INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2008; **56**:2091–2110
Published online 20 September 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1615

Simple treatment of non-aligned boundaries in a Cartesian grid shallow flow model

Qiuhua Liang[1,*,†] and Alistair G. L. Borthwick[2]

[1] *School of Civil Engineering and Geosciences, University of Newcastle upon Tyne, NE1 7RU, U.K.*
[2] *Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, U.K.*

SUMMARY

A simple method is proposed for treating curved or irregular boundaries in Cartesian grid shallow flow models. It directly evaluates fictional values in 'ghost' cells adjacent to boundary cells and requires no interpolation or generation of cut cells. The boundary treatment is implemented in a dynamically adaptive quadtree grid-based solver of the hyperbolic shallow water equations and validated against several test cases with analytical or alternative numerical solutions. The method is easy to code, accurate, and demonstrably effective in dealing with irregular computational domains in shallow flow simulations. Results are presented for still water in a basin of complicated geometry, steady hydraulic jump in an open channel with a converging sidewall, wind-induced circulation in a circular shallow lake, and shock wave diffraction in a channel containing a contraction and expansion. Copyright © 2007 John Wiley & Sons, Ltd.

Received 16 April 2007; Revised 6 August 2007; Accepted 7 August 2007

KEY WORDS: Cartesian method; boundary fitness; shallow flow; Godunov-type scheme; quadtree grid; Riemann solver

1. INTRODUCTION

Accuracy, efficiency, and boundary fitness are three of the most important prerequisites of a successful numerical model. This paper focuses on boundary fitness in the context of shallow flow simulation, where the natural domain geometry can be very complicated. Shallow free surface flows occur wherever the flow is predominantly horizontal and gravity waves are long with respect to the depth, such as in channels, rivers, floodplains, lakes, lagoons, estuaries, and coastal waters.

Much attention has been given to devising methods for overcoming the problem of fitting computational grids to domains with curved or irregular boundaries. Automatic boundary fitting

[*] Correspondence to: Qiuhua Liang, School of Civil Engineering and Geosciences, University of Newcastle upon Tyne, NE1 7RU, U.K.

[†] E-mail: Qiuhua.Liang@ncl.ac.uk, qiuhua.liang@newcastle.ac.uk

is usually achieved using one of three grid generation methods: structured curvilinear grids (see e.g. [1–3]); unstructured triangular grids (see e.g. [4–6]); and Cartesian grids with local treatment of near-boundary cells (see e.g. [7, 8]). The curvilinear systems approach involves mapping the physical flow geometry onto a computational grid composed of rectangular blocks. The transformed governing equations are more complicated in curvilinear coordinates, and this complexity can influence the stability and convergence properties of the solver [9]. Furthermore, the quality of the curvilinear grid obviously depends on the shape of the boundary, and hence it is awkward to apply the method to problems with highly irregular boundaries. The second approach is to utilize an unstructured mesh, mostly created by means of a moving front or Voronoi scheme (see e.g. [10]). Such meshes are widely utilized with finite volume or finite element methods, and can conform to nearly any desired geometry (see e.g. [6]). Although unstructured meshes are straightforward to coarsen and refine, remeshing can carry a large computational overhead with regard to node neighbour identification and measures to ensure local cell quality. A third approach is to implement special boundary-fitting techniques directly on a Cartesian-type grid. Examples include irregular stars for finite differences (see e.g. [11]), the Cartesian cut cell method [7, 12, 13], and immersed boundary methods (e.g. [8, 9]). In the Cartesian cut cell method, all cell faces are aligned parallel to local Cartesian coordinate axes and the governing equations are solved directly in Cartesian coordinates.

In this paper, we propose a simple, yet effective method for boundary fitting in the context of a dynamically adaptive quadtree grid-based shallow water equation solver. The proposed boundary treatment method calculates flow variables in ghost cells directly from the neighbouring cell under consideration, and thus it is very simple and easy to implement. Despite its simplicity, it will be demonstrated that the method is effective and accurate. Another important feature is that the method has no effect on computational efficiency. The structure of the paper is as follows. Section 2 briefly presents a Godunov-type finite volume solver of the hyperbolic matrix form of the shallow water equations, with mathematically balanced flux gradient and source terms. The solver is designed to model steep-fronted shallow flows over non-uniform terrain. Section 3 describes the boundary approximation used to fit Cartesian grids to complicated boundaries. Section 4 discusses results from four validation tests: still water in a basin of complicated bed topography and irregular side walls, the development of an oblique hydraulic jump at a converging wall junction, wind-induced circulation patterns in a circular lake of varying bathymetry, and shock-like bore propagation in a channel containing a contraction followed by an expansion. Section 5 outlines the main conclusions.

2. QUADTREE GODUNOV-TYPE SHALLOW FLOW MODEL

Using Green's theorem, a matrix-hyperbolic conservation form of the shallow water equations is

$$\frac{\partial}{\partial t}\int_{\Omega}\mathbf{u}\,\mathrm{d}\Omega+\oint_{S}\mathbf{F}\,\mathrm{d}S=\int_{\Omega}\mathbf{s}\,\mathrm{d}\Omega \tag{1}$$

where t is time, $\mathbf{u}$ is the vector of conserved variables, and $\mathbf{F}$ is the vector of fluxes passing through the lateral boundary (S) of the problem domain (Ω) defined by

$$\mathbf{F}=\mathbf{f}n_x+\mathbf{g}n_y \tag{2}$$

DOI: 10.1002/fld

where **f** and **g** are the flux component vectors, n_x and n_y are Cartesian components, and **s** is the vector of source terms. In full, the constituent vectors are

$$\mathbf{u}=\begin{bmatrix}\eta\\ uh\\ vh\end{bmatrix},\quad \mathbf{f}=\begin{bmatrix}uh\\ u^2h+g(\eta^2-2\eta z_\mathrm{b})/2-\varepsilon h\,\partial u/\partial x\\ uvh-\varepsilon h\,\partial v/\partial x\end{bmatrix}$$

$$\mathbf{g}=\begin{bmatrix}vh\\ uvh-\varepsilon h\,\partial u/\partial y\\ v^2h+g(\eta^2-2\eta z_\mathrm{b})/2-\varepsilon h\,\partial v/\partial y\end{bmatrix}\quad\text{and}\quad \mathbf{s}=\begin{bmatrix}0\\ (\tau_{wx}-\tau_{\mathrm{b}x})/\rho-g\eta S_{ox}+hfv\\ (\tau_{wy}-\tau_{\mathrm{b}y})/\rho-g\eta S_{oy}-hfu\end{bmatrix}$$

where η is the water surface elevation and z_b is the bed elevation above datum, such that the total water depth is $h=\eta-z_\mathrm{b}$; u and v are the depth-averaged Cartesian velocity components; ε is the depth-averaged kinematic eddy viscosity coefficient; τ_{wx} and τ_{wy} are the surface (wind) stress components; $\tau_{\mathrm{b}x}$ and $\tau_{\mathrm{b}y}$ are the bed friction stresses; f is the Coriolis parameter; g is the acceleration due to gravity; ρ is the water density; and S_{ox} $(=\partial z_\mathrm{b}/\partial x)$ and S_{oy} $(=\partial z_\mathrm{b}/\partial y)$ are the bed gradients in the horizontal Cartesian directions. Liang and Borthwick [14] provide a full derivation of the above shallow water equations. The flux gradient and source terms are mathematically balanced and so there is no need for additional numerical treatment for spatially non-uniform bed topography. The flux vector can also be expressed in terms of inviscid and viscous fluxes as

$$\mathbf{F}=\mathbf{F}^\mathrm{I}-\varepsilon\mathbf{F}^\mathrm{V} \tag{3}$$

where $\mathbf{F}^\mathrm{I}$ and $\mathbf{F}^\mathrm{V}$ are inviscid and viscous flux vectors given by

$$\mathbf{F}^\mathrm{I}=\mathbf{f}^\mathrm{I}n_x+\mathbf{g}^\mathrm{I}n_y=\begin{bmatrix}(uh)n_x+(vh)n_y\\ (u^2h+g(\eta^2-2\eta z_\mathrm{b})/2)n_x+(uvh)n_y\\ (uvh)n_x+(v^2h+g(\eta^2-2\eta z_\mathrm{b})/2)n_y\end{bmatrix} \tag{4}$$

and

$$\mathbf{F}^\mathrm{V}=\mathbf{f}^\mathrm{V}n_x+\mathbf{g}^\mathrm{V}n_y=\begin{bmatrix}0\\ (h\,\partial u/\partial x)n_x+(h\,\partial u/\partial y)n_y\\ (h\,\partial v/\partial x)n_x+(h\,\partial v/\partial y)n_y\end{bmatrix} \tag{5}$$

The conservation law formed by the shallow water equations (1) is solved using a Godunov-type finite volume scheme based on adaptive quadtree grids. Second-order accuracy in both time and space is achieved by employing the MUSCL-Hancock method [15], where MUSCL stands for Monotone Upwind Schemes for Conservation Laws. In an explicit scheme, the MUSCL-Hancock method updates flow variables over a time interval *via* predictor and corrector steps.

DOI: 10.1002/fld

From (1), the time-marching formula for updating the cell-centred flow variables over the predictor step is

$$\mathbf{u}_i^{n+1/2}=\mathbf{u}_i^n-\frac{\Delta t}{2}\left[\frac{(\mathbf{F}_{i,\mathrm{E}}-\mathbf{F}_{i,\mathrm{W}})}{\Delta x}+\frac{(\mathbf{F}_{i,\mathrm{N}}-\mathbf{F}_{i,\mathrm{S}})}{\Delta y}-\mathbf{s}_i\right] \tag{6}$$

where subscript i is the cell index; superscript n denotes the time level; Δt is the time step; $\mathbf{F}_{i,\mathrm{E}}$, $\mathbf{F}_{i,\mathrm{W}}$, $\mathbf{F}_{i,\mathrm{N}}$ and $\mathbf{F}_{i,\mathrm{S}}$ are the flux terms evaluated at the east, west, north and south cell interfaces; and Δx and Δy are the cell sizes in the x- and y-directions. During this half time step, the bed slope terms are approximated by central differences. Interface fluxes are determined at the midpoint of each cell face by linear interpolation, such that

$$\mathbf{u}(x,y)=\mathbf{u}_i+\mathbf{r}\nabla\mathbf{u}_i \tag{7}$$

where $\mathbf{r}$ is the distance vector from the cell centre to the point of interest (x,y); and $\nabla\mathbf{u}_i$ is the gradient vector. A slope limiter is used to prevent numerical oscillations in regions of high gradient. The interpolation formula for face values in the x-direction for cell i then becomes

$$\mathbf{u}_{i,\mathrm{W}}=\mathbf{u}_i-\tfrac{1}{2}\boldsymbol{\Psi}(r)(\mathbf{u}_i-\mathbf{u}_\mathrm{w}) \quad\text{and}\quad \mathbf{u}_{i,\mathrm{E}}=\mathbf{u}_i+\tfrac{1}{2}\boldsymbol{\Psi}(r)(\mathbf{u}_i-\mathbf{u}_\mathrm{w}) \tag{8}$$

in which $\boldsymbol{\Psi}(r)$ is the slope limiter, $\mathbf{u}_{i,\mathrm{W}}$ and $\mathbf{u}_{i,\mathrm{E}}$ are the west and east face flow variable vectors of cell i; and $\mathbf{u}_\mathrm{w}$ is the flow variable vector at the centre of the west neighbour of cell i. A minmod slope limiter is chosen because it gives more stable solutions:

$$\Psi(r)=\max[0,\min(r,1)] \tag{9}$$

where r is the ratio of successive gradients [13]. The y-direction face values for cell i can be derived in a similar way. Velocity gradients in the viscous interface fluxes defined by (3) and (5) are approximated using central differences.

In the corrector step, the flow variables are calculated over a full time step, based on flow data from the predictor step. The explicit updating formula is

$$\mathbf{u}_i^{n+1}=\mathbf{u}_i^n-\Delta t\left[\frac{(\mathbf{F}_{i,\mathrm{E}}^{n+1/2}-\mathbf{F}_{i,\mathrm{W}}^{n+1/2})}{\Delta x}+\frac{(\mathbf{F}_{i,\mathrm{N}}^{n+1/2}-\mathbf{F}_{i,\mathrm{S}}^{n+1/2})}{\Delta y}-\mathbf{s}_i^{n+1/2}\right] \tag{10}$$

Inviscid fluxes are calculated using the Harten, Lax, van Leer Contact (HLLC) approximate Riemann solver proposed by Toro *et al.* [16], where the face values of the flow variables are again obtained using slope-limited interpolation formulae. In the x-direction,

$$\mathbf{u}_{i,\mathrm{W}}^{n+1/2}=\mathbf{u}_i^{n+1/2}-\tfrac{1}{2}\boldsymbol{\Psi}(r)(\mathbf{u}_i-\mathbf{u}_\mathrm{w}),\quad \mathbf{u}_{i,\mathrm{E}}^{n+1/2}=\mathbf{u}_i^{n+1/2}+\tfrac{1}{2}\boldsymbol{\Psi}(r)(\mathbf{u}_i-\mathbf{u}_\mathrm{w}) \tag{11}$$

In evaluating $\boldsymbol{\Psi}(r)(\mathbf{u}_i-\mathbf{u}_\mathrm{w})$ in the corrector step, Hu *et al.* [17] find that more accurate results are obtained by using flow information at time level n than at time level $n+\frac{1}{2}$. The y-direction face values are calculated in a similar manner. Details of the HLLC approximate Riemann solver are given by Harten *et al.* [18], Fraccarollo and Toro [19], and Liang *et al.* [20]. Viscous fluxes are calculated by central differencing the velocity components obtained from the predictor step.

On the quadtree grid, the flux calculations are carried out on a locally uniform grid template. Natural neighbour interpolation formulae are used to obtain values for flow variables at appropriate locations so as not to violate the conservation laws [20].

DOI: 10.1002/fld

At open boundaries, gradients of surface water level, normal and tangential velocity components are set to zero. Inflow and outflow conditions are imposed using Riemann invariants set according to the local Froude number (evaluated at the boundary fluid cell under consideration). At solid walls, the normal velocity component and the gradients of the tangential velocity component and surface water level are set to zero.

The scheme is explicit, where its stability property is determined by the Courant–Friedrichs–Lewy (CFL) criterion. Here, the Courant number is set to 0.8.

3. BOUNDARY TREATMENT

In Cartesian grids, staircase approximations are often used for curved or irregular boundaries. Typically, when a solid body is placed in the fluid domain, the background grid is cut by the boundary curve. Fluid cells included in the flow calculation are those cells whose centroid lies within the fluid. Other cells, whose centroid lies within the solid body, are known as solid cells and excluded from the flow calculation. Boundary cells are fluid cells with at least one neighbour sharing a common interface that is a solid cell. Figure 1 shows three configurations commonly encountered when a solid boundary curve cuts a grid cell. In the present work, a similar boundary treatment is applied to each configuration. Consider Case (b) in Figure 1, with the cell notation given in more detail in Figure 2. If the boundary curve happens to be aligned with the common interface of the two cells $C$and G, the slip boundary conditions are simply,

$$\zeta_G = \zeta_C, \quad u_G = -u_C \quad \text{and} \quad v_G = v_C \tag{12}$$

For curved or irregular boundaries, however, the above treatment introduces spurious viscosity, local separation, and spikes in the free surface, to the numerical solutions due to the staircase approximation. For example, Rogers *et al.* [21] discuss near-boundary inaccuracies caused by the staircase grid for the oblique hydraulic jump.

We propose a simple, yet effective method to resolve the problem. For the grid configuration in Figure 2, the fluxes through the mid-point O' of the western cell interface of the boundary fluid cell C have to be evaluated in a finite volume scheme. Point O, which is the nearest boundary point to O', is first sought. The tangent T at point O is at an angle θ anticlockwise from the x-axis. For a slip boundary, the normal velocity component and the gradient of the tangential velocity component are zero at O, and hence

$$u_{\mathrm{N}}|_O = 0 \quad \text{and} \quad \left.\frac{\partial u_{\mathrm{T}}}{\partial N}\right|_O = 0 \tag{13}$$

where the subscripts N and T denote the normal and tangential directions, respectively. Now assume that the boundary point O is moved to O', which has the same tangential and normal

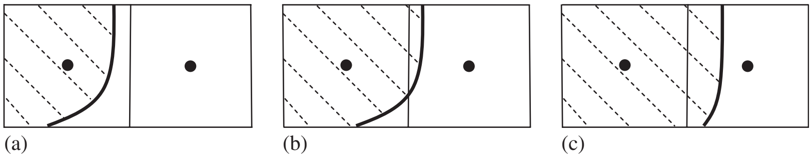

Figure 1. Three different types of boundary configurations.

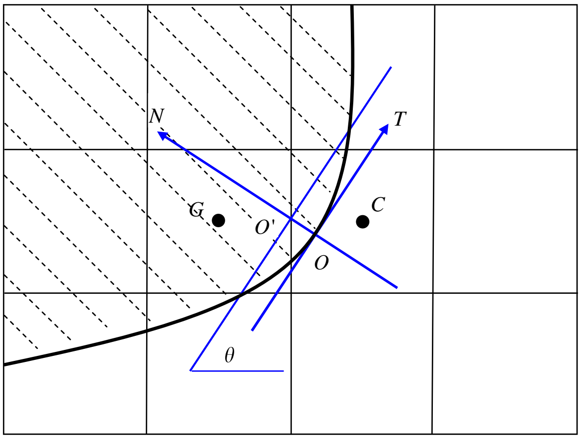

Figure 2. Simple method to treat a curved boundary on Cartesian grids.

directions as O. Hence, the slip boundary conditions at O' can be written as

$$u_{\mathrm{N}}|_{O'}=0 \quad \text{and} \quad \left.\frac{\partial u_{\mathrm{T}}}{\partial N}\right|_{O'}=0 \tag{14}$$

Decomposing the velocities at cells G and C into normal (N) and tangential (T) components, the slip boundary conditions at point O' suggest that

$$\begin{aligned} &\text{N direction: } -u_C\sin\theta+v_C\cos\theta=-(-u_G\sin\theta+v_G\cos\theta)\\ &\text{T direction: } u_C\cos\theta+v_C\sin\theta=u_G\cos\theta+v_G\sin\theta \end{aligned} \tag{15}$$

Therefore, the slip boundary conditions at point O' are given by

$$\begin{aligned} &\zeta_G=\zeta_C, \quad u_G=u_C-2(u_C\sin\theta-v_C\cos\theta)\sin\theta\\ &v_G=v_C+2(u_C\sin\theta-v_C\cos\theta)\cos\theta \end{aligned} \tag{16}$$

To implement the method, θ has to be estimated. For certain simple boundary geometries, such as the arc of a circle, θ can be determined analytically. In general, however, the following numerical method is used to evaluate θ. First, the nearest boundary point (O) to the mid-point (O') of the cell interface under consideration is identified. In the present work, the boundary curve is defined by the set of seeding points used in the generation of the background quadtree grid. After the nearest seeding point (denoted by the index i) to the mid-point O' has been located, the local tangent is approximated by the line connecting the two neighbouring seeding points. The value of θ is therefore estimated from

$$\theta=\tan^{-1}\frac{y_{\mathrm{s}}(i+1)-y_{\mathrm{s}}(i-1)}{x_{\mathrm{s}}(i+1)-x_{\mathrm{s}}(i-1)} \tag{17}$$

where x_{s} and y_{s} are the coordinates of the seeding points. Obviously, the use of more finely resolved seeding points improves both the accuracy of the boundary representation and that of θ. Values of θ are calculated immediately after generating the quadtree grid and stored for use throughout

DOI: 10.1002/fld

the simulation. During grid generation, the boundary cells are generated at the finest subdivision level in order to obtain an accurate representation of the solid boundary geometry. The boundary cells are fixed during grid adaptation, and hence the values of θ remain constant during the entire simulation. Hence, the method is computationally efficient when implemented on a dynamically adaptive quadtree grid and can be used directly on a general Cartesian solver.

A key step in the above curved boundary treatment is the shift of the actual boundary point to the mid-point of the cell interface under consideration, which could lead to local inaccuracy near boundaries. However, the validation results in Section 4 demonstrate that the errors introduced by the shifts are small and do not affect the overall accuracy of the global solution. This is especially the case for a quadtree grid, where cells of the finest resolution approximate solid boundaries. The present simple boundary method can be also extended to more complicated boundary conditions. For example the log-style boundary condition presented in Zedler and Street [22] can be utilized by replacing the equations for the normal gradient of tangential velocity component given by (13) and (14) with that for a log-law Neumann boundary condition. However, the merit of the current method vanishes when dealing with no-slip boundary conditions as solutions to Equation (14) give the same values as those from a staircase approximation.

4. RESULTS

In all cases, $\rho = 1000\,\mathrm{kg/m^3}$, $g = 9.81\,\mathrm{m/s^2}$, and the Coriolis effect is ignored. When dynamic grid adaptation is used, the criteria for grid enrichment and coarsening are based on the r.m.s. free surface gradient,

$$\Theta = \sqrt{\left(\frac{\partial \zeta}{\partial x}\right)^2 + \left(\frac{\partial \zeta}{\partial y}\right)^2} \tag{18}$$

A cell is subdivided into four sub-cells, whenever Θ is greater than a prescribed upper threshold value and the subdivision level of the cell is less than the set maximum. When the values of Θ of the four children of a parent cell are all smaller than a prescribed lower threshold limit and the subdivision level of the child cells is greater than the set minimum, the grid is locally coarsened by removing the four child cells. At present, the adaptation criteria based on the values of Θ are determined by trial and error. However, a more general method is worth future investigation.

4.1. Still water in a basin of complicated geometry

We first consider still water at steady state in a basin of complicated geometry in order to examine the ability of the balanced shallow water solver to handle those source terms related to spatially varying bed slopes. The test was suggested by Goutal and Maurel [23] at a Workshop on Dam-break Wave Simulation and relates to a 1500 m long channel whose geometry is described by Table I and Figure 3. The boundaries are all slip, with the irregular shape of the lateral walls modelled using the method outlined in Section 3. The initial conditions are that the discharge is zero everywhere and the free surface is horizontal, with its level 12 m above the $z = 0$ datum indicated in Figure 3(b). The water is therefore at rest everywhere in the channel, and without any external momentum input, the flow should remain motionless. However, the test is a stern one of a Godunov-type solver, because the bed slopes are variable, and without damping present from viscous effects or bed friction, a numerical scheme that does not exactly balance the contributions

DOI: 10.1002/fld

Table I. Still water test: description of the channel geometry.

X (m)	z_b (m)	Width (m)
0	0	40
50	0	40
100	2.5	30
150	5	30
200	5	20
250	5	30
300	3	30
350	5	25
400	5	25
425	7.5	30
435	8	35
450	9	35
470	9	40
475	9	40
500	9.1	40
505	9	45
530	9	50
550	6	50
565	5.5	45
575	5.5	40
600	5	40
650	4	30
700	3	40
750	3	40
800	2.3	5
820	2	40
900	1.2	35
950	0.4	25
1000	0	40
1500	0	40

from source and flux gradient terms would generate spurious fluxes and become unstable (see e.g. [24]). Simulations on uniform and non-uniform quadtree grids of different subdivision levels from 4 to 7 were undertaken until $t = 200$ s. The water invariably remained stationary, confirming that the numerical scheme correctly handles flux gradient and bed source terms when the domain has irregular topography. Figure 4 shows the results obtained on a quadtree grid with highest and lowest subdivision levels of 7 and 4, with the finest mesh produced in areas near the domain boundaries.

4.2. Oblique hydraulic jump

The second case concerns the steady hydraulic jump that develops in an open channel, when supercritical inviscid flow is deflected by an inclined wall. The flow domain is $40\,\text{m} \times 30\,\text{m}$ in plan with a frictionless and horizontal bed. The upstream supercritical flow is directed from left to right and has a uniform depth of 1 m and velocity of 8.57 m/s corresponding to a Froude number of about 2.74. At a distance of 10 m from the inflow boundary, the southern channel wall inclines inwards at an angle of 8.95° to the x-direction, causing a hydraulic jump to occur. This problem

DOI: 10.1002/fld

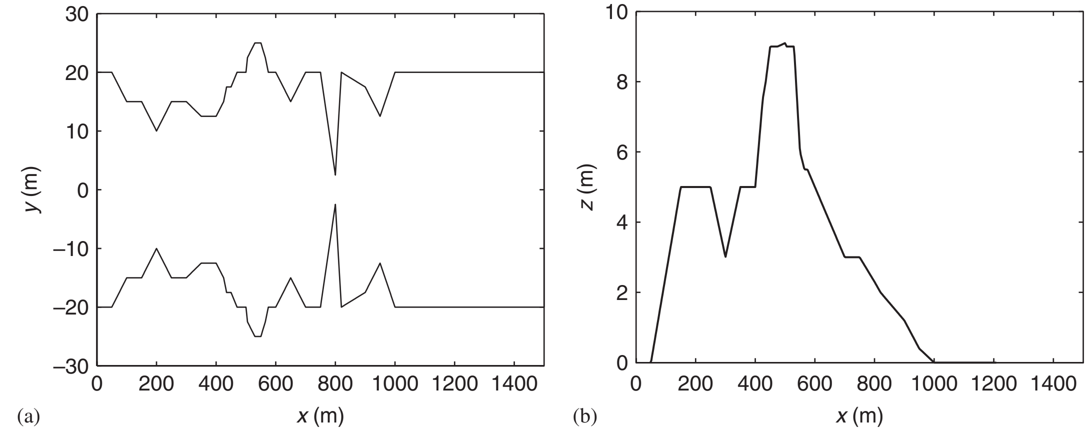

Figure 3. Still water test: channel geometry; (a) plan view and (b) side elevation.

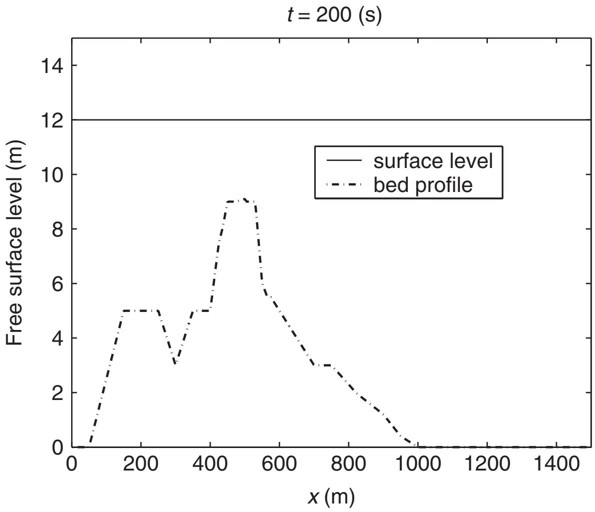

Figure 4. Still water test: free surface elevation along the channel at $t = 200$s.

has an analytical solution, whereby the jump is oriented across the domain at an angle of 30° to the x-direction, and the water depth increases abruptly from 1 to 1.5 m across the jump.

In the numerical model, slip conditions are imposed at the north and south walls. At the western inflow boundary, the velocity is fixed at $u = 8.57\,\mathrm{m/s}$. At the eastern outflow boundary transmissive conditions are applied (see e.g. [25]). Bed friction and viscosity have zero values. An initial quadtree grid of 1357 leaf cells is generated about boundary seeding points, with highest and lowest subdivision levels of 7 and 4. The finest cells at level 7 have a side dimension equal to 0.3125 m, which is $\frac{1}{128}$ of the unit square side dimension (40 m). In this case, dynamic grid adaptation is based on free surface criteria with thresholds of $\Theta = 0.1$ and 0.08 used for grid enrichment and

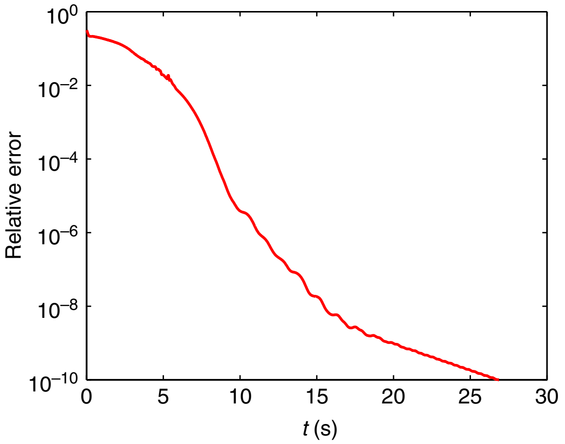

Figure 5. Oblique hydraulic jump: temporal convergence history.

coarsening, respectively. A converged solution is assumed to have been reached when the relative error,

$$R=\sqrt{\sum_i \left(\frac{h_i^n-h_i^{n-1}}{h_i^n}\right)^2} \tag{19}$$

is less than 1.0×10^{-10}. Figure 5 plots R against time, indicating the temporal convergence history, where steady state is estimated to be achieved by 27 s ($R<1.0\times10^{-10}$).

Figure 6 shows the predicted steady-state surface water level contours, 3D water surface, and adapted quadtree grid that consists of 2356 cells. The oblique jump is sharply captured, with its position and height in excellent agreement with the analytical solution. In Figure 6(a) the dashed line indicates the analytical location of the jump. Near the inclined boundary wall, the contour lines and 3D surface are properly defined in keeping with the flow physics, again confirming the effectiveness of the simple boundary treatment proposed in Section 3. Figure 7 illustrates the predicted velocity field in the entire domain and in the region near the inclined boundary. The flow upstream and downstream of the jump is directed parallel to the lateral boundary.

The effect on accuracy of the boundary treatment is ascertained using the L_1 norm of the error, e, defined as

$$e=\frac{\sum_i |h_i-\tilde{h}_i|}{\sum_i \tilde{h}_i} \tag{20}$$

where h_i and $\tilde{h}_i$ are the predicted and analytical water depths at cell i. L_1 errors are computed on uniform quadtree grids of subdivision levels 4–8 for all the grid cells and also for a subset of boundary cells near the inclined wall. Figure 8 gives a plot of $\ln(e)$ against $\ln(\mathrm{d}x)$. The cell size is $\mathrm{d}x=x_l/2^{\mathrm{lev}}$ where $x_l=40\,\mathrm{m}$ is the length of the domain and lev is the grid subdivision level. Both series of L_1 errors lie on straight lines of slope slightly greater than 1, indicating that the order of accuracy of the scheme is about 1 both globally and locally near the boundary. This is to be expected because the otherwise second-order accurate solution is smeared by the shock-like

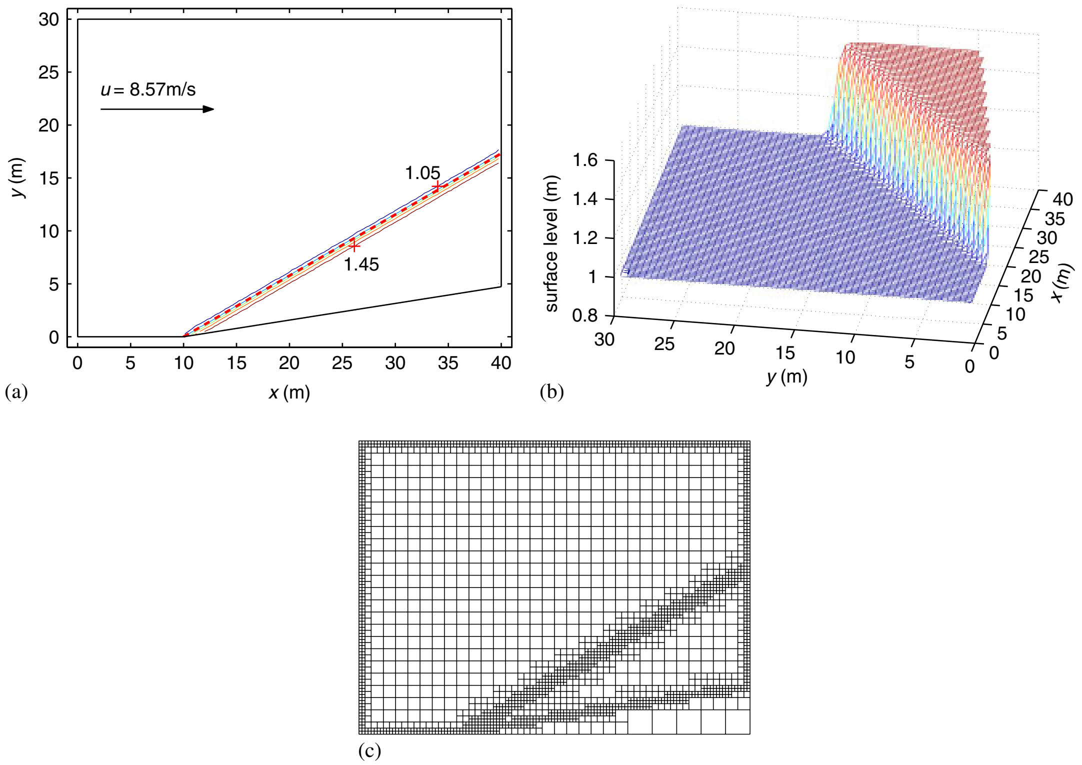

Figure 6. Oblique hydraulic jump: predicted results on adaptive quadtree grid; (a) surface water level contours, (b) 3D water surface and (c) adapted quadtree grid.

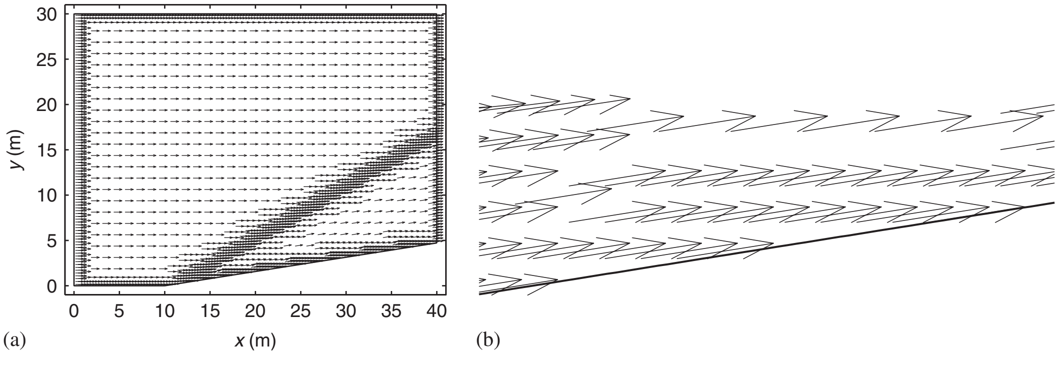

Figure 7. Oblique hydraulic jump: velocity vectors; (a) entire domain and (b) near the inclined wall.

flow discontinuity at the hydraulic jump. Similarly, the smoothing effect on the flow of the present simple boundary treatment again contributes to the numerical solution being first-order accurate. In general applications, the present boundary treatment should maintain global second-order accuracy, in accordance with the findings of Tseng and Ferziger [8].

DOI: 10.1002/fld

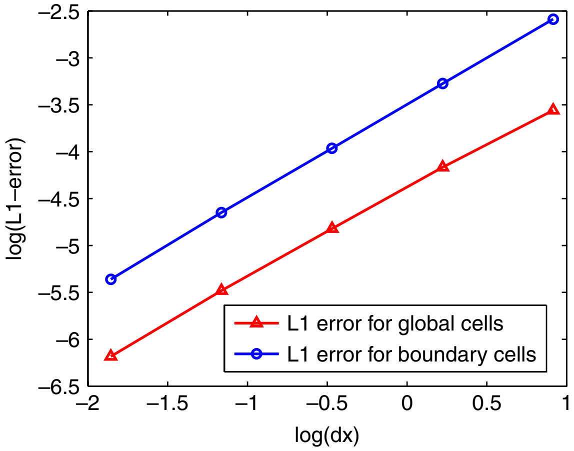

Figure 8. Oblique hydraulic jump: error analysis.

Figure 9 shows the results obtained using the same adaptive quadtree grid but without boundary treatment. The inclined boundary is instead approximated as a Cartesian staircase. From the surface water level contours in Figure 9(a), it is obvious that flow spikes appear near the inclined solid boundary. The location of the jump is also wrongly predicted. From the close-up view of the velocity vectors near the inclined wall in Figure 9(c), numerical dissipation arises in the near wall region where the velocity vectors are smaller and not aligned with the boundary. Rogers *et al.* [21] also present similar findings. Taken overall, this provides further evidence that the simple boundary treatment suggested in Section 3 is a very useful adjunct to the Cartesian grid-based shallow flow models.

4.3. *Wind-induced circulation in a circular shallow lake*

Kranenburg [26] has derived an analytical approximation to wind-induced flow in a circular lake with axially symmetric bed topography by simplifying the momentum equations and fitting a stream function solution representing the steady-state flow pattern. The bed elevation of Kranenburg's lake is given by

$$z_{\mathrm{b}}=\left(\frac{1}{2}\right)^{1/2}H\left(1-\left(1-\frac{r}{R_0}\right)^{1/2}\right) \tag{21}$$

where r is the distance from the basin centre, $H=0.5\,\mathrm{m}$ is a weighted averaged water depth, and $R_0=120\,\mathrm{m}$ is the radius of the lake. Figure 10(a) illustrates the basin bathymetry at a cross-section passing through the centre of the lake. The water is initially quiescent, and has surface elevation,

$$\eta=H\left(\tfrac{1}{2}+\sqrt{\tfrac{1}{2}}\right) \tag{22}$$

A uniformly distributed wind shear of constant magnitude $\tau_{\mathrm{w}}=0.002\,\mathrm{N/m^2}$ is then applied. Bed stresses are estimated from

$$\tau_{\mathrm{b}x}=\rho C_f u\sqrt{u^2+v^2} \quad \text{and} \quad \tau_{\mathrm{b}y}=\rho C_f v\sqrt{u^2+v^2} \tag{23}$$

DOI: 10.1002/fld

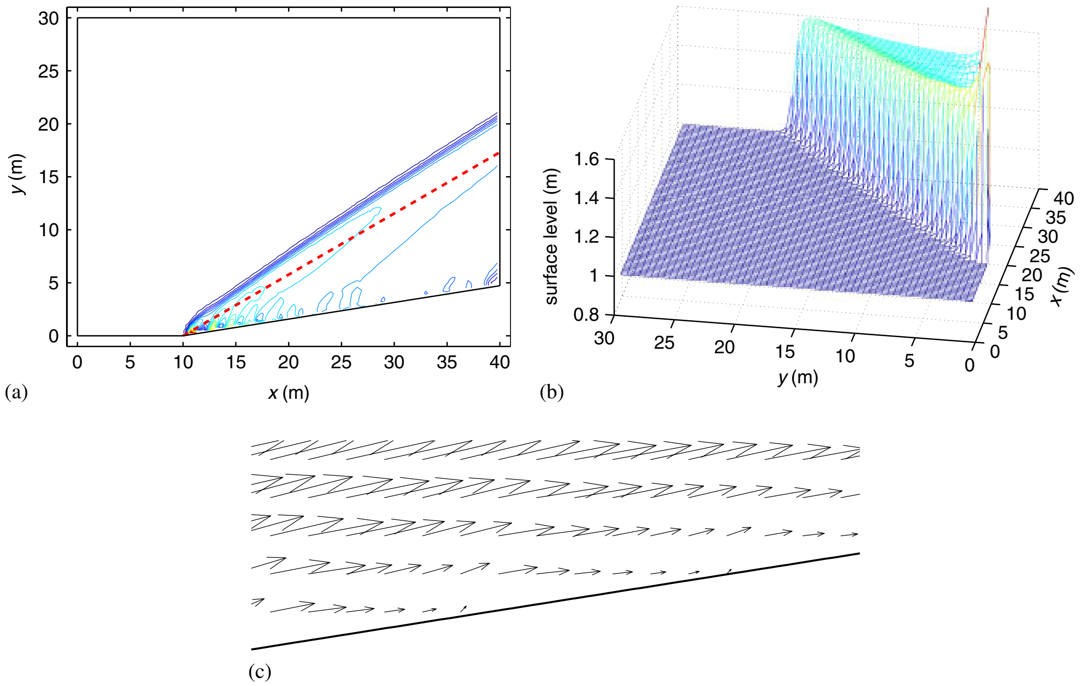

Figure 9. Oblique hydraulic jump: results with staircase boundary treatment; (a) surface water level contours, (b) 3D water surface, and (c) velocity vectors near inclined wall.

The bed friction coefficient is determined as

$$C_f = \left[\frac{\kappa}{1+\ln(z_0/h)}\right]^2 \tag{24}$$

in which $\kappa = 0.4$ is the von Kármán constant, and $z_0 = 2.8\,\text{mm}$ is the roughness height. The depth-averaged eddy viscosity coefficient is

$$\varepsilon = \tfrac{1}{6}\kappa u_* h \tag{25}$$

where $u_* = \sqrt{\tau_\mathrm{w}/\rho}$ is the friction velocity at the free surface. Slip boundary conditions are imposed at the solid wall at the lake perimeter, in accordance with Kranenburg's model. Then under these conditions, the shallow water equations are solved with the circular basin discretized on the quadtree grid shown in Figure 10(b). The quadtree grid has subdivision levels between 6 and 8 inclusive, and 2796 cells. A fixed grid is used because the flow remains smoothly varying as it approaches steady state, which is assumed to occur at $t = 6\,\text{h}$.

Figure 11(a) presents the steady-state flow pattern due to wind blowing from the north-west. A symmetric pair of recirculating gyres occupies the land, with the flow opposing the wind as it crosses the deep water in the middle of the lake. The pair of gyres may be explained using the vorticity equation (see [27, 28]). It should be noted that the direction of rotation of the gyres is consistent with the shallow water equations, but would be opposite if the full three-dimensional Reynolds-averaged Navier–Stokes equations were to be solved. The fastest velocities occur in the

DOI: 10.1002/fld

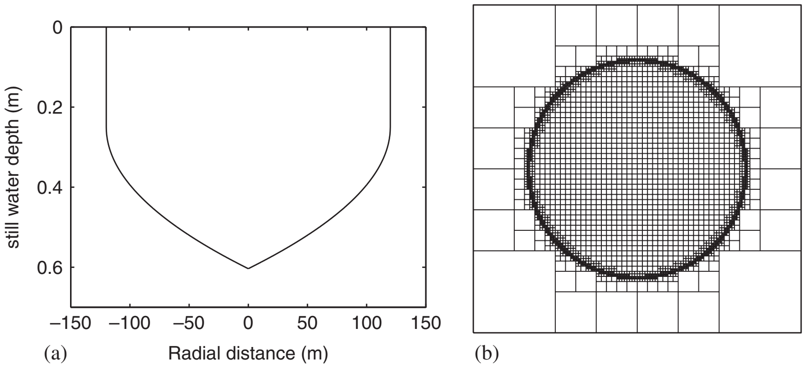

Figure 10. Wind-induced flow in a circular lake: geometry and quadtree grid; (a) cross-section and (b) top view.

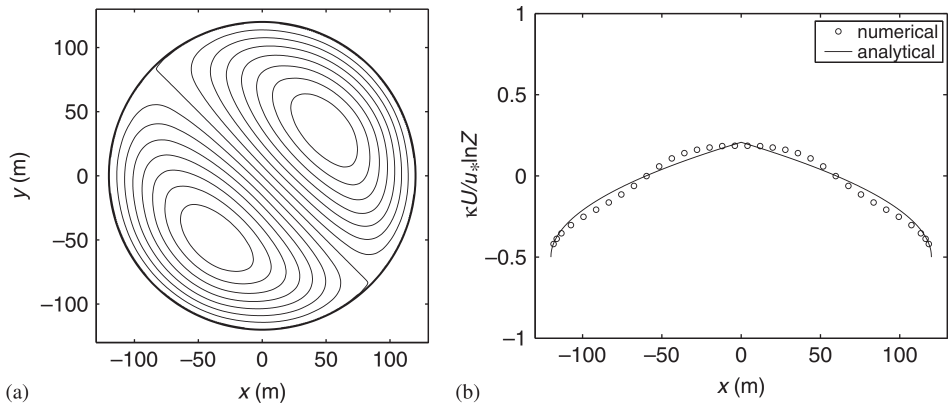

Figure 11. Wind-induced flow in a circular lake: steady-state results; (a) streamlines and (b) normalized velocity profile.

shallowest zone near the perimeter wall, away from the stagnation points that are located at the ends of the dividing streamline. The centres of rotation of the predicted gyres are slightly downwind of the northeast–southwest axis of the lake, in agreement with Rogers *et al.* [21]. The numerical prediction and analytical approximation [26] of the normalized depth-averaged velocity profile $\kappa U/(u_* \ln Z)$ across the lake along an axis normal to the wind direction are in agreement (Figure 11(b)). Note that $U = u_*(h/H-1)\ln(Z)/\kappa$ for the analytical approximation and $U=(u+v)/\sqrt{2}$ for the numerical prediction, with $Z = H/z_0$. The results provide further validation of the present numerical model with respect to the bed slope, bed friction, surface stress, and viscous flux terms in the shallow water equations. The simple boundary method is also further justified as no distortion of streamlines is found near the lake perimeter wall.

In order to assess further the accuracy associated with the simple boundary treatment method, this test case has been simulated on uniform quadtree grids of subdivision levels of 6–9 and the global L_1 error is estimated, based on the reference solution on a level 9 grid. Figure 12 plots the L_1 error obtained for the different quadtree grids. On the level 6 coarse grid, the numerical results predict an L_1 error that is smaller than expected and the solution fails to converge. However, on a

DOI: 10.1002/fld

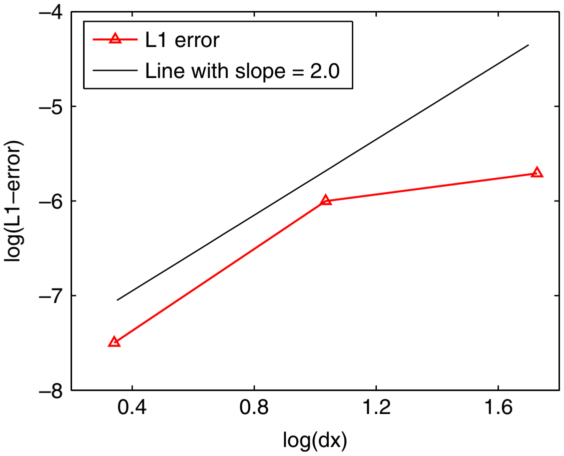

Figure 12. Wind-induced flow in a circular lake: error analysis.

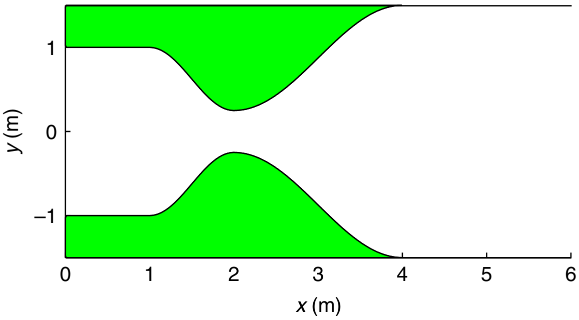

Figure 13. Shock diffraction in a contraction–expansion channel: channel geometry.

sufficiently fine grid (at least level 7), the accuracy of the numerical scheme converges to second order. This confirms that the present simple boundary treatment method translates into a globally second-order accurate scheme, despite its simplicity.

4.4. Shock diffraction in a contraction–expansion channel

Figure 13 depicts the side walls of a flat, frictionless, and rectangular open channel considered by Causon *et al*. [12] in the context of a Cartesian cut cell-based shallow flow model. The channel is symmetric about $y=0$, with its upper wall defined by

$$y=\begin{cases} 1.0 & 0\leqslant x\leqslant 1 \\ -0.375\cos(\pi(x-2))+0.625 & 1<x\leqslant 2 \\ -0.625\cos\left(\frac{\pi}{2}(x-2)\right)+0.875 & 2<x\leqslant 4 \\ 1.5 & 4<x\leqslant 6 \end{cases} \tag{26}$$

DOI: 10.1002/fld

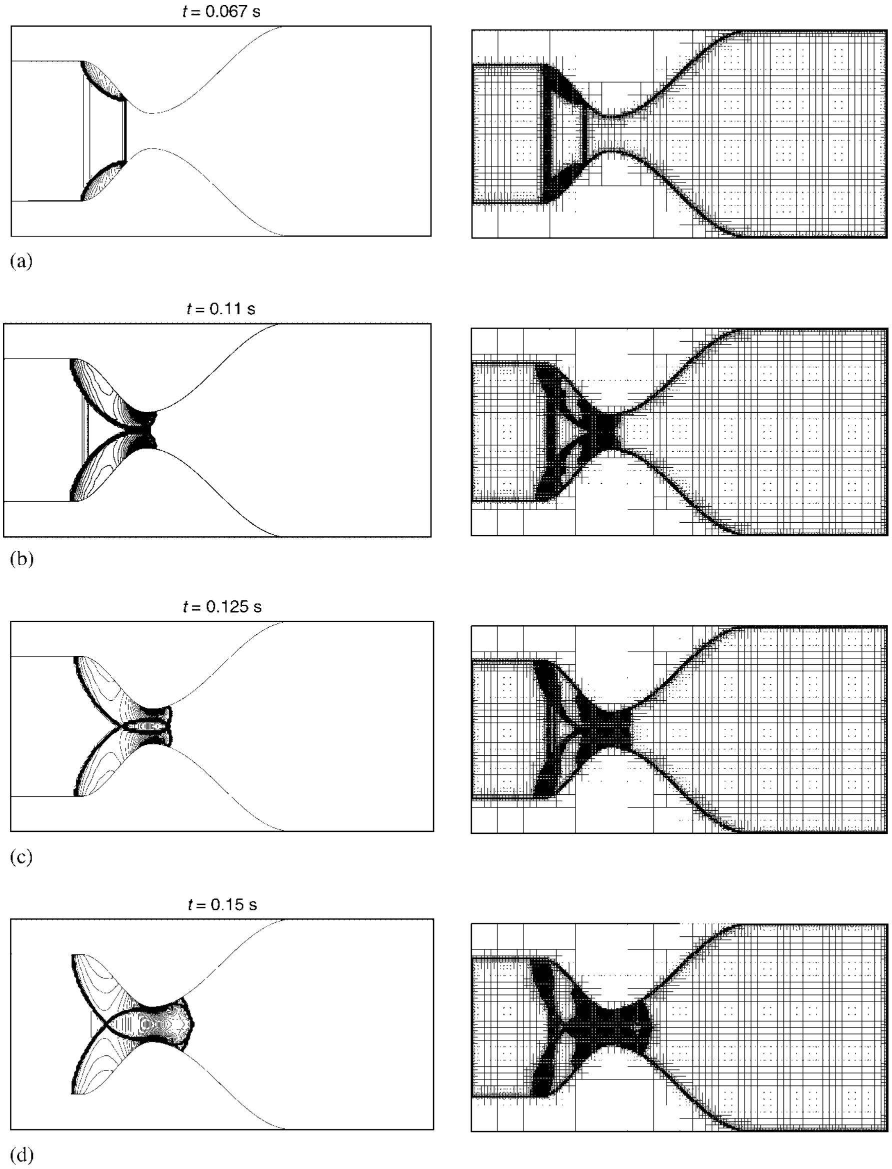

Figure 14. Shock diffraction in a contraction–expansion channel: surface water level contours and adapted quadtree grids at different times.

The channel contracts smoothly from an upstream straight section of width 2 m to a throat section of width 0.5 m, after which it expands smoothly to a downstream straight section of width 3 m. A shock-like bore is introduced at the entrance to the curved constriction, with initial conditions

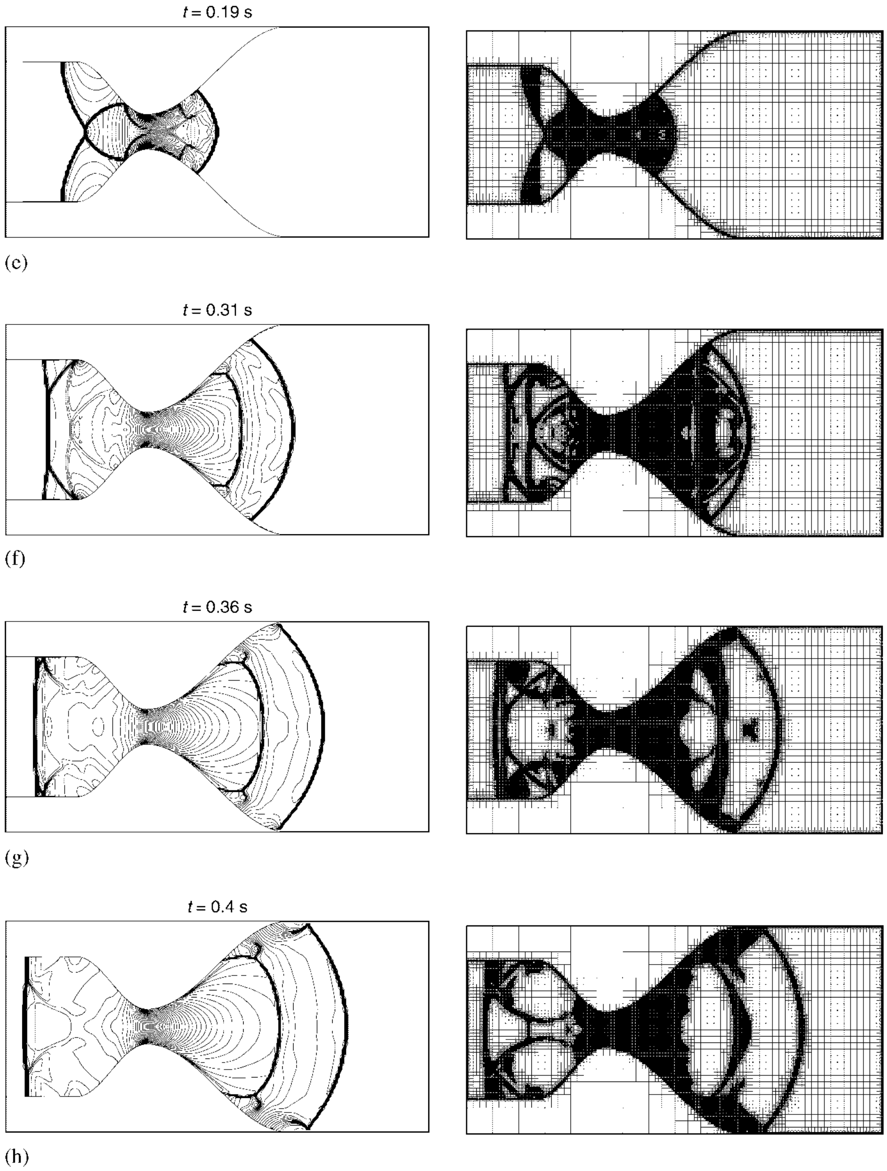

Figure 14. *Continued.*

determined from [12]

$$
\begin{aligned}
h_{\mathrm{L}} &= l h_{\mathrm{R}}\\
u_{\mathrm{L}} &= c+(u_{\mathrm{R}}-c)/l \qquad (27)\\
v_{\mathrm{L}} &= v_{\mathrm{R}}
\end{aligned}
$$

DOI: 10.1002/fld

where c is the speed of the shock wave, subscripts L and R represent the left and right sides of the shock wave, and

$$l=\frac{1}{2}\left(\sqrt{1+8(Fr_{\mathrm{R}}-Fr_{\mathrm{S}})^2}-1\right) \tag{28}$$

in which $Fr_{\mathrm{R}}=u_{\mathrm{R}}/\sqrt{gh_{\mathrm{R}}}$ is the Froude number of the flow before the shock wave arrives, $Fr_{\mathrm{S}}=3$ is the prescribed Froude number of the shock and $c=Fr_{\mathrm{S}}\sqrt{gh_{\mathrm{R}}}$. On the right-hand side of the shock, the water is initially at rest with a depth of 1 m.

Slip boundary conditions are imposed at the channel walls. The left and right ends of the channel are transmissive. An initial quadtree grid with highest and lowest subdivision levels of 9 and 6, respectively, is generated with the finest mesh near the boundaries and in the vicinity of the initial shock. For grid adaptation, the upper and lower limits of Θ are prescribed to be 1.4 and 1.2. Figure 14 presents the surface water level contours and corresponding adapted quadtree grid at times $t=0.067, 0.11, 0.125, 0.15, 0.19, 0.31, 0.36,$ and 0.4 s. The contours illustrate the complex wave patterns that develop due to bore diffraction and repeated reflections at the channel walls as the incident shock-like wave propagates along the channel. The evolution of the adaptive quadtree grid is consistent with the developing wave patterns. The numerical predictions shown in Figure 14 are very similar to the sequences of depth contours obtained by Yang and Hsu [29] and Causon *et al.* [12].

5. CONCLUSIONS

This paper presents a simple method for representing curved or irregular boundaries in Cartesian grid numerical models. Fictional boundary values are calculated from the flow variables in each boundary cell of interest. There is no requirement for complicated interpolation or specific generation of cut cells, and hence the method hardly increases the computational overhead of an existing model. This boundary method is a major improvement on staircase-type boundary approximations, which introduce large near-boundary errors.

The boundary treatment is applied to Cartesian quadtree grids on which the hyperbolic shallow water equations are solved using a Godunov-type HLLC solver with MUSCL-Hancock time integration. The governing equations are presented in a form that naturally balances the flux gradient and source terms, for steep-fronted flow over non-uniform terrain. Validation is undertaken for still water in a basin of complicated geometry, oblique hydraulic jump in a channel with a convergent side wall, the steady-state wind-induced flow pattern in a circular lake of concentrically varying bathymetry, and the diffraction and repeated reflection of a bore wave in a rectangular channel with a contraction and expansion. In all cases, the results are in agreement with other theoretical solutions, and confirm that the model properly represents curved and irregular boundaries, while also correctly balancing flux gradient and source terms in the hyperbolic governing equations. Numerical experiments indicate that the boundary treatment is locally first-order accurate for an oblique hydraulic jump. It is reasonable to expect that the method should achieve near second-order global accuracy, when applied to practical shallow flow simulations. By combining the boundary treatment with a dynamically adaptive quadtree grid, it is feasible to obtain accurate predictions of complicated shallow flows, such as large-scale flood inundation, at reasonable computational cost. This simple boundary treatment can be directly implemented in a general Cartesian model.

DOI: 10.1002/fld

By modifying the model to include the Coriolis force, etc., the method should also be useful for simulating circulation patterns in estuaries and bays where the coastline is naturally complicated. It would be interesting to compare the results of such simulations with those of Csanady [27] and Sankaranarayanan [28], for example.

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. Fujihara M, Borthwick AGL. Godunov-type solution of curvilinear shallow-water equations. *Journal of Hydraulic Engineering* (ASCE) 2000; **126**(11):827–836.
3. 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.
4. 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.
5. 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.
6. 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.
7. Yang G, Causon DM, Ingram DM, Saunders R, Batten P. A cartesian cut cell method for compressible flows. Part A: static body problems. *Aeronautical Journal* 1997; **101**(1002):47–56.
8. Tseng Y-H, Ferziger JH. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics* 2003; **192**:593–623.
9. 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.
10. Turnbull MS, Borthwick AGL, Eatock Taylor R. Wave–structure interaction using coupled structured–unstructured finite element meshes. *Applied Ocean Research* 2003; **25**(2):63–77.
11. Sabersky RH, Acosta AJ, Hauptmann EG. *Fluid Flow* (3rd edn). Macmillan: New York, 1989; 230.
12. 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.
13. Liang Q, Zang J, Borthwick AGL, Taylor PH. Shallow flow simulation on dynamically adaptive cut-cell quadtree grids. *International Journal for Numerical Methods in Fluids* 2007; **53**:1777–1799.
14. Liang Q, Borthwick AGL. Adaptive quadtree simulation of shallow flows with wet–dry fronts over complex topography. *Computers and Fluids* 2007, accepted.
15. van Leer B. On the relation between the upwind-differencing schemes of Godunov, Enguist–Osher and Roe. *SIAM Journal on Scientific and Statistical Computing* 1984; **5**:1–20.
16. Toro EF, Spruce M, Speares W. Restoration of the contact surface in the HLL Riemann solver. *Shock Waves* 1994; **4**:25–34.
17. Hu K, Mingham CG, Causon DM. Numerical simulation of wave overtopping of coastal structures using the non-linear shallow water equations. *Coastal Engineering* 2000; **41**(4):433–465.
18. Harten A, Lax PD, van Leer B. On upstream differencing and Godunov-type schemes for hyperbolic conservation-laws. *SIAM Review* 1983; **25**(1):35–61.
19. Fraccarollo L, Toro EF. Experimental and numerical assessment of the shallow water model for two-dimension dam-break type problems. *Journal of Hydraulic Research* 1995; **33**(6):843–863.
20. 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.
21. Rogers BD, 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.
22. Zedler EA, Street RL. Sediment transport over ripples in oscillatory flow. *Journal of Hydraulic Engineering* (ASCE) 2006; **132**(2):180–193.
23. Goutal N, Maurel F (eds). *Proceedings of the 2nd Workshop on Dam-Break Wave Simulation*. HE 43/97/016/B, Départment Laboratoire National d'Hydraulique, Groupe Hydraulique Fluviale Electricité de France, France, 1997.

DOI: 10.1002/fld

24. 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.
25. Toro EF. *Shock-Capturing Methods for Free-Surface Shallow Flows*. Wiley: Chichester, 2001.
26. Kranenburg C. Wind-driven chaotic advection in a shallow model lake. *Journal of Hydraulic Research* 1992; **30**(1):29–46.
27. Csanady GT. Barotropic currents over the continental shelf. *Journal of Physical Oceanography* 1974; **4**:357–371.
28. Sankaranarayanan S. Modeling the tide and wind–induced circulation in Buzzards Bay Estuarine. *Coastal and Shelf Science* 2007; **73**(3-4):467–480.
29. Yang JY, Hsu CA. Computation of free surface flows. *Journal of Hydraulic Research* 1993; **31**(3):403–413.

DOI: 10.1002/fld