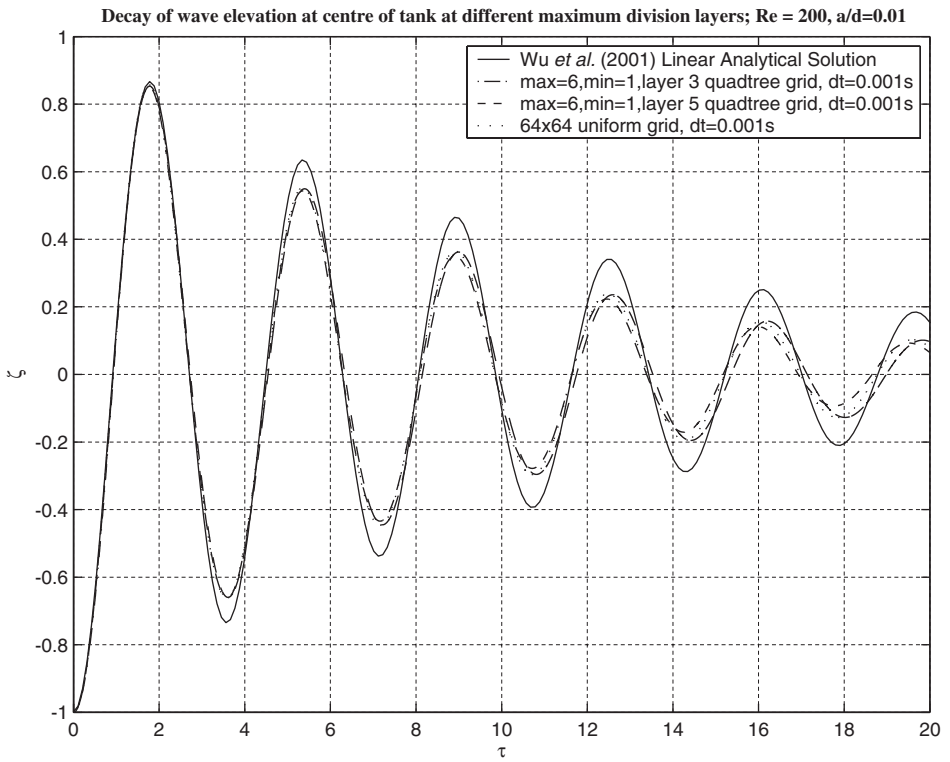

Figure 10. Comparison of the calculated results at the centre of the tank at different layers of maximum subdivision levels on a dynamically adaptive quadtree grid, $Re = 200$.

predicted damped free surface oscillations at the centre of the tank for $Re = 200$. Four time histories are presented: numerical predictions based on quadtree grids with 3 and 5 layers at finest resolution either side of the free surface, numerical prediction on an equivalent 64×64 uniform grid, and the analytical solution of Wu *et al.* [28]. The results obtained using the quadtree grids are very close to those computed on the equivalent uniform grid. Although the numerical predictions are slightly more damped than the analytical solution, there is excellent agreement in the phase of the decaying oscillations. In the computations, the time step is 1×10^{-3} s.

In order to show the effect of the ratio of the number of cells of an equivalent uniform grid to the number of leaf cells of the quadtree, the $Re = 200$ case has been simulated on a maximum-7, minimum-1, and 3 layers maximum subdivision level quadtree grid. The total number of cells generated at $\tau = 0$ is 3029, and the number of leaf cells is 1504. The ratio of the number of leaf cells on the equivalent 128×128 uniform grid to those on the quadtree is $16384/1504 \approx 10.89$. The CPU time for the calculations on the quadtree grid is 0.17 s per time step, whereas the same case run on a 128×128 uniform grid requires 0.30 s per time step. This indicates that the use of dynamically adaptive quadtree grids can lead to efficiency gains of more than a factor of two when compared with simulations on an equivalent uniform grid.

Table I. CPU time for $Re = 20$.

$Re = 20$	Unit: s (per time step)
64×64-Uniform grid	0.088
5-Layer max. subdivision level-6 quadtree grid	0.090

Table II. CPU time for $Re = 200$.

$Re = 200$	Unit: s (per time step)
64×64-Uniform grid	0.073
3-Layer max. subdivision level-6 quadtree grid	0.079
128×128-Uniform grid	0.301
3-Layer max. subdivision level-7 quadtree grid	0.170

Tables I and II summarize the CPU times recorded for the viscous sloshing simulations on uniform and dynamically adaptive quadtree grids.

7. CONCLUSIONS

A finite-volume-type VOF method has been implemented on a dynamically adaptive quadtree grid, using direct Gaussian elimination to solve the sparse matrix system of simultaneous linear equations. The model has been validated for pure advection of a patch of dyed fluid in unidirectional and rotating flow fields. Results obtained on quadtree grids that are adapted according to the volume-fraction function are in excellent agreement with simulations on a uniform grid of the same resolution as the finest level of the quadtree grid. It is found that the use of adaptive grids leads to improved computational efficiency. The numerical model has also been used to simulate the free surface sloshing of viscous liquid in a rectangular tank. The results show the exponential decay in oscillation amplitude in reasonable agreement with a linear analytical solution of the Navier–Stokes equations derived by Wu *et al.* [28]. Several points should be noted.

- Dynamically adaptive quadtree grids can achieve very fine grid resolution in zones where physical variables have high gradients or change abruptly, such as occur at the gas–liquid interface. In zones where the flow field is hardly varying, the saving in cells due to grid coarsening can make numerical models based on quadtree grids more efficient than those based on a uniform grid (of grid density equal to the maximum level of the quadtree grid).
- An implicit scheme can be less efficient to implement on a dynamically adaptive quadtree grid than on a uniform grid because a sparse matrix solver is needed.
- If the ratio of the number of leaf cells on the equivalent uniform grid to that on the quadtree is above about five, it is likely that computations will be more efficient on the dynamically adaptive quadtree grid than on the equivalent fine resolution uniform grid. Cases like a rising inviscid air bubble in water, solitary wave propagation and casting

processes, etc. may be modelled extremely efficiently on dynamically adaptive quadtree grids, because only localized regions of the computational domain (such as wave fronts) are of interest. However, for more dispersed multiphase flows (such as Rayleigh–Taylor instability [30]), the widespread non-zero and non-unity volume-fraction values lead to fine grid resolution over most of the computational domain. In such cases, there is no advantage to be gained by using dynamically adaptive grids.

- In a viscous flow, it is important to resolve accurately the velocity profile in the boundary layer. Hence, it is recommended that quadtree grid methods have a layer of cells adjacent to the boundary at maximum resolution. In a highly viscous flow, the presence of a thick boundary layer implies that a correspondingly thick layer of cells at finest resolution should be used in the quadtree grid.
- It should be noted that the present scheme is well suited for parallelization along the lines proposed by Bell *et al.* [31] because the quadtree grid data structure is simple to interrogate, information is stored at distributed nodes and parallel sparse matrix solvers are available.

ACKNOWLEDGEMENTS

This research has been supported by the Government of Taiwan, R.O.C. Substantial computational resources have been provided by the Oxford Supercomputing Centre (OSC), University of Oxford, U.K. The second author acknowledges support from the U.K. EPSRC under grant GR/N22595.

REFERENCES

1. 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.
2. Gáspár C, Józsa J. A coupled Lagrangian particle tracking and quadtree-based adaptive method with application to shear layer evolution. *Proceedings of the XXIV IAHR Congress*, Madrid, Spain, 1991.
3. De Zeeuw D, Powell KG. An adaptively refined Cartesian mesh solver for the Euler equations. *Journal of Computational Physics* 1993; **104**:55–68.
4. Wille SØ. A structured tri-tree search method for generation of optimal unstructured finite element grids in two and three dimensions. *International Journal for Numerical Methods in Fluids* 1992; **14**:861–881.
5. 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.
6. Hirt CW, Nichols BD. Volume of fluid (VOF) method for dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**:201–221.
7. 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.
8. Rudman M. Volume tracking methods for interfacial flow calculations. *International Journal for Numerical Methods in Fluids* 1997; **24**:671–691.
9. Harvie DJE, Fletcher DF. A new volume of fluid advection algorithm: the stream scheme. *Journal of Computational Physics* 2000; **162**:1–32.
10. Harvie DJE, Fletcher DF. A new volume of fluid advection algorithm: the defined donating region scheme. *International Journal for Numerical Methods in Fluids* 2001; **35**:151–172.
11. Rogers JCW, Szymczak WG. Computations of violent surface motions: comparisons with theory and experiment. *Philosophical Transactions of The Royal Society, Series A* 1997; **355**:649–663.
12. Muzaferija S, Perić M. *Computation of Free Surface Flows Using Interface-Tracking and Interface-capturing Methods; Nonlinear Water Wave Interaction*, Chapter 2. Computational Mechanics Publications: Southampton, U.K., 1998.
13. Lin P, Liu PLF. *Free Surface Tracking Methods and Their Applications to Wave Hydrodynamics; Advances in Coastal and Ocean Engineering*, vol. 5. World Scientific: Singapore, London, 1999.
14. Kothe DB, Rider WJ. Comments on modeling interfacial flows with volume-of-fluid methods. *Technical Report* LA-UR-94-3384, National Laboratory, Los Alamos, 1994.
15. Rider WJ, Kothe DB. Reconstructing volume tracking. *Journal of Computational Physics* 1998; **141**:112–152.

16. Ubbink O, Issa RI. A method for capturing sharp fluid interfaces on arbitrary meshes. *Journal of Computational Physics* 1999; **153**:26–50.
17. Sussman M, Almgren AS, Bell JB, Colella P, Howell LH, Welcome ML. An adaptive level set approach for incompressible two-phase flows. *Journal of Computational Physics* 1999; **148**:81–124.
18. Ferziger JH, Perić M. *Computational Methods for Fluid Dynamics* (2nd edn). Springer: Berlin, 1999.
19. Crowley WP. Numerical advection experiments. *Monthly Weather Review* 1968; **96**(1):1–11.
20. 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.
21. Rogers B, Fujihara M, Borthwick AGL. Adaptive Q-tree Godunov-type scheme for shallow water equations. *International Journal for Numerical Methods in Fluids* 2001; **35**:247–280.
22. Samet H. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley: Reading, MA, 1990.
23. Spalding DB. A method for computing steady and unsteady flows possessing discontinuities of density. *CHAM Report*, 1974; 910/2.
24. Richardson SM. *Fluid Mechanics*. Hemisphere Publishing Corporation: New York, 1989.
25. Jasak H. Error analysis and estimation for finite volume method with applications to fluid flows. *Ph.D. Thesis*, Department of Mechanical Engineering, Imperial College of Science, Technology & Medicine, University of London, U.K., 1996.
26. 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.
27. Issa RI. Solution of implicitly discretised fluid flow equations by operator-splitting. *Journal of Computational Physics* 1986; **62**:40–65.
28. Wu GX, Eatock Taylor R, Greaves DM. The effect of viscosity on the transient free surface waves in a two dimensional tank. *Journal of Engineering Mathematics* 2001; **40**(1):77–90.
29. Mei CC. *The Applied Dynamics of Ocean Surface Waves (Advanced series on Ocean Engineering; V.1)*. World Scientific: Singapore, 1989.
30. Černe G, Petelin S, Tiselj I. Coupling of the interface tracking and the two-fluid models for the simulation of incompressible two-phase flow. *Journal of Computational Physics* 2001; **171**:776–804.
31. Bell JB, Day MS, Almgren AS, Lijewski MJ, Rendleman CA. A parallel adaptive projection method for low Mach number flows. *International Journal for Numerical Methods in Fluids* 2002; **40**:209–216.

