INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2007; **53**:1777–1799
Published online 8 November 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1363

Shallow flow simulation on dynamically adaptive cut cell quadtree grids

Qiuhua Liang[‡, ‖], Jun Zang[*, †], Alistair G. L. Borthwick[§] and Paul H. Taylor[¶]

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

SUMMARY

A computationally efficient, high-resolution numerical model of shallow flow hydrodynamics is described, based on dynamically adaptive quadtree grids. The numerical model solves the two-dimensional non-linear shallow water equations by means of an explicit second-order MUSCL-Hancock Godunov-type finite volume scheme. Interface fluxes are evaluated using an HLLC approximate Riemann solver. Cartesian cut cells are used to improve the fit to curved boundaries. A ghost-cell immersed boundary method is used to update flow information in the smallest cut cells and overcome the time step restriction that would otherwise apply. The numerical model is validated through simulations of reflection of a surge wave at a wall, a low Froude number potential flow past a circular cylinder, and the shock-like interaction between a bore and a circular cylinder. The computational efficiency is shown to be greatly improved compared with solutions on a uniform structured grid implemented with cut cells. Copyright © 2006 John Wiley & Sons, Ltd.

Received 8 December 2005; Revised 21 July 2006; Accepted 6 August 2006

KEY WORDS: non-linear shallow water equations; quadtree; cut cell; Godunov method; approximate Riemann solver

1. INTRODUCTION

The non-linear shallow water equations (NSWEs) are commonly used to simulate free surface flows when vertical accelerations are small. Shallow flow simulations are of considerable value

*Correspondence to: Jun Zang, Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, U.K.
†E-mail: jun.zang@eng.ox.ac.uk
‡E-mail: qiuhua.liang@newcastle.ac.uk
§E-mail: alistair.borthwick@eng.ox.ac.uk
¶E-mail: paul.taylor@eng.ox.ac.uk
‖Now at School of Civil Engineering & Geosciences, University of Newcastle upon Tyne, Newcastle upon Tyne NE1 7RU, U.K.

Contract/grant sponsor: U.K. EPSRC; contract/grant number: GR/T07220/01

in environmental engineering. Applications include the prediction of fluvial hydrodynamics, lake circulation, flood inundation, estuarine dynamics, and nearshore processes. The accurate modelling of shallow flow interactions with structures is important in the design of nearshore wind turbine foundations, bridge piers, etc.

In combination with Godunov-type shock-capturing schemes, finite volume numerical solvers of the non-linear shallow water equations model steep-fronted shallow flows [1–13]. Godunov-type finite volume schemes are able to provide high-resolution upwinded numerical solutions for a wide variety of subcritical, supercritical, and transcritical flows. By incorporating a limiter, Godunov-type solvers can also be used to predict discontinuous flows, such as dam-breaks, in a *total-variation-diminishing* (TVD) way without causing spurious numerical oscillations.

Shallow free surface flows often occur in domains with complicated boundaries, and may involve shock-like bores, local shear, eddies, and wet–dry fronts. Boundary fitting is then of particular importance, and is normally dealt with in one of three ways dictated by the type of grid being used; boundary-fitted structured [1, 5, 14], boundary-fitted unstructured [2, 8, 15], and Cartesian [3, 7, 16]. The first type, the boundary-fitted structured grid, has the following advantages: it is relatively straightforward to impose proper boundary conditions; the solution is accurate near the boundary; and the conservation property is observed. However, the quality of the structured boundary-fitted grid generated depends on the shape of the boundary. In addition, the governing equations are more difficult to solve when re-written in a curvilinear coordinate system, and this may affect the stability and convergence properties of the solver [17]. The second type, the boundary-fitted unstructured grid, is well suited to dealing with awkward boundary geometries, particularly when implemented with finite volume or finite element schemes. In fact, boundary-fitted unstructured grids can be made to conform to nearly any desired geometry (see, e.g. Reference [15]). However, the grid generation process for this type of grid is not completely automatic. User intervention is often needed to produce a grid of satisfactory quality, in terms of adequate local resolution without too severe distortion.

The most popular type of unstructured grid consists of triangular elements in two dimensions, and is normally easier to generate than with quadrilateral elements. However, triangular unstructured grids generally result in lower numerical accuracy. For example, it is difficult to construct approximations that maintain accurate propagation of one-dimensional flow disturbances because a triangular grid cell does not have parallel faces.

In the third, structured, Cartesian-type grid, all cell faces are aligned parallel to local Cartesian coordinate axes. To approximate the boundary with piece-wise linear segments, input geometries (solid bodies) are cut out of the background grid. The connectivity between neighbouring cells is implicit, and so there is no requirement for grid generation in the conventional sense. The governing equations are solved directly in Cartesian coordinates. However, use of this scheme requires special care to avoid instability arising from small boundary cells while retaining high-order accuracy along the boundary. With a solver specified on a structured grid it is difficult to perform dynamic grid adaptation.

In recent years, attention has been given to the development of shallow flow models based on dynamically adaptive quadtree grids [7, 18]. A quadtree grid is a hierarchical form of Cartesian grid with no rigid cell connectivity. Neighbour information is stored in a data tree [19]. Since a quadtree grid consists of axis-oriented Cartesian cells, the benefits of accurate spatial derivatives and high-order cancellation of discretization errors may be exploited. It is straightforward to add and remove grid cells according to user-specified criteria at intervals during the simulation, thus obtaining a locally high resolution, dynamically adaptive grid. This is ideal when simulating free

DOI: 10.1002/fld

surface flows containing zones of locally high hydrodynamic gradient (e.g. fronts, transitions, bores, and localised eddies). The quadtree grid can provide a high resolution approximation to complicated boundary geometry, but has the drawback that the approximation of any curved boundary is in the form of a staircase, no matter how fine the mesh is near the boundary. This can introduce additional numerical errors, which can appear as spurious local reflections and the release of spurious vorticity into the flow. Thus, the coupling of a quadtree gridding and boundary fitting scheme with a high resolution shallow water equation solver would make the numerical simulations more robust and represent a significant advance.

In this work, we present a numerical solver based on dynamically adaptive quadtree grids, implemented with a Cartesian cut cell technique for boundary fitting. Section 2 outlines the grid generator. Section 3 describes the Godunov-type finite volume solver of the non-linear shallow water equations in a cut cell quadtree context. Section 4 discusses the model validation tests. Brief conclusions are drawn in Section 5.

2. GENERATION OF QUADTREE GRID AND CARTESIAN CUT CELLS

A quadtree grid consists of a set of non-uniform Cartesian cells obtained from automatic recursive subdivision of a unit square called the root cell. First, the external boundary geometry of the flow domain is rescaled to fit within the unit square, and the external and any internal boundaries are discretized into sets of seeding points. Detailed descriptions of the quadtree grid generation procedure are given by Rogers *et al.* [7] and Liang *et al*. [18]. In cases involving islands or surface-piercing solid bodies, we require that each set of body boundary seeding points is ordered anticlockwise to form a closed loop surrounding the internal body or bodies. This facilitates Cartesian cut cell generation around the body contour. After dividing the root cell into four equal quadrant cells, each new cell is checked in turn and recursively subdivided if it contains two or more seeding points and its subdivision level is less than the maximum specified. Finally, grid regularization is performed to ensure that no cell has a side length more than twice the size of its neighbours (including corner and face neighbours).

The Cartesian cut cell technique [3, 20] is incorporated with the quadtree grid to provide boundary fitting, and avoid the artificial boundary staircase that would otherwise occur. Solid bodies are 'cut' out of the background Cartesian quadtree grid so that the boundaries are approximated by straight line segments. In practice, the cut cells are generated before the quadtree grid is regularized [7, 18]. Boundary cells (containing seeding points) and their direct corner and face neighbours are defined at the highest subdivision to give the most accurate description of the boundary. Cut cell generation essentially reduces to finding the intersections between line segments formed by seeding points and cell interfaces of the background mesh. The first step is to locate the grid cells where the required seeding points are positioned. The cell (i) in which the first seeding point located is found by searching all the leaf cells in the background quadtree grid. After the first seeding point has been located, the search procedure can be accelerated by firstly examining cell i and its neighbouring cells. If two neighbouring seeding points are located at different leaf cells, there are intersections between the line segment connecting the two seeding points and cell interfaces. The coordinates of the intersections are determined by linear interpolation. Figure 1 illustrates the procedure for determining the intersections, where the seeding points $(a, b, c, \ldots)$ are represented by solid black dots. After locating the start seed a and end seed b, the location of the intersection A between line segment ab and the cell interface is found by linear interpolation; in Figure 1 the intersection

DOI: 10.1002/fld

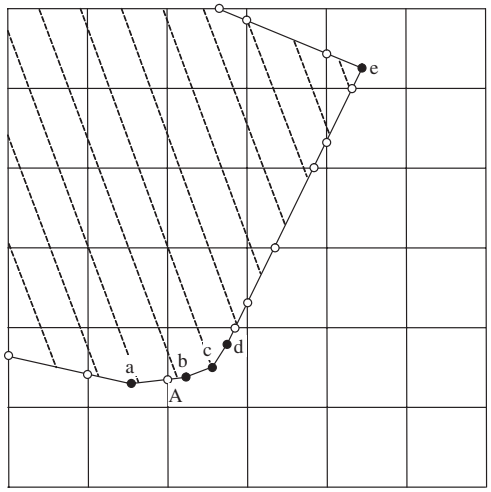

Figure 1. Intersection points between line segment and cell interfaces: solid black dots = seeding points; white circles = intersections.

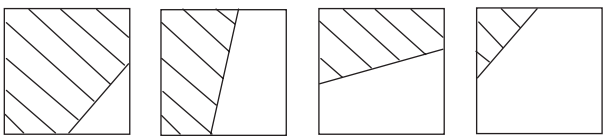

Figure 2. Grid generation: sub-types of cut cell for case with $\varphi \in (0°, 90°)$.

is denoted by a white circle. Then the end seed b becomes the start seed of the next line segment bc. Because both seeding points (b and c) lie within a single cell, there is no intersection, and so the intersection search moves to the next line segment cd, and this is repeated until all the seeding points have been considered. After all the intersections have been identified, they are joined to form cut cells, whose indices are then flagged. Herein a cut cell refers to the polygon formed by the fluid part of the original cell and the cut edge (line segment connecting two intersections). A cut cell can have 3–5 faces. According to the angle (φ) formed by the cut edge and the x-axis: each cut cell can be categorized into one of four types: $\varphi \in (0°, 90°)$; $\varphi \in (90°, 180°)$; $\varphi \in (180°, 270°)$; and $\phi \in (270°, 360°)$. Each type has four sub-types. The four sub-types for the case with $\varphi \in (0°, 90°)$ are illustrated in Figure 2; the sub-types for other cases can be obtained by rotation.

After all the cut cells have been produced, a boundary identification technique is performed to identify which cells contain fluid, are solid or are on the interface as cut cells. Each leaf cell, apart from cut cells, in the background quadtree grid is checked in turn by drawing two lines from the cell centre to the faces of the initial unit square in positive and negative coordinate directions, and which can be performed in either x or y-direction. The number of intersections of each line and the boundaries is evaluated. We assume the computational domain consists of the unit square with solid bodies located inside. If the numbers of intersection points for both lines are even, the cell

DOI: 10.1002/fld

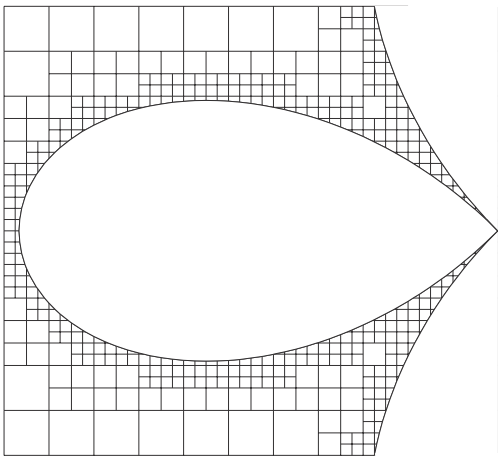

Figure 3. Cut cell quadtree grid generated about a petal.

is flagged as a fluid cell; otherwise it is a solid cell. The boundary perimeter, which is described by seeding points, is converted into continuous form by linear interpolation. After this procedure, all the leaf cells have been flagged and belong to one of three types: fluid cells, cut cells and solid cells. Grid regularization [7, 18] is then applied to all the fluid cells.

In order to evaluate the performance of the grid generator, a quadtree grid with cut cells is generated for a flower-shaped geometry. The boundary of the flower-shaped solid body is described by the following formula:

$$r = c\cos(2\theta) \tag{1}$$

where (r, θ) are the polar coordinates and c is set to $\frac{1}{3}$. The boundary curve is approximated by 800 seeding points. Figure 3 shows the resulting mesh in a quadrant for one petal of the flower. The grid has maximum and minimum subdivision levels of 7 and 5, respectively. A total of 3365 cells are generated, of which 2068 are undivided leaf cells. 521 cut cells are used to describe the curved boundary. The total CPU time for grid generation is 0.6 s on a Pentium 3 GHz (1G RAM) personal computer. Examples of quadtree grids incorporated with cut cells for boundary fitting are also presented by Causon *et al*. [3, 21].

This example demonstrates that the present quadtree and cut cell grid generator has the ability to provide locally high resolution where the geometry is most complicated. Cartesian cut cells essentially approximate curved boundaries in a line segment manner, thus giving a better boundary fit than the staircase approximation.

3. SHALLOW WATER EQUATION SOLVER ON CUT CELL QUADTREE GRIDS

3.1. Governing equations

The two-dimensional non-linear shallow water equations are derived by integrating the Reynolds equations over the flow depth while assuming that vertical accelerations are insignificant. In matrix

DOI: 10.1002/fld

form, the hyperbolic conservation law formed by the equation set can be written as

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} = \mathbf{s} \tag{2}$$

where t denotes time, x and y are Cartesian coordinates, and $\mathbf{u}$, $\mathbf{f}$, $\mathbf{g}$ and $\mathbf{s}$ are vectors representing conserved variables, fluxes in the x and y-directions, and source terms, respectively. Neglecting the viscous fluxes, surface and bed stresses and the Coriolis effects, the vectors can be written as

$$\mathbf{u} = \begin{bmatrix} \zeta \\ uh \\ vh \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} uh \\ u^2h + g(\zeta^2 + 2\zeta h_s)/2 \\ uvh \end{bmatrix}$$

$$\mathbf{g} = \begin{bmatrix} vh \\ uvh \\ v^2h + g(\zeta^2 + 2\zeta h_s)/2 \end{bmatrix} \quad \text{and} \quad \mathbf{s} = \begin{bmatrix} 0 \\ -g\zeta S_{ox} \\ -g\zeta S_{oy} \end{bmatrix}$$

Here, ζ is the water elevation above the still water level datum; the total depth h $(= h_s + \zeta)$ where h_s is the still water depth; u and v are depth-averaged velocity components in the two Cartesian directions; g is the acceleration due to gravity; S_{ox} $(= -\partial h_s/\partial x)$ and S_{oy} $(= -\partial h_s/\partial y)$ are bed slopes in the x and y-directions, respectively. The above non-linear shallow water equations are in deviatoric form, obtained using the generalized flux-source term balancing technique proposed by Rogers *et al*. [22], and are directly applicable to shallow flow hydrodynamics in domains with varying bed topography.

3.2. High-resolution finite volume method

Using the finite volume method, the shallow water equations (2) are solved in integral form

$$\frac{\partial}{\partial t}\int_\Omega \mathbf{u}\,\mathrm{d}\Omega + \int_\Omega \left(\frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y}\right)\mathrm{d}\Omega = \int_\Omega \mathbf{s}\,\mathrm{d}\Omega \tag{3}$$

in which Ω is the numerical domain. Applying Green's theorem to the second term of the above equation gives

$$\frac{\partial}{\partial t}\int_\Omega \mathbf{u}\,\mathrm{d}\Omega + \oint_S \mathbf{F}\,\mathrm{d}S = \int_\Omega \mathbf{s}\,\mathrm{d}\Omega \tag{4}$$

where S is the lateral boundary of the problem domain Ω, and $\mathbf{F}$ is the vector of fluxes passing through S, given by

$$\mathbf{F} = \mathbf{f}n_x + \mathbf{g}n_y \tag{5}$$

with n_x and n_y being the Cartesian components of the unit normal vector ($\mathbf{n}$) of S. Equation (4) is solved using a Godunov-type finite volume scheme on a cut cell quadtree grid. The two-step

DOI: 10.1002/fld

MUSCL-Hancock method is used to achieve overall second-order accuracy. For the predictor step, the discretized time-marching formula for Equation (4) is

$$(A\mathbf{u})_{i,j}^{n+1/2} = (A\mathbf{u})_{i,j}^{n} - \frac{\Delta t}{2}\left(\sum_{k=1}^{m} \mathbf{F}_k^n \cdot \mathbf{S}_k - (A\mathbf{s})_{i,j}^{n}\right) \tag{6}$$

where A is the area of cell (i, j), superscript n denotes the time level; Δt is the time step; $\mathbf{S}_k$ is the kth cell side vector defined as the cell side length multiplied by its outgoing unit normal vector ($\mathbf{n}$); m is the number of cell sides for an arbitrary grid cell where $m = 4$ for a flow cell and m ranges from 3 to 5 for a cut cell; and $\mathbf{F}_k^n$ is the flux vector through cell side k, evaluated based on the flow variables at time level n. The calculation of the interface fluxes is performed within each cell using interpolated values at the central extremities of the inner interfaces. There is no requirement for Riemann solutions in the predictor step.

During the corrector step, the interface fluxes are calculated over the full time increment, with the fluxes evaluated by solving a local Riemann problem based on data from the predictor step. The explicit updating formula in the corrector step reads

$$(A\mathbf{u})_{i,j}^{n+1} = (A\mathbf{u})_{i,j}^{n} - \Delta t\left(\sum_{k=1}^{m} \mathbf{F}_k^{n+1/2} \cdot \mathbf{S}_k - (A\mathbf{s})_{i,j}^{n+1/2}\right) \tag{7}$$

where $\mathbf{F}_k^{n+1/2}$ is the flux vector through cell side k, which is evaluated based on the flow variables resulting from the predictor step. The flux vector $\mathbf{F}_k^{n+1/2}$ is calculated by solving a local Riemann problem at each cell interface using a slope limited HLLC approximate Riemann solver [18, 23].

3.3. Stability criterion

Our non-linear shallow water equation solver is an explicit scheme, whose stability is determined by the Courant–Friedrichs–Lewy (CFL) criterion whereby the time step, Δt, is restricted such that

$$\Delta t = C \min(\Delta t_x, \Delta t_y) \tag{8}$$

with

$$\Delta t_x = \min_i \frac{\Delta x_i}{|u_i| + \sqrt{g h_i}} \quad \text{and} \quad \Delta t_y = \min_i \frac{\Delta y_i}{|v_i| + \sqrt{g h_i}}$$

where C is the Courant number specified in the range $0 < C \leqslant 1$, subscript i is the cell index, and Δx_i and Δy_i are the cell sizes in the x and y-directions, respectively. In the present work, an adaptive time step is used during the simulations, which is obtained by evaluating Equation (8) in the previous iteration with the Courant number C set to 0.65 for all the cases.

3.4. Boundary conditions

For the cases considered in this paper, the boundary conditions are either slip (reflective) wall or open flow. For slip (reflective) boundary conditions, the normal velocity components and the normal gradient of the tangential velocity components on the boundary wall are set to zero. The open boundary conditions are implemented using the following Riemann invariants according to

DOI: 10.1002/fld

the local Froude number ($Fr = u/\sqrt{gh}$):

(1) $Fr < 1$ (sub-critical flow)

$$\hat{u}_B = \hat{u}_I \pm 2\sqrt{g}(\sqrt{h_I} - \sqrt{h_B}), \quad \hat{v}_B = \hat{v}_I \quad \text{where } h_B \text{ is prescribed} \tag{9}$$

or

$$\hat{h}_B = \left(\sqrt{h_I} \pm \frac{1}{2\sqrt{g}}(\hat{u}_I - \hat{u}_B)\right)^2, \quad \hat{v}_B = \hat{v}_I \quad \text{where } \hat{u}_B \text{ is prescribed} \tag{10}$$

with the sign dependent on the flow direction: outflow +; inflow −.

(2) $Fr > 1$ (supercritical flow)
For inflow, the variables h_B, $\hat{u}_B$, and $\hat{v}_B$ are normally prescribed; and for outflow

$$h_B = h_I, \quad \hat{u}_B = \hat{u}_I, \quad \hat{v}_B = \hat{v}_I \tag{11}$$

In the above equations, $\hat{u}$ and $\hat{v}$ are the depth-averaged velocity components normal and tangential to the boundary. Subscripts B and I denote the boundary value and the inner Riemann state, respectively, at a given boundary point.

3.5. Modifications for cut cells

The high resolution finite volume numerical solver is constructed on a square fluid cell template. Near the solid boundary, the cut cells contain both fluid and solid faces and a single cell interface may have both fluid and solid parts. Hence, an alternative method is required to implement the boundary conditions properly in the cut cells, which is outlined next. We first classify each cut cell according to its size as either large or small. A large cut cell is defined as having an area larger than or equal to half that of the host grid cell from which it is cut. Examples of large cells are cells $(i-1, j-1)$, (i, j), $(i+1, j+1)$ in Figure 4. When the area of a cut cell is smaller than half its host cell, such a cut cell is defined as being small. Examples of small cells are cells $(i+1, j)$ and $(i, j-1)$ in Figure 4.

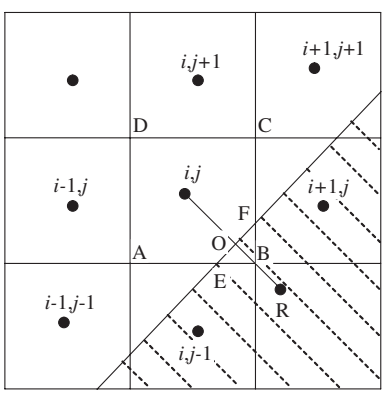

Figure 4. Calculation of interface flux on a cut cell.

DOI: 10.1002/fld

When evaluating interface fluxes for a large cut cell, the flow data at the cell face need to be approximated. The reconstruction procedure is similar to that for a normal fluid cell and is described as follows:

$$\mathbf{u}(x, y) = \mathbf{u}_{i,j} + \mathbf{r} \cdot \nabla \mathbf{u}_{i,j}^{\text{mod}} \tag{12}$$

where $\mathbf{r}$ is the normal distance vector from the cell centre to a specific point (x, y) on a cell interface, which is defined to be positive in east and north directions, $\nabla \mathbf{u}_{i,j}^{\text{mod}}$ is the limited gradient vector modified for calculating fluxes across the flow boundaries of any given large cut cell. According to Yang *et al*. [20] and Causon *et al*. [3], there are two types of gradients to be evaluated for a cut cell: fluid and solid. Taking the x-direction gradient vector on cell (i, j) in Figure 4 as an example, the fluid and solid gradients can be expressed by

$$\nabla \mathbf{u}_{i,j}^{\text{mod f}} = \Psi(l) \frac{\mathbf{u}_{i,j} - \mathbf{u}_{i-1,j}}{\Delta x_{i-1/2,j}} \quad \text{and} \quad \nabla \mathbf{u}_{i,j}^{\text{mod s}} = \Psi(l) \frac{\mathbf{u}_{i,j} - \mathbf{u}_{i-1,j}}{\Delta x_{i-1/2,j}} \tag{13}$$

in which $\Psi(l)$ is a slope limiter, $\Delta x_{i-1/2,j} = x_{i,j} - x_{i-1,j}$. There are several choices for the slope limiter and a concise expression is [24]

$$\Psi(l) = \max[0, \min(\beta l, 1), \min(l, \beta)] \tag{14}$$

in which l is the ratio of successive gradients and β ($1 \leqslant \beta \leqslant 2$) is the limiter parameter. Toro [23] describes the use of various limiters for Riemann-type solvers: $\beta = 1$ gives the min-mod limiter and $\beta = 2$ gives Roe's superbee limiter. In the present work, the min-mod limiter is used because of its superior properties for numerical stability. For fluid gradients, the ratio of successive gradients is given by

$$l = \frac{(q_{i+1,j} - q_{i,j})/\Delta x_{i+1/2,j}}{(q_{i,j} - q_{i-1,j})/\Delta x_{i-1/2,j}} \tag{15}$$

where q represents one of the principal flow variables, ζ, (uh) and (vh), and $\Delta x_{i+1/2,j} = x_{i+1,j} - x_{i,j}$. For solid gradients, the ratio is defined as

$$l = \frac{(q_{\text{R}} - q_{i,j})/\Delta x_{i,\text{R}}}{(q_{i,j} - q_{i-1,j})/\Delta x_{i-1/2,j}} \tag{16}$$

in which $\Delta x_{i,\text{R}} = x_{\text{R}} - x_{i,j}$, and flow information for fictional cell R is obtained using linear interpolation based on values at cell (i, j) and at the boundary point O that is prescribed according to boundary conditions. It is noted that if cell face DA is the edge containing both fluid and solid parts, the calculation is similar with vector $(\mathbf{u}_{i,j} - \mathbf{u}_{\text{R}})/\Delta x_{i,\text{R}}$ taking the place of the corresponding part in Equation (13) when evaluating solid gradients, with $\Delta x_{i,\text{R}} = x_{i,j} - x_{\text{R}}$. The x-direction limited gradient vector for the large cut cell (i, j) is given by

$$\nabla \mathbf{u}_{i,j}^{\text{mod}} = \frac{\nabla y_s \nabla \mathbf{u}_{i,j}^{\text{mod}\,s} + \Delta y_f \nabla \mathbf{u}_{i,j}^{\text{mod}\,f}}{\Delta y} \tag{17}$$

where $\Delta y_s = |BF|$, $\Delta y_f = |FC|$, $\Delta y = |BC|$. Now $\Delta y = \Delta y_s + \Delta y_f$, so Equation (17) includes the cases with $\Delta y_s = 0$ or $\Delta y_f = 0$, in which the modified limited gradient vector $\nabla \mathbf{u}_{i,j}^{\text{mod}}$ simply reduces to either the fluid or solid gradient term. The above formula can be extended to the

DOI: 10.1002/fld

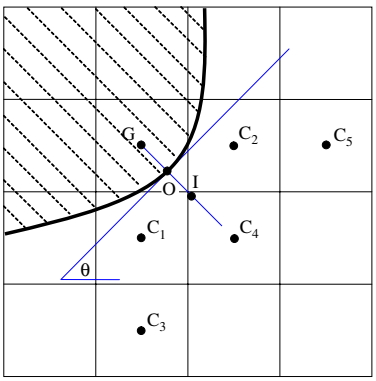

Figure 5. Updating flow information on small cut cells using GCIBM.

y-direction in a similar manner. After all the face values of the conservative flow variables are obtained, the interface fluxes through cell faces AE, FC, CD and DA can be evaluated using the corresponding formulations for the predictor and corrector steps.

The fluxes through the cut edge (EF in Figure 4) must also be properly evaluated in order to maintain the conservation properties of the numerical scheme. However, the normal velocity component on the cut edge EF is zero for slip (reflective) boundary conditions. Therefore, there is no convective flux through the cut edge. The only component of the inviscid fluxes through the cut edge is $g(\zeta^2 + 2\zeta h_s)/2$ due to gravitational acceleration.

Using Cartesian cut cells to approximate curved boundaries, very small cut cells are often generated. In an explicit numerical scheme, these small cells lead to the requirement of a very small time step to avoid numerical instability. A conventional means of overcoming this computationally inefficient requirement is to merge the small cells with a neighbouring larger cell [3, 20]. However, the increased cell sizes near the boundary may damage the solution accuracy. Thus, we use the Ghost-cell Immersed Boundary Method (GCIBM) proposed by Tseng and Ferziger [25] to treat the small cut cells. The flow information at the small cut cells is interpolated from the boundary conditions and neighbouring cells as explained next.

As shown in Figure 5, cell G is a small cut cell (ghost cell). Cell-centred values of the flow variables for this cell are obtained by interpolation from the neighbouring cells and appropriate boundary conditions at O in order to avoid the small time step requirement caused by the presence of the small cell. Here, a linear two-dimensional interpolation is used

$$\Phi = a_0 + a_1 x + a_2 y \tag{18}$$

where Φ represents values of flow variables at (x, y), and a_0, a_1 and a_2 are the coefficients to be determined. In order to construct the interpolation scheme, flow variables at three data points are needed, e.g. O, C_1 and C_2 in Figure 5. Using Equation (18) for the three nodes and written in matrix form gives

$$\mathbf{\Phi} = \mathbf{Ba} \tag{19}$$

DOI: 10.1002/fld

where $\mathbf{\Phi} = [\Phi_0\ \Phi_1\ \Phi_2]^{\mathrm{T}}$, $\mathbf{a} = [a_0\ a_1\ a_2]^{\mathrm{T}}$, and $\mathbf{B}$ is a 3×3 matrix given by

$$\mathbf{B} = \begin{bmatrix} 1 & x_0 & y_0 \\ 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \end{bmatrix} \tag{20}$$

Herein the flow information at cells C_1 and C_2 is known from the numerical solutions. If the flow variables at boundary point O are prescribed, the coefficients $\mathbf{a}$ can then be determined from

$$\mathbf{a} = \mathbf{B}^{-1}\mathbf{\Phi} \tag{21}$$

Flow information at point (x, y) is obtained using Equation (18). In order to improve numerical stability, Tseng and Ferziger [25] suggest calculating flow information at the image point of the ghost node first. The value at the ghost node is then given by $\Phi_{\mathrm{G}} = 2\Phi_0 - \Phi_{\mathrm{I}}$.

The above GCIBM is based on a two-dimensional linear interpolation scheme, which is first-order accurate. Tseng and Ferziger [25] prove that this preserves the accuracy of the overall numerical scheme.

In certain cases, boundary conditions are given in derivative form and the interpolation scheme can be easily modified for different boundary conditions. For example, slip boundary conditions give

$$\frac{\partial h_O}{\partial n} = 0, \quad \hat{u}_O = 0 \quad \text{and} \quad \frac{\partial \hat{v}_O}{\partial n} = 0 \tag{22}$$

where $\hat{u}_O$ and $\hat{v}_O$ are the normal and tangential components of depth-averaged velocity, and h_O is the water depth at boundary point O. For $\partial h_O/\partial n = 0$, the normal derivative on the boundary can be decomposed into

$$\frac{\partial h_O}{\partial n} = \frac{\partial h_O}{\partial x}\,\hat{n}_x + \frac{\partial h_O}{\partial y}\,\hat{n}_y \tag{23}$$

where $\hat{n}_x = -\sin(\theta)$ and $\hat{n}_y = \cos(\theta)$ are the components of the unit normal vector at the surface O, where θ is defined such that $\tan(\theta)$ is the slope of the normal, as in Figure 5. $\partial h_O/\partial x$ and $\partial h_O/\partial y$ can be easily obtained by partial differentiation of (18). The matrix $\mathbf{B}$ is then given by

$$\mathbf{B} = \begin{bmatrix} 0 & \hat{n}_x & \hat{n}_y \\ 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \end{bmatrix} \tag{24}$$

and the coefficient vector $\mathbf{a}$ is

$$\mathbf{a} = [a_0\ \ a_1\ \ a_2]^{\mathrm{T}} = \mathbf{B}^{-1}[\partial h_O/\partial n\ \ h_1\ \ h_2]^{\mathrm{T}} \tag{25}$$

For $\hat{u}_O = 0$ and $\partial \hat{v}_O/\partial n = 0$, we let

$$u = a_{u0} + a_{u1}x + a_{u2}y, \quad v = a_{v0} + a_{v1}x + a_{v2}y \tag{26}$$

DOI: 10.1002/fld

where $a_{u0}, a_{u1}, a_{u2}, a_{v0}, a_{v1}$ and a_{v2} are the coefficients required to be evaluated for the interpolation scheme. Expressing the boundary conditions in terms of Cartesian velocity components, we have

$$\hat{u}_O = -u_O \sin(\theta) + v_O \cos(\theta) = 0, \quad \frac{\partial \hat{v}_O}{\partial n} = \frac{\partial [u_O \cos(\theta) + v_O \sin(\theta)]}{\partial n} = 0 \tag{27}$$

After replacing the Cartesian velocity components u_O and v_O using Equation (26), the matrix $\mathbf{B}$ is derived as follows:

$$\mathbf{B} = \begin{bmatrix} \hat{n}_x & x_0\hat{n}_x & y_0\hat{n}_x & \hat{n}_y & x_0\hat{n}_y & y_0\hat{n}_y \\ 0 & \hat{n}_x\hat{n}_y & \hat{n}_y\hat{n}_y & 0 & -\hat{n}_x\hat{n}_x & -\hat{n}_x\hat{n}_y \\ 1 & x_1 & y_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_1 & y_1 \\ 1 & x_2 & y_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_2 & y_2 \end{bmatrix} \tag{28}$$

The coefficients a_{u0}, a_{u1}, a_{u2}, a_{v0}, a_{v1} and a_{v2} are then computed from

$$\mathbf{a} = [a_{u0} \;\; a_{u1} \;\; a_{u2} \;\; a_{v0} \;\; a_{v1} \;\; a_{v2}]^{\mathrm{T}} = \mathbf{B}^{-1}[\hat{u}_O \;\; \partial\hat{v}_O/\partial n \;\; u_1 \;\; v_1 \;\; u_2 \;\; v_2]^{\mathrm{T}} \tag{29}$$

4. RESULTS

Three validation simulations will be presented to illustrate the accuracy and efficiency of the numerical model. In cases where grid adaptation is invoked, the adaptation criterion is based on the non-dimensional root mean square values of the free surface gradient

$$\Theta = \sqrt{\left(\frac{\partial \zeta}{\partial x}\right)^2 + \left(\frac{\partial \zeta}{\partial y}\right)^2} \tag{30}$$

During dynamic grid adaptation, a grid cell is subdivided into four higher level cells if Θ is greater than a prescribed maximum value and the subdivision level of the cell under consideration is less than the maximum level. Cell removal is undertaken when four child cells having the same parent cell have Θ less than a prescribed minimum value and the child cells' subdivision level is greater than the minimum subdivision level.

4.1. Reflection of a surge wave at a wall

The first simulation involves the propagation and reflection of a surge wave initially travelling from left to right along a rectangular channel with a horizontal and frictionless bed. The channel is 10 000 m long and 1250 m wide. An open inlet is situated at the left-hand end of the channel, and a solid vertical wall at the right-hand end. In this and all subsequent simulations, $g = 9.81\,\mathrm{m/s^2}$. The initial still water depth is 5 m throughout the channel. At the left-hand boundary, a surge wave of 5 m amplitude is released. Before hitting the solid wall, the analytical solution of

DOI: 10.1002/fld

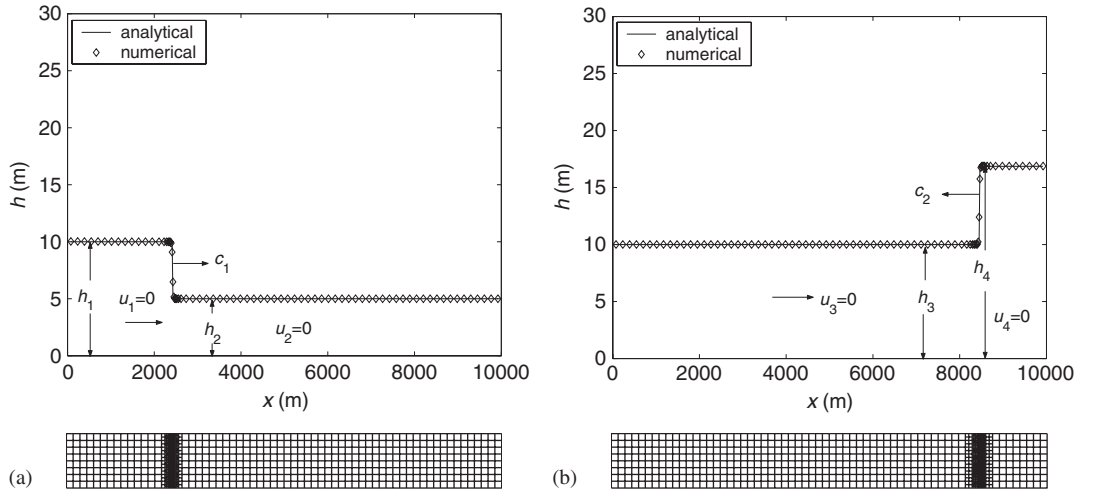

Figure 6. Surge reflection: water depth profile and quadtree grid at (a) $t = 200$ s and (b) $t = 1000$ s.

the flow is [26]

$$h_1 u_1 = c_1(h_1 - h_2), \quad c_1 = \sqrt{\frac{g h_1 (h_1 + h_2)}{2 h_2}} \tag{31}$$

where u_1 is the incoming depth-averaged velocity, $h_1 = 10$ m is the total water depth of surge, $h_2 = 5$ m is the undisturbed water depth (i.e. the initial still water depth), c_1 is the wave celerity of the propagating surge. According to Chow [26], the reflected flow is given by

$$h_3 u_3 = c_2(h_4 - h_3), \quad c_2 = \sqrt{\frac{g h_4 (h_3 + h_4)}{2 h_3}} - u_3 \tag{32}$$

in which $h_3 = h_1$ and $u_3 = u_1$ are the water depth and depth-averaged velocity of the surge before reflection, h_4 and c_2 are the water depth and wave celerity of the reflected surge, respectively.

Figure 6 shows the analytical and computational wave profiles at $t = 200$ s (before reflection) and $t = 1000$ s (after reflection). At all times, the computational water depth is almost identical to the analytical solution, and the discontinuous solutions caused by the surge wave are correctly captured. The region of finest resolution of the dynamically adapted quadtree grids is located at the surge wave front, and provides high-resolution gridding (with highest and lowest subdivision levels of 9 and 6) in those areas near to the solution discontinuity. $\Theta_{\max} = 0.0005$ and $\Theta_{\min} = 0.00025$ are prescribed for grid adaptation.

In order to evaluate the computational efficiency and grid convergence of the present flow model, calculations have also been performed on uniform quadtree grids with different subdivision levels. The average error (L_1 norm) is defined as

$$e = \frac{\sum_{i,j} |h_{i,j} - \tilde{h}_{i,j}| A_{i,j}}{\sum_{i,j} \tilde{h}_{i,j} A_{i,j}} \tag{33}$$

DOI: 10.1002/fld

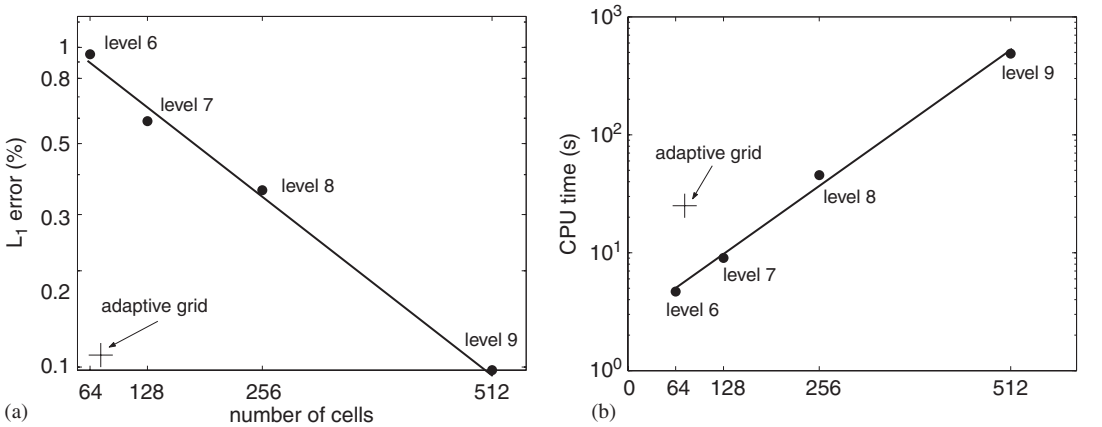

Figure 7. Surge reflection: (a) average error versus number of grid cells in x-direction for different grids; and (b) CPU time for different grids.

where $h_{i,j}$ and $\tilde{h}_{i,j}$ are the cell-centred computational and analytical water depths at cell (i, j), respectively, and $A_{i,j}$ is the cell area. The average error of the solution at $t = 1000$ s against the number of grid cells along the channel is shown in Figure 7(a) for the adaptive and different uniform quadtree grids. It is evident that the solution converges as the grids become finer. The symbol '+' in the figure represents the average error as a function of the number of cells along the channel for the solution on the adaptive quadtree grid. Using the dynamically adaptive quadtree grid, the numerical solution is almost as accurate as that obtained on a level 9 uniform grid, even though the number of grid cells used is similar to that of a level 6 uniform grid. The computational efficiency gain achieved by using an adaptive quadtree grid is indicated in Figure 7(b), showing the CPU time required for the adaptive grid simulation is $\sim 5\%$ of that for a level 9 uniform grid.

4.2. Low Froude number potential flow past a circular cylinder

A validation case is proposed for a low Froude number potential flow past a circular cylinder. The flow occurs in a 20 m × 10 m horizontal and frictionless channel, with a 2 m diameter circular cylinder located in the middle. Based on potential flow theory [27], the velocity potential of a dipole between parallel walls is

$$\Omega = U_0 z + U_0 r^2 \sum_{n=-\infty}^{n=\infty} \frac{1}{z + 2ibn} \tag{34}$$

where U_0 is the background flow velocity, r is the radius of the cylinder in an infinite domain, b is the width of the channel and $z = x + iy$ is the complex position co-ordinate. Then the velocity field is given by

$$u - iv = U_0 \left(1 - \frac{r^2\pi^2}{4b^2} \operatorname{csch}\left(\frac{\pi z}{2b}\right)^2\right) \tag{35}$$

It is stressed that this is the exact potential flow velocity field due to the embedded dipole. It does not exactly correspond to flow around a circle in a channel—the dividing streamline is equivalent

DOI: 10.1002/fld

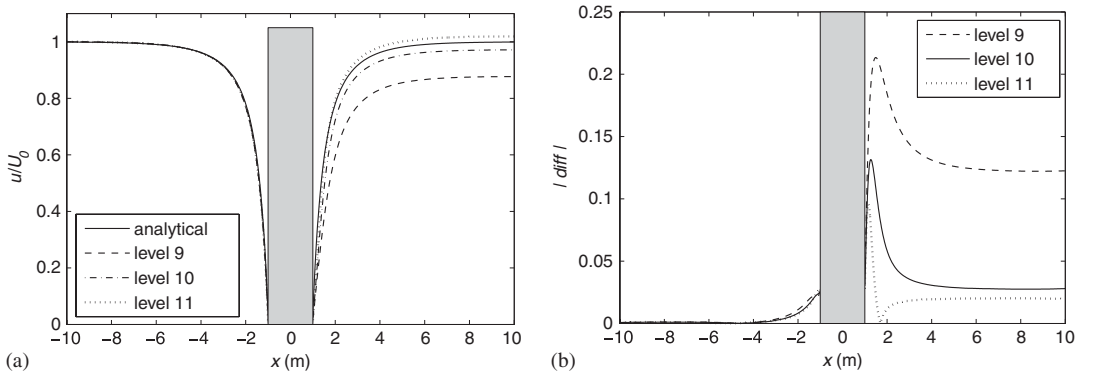

Figure 8. Low Froude number flow past cylinder: (a) centre-line velocity profile along the channel; and (b) absolute error between the analytical and numerical profiles.

to a body of length $1.9685r$ and transverse width $1.9677r$—due to the images of the dipole in the walls. However, this solution is deemed to be a sufficiently accurate model for flow past the cylinder to use as a benchmark for the quadtree solver.

Because the flow is strictly symmetric about the horizontal central line of the channel ($y = 0$), numerical calculations are performed in half of the original physical domain. The potential flow is driven by the western inlet boundary with a uniform incoming velocity of U_0. In order not to induce significant free surface displacement, the Froude number (Fr) of the undisturbed incoming flow is chosen to be very low (0.08), and the incoming uniform velocity is determined by $U_0 = Fr\sqrt{gh_s}$, with $h_s = 1$ m being the still water level. The eastern boundary is transmissive and the southern and northern boundaries are slip. Because the flow is smooth everywhere inside the computational domain, calculations are considered on uniform quadtree grids without adaptation. Cut cells are incorporated with quadtree grids to fit the circular boundary. For quadtree grids of levels 9, 10 and 11, there are approximately 150, 300, 600 cut cells, respectively, around half of the circular boundary. Simulations start from the analytical velocity field and run until steady state (run for 150 s for all the cases considered herein).

Figure 8 presents a comparison of the normalized central line velocity profiles along the channel obtained on different level quadtree grids with the corresponding analytical solution. In the mean flow direction, the normalized analytical velocity decelerates from 1 to 0 at the front stagnation point, while it accelerates from 0 at the rear stagnation point to 1 far downstream. The numerical results presented in Figure 8(a) show that the deceleration of the flow ahead of the front stagnation point is properly simulated even on the relatively coarse level 9 quadtree grid. The acceleration downstream of the rear stagnation represents the most challenging part of the simulation, where the difference between the analytical solution and the numerical predicted profile on a coarse grid of level 9 is evident. The numerical predictions on finer grids of level 10 and 11 provide much better agreement with the analytical solution, even though there is a slight overshoot of the numerical profile on the level 11 quadtree grid. The absolute differences between the analytical solution and the numerical profiles are presented in Figure 8(b). With the increasingly finer grids, the absolute error becomes smaller, indicating the numerical predictions are converging. Figure 9 illustrates the analytical and numerical transverse profiles of the u-velocity from the shoulder of the body outwards to the wall of the channel. The predicted transverse profile obtained on the level 11 grid

DOI: 10.1002/fld

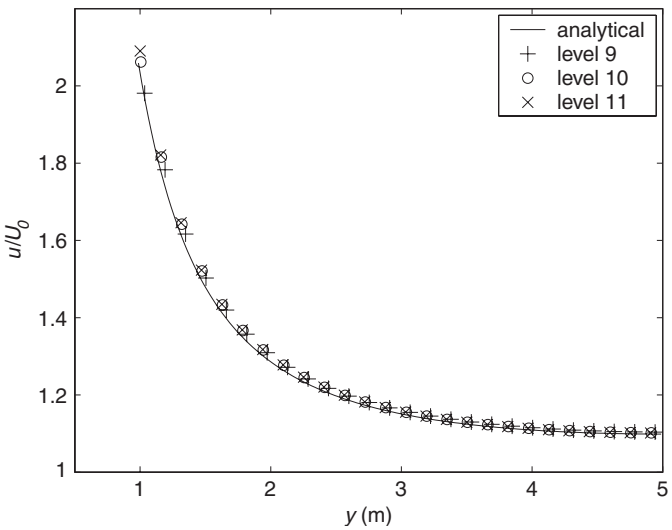

Figure 9. Low Froude number flow past cylinder: transverse velocity profile from the shoulder of the body outwards to the wall of the channel.

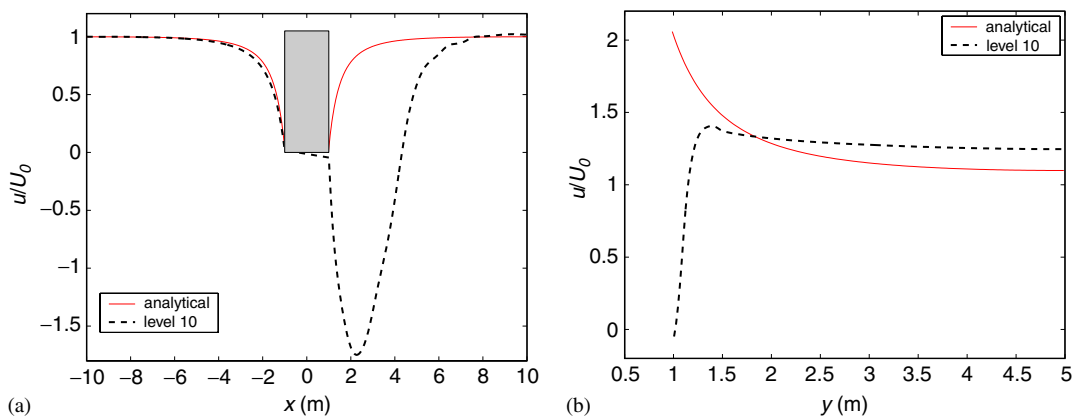

Figure 10. Low Froude number flow past cylinder on quadtree grid without cut cell: (a) centre-line velocity profile along the channel; and (b) velocity profile from the shoulder of the body outwards to the wall of the channel.

contains a slight velocity overshoot at the shoulder point ($y = 1$ m). The level 11 grid is the finest on which we were able to obtain a solution without incurring an excessive CPU time penalty on a Pentium 3 GHz (1 G RAM) PC.

Figure 10 presents a comparison between the analytical solution and the predicted centre-line longitudinal and transverse velocity profiles on a level 10 quadtree grid without cut cells. It is evident that a spurious vortex wake has been produced downstream and the velocity at the shoulder point is reduced to zero because of the stair-case approximation of the curved boundary. Therefore,

DOI: 10.1002/fld

in simulating inviscid flow with curved boundaries, spurious numerical viscosity induced by the stair-case boundary approximation can contaminate the predictions, and so cut cells are essential in order to obtain reasonably accurate solutions.

4.3. Shock reflection by a circular cylinder

The interaction of a shock-like bore with a circular cylinder has been intensively investigated (see References [3, 28, 29]). In the present work, the calculation is carried out in a 5 m × 5 m domain, with a horizontal, frictionless bed. At the centre is placed a circular surface-piercing cylinder of diameter 1 m. The domain has open inlet and outlet boundaries at its west and east ends. The north and south lateral boundaries are open. The shock-like bore propagates from the western (left) boundary, according to flow states given by [3]

$$\begin{aligned} h_{\mathrm{L}} &= l h_{\mathrm{R}} \\ u_{\mathrm{L}} &= c + (u_{\mathrm{R}} - c)/l \\ v_{\mathrm{L}} &= v_{\mathrm{R}} \end{aligned} \tag{36}$$

where c is the speed of the shock wave and l is defined as

$$l = \tfrac{1}{2}\left(\sqrt{1 + 8(Fr_{\mathrm{R}} - Fr_{\mathrm{S}})^2} - 1\right) \tag{37}$$

in which $Fr_{\mathrm{R}} = u_{\mathrm{R}}/\sqrt{gh_{\mathrm{R}}}$ is the Froude number of the flow before the shock wave arrives, Fr_{S} is the prescribed Froude number of the shock and $c = Fr_{\mathrm{S}}\sqrt{gh_{\mathrm{R}}}$. Initially, the water is at rest, and has a still water depth equal to 1 m. The flow states before the shock are: $h_{\mathrm{R}} = 1.0$ m; $u_{\mathrm{R}} = 0$ m/s; and $v_{\mathrm{R}} = 0$ m/s. The Froude number of the incident shock (Fr_{S}) is set to 2.81, identical to that studied by Causon *et al.* [3]. The flow is assumed to be inviscid. The initial quadtree grid has highest and lowest subdivision levels of 9 and 7, respectively. The level 9 fine mesh is generated about the circular boundary of the cylinder and the region where the initial shock front is located. The resulting grid has 29 260 leaf cells, of which 410 are cut cells used to approximate the circular boundary of the cylinder. During grid adaptation, $\Theta_{\max} = 1.8$ and $\Theta_{\min} = 1.5$.

Figure 11 presents a series of water depth contours at different output times, where $t = 0$ refers to the moment when the shock hits the rigid surface of the cylinder. Once the collision occurs, a circular reflected shock wave is instantaneously induced and the flow forms a two-shock system. The flow is strictly symmetric and the confluence points of the two shocks move along the cylinder surface. Considering the upper part of the flow, when the angle between the front stagnation point and the shock collision point increases to about 45° at $t = 0.025$ s, the collision point separates from the cylinder surface and changes into a triple point. A diffracted shock appears and a three-shock system is developed together with the incident and reflected shocks, indicating the flow dynamics regime changing from regular reflection to Mach reflection [29]. Later, the diffracted shock fronts from upper and lower sides of the cylinder curl up and meet at the rear stagnation point behind the cylinder, creating vortices, and resulting in an instantaneous increase of water depth in the wake that is then transported downstream. With the complicated wave pattern caused by the shock–shock interaction propagating downstream, together with the vortex motion, a cavity develops in the near wake, and remains behind the cylinder for a considerable time. The adapted quadtree grid at $t = 0.3$ s shown in Figure 11 indicates how the grid has adapted to the flow field. Figure 12 illustrates the evolution of the upper triple point. It is evident that the triple point develops at the

DOI: 10.1002/fld

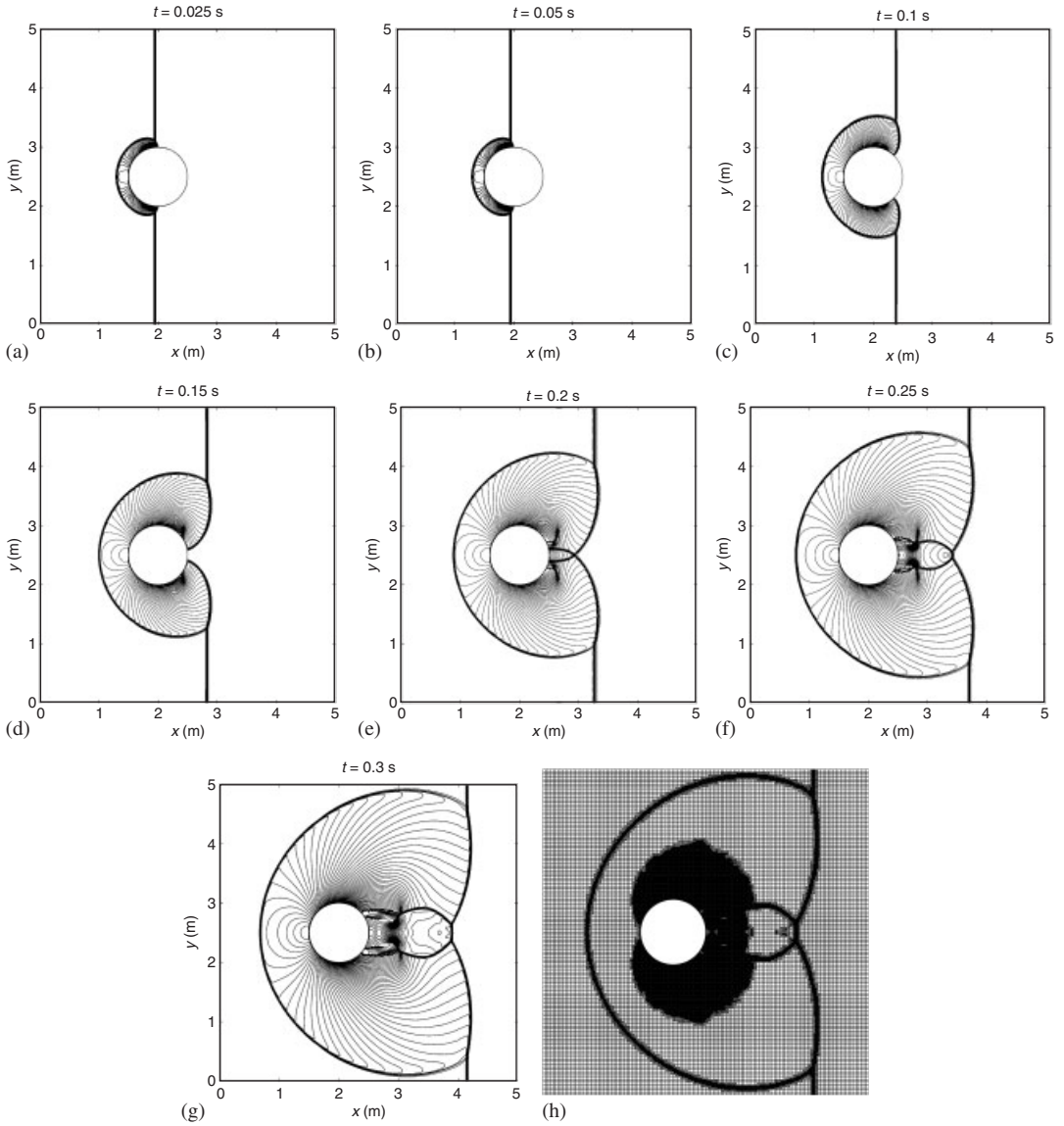

Figure 11. Shock reflection by a circular cylinder: depth contours at different times, and adaptive quadtree grid at $t = 0.3$ s.

surface of the cylinder at about 45° from the front stagnation point and then propagates outwards tangentially to the cylinder from this point. Figure 13 shows the time histories of water depth at different gauge points, located every 30° on the cylinder surface starting from the front stagnation point, as illustrated in Figure 13(a). The water depth histories of those upstream gauge points (point 1, 2, 3 and 4) exhibit similar behaviour, with the peak depths decreasing away from the

DOI: 10.1002/fld

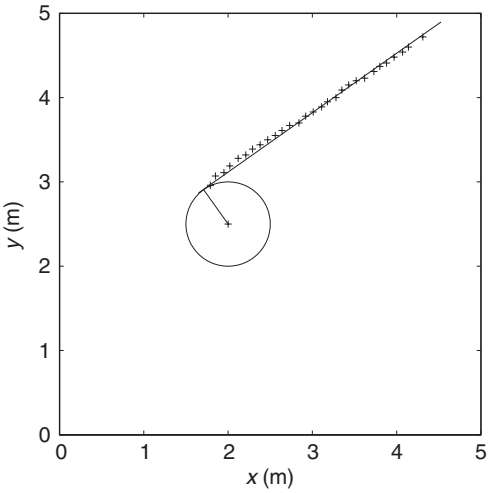

Figure 12. Shock reflection by a circular cylinder: evolution of the triple point, where the + symbols represent the positions of the triple point at different times and the line is a best fit.

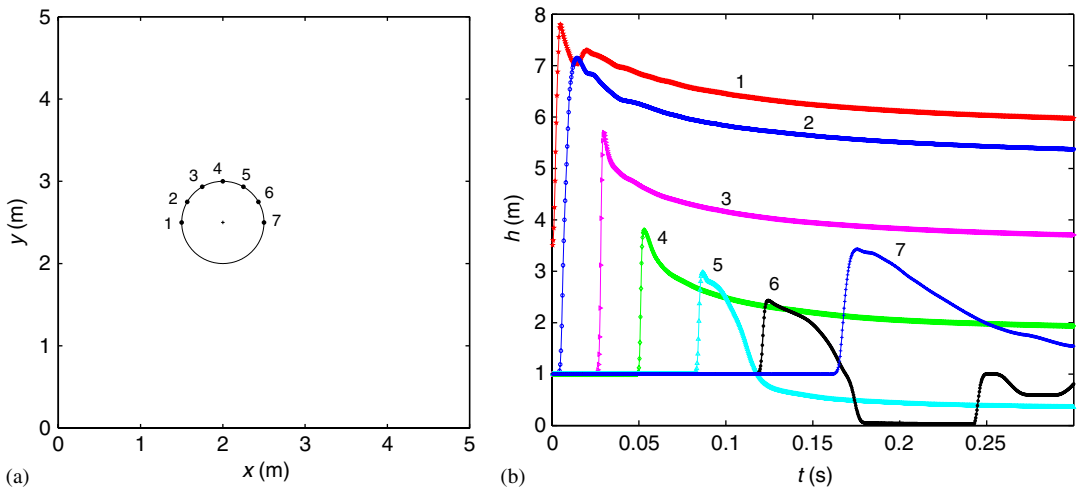

Figure 13. Shock reflection by a circular cylinder: time histories of water depth at different sample points.

front stagnation point. A sudden increase of water depth is observed at each of these points when the incident shock (or diffracted shock for gauges 3 and 4) hits the cylinder at the corresponding points. The depths abruptly drop, immediately after the collision, followed by a much more gentle decrease almost leading to a steady depth as the shock propagates outwards from the cylinder. The water depth at gauge point 5 also jumps suddenly as the diffracted shock arrives. Afterwards, the depth decreases rapidly, falling to a near constant level less than the initial still water depth. At gauge point 6, after the increase of water depth caused by the diffracted shock, the water

DOI: 10.1002/fld

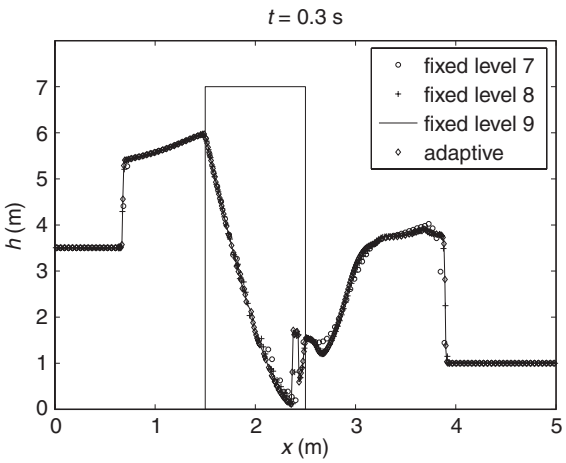

Figure 14. Shock reflection by a circular cylinder: depth profile along the centre-line and around the cylinder surface, computed on different grids.

depth then decreases to almost zero as the shock travels outwards. Another sudden increase and quick decrease of water depth occur as the interacted shock, formed after the meeting of the two diffracted shock fronts, passes by the gauge point. Further fluctuations of the water depth occur due to complicated wave interactions in the wake. At the rear stagnation point (gauge point 7), a gain of depth occurs as the diffracted shock fronts meet, where the largest water depth is higher than those at gauge points 5 and 6. The complicated wave pattern formed by the collision of the shock fronts and the resulting high water level then moves outwards, causing a gradual decrease of water depth at gauge point 7.

The predicted results agree closely with those obtained by Causon *et al*. [3], who solved the shallow water equations on a uniform Cartesian grid with cut cells. Excellent agreement is also achieved with the results of Yang *et al*. [29], who solved the gas dynamics equations using finite differences. Of course, the shallow water equations can be viewed as a special version of the compressible gas dynamics equations with the ratio of the specific heats ($\gamma = 2$).

Calculations have also been carried out on different levels of uniform quadtree and structured Cartesian grids. In all cases, Cartesian cut cells have been used to provide boundary-fitting. Figure 14 shows the depth profile along the central line of the channel and around the surface of the cylinder at $t = 0.3$ s on the adaptive and three different levels of uniform quadtree grids. Herein the solution on the finest level 9 uniform grid is used as a reference. The predictions on coarser grids converge to the reference solution, but the depth profile obtained on the level 7 grid is more diffusive, as high surface gradients occur. The depth profile produced on the adaptive quadtree grid is almost identical to the reference solution. In order to evaluate the computational efficiency of the present model, the computation has also been performed on a 480×480 structured Cartesian grid, which has similar resolution to the level 9 uniform quadtree grid. On a Pentium 3 GHz (1 G RAM) PC, the CPU time for the simulation is 36.7 min on the structured grid, whereas it is only 6.0 min using the dynamically adaptive quadtree grid.

Roache's [30] Grid Convergence Index (GCI) has been evaluated based on the water depth predictions obtained at $t = 0.3$ s on the three uniform quadtree grids at levels 7, 8 and 9. This is

DOI: 10.1002/fld

Table I. Shock reflection by a circular cylinder: water depth based GCI at 11 sample points on four continuously doubling uniform quadtree grids.

Sample points		h_1 (m)	h_2 (m)	h_3 (m)		
x (m)	y (m)	$\Delta x \approx 0.01$ m	$\Delta x \approx 0.02$ m	$\Delta x \approx 0.04$ m	GCI_{12}	GCI_{23}
0.80	2.52	5.46	5.46	5.47	0.11	0.21
3.42	2.52	3.75	3.76	3.79	0.31	0.83
3.81	4.51	3.76	3.76	3.78	0.12	0.64
3.81	4.32	3.64	3.65	3.66	0.23	0.30
3.81	4.08	3.50	3.50	3.53	0.17	0.82
3.42	4.71	3.92	3.92	3.90	0.09	0.51
2.40	4.71	4.33	4.32	4.16	0.28	3.73
2.13	4.51	4.40	4.41	4.43	0.11	0.41
1.70	4.32	4.66	4.68	4.59	0.33	1.90
1.50	4.12	4.76	4.76	4.76	0.01	0.12
2.71	3.07	1.44	1.45	1.51	0.76	4.15

defined as

$$GCI_{21} = 3\,\frac{|e_{21}|}{r^p - 1} \tag{38}$$

where $r = \Delta x_2/\Delta x_1 = 2$ is the grid refinement ratio, $p = 2$ is the formal order of accuracy of the numerical scheme, and e_{21} is the percentage relative error, defined as

$$e_{21} = 100\,\frac{h_1 - h_2}{h_1} \tag{39}$$

in which h_1 and h_2 are the water depths at the same location on a fine and coarse grid, respectively. The values of GCI at all the grid points have been checked and Table I lists the values of GCI obtained at eleven sample points on level 9 and 8 grids (GCI_{21}) and on level 8 and 7 grids (GCI_{32}). The results indicate that the calculations are convergent to the asymptotic value on the grids being used in the present study.

5. CONCLUSIONS

This paper describes a Godunov-type shallow water equation solver based on dynamically adaptive quadtree grids with boundary-fitting provided by Cartesian cut cells. Previous work has shown that dynamically adaptive quadtree grids allow locally high gradient shallow flows to be resolved [7, 18]. However, the stair-case approximation of a curved boundary can introduce spurious reflections and local separation effects akin to artificial viscosity, grossly compromising the accuracy of the scheme. Here curved boundaries are approximated as straight line segments with Cartesian cut cells within the quadtree grid. The present combined scheme is computationally fast and provides an accurate representation of the inviscid flow field near curved boundaries.

One of the most difficult aspects of applying Cartesian cut cells is the avoidance of the requirement for a tiny time-step triggered by the very small cut cells which can arise on boundaries inclined to the main grid axes. In the present paper, the Ghost-cell Immersed Boundary Method

DOI: 10.1002/fld

(GCIBM) of Tseng and Ferziger [25] has been implemented to update flow variables at each small cut cell by interpolation from nearest neighbour cells and boundary values. Without cell merging, the GCIBM effectively eliminates the small time step restriction associated with small cut cells and preserves the overall second-order accuracy of the numerical scheme.

The dynamically adaptive quadtree shallow flow model is validated for surge reflection from a wall, and agreement obtained between the predicted and analytical solutions. For the much more challenging problem of low Froude number inviscid shallow flow past a circular cylinder, the combined quadtree cut cell model predicts a steady velocity field in agreement with an approximate potential flow analytical solution. However, simulations produced without cut cells at the cylinder boundary give grossly inaccurate results due to numerical viscosity induced by the stair-case representation of the curved boundary even on a very fine grid.

The global performance of the numerical solver is evaluated by simulating the reflection and diffraction phenomena produced by a strong hydraulic jump impacting a circular cylinder. Numerical convergence is confirmed using Roache's grid convergence index. The predicted results match those given in the literature using alternative discretization schemes. Numerical experiments indicate that the quadtree cut cell model simulation is about six times faster than its counterpart on a structured Cartesian grid for the case considered. Thus, the present numerical model represents an advance on shallow water equation solvers based solely either on quadtree grids [7, 18] or else on uniform grids with cut cells [3].

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial support of UK EPSRC (Grant no. GR/T07220/01). Thanks are also due to Dr Yu-Heng Tseng of the Lawrence Berkeley Laboratory for advice concerning the Ghost-cell Immersed Boundary Method and to Professor Derek Causon of Manchester Metropolitan University for useful discussions on the Cartesian cut cell method.

REFERENCES

1. Alcrudo F, García-Navarro P. A high-resolution Godunov-type scheme in finite volumes for the 2D shallow-water equations. *International Journal for Numerical Methods in Fluids* 1993; **16**(6):489–505.
2. Anastasiou K, Chan CT. Solution of the 2D shallow water equations using the finite volume method on unstructured triangular meshes. *International Journal for Numerical Methods in Fluids* 1997; **24**(11):1225–1245.
3. Causon DM, Ingram DM, Mingham CG, Yang G, Pearson RV. Calculation of shallow water flows using a Cartesian cut cell approach. *Advances in Water Resources* 2000; **23**(5):545–562.
4. Fraccarollo L, Toro EF. Experimental and numerical assessment of the shallow water model for two-dimensional dam-break type problems. *Journal of Hydraulic Research* 1995; **33**(6):843–863.
5. Fujihara M, Borthwick AGL. Godunov-type solution of curvilinear shallow-water equations. *Journal of Hydraulic Engineering* (ASCE) 2000; **126**(11):827–836.
6. Hu K, Mingham CG, Causon DM. A bore-capturing finite volume method for open-channel flows. *International Journal for Numerical Methods in Fluids* 1998; **28**(8):1241–1261.
7. 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**(3):247–280.
8. Sleigh PA, Gaskell PH, Berzins M, Wright NG. An unstructured finite-volume algorithm for predicting flow in rivers and estuaries. *Computers and Fluids* 1998; **27**(4):479–508.
9. Zhao DH, Shen HW, Tabious III GQ, Lai SJ, Tan WY. Finite-volume two-dimensional unsteady-flow model for river basins. *Journal of Hydraulic Engineering* (ASCE) 1994; **120**(7):864–883.
10. Zoppou C, Roberts S. Numerical solution of the two-dimensional unsteady dam break. *Applied Mathematical Modelling* 2000; **24**(7):457–475.

DOI: 10.1002/fld

11. Zhou JG, Ingram DM, Mingham CG. Numerical solutions of the shallow water equations with discontinuous bed topography. *International Journal for Numerical Methods in Fluids* 2002; **38**(8):769–788.
12. Hubbard ME, Dodd N. A 2D numerical model of wave run-up and overtopping. *Coastal Engineering* 2002; **47**:1–26.
13. Gallouët T, Hérard J-M, Seguin N. Some approximate Godunov schemes to compute shallow-water equations with topography. *Computers and Fluids* 2003; **32**:479–513.
14. Sankaranarayanan S, Spaulding ML. A study of the effects of grid non-orthogonality on the solution of shallow water equations in boundary-fitted coordinate systems. *Journal of Computational Physics* 2003; **184**:299–320.
15. Namin M, Lin B, Falconer RA. Modelling estuarine and coastal flows using an unstructured triangular finite volume algorithm. *Advances in Water Resources* 2004; **27**:1179–1197.
16. Rosatti G, Cesari D, Bonaventura L. Semi-implicit, semi-Lagrangian modelling for environmental problems on staggered Cartesian grids with cut cells. *Journal of Computational Physics* 2005; **204**:353–377.
17. Ye T, Mittal R, Udaykumar HS, Shyy W. An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries. *Journal of Computational Physics* 1999; **156**(2):209–240.
18. Liang Q, Borthwick AGL, Stelling G. Simulation of dam- and dyke-break hydrodynamics on dynamically adaptive quadtree grids. *International Journal for Numerical Methods in Fluids* 2004; **46**(2):127–162.
19. Samet H. Neighbor finding techniques for images represented by quadtrees. *Computer Graphics and Image Processing* 1982; **18**(1):37–57.
20. Yang G, Causon DM, Ingram DM, Saunders R, Batten P. A Cartesian cut cell method for compressible flows: A. Static body problems. *Aeronautical Journal* 1997; **101**(1002):47–56.
21. Causon DM, Ingram DM, Mingham CG. A Cartesian cut cell method for shallow water flows with moving boundaries. *Advances in Water Resources* 2001; **24**:899–911.
22. Rogers BD, Borthwick AGL, Taylor PH. Mathematical balancing of flux gradient and source terms prior to using Roe's approximate Riemann solver. *Journal of Computational Physics* 2003; **192**(2):422–451.
23. Toro EF. *Riemann Solvers and Numerical Methods for Fluid Dynamics*: *A Practical Introduction*. Springer: Berlin, New York, 1997.
24. Hirsch C. *Numerical Computation of Internal and External Flows. Vol. 2*: *Computational Methods for Inviscid and Viscous Flows*. Wiley: New York, U.S.A., 1990.
25. Tseng Y-H, Ferziger JH. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics* 2003; **192**:593–623.
26. Chow VT. *Open-Channel Hydraulics*. McGraw-Hill: New York, 1959.
27. Milne-Thomson LM. *Theoretical Hydrodynamics*. Macmillan: London, 1938.
28. LeVeque RJ. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press: Cambridge, MA, 2002.
29. Yang JY, Liu Y, Lomax H. Computation of shock wave reflection by circular cylinders. *AIAA Journal* 1987; **25**(5):683–689.
30. Roache PJ. Perspective: a method for uniform reporting of grid refinement studies. *Journal of Fluids Engineering* 1994; **116**:405–413.

DOI: 10.1002/fld