INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 24, 1225–1245 (1997)

SOLUTION OF THE 2D SHALLOW WATER EQUATIONS USING THE FINITE VOLUME METHOD ON UNSTRUCTURED TRIANGULAR MESHES

K. ANASTASIOU* AND C. T. CHAN

Department of Civil Engineering, Imperial College of Science, Technology and Medicine, Imperial College Road, London SW7 2BU, U.K.

SUMMARY

A 2D, depth-integrated, free surface flow solver for the shallow water equations is developed and tested. The solver is implemented on unstructured triangular meshes and the solution methodology is based upon a Godunov-type second-order upwind finite volume formulation, whereby the inviscid fluxes of the system of equations are obtained using Roe's flux function. The eigensystem of the 2D shallow water equations is derived and is used for the construction of Roe's matrix on an unstructured mesh. The viscous terms of the shallow water equations are computed using a finite volume formulation which is second-order-accurate. Verification of the solution technique for the inviscid form of the governing equations as well as for the full system of equations is carried out by comparing the model output with documented published results and very good agreement is obtained. A numerical experiment is also conducted in order to evaluate the performance of the solution technique as applied to linear convection problems. The presented results show that the solution technique is robust. © 1997 by John Wiley & Sons, Ltd.

Int. J. Numer. Meth. Fluids, **24**: 1225–1245, 1997.

No. of Figs: 11. No. of Tables: 3. No. of Refs: 21.

KEY WORDS: free surface flow; unstructured triangular mesh; Roe's matrix; upwind finite volume method

1. INTRODUCTION

In recent years, algorithms for solving the Euler and Navier–Stokes equations on unstructured meshes have gained in popularity and impressive results have been reported.[1–6] By exploiting the high flexibility of unstructured meshes and by adopting a solution methodology based on the finite volume method, the governing systems of equations can be solved on a Cartesian co-ordinate system without having to use co-ordinate transformations; moreover, discontinuous flows such as those associated with shock propagation can be successfully simulated. In terms of versatility the finite volume method as implemented on unstructured meshes compares favourably with solution techniques based on structured meshes or the finite element method.

A very useful application of the above solution methodology in problems involving environmental flows would be to solve the system of 2D shallow water equations used to describe free surface flow in shallow water bodies. Recently, Zhao *et al.*[7] have reported a solution of the inviscid 2D shallow water equations using Osher's scheme[8] on an unstructured mesh. The scheme they used is first-order-accurate and as a result it cannot perform satisfactorily in problems involving contact discontinuities and oblique shocks.[9] Alcrudo and Garcia-Navarro[10] have also solved the inviscid 2D shallow water

* Correspondence to: K. Anastasiou.

CCC 0271–2091/97/111225–21 $17.50

Received April 1996
Revised September 1996

equations using a second-order-accurate scheme on quadrilateral body-fitted grids using Roe's scheme and have reported impressive results for rapidly varying inviscid flow.

The present work reports on the construction and testing of a numerical model of the full system of depth-integrated 2D shallow water equations which has the following attributes.

1. It is able to handle problem domains of arbitrary complexity.
2. It is able to simulate steady or unsteady subcritical or supercritical flow.
3. It is able to simulate discontinuous flows such as those associated with the classical dam break problem or an oblique hydraulic jump.

The present numerical model is second-order-accurate and is based on a cell-centred finite volume upwind scheme implemented on a unstructured triangular mesh. The description of the flow variables over each triangular cell is based upon the technique proposed by Barth and Jespersen.[4] In addition, Roe's flux function is used for the evaluation of the inviscid fluxes at the triangular cell faces, assuming a local Riemann problem in the direction normal to a cell face. The eigensystem required by Roe's flux function is derived from the flux Jacobian matrix with the x-axis projected normal to the cell face. The viscous terms are computed in a manner similar to that proposed by Pan and Cheng,[5] which is second-order-accurate.

Section 2 of the paper describes the finite volume discretization of the 2D shallow water equations on an unstructured triangular mesh. A detailed account of the formulation of the inviscid and viscous fluxes is given in Section 3. In Section 4 a description of the limiter used to minimize numerical oscillations around solution discontinuities is given and the boundary conditions used are discussed. The time integration of the discretized equations is briefly described in Section 6. Results for rapidly varying 2D flows, a steady flow over a step and a jet-forced flow in a reservoir are presented and discussed in Section 6 for the purpose of validating the numerical model. In the same section a numerical experiment is conducted which aims at evaluating the performance of the scheme as applied to linear convection problems. Two benchmark tests are carried out and the presented results show that the present numerical scheme performs satisfactorily.

2. FINITE VOLUME DISCRETIZATION OF 2D SHALLOW WATER EQUATIONS ON AN UNSTRUCTURED MESH

The integral form of the system of 2D shallow water equations can be written as

$$\frac{\partial}{\partial t}\int\int_{\Omega} Q\,\mathrm{d}\Omega + \oint_{S} \mathbf{F}\cdot\mathbf{n}\,\mathrm{d}S = \int\int_{\Omega} H\,\mathrm{d}\Omega, \tag{1}$$

where Ω is the domain of interest, S is the boundary surrounding Ω, $\mathbf{n}$ is the normal vector to S in the outward direction, Q is the vector of conserved variables, $\mathbf{F}$ is the vector of flux functions through S and H is the vector of forcing functions. Q, $\mathbf{F}$ and H are given as

$$\mathbf{F}\cdot n = F^{\mathrm{I}} - F^{\mathrm{V}} = (f^{\mathrm{I}} - vf^{\mathrm{V}})n_x + (g^{\mathrm{I}} - vg^{\mathrm{V}})n_y,$$

$$Q = \begin{bmatrix} h \\ uh \\ vh \end{bmatrix}, \qquad f^{\mathrm{I}} = \begin{bmatrix} uh \\ u^2h + gh^2/2 \\ uvh \end{bmatrix}, \qquad g^{\mathrm{I}} = \begin{bmatrix} vh \\ uvh \\ v^2h + gh^2/2 \end{bmatrix}, \tag{2}$$

$$f^{\mathrm{V}} = \begin{bmatrix} 0 \\ hu_x \\ hv_x \end{bmatrix}, \qquad g^{\mathrm{V}} = \begin{bmatrix} 0 \\ hu_y \\ hv_y \end{bmatrix}, \qquad H = \begin{bmatrix} 0 \\ -gh(Sf_x + S_{\mathrm{o}x}) + hC_{\mathrm{f}}v \\ -gh(Sf_y + S_{\mathrm{o}y}) - hC_{\mathrm{f}}u \end{bmatrix},$$

where the superscripts I and V denote the inviscid and viscous fluxes respectively, n_x and n_y denote the components of the normal vector $\mathbf{n}$, u_x, u_y and v_x, v_y are the derivatives of the velocity components in the x- and y-direction respectively, h is the water depth, u and v are the velocity components, g is the gravitational acceleration, (Sf_x, Sf_y) represent the bottom friction terms, (S_{ox}, S_{oy}) are the bed slope terms, C_f is the Coriolis parameter and v is the eddy viscosity. The bottom friction term Sf is described by $h^{-1}C^{-2}v|V|$, where C is the Chẽzy coefficient.

The problem domain is first discretized into a set of triangular cells forming an unstructured computational mesh. The average of conserved variables is stored at the centre of each cell and the edges of each cell define the faces of a triangular control volume. For each triangular control volume, equation (1) is written as

$$\frac{\partial Q_i V_i}{\partial t} = -\oint_{\partial C_i} \mathbf{F} \cdot \mathbf{n} \mathrm{d}S + H_i V_i = -R(Q_i), \tag{3}$$

where Q_i and H_i are the average quantities of cell i stored at the cell centre and ∂C_i and V_i denote the boundary and area of cell i respectively. The right-hand side of equation (3) is evaluated by summing the flux vectors over each edge of the triangular cell and the discrete form of the integral is

$$\oint_{\partial C_i} \mathbf{F} \cdot \mathbf{n} \mathrm{d}S = \sum_{j=k(i)} F_{i,j} \Delta l_j, \tag{4}$$

where $k(i)$ is a list of the neighbouring cells to cell i, $F_{i,j}$ is the numerical flux through the interface of cells i and j and Δl_j is the length of edge j. For a cell-centred scheme based on a triangular mesh, $k(i)$ is a list of three cells.

3. EVALUATION OF INVISCID AND VISCOUS FLUXES

3.1. Inviscid fluxes

In order to evaluate the inviscid numerical fluxes $F^{\mathrm{I}}_{i,j}$, Roe's[11–13] flux function is adopted locally at each cell edge, assuming a one-dimensional Riemann problem in the direction normal to the cell edge, as follows:

$$\begin{aligned} F^{\mathrm{I}}_{i,j} &= \tfrac{1}{2}[F^{\mathrm{I}}(Q^+_{i,j}) + F^{\mathrm{I}}(Q^-_{i,j}) - |A|(Q^+_{i,j} - Q^-_{i,j})], \\ |A| &= R|\Lambda|L, \end{aligned} \tag{5}$$

where $Q^+_{i,j}$ and $Q^-_{i,j}$ are the reconstructed right and left states respectively of the cell face between cells i and j and A is the flux Jacobian evaluated by Roe's average state, which is the average of Q^+ and Q^-.[14] The quantities R, L and Λ are the right and left eigenvectors and the eigenvalues of A respectively. The flux Jacobian A is given by

$$A = \frac{\partial(\mathbf{F} \cdot \mathbf{n})}{\partial Q} = \begin{bmatrix} 0 & n_x & n_y \\ (c^2 - u^2)n_x - uvn_y & 2un_x + vn_y & un_y \\ -uvn_x + (c^2 - v^2)n_y & vn_x & un_x + 2vn_y \end{bmatrix}, \tag{6}$$

its eigenvalues are given by

$$\lambda_1 = un_x + vn_y, \qquad \lambda_2 = un_x + vn_y - cn, \qquad \lambda_3 = un_x + vn_y + cn, \tag{7}$$

the right eigenvector matrix is given by

$$\begin{bmatrix} 0 & 1 & 1 \\ n_y & u - cn_x/n & u + cn_x/n \\ -n_x & v - cn_y/n & v + cn_y/n \end{bmatrix} \tag{8}$$

and the left eigenvector matrix is given by

$$\begin{bmatrix} -(un_y - vn_x)/n & n_y/n^2 & -n_x/n^2 \\ (un_x + vn_y)/2cn + \frac{1}{2} & -n_x/2cn & -n_y/2cn \\ -(un_x + vn_y)/2cn + \frac{1}{2} & n_x/2cn & n_y/2cn \end{bmatrix}, \tag{9}$$

where c denotes the wave celerity $(gh)^{1/2}$ and n denotes the magnitude of the normal vector $\mathbf{n}$ to the cell edge; for a unit normal the magnitude is equal to one.

In order to obtain second-order accuracy, a piecewise linear model for the cell variables must first be reconstructed from the solution data before the two Riemann states at each cell edge are computed. For a given cell with centre A, for example, this requires the construction of the cell variables in the form

$$Q(x, y) = Q_{\mathrm{A}} + \nabla Q_{\mathrm{A}} \cdot \mathbf{r}, \tag{10}$$

where $\mathbf{r}$ is the vector from cell centre A to any point (x, y) within the cell, Q_{A} is the cell centre value of cell A and ∇Q_{A} is the gradient of cell A evaluated using the cell centre values.

By appling Gauss's theorem, the solution gradient of cell A can be estimated by evaluating a boundary integral over the path surrounding A as

$$\nabla Q = \frac{1}{A_\Omega} \oint_{\partial A} Q \mathbf{n} \mathrm{d}S, \tag{11}$$

where ∂A is an integration path connecting points with known variables and A_Ω is the area contained in the path. A direct simple integration path for ∇Q is the path connecting three neighbouring cell centres B, C and D as shown in Figure 1. As suggested by Pan and Cheng,[6] the estimated gradient accuracy can be further improved by carrying out a weighted averaging process as follows:

$$\nabla Q_{\mathrm{A}}^{\mathrm{mod}} = \tfrac{1}{2}(\nabla Q_{\mathrm{A}} + w_{\mathrm{B}} \nabla Q_{\mathrm{B}} + w_{\mathrm{C}} \nabla Q_{\mathrm{C}} + w_{\mathrm{D}} \nabla Q_{\mathrm{D}}), \tag{12}$$

with

$$\begin{aligned} w_{\mathrm{B}} &= (\mathbf{r}_{\mathrm{AC}} \times \mathbf{r}_{\mathrm{CD}})/(\mathbf{r}_{\mathrm{BC}} \times \mathbf{r}_{\mathrm{CD}}), \\ w_{\mathrm{C}} &= (\mathbf{r}_{\mathrm{AD}} \times \mathbf{r}_{\mathrm{DB}})/(\mathbf{r}_{\mathrm{BC}} \times \mathbf{r}_{\mathrm{CD}}), \\ w_{\mathrm{D}} &= (\mathbf{r}_{\mathrm{AB}} \times \mathbf{r}_{\mathrm{BC}})/(\mathbf{r}_{\mathrm{BC}} \times \mathbf{r}_{\mathrm{CD}}), \end{aligned} \tag{13}$$

where $\nabla Q_{\mathrm{B,C,D}}$ are the gradient vectors of cells B, C and D respectively, evaluated by equation (11), and the quantities $\mathbf{r}$ are the distance vectors from cell centre A to the centre of the neighbouring cells.

3.2. *Viscous fluxes*

The viscous numerical flux $F_{i,j}^{\mathrm{V}}$ is evaluated by computing the gradient of the velocity at each cell edge using the expression

$$\nabla Q_{\mathrm{PQ}} = \frac{1}{A_{\mathrm{APBQ}}}(A_{\mathrm{BPQ}} \nabla Q_{\mathrm{APQ}} + A_{\mathrm{APQ}} \nabla Q_{\mathrm{BPQ}}), \tag{14}$$

where, with reference to Figure 2(a), ∇Q_{APQ} and ∇Q_{BPQ} are the gradient vectors of path APQ and BPQ respectively, evaluated by equation (11), and A_{APBQ}, A_{APQ} and A_{BPQ} are the areas of

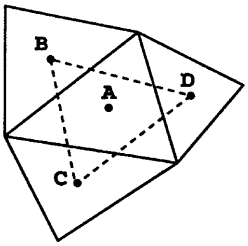

Figure 1. BCD defines integration path for ∇Q_A

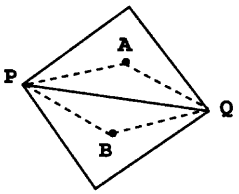

Figure 2(a). APBQ defines integration path for viscous stresses at interior cell

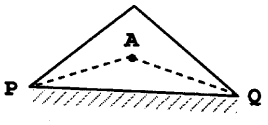

Figure 2(b). APQ defines integration path for viscous stresses at boundary cell

quadrilateral APBQ and triangles APQ and BPQ respectively. The variable values at cell vertices P and Q are obtained by a distance-weighted average of cell centre values surrounding that particular vertex. Once the gradient vectors of velocity are obtained, the viscous fluxes at cell edge PQ can then be combined with the evaluated inviscid flux and put into equation (4) for integration. For a triangle on the boundary as shown in Figure 2(b), the gradient at the boundary PQ is evaluated by the triangular contour APQ only. Assuming accurate values at P and Q, the above formulation is second-order-accurate for a parallelogram and first-order-accurate for a general quadrilateral element.[5]

4. LIMITER AND BOUNDARY CONDITIONS

4.1. Limiter for second-order scheme

The ability to simulate discontinuities and eliminate numerical oscillations is highly desirable in a numerical scheme. Methods currently used to control or limit spurious oscillations and diffusion around discontinuities do so either by introducing artificial viscosity or by making use of non-linear limiters. The use of artificial viscosity suppresses numerical oscillations which otherwise would have been generated. The use of non-linear limiters suppresses oscillations by controlling the numerical solution in a non-linear way, so that the appearance of any new local extreme is prohibited during the process of reconstruction of the cell variables.

In all second- and higher-order schemes the application of non-linear limiters involves the introduction of a parameter termed the 'limiter' into the gradient terms that appear in the process of cell variable reconstruction. This step is necessary for higher-order schemes in order to maintain monotonicity. As a result the reconstruction of the second-order cell variables is *limited* in the form

$$Q(x, y) = Q_o + \Phi \nabla Q \cdot \mathbf{r}, \tag{15}$$

where Φ is a chosen limiter. When Φ is set to zero, equation (15) is a first-order-accurate reconstruction. In a manner similar to that proposed by Barth and Jespersen,[4] the limiter is defined as

$$\Phi = \min(\Phi_j), \quad j = k(i), \tag{16}$$

where

$$\Phi_j(r_j) = \max[\min(\beta r_j, 1), \min(r_j, \beta)], \tag{17}$$

with

$$r_j(Q_j) = \begin{cases} (Q_o^{\max} - Q_o)/(Q_j - Q_o) & \text{if} \quad Q_j - Q_o > 0, \\ (Q_o^{\min} - Q_o)/(Q_j - Q_o) & \text{if} \quad Q_j - Q_o < 0, \\ 1 & \text{if} \quad Q_j - Q_o = 0 \end{cases} \tag{18}$$

and

$$Q_o^{\min} = \min(Q_o, Q_{\text{neighbour}}), \qquad Q_o^{\max} = \max(Q_o, Q_{\text{neighbour}}), \tag{19}$$

where Q_j is the unlimited cell edge value and Q_o is the cell centroid value. The quantity β in equation (17) can take any values between one and two. In particular, $\beta = 1$ is the minmod limiter and $\beta = 2$ is Roe's 'Superbee' limiter.[15]

4.2. Boundary conditions

Boundary conditions are imposed on two different types of boundary, namely the wall boundary and the open boundary. Boundary conditions are applied at the interface of a cell. The boundary conditions for a wall boundary are set as follows:

slip wall	no-slip wall
$h_R = h_L$	$h_R = h_L$
$u_R = 0$	$u_R = 0$
$v_R = v_L$	$v_R = 0.$

(20)

Here the subscripts R and L denote the right and left Riemann states respectively at a cell interface, u denotes the velocity in the outward normal direction to the cell edge and v denotes the velocity parallel to the cell edge. The no-slip wall condition is applicable when the eddy viscosity $\nu \neq 0$. The boundary conditions for an open boundary are set by making use of the outgoing Riemann invariants and are as follows:

subcritical inflow	subcritical outflow
$h_R = h_B$	$h_R = h_B$
$u_R = u_L + \sqrt{g}(\sqrt{h_L} - \sqrt{h_R})$	$u_R = u_L + \sqrt{g}(\sqrt{h_L} - \sqrt{h_R})$
$v_R = 0$	$v_R = v_L$

(21)

supercritical inflow	supercritical outflow
$h_R = h_B$	$h_R = h_L$
$u_R = u_B$	$u_R = u_L$
$v_R = 0$	$v_R = v_L.$

(22)

The subscript B above denotes prescribed physical boundary values. In subcritical flow, if u_B is prescribed instead of h_B, then h_R is calculated from $u_R = u_L + \sqrt{g}(\sqrt{h_L} - \sqrt{h_R})$. The local values of the Froude number are used to determine whether the flow is subcritical or supercritical at a given open boundary.

5. TIME INTEGRATION

For the purpose of time integration, equation (3) is expressed as

$$\frac{(QV)^{n+1} - (QV)^n}{\Delta t} = -[\alpha R(Q^{n+1}) + (1-\alpha)R(Q^n)]. \tag{23}$$

When $\alpha = 0$, equation (23) is the Euler explicit scheme, when $\alpha = 1$, it is the first-order Euler implicit method, and when $\alpha = 0{\cdot}5$, it is the second-order trapezoidal implicit scheme. For implicit time integration, Newton's linearization of equation (23) can be used about a known state Q^n to yield

$$\left(\frac{IV}{\Delta t} + \alpha\frac{\partial R(Q^*)}{\partial Q}\right)(Q^{s+1} - Q^s) = -\left(\frac{(Q^s - Q^n)V}{\Delta t} + \alpha R(Q^s) + (1-\alpha)R(Q^n)\right), \tag{24}$$

where s is a subiteration index and I is the identity matrix. When the RHS of equation (24) is subiterated towards zero, Q^{s+1} approaches the time-accurate Q^{n+1}. It may be noted that if $R(Q^*)$ is set to $R(Q^s)$, the above scheme belongs to the class of standard Newton–Raphson iteration methods, whereas if $R(Q^*)$ is set to $R(Q^n)$, then it is a modified Newton method.

In implicit time integration, some computational work relating to the inversion process can be saved by letting the inviscid Jacobian on the implicit side of the equation be first-order-accurate and by simulating the viscous Jacobian using a Laplacian-like dissipation operator as follows:

$$-v\sum_{j=k(i)} F^{\mathrm{V}}_{i,j}\Delta l_j = -v\sum_{j=k(i)} h_i \nabla Q^{\mathrm{V}}_{i,j}\cdot\mathbf{n}\Delta S_j \approx -v\sum_{j=k(i)}\frac{h_i\mathbf{r}_{j,i}\cdot\mathbf{n}\Delta l_j}{|\mathbf{r}_{j,i}|^2}(Q^{\mathrm{V}}_j - Q^{\mathrm{V}}_i), \tag{25}$$

where Q^{V} denotes the vector $[0, u, v]^{\mathrm{T}}$ and $\mathbf{r}_{j,i}$ is the distance vector from cell centre i to cell centre j. The fully discretized implicit time integration equation can be written in matrix form as

$$(D + L + U)\Delta Q^s = \mathrm{RHS}, \tag{26}$$

where D is a block diagonal matrix, L is a block lower triangular matrix and U is a block upper triangular matrix. Each of the elements in D, L and U is a 3×3 matrix. An approximate LU factorization (ALU) scheme as proposed by Pan and Lomax[16] can be adopted to invert the implicit equation (26) in the form

$$(D + L)D^{-1}(D + U)\Delta Q^s = \mathrm{RHS}, \tag{27}$$

where the error of the factorization is equal to $LD^{-1}U\Delta Q^s$. Within each time step of the implicit integration, subiteration is terminated when the $L2$ norm of the iteration process,

$$L2 = \left[\left(\sum_{i=1}^{N}(Q^{s+1} - Q^s)^2\right)\Big/ N\right]^{1/2}, \tag{28}$$

is less than a specified limit value.

6. NUMERICAL RESULTS

6.1. 2D dam break problem

In order to validate the inviscid part of the shallow water equations, two dam break test cases are examined. First the model is applied to the 2D frictionless partial dam break problem studied by Fennema and Chaudhry.[17] The geometry of the problem consists of a 200×200 m^2 basin as illustrated in Figure 3(a). The initial water level of the dam is 10 m and the tail water is 5 m high. At the instant of dam failure, water is released into the downstream side through a breach 75 m wide,

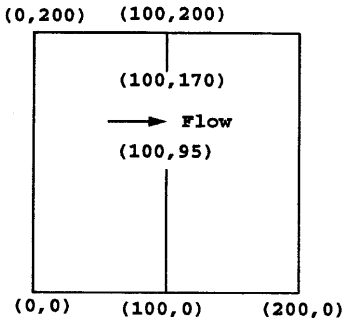

Figure 3(a). Definition of problem domain for partial dam break test

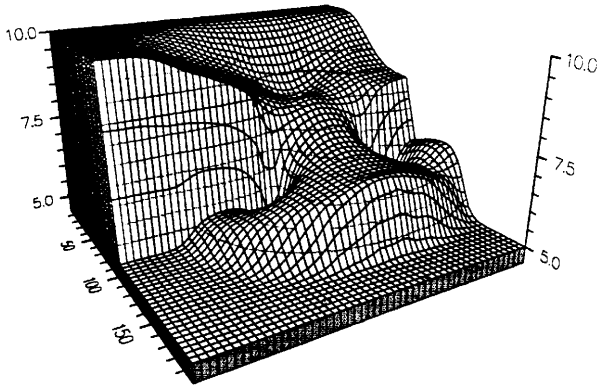

Figure 3(b). Water surface profile at $t = 7{\cdot}2$ s after breaking of dam

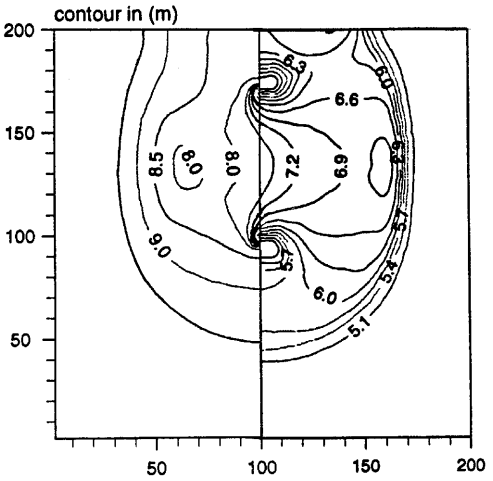

Figure 3(c). Contour plot of water surface elevation for partial dam break test

forming a wave that propagates while spreading laterally. At the same time a negative wave propagates upstream. The problem domain was triangulated into 5440 cells and the computational model was run for up to 7·2 s after the dam break using the minmod limiter. Figures 3(b) and 3(c) show the 3D view of the water surface elevation and the contour map of water depths respectively. These results agree well with those reported by Fennema and Chaudhry,[17] Alcrudo and Garcia-Navarro[10] and Zhao *et al.*[7]

The second dam break test case is based on the hypothetical test case studied by Alcrudo and Garcia-Navarro[10] which involves the breaking of a circular dam. The initial conditions consist of two regions of still water separated by a cylindrical wall of radius 11 m. The water depth inside the dam is 10 m, whilst outside the dam it is 1 m. At the instant of dam failure the circular wall is assumed to be removed completely and the subsequent time evolution of the radially spreading waves is studied. The problem domain is defined in Figure 4(a). The computational domain was triangulated into 7490 cells and the model was run for up to 0·69 s after the dam break using the minmod limiter. Figures 4(b) and 4(c) show the numerical results for the water surface elevation in 3D and 2D respectively. It is noted that the 'priviledged flow in the direction of co-ordinate axes' phenomenon as reported by Alcrudo and Garcia-Navarro in their rectangular grid model does not appear here. The advancing front is seen to be circularly symmetrical and a ring of irregular depressing waves is noted next to the

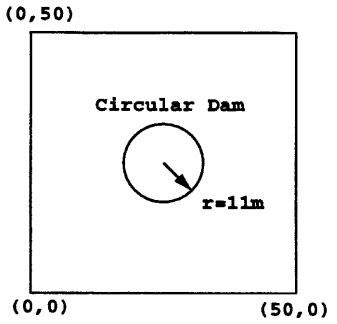

Figure 4(a). Definition of problem domain for circular dam break test

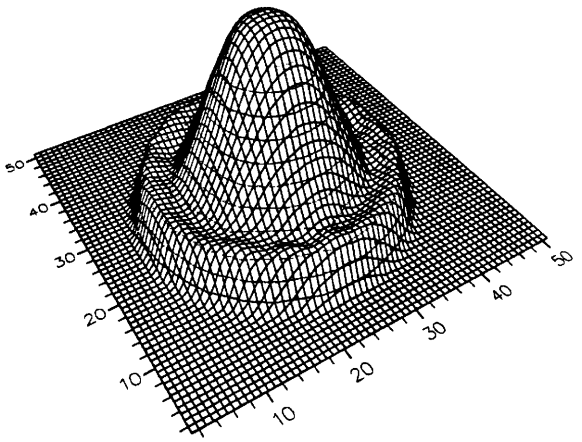

Figure 4(b). Water surface profile at $t = 0{\cdot}69$ s after breaking of circular dam

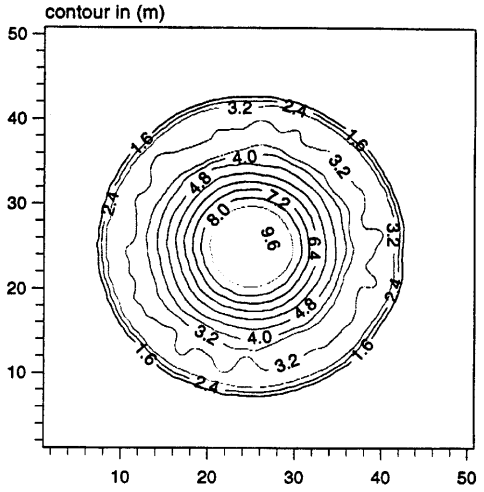

Figure 4(c). Contour plot of water elevation for circular dam break test

advancing front. In general the contours of water surface elevation appear in good agreement with those reported by Alcrudo and Garcia-Navarro.

6.2. *Oblique hydraulic jump*

A further inviscid flow test case is examined which corresponds to an oblique hydraulic jump problem for which an exact solution is available. The definition of the problem domain is shown in Figure 5(a). An oblique hydraulic jump is induced in the flow field by a supercritical flow which is deflected by a converging wall at an angle θ. A schematic diagram of the induced shock front is given in Figure 5(b), where α is the angle formed by the shock front with the x- axis.

The initial conditions are set as $\theta = 8{\cdot}95^\circ$, $h_o = 1$ m, $u_o = 8{\cdot}57$ m s^{-1} and $v_o = 0$. Supercritical flow boundary conditions are applied at the inflow and outflow boundaries of the computational domain. The computational domain was triangulated into 2448 cells and the model was run to steady state

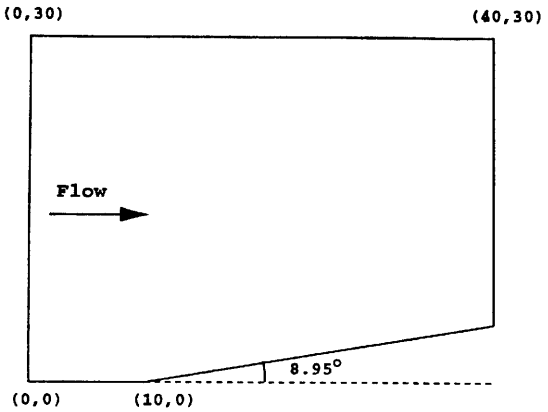

Figure 5(a). Definition of problem domain for oblique hydraulic jump test

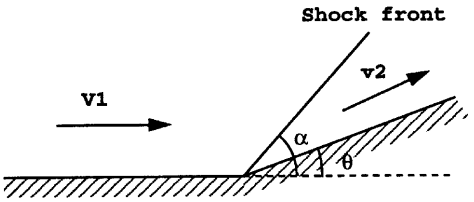

Figure 5(b). Schematic diagram showing oblique shock front in supercritical flow

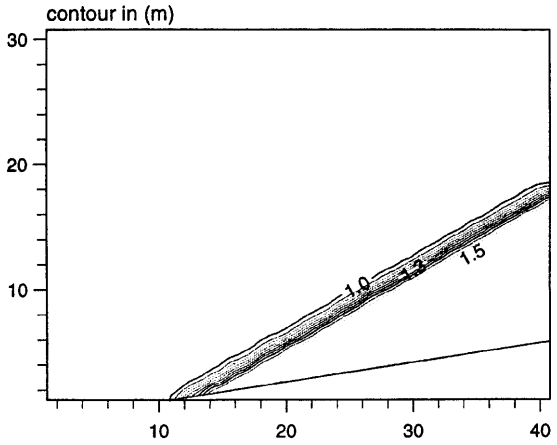

Figure 5(c). Water surface profile showing oblique hydraulic jump

using the minmod limiter. The numerical results obtained are $\alpha = 30^\circ$, with the downstream flow variables across the jump being $h = 1{\cdot}5$ m and $|\mathbf{u}| = 7{\cdot}954$. These results compare very well with those obtained by Alcrudo and Garcia-Navarro and the published analytical solution[10] ($h = 1{\cdot}5049$ m, $|\mathbf{u}| = 7{\cdot}9419$). The depth contour plot showing the oblique hydraulic jump is given in Figure 5.

6.3. Flow past a backward step

In order to test the handling by the algorithm of the full set of shallow water equations including the viscous terms, two test cases relating to flow past a backward step are examined. First an unsteady flow past a backward step in a laboratory tidal flume studied by Stelling[18] is modelled. The geometry of the problem domain is shown in Figure 6(a). The computational domain was uniformly triangulated into 2880 cells. The initial conditions are $h = 0{\cdot}1$ m, $u = 0$ and $v = 0$. The boundary conditions at the inflow boundary are set as

$$v = 0, \qquad u(t) = \sum_{i=1}^{3} \hat{u}_i \sin(w_i t), \tag{29}$$

where $w_i = 2\pi/150$, $\hat{u}_1 = 0{\cdot}0375$ m s^{-1}, $\hat{u}_2 = 0{\cdot}05$ m s^{-1} and $\hat{u}_3 = 0{\cdot}01$ m s^{-1}. The boundary conditions at the outflow boundary are set as

$$h(t) = \begin{cases} 0{\cdot}1, & t < 5 \text{ s}, \\ 0{\cdot}1 + \sum_{i=1}^{3} \hat{h}_i \sin[w_i(t-5)], & t \geq 5 \text{ s}, \end{cases} \tag{30}$$

where $\hat{h}_1 = 0{\cdot}021$, $\hat{h}_2 = 0{\cdot}001$ and $\hat{h}_3 = 0{\cdot}0005$. The Chẽzy and eddy viscosity coefficients are set equal to 62·64 and $2{\cdot}3 \times 10^{-4}$, respectively and a slip wall boundary condition is used. The numerical output at $t = 35$ s is shown in Figure 6(b). The presented flow field is found to compare very well with the results obtained by Stelling. It should be noted that the velocity vectors are plotted by applying interpolation to the cell-centred values.

In the second test a steady flow past a backward step is dealt with and the reattachment length of the flow field downstream from the step is calculated. The geometry of the problem domain is similar to that used by Denham and Patrick[19] (Figure 7(a)) and the computational domain was uniformly triangulated into 3050 cells. The step dimension is h, the upstream boundary is located at a distance $11h$ from the step and the downstream boundary is at a distance $13h$ from the step. The step dimension and water depth are both taken to be unity. A slip wall boundary condition is used, the bed stress is set to zero, a velocity of 0·5 m s^{-1} is prescribed at the inflow boundary and a still water level of 1·0 m is prescribed at the outflow boundary. The eddy viscosity coefficient is set to 0·00667 to give an inlet Reynolds number of 75.

Figure 7(b) depicts the computed flow field, while the computed streamfunction contours near the step are shown in Figure 7(c). The presented results indicate that the reattachment length in this example is about 3·8 m downstream from the step. This is in good agreement with the experimental data reported by Denham and Patrick.

6.4. *Jet-forced flow in a circular reservoir*

A further test case with the viscous terms included is examined which simulates a jet-forced flow in a circular reservoir. The same test case has been studied by Akponasa[20] in validating a boundary-fitted grid model. The domain geometry is shown in Figure 8(a). For this test case a no-slip wall boundary condition is used, the bed stress is set to zero and the coefficient of eddy viscosity is set to

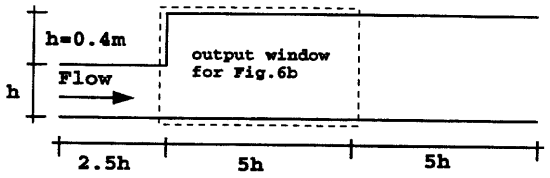

Figure 6(a). Definition of problem domain for flow past backward step: test case 1

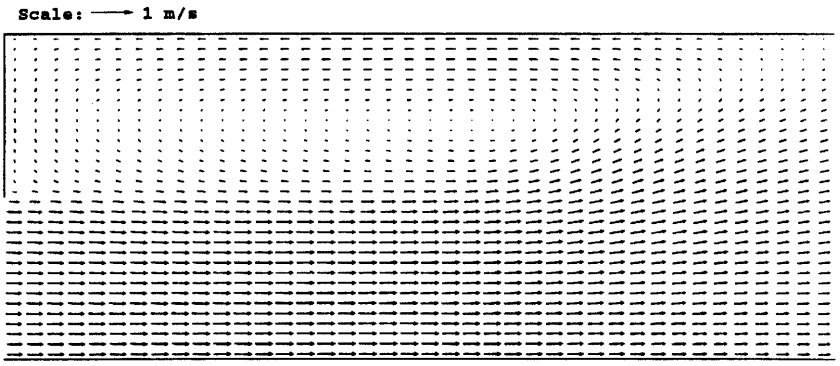

Figure 6(b). Flow field at $t = 35$ s for flow past backward step: test case 1

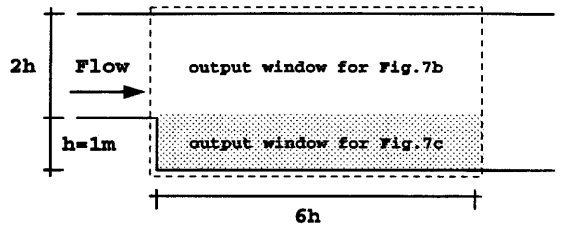

Figure 7(a). Definition of problem domain for flow past backward step: test case 2

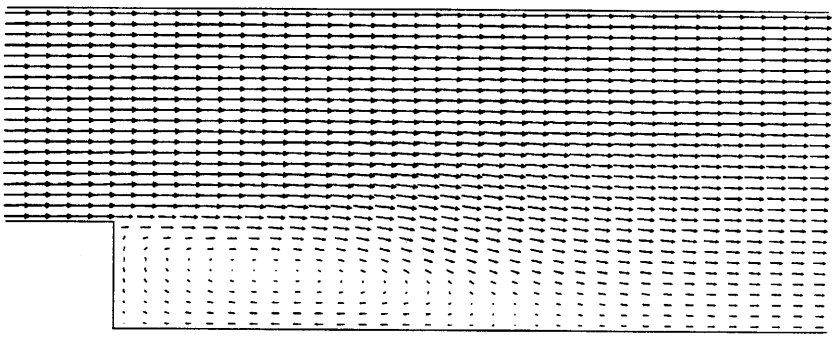

Figure 7(b). Flow field at steady state for flow past backward step: test case 2

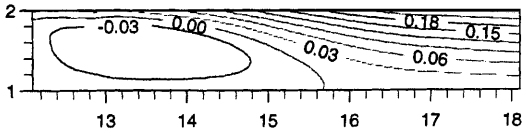

Figure 7(c). Streamfunction contours showing reattachment length

0·00078. The inflow velocity is 0·1 m s^{-1}, which corresponds to an inlet Reynolds number of 10. A still water level of 0·1 m is prescribed at the outflow boundary. For this test case the computational domain was triangulated into 3652 cells. The computed results for the flow field at steady state are shown in Figure 8(b) and the streamfunction plot for the upper half of the computational domain is shown in Figure 8(c). The results from the present model are in very good agreement with those obtained by Akponasa.

6.5. Numerical experiments on linear convection problems

A series of numerical experiments has also been undertaken in order to assess the performance of the algorithm in dealing with linear convection problems. The governing equation is

$$\frac{\partial q}{\partial t}+\frac{\partial uq}{\partial x}+\frac{\partial vq}{\partial y}=q\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right), \tag{31}$$

where q is a scalar field. The RHS of equation (31) is equal to zero for divergence-free flow fields.

The first test case involves the advection of a square-shaped scalar field as studied by Tamamidis and Assanis.[21] The square scalar field is advected in a uniform velocity field in a 6×6 units

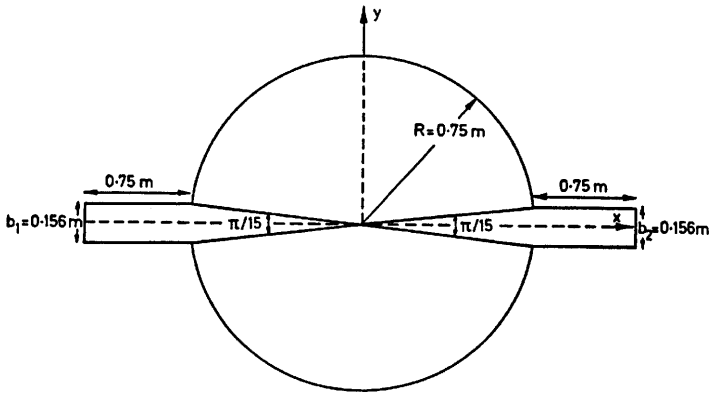

Figure 8(a). Definition of problem domain for jet-forced flow in circular reservoir

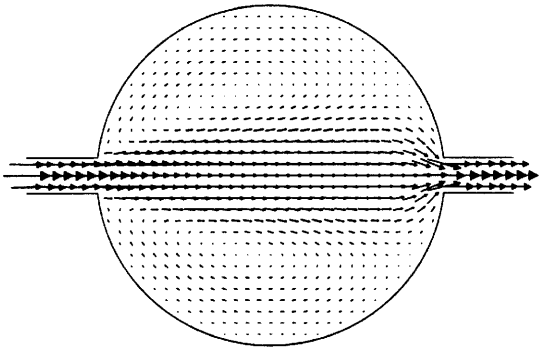

Figure 8(b). Vector plot of jet-forced flow field in circular reservoir

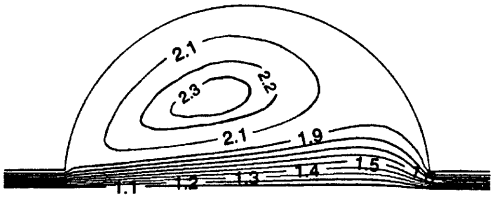

Figure 8(c). Streamfunction contour plot in upper half of circular reservoir

computational domain. The definition of the problem domain and the 3D view of the initial conditions are shown in Figures 9(a) and 9(b) respectively. The initial size of the scalar field is a $1{\cdot}5 \times 1{\cdot}5$ units square. The maximum and minimum values of the field are $q_{\max} = 10$ and $q_{\min} = 0$ and the field is advected a distance of 3·9 units from its initial position. The values of the flow variables u and v are both set to 1 unit. The computational domain was triangulated into 7098 cells and three runs were carried out using the minmod limiter, the β limiter with β set to 1·5 and the Superbee limiter. The maximum and minimum values as well as the root-mean-square (RMS) error of the scalar field after the advection are recorded in Table I. The RMS values are obtained by

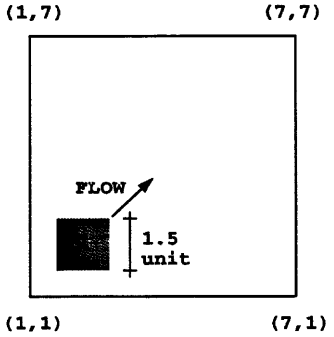

Figure 9(a). Definition of problem domain for convection of 'square' scalar field

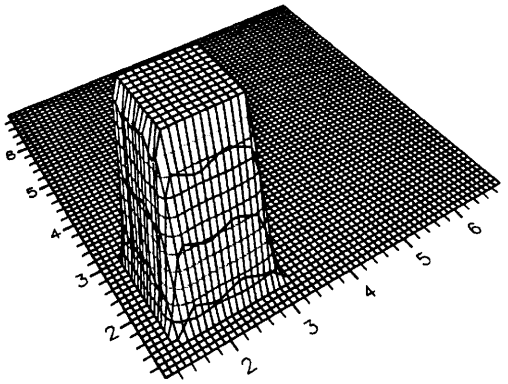

Figure 9(b). 3D view of initial profile of square scalar field

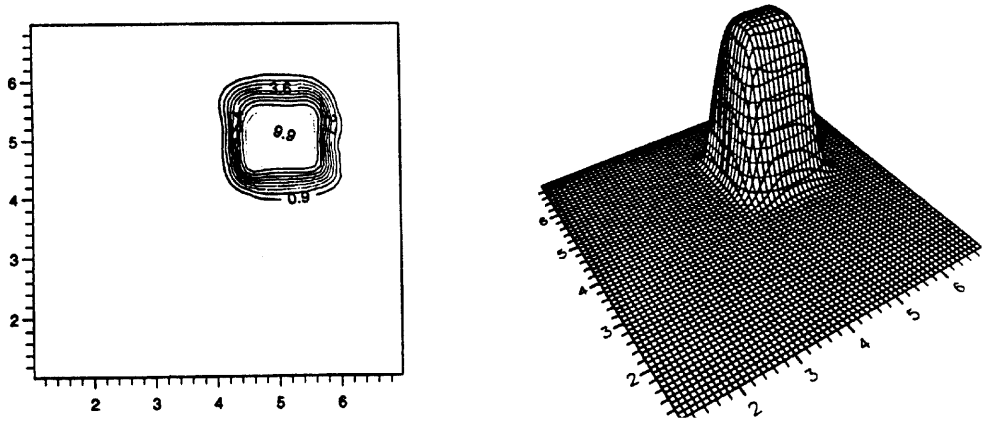

Figure 9(c). 3D view and contour plot of advected square scalar field: test with minmod limiter

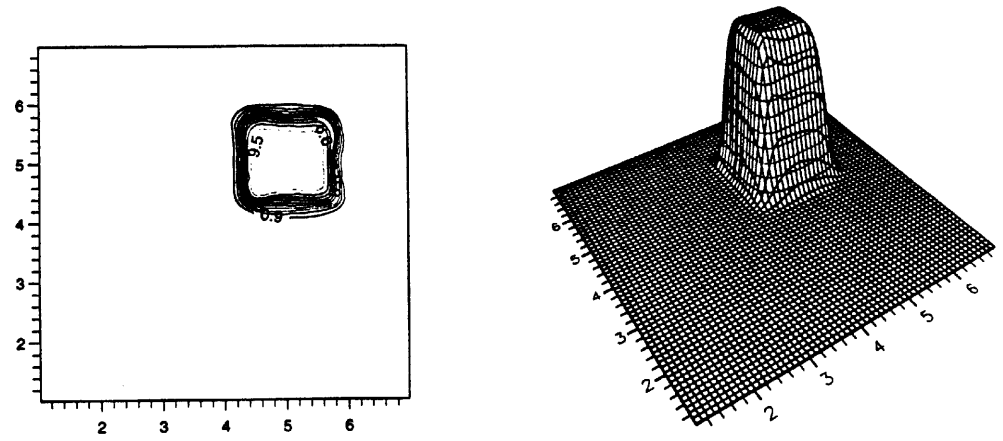

Figure 9(d). 3D view and contour plot of advected square scalar field: test with $\beta = 1{\cdot}5$ limiter

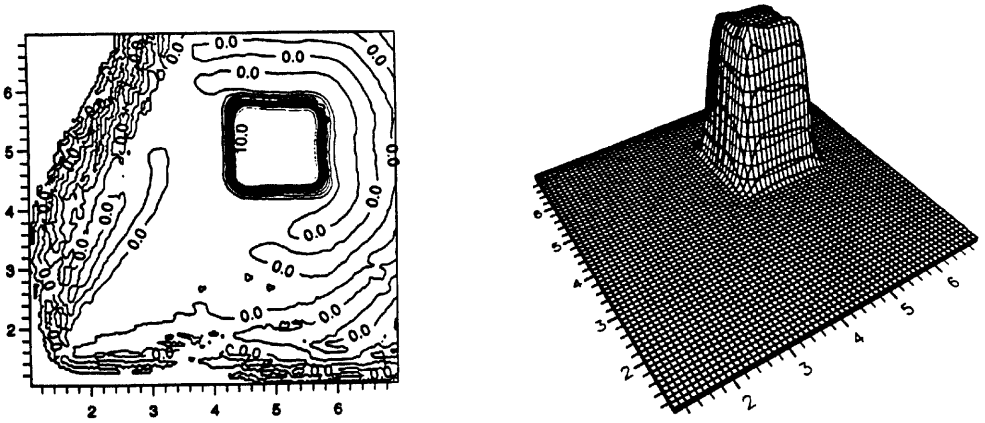

Figure 9(e). 3D view and contour plot of advected square scalar field: test with Superbee limiter

comparing the numerical results with the exact solution. The 3D views and the corresponding contour plots of the advected fields are shown in Figures 9(c)–9(e). It is noted from these results that the minmod limiter produces very satisfactory results, but numerical diffusion is evident, as is always the case with any dissipative algorithm with a limiter. On the other hand, the Superbee limiter in this case is seen to be overly 'antidiffusive' and a large number of ripples are seen to be generated throughout the problem domain. The β limiter with β set to 1·5 gives very good results in maintaining the field gradient, but some numerical oscillation is produced.

Table I.

	$q_{\max}$	$q_{\min}$	RMS
Minmod	9·997	0	0·789
$\beta = 1{\cdot}5$	10·286	−0·192	0·681
Superbee	11·098	−0·626	0·633

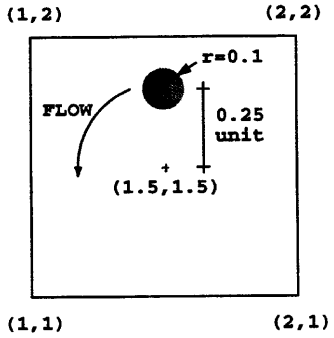

Figure 10(a). Definition of problem domain for convection of 'cone' scalar field

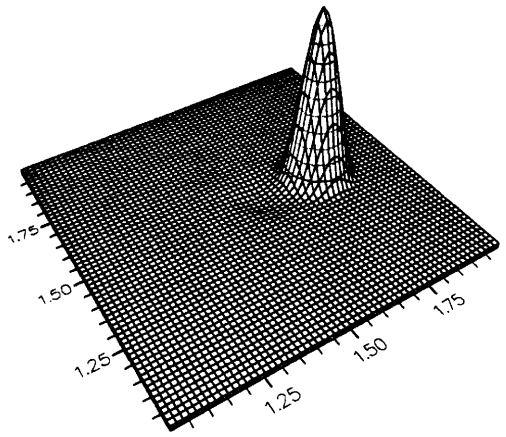

Figure 10(b). 3D view of initial profile of cone scalar field

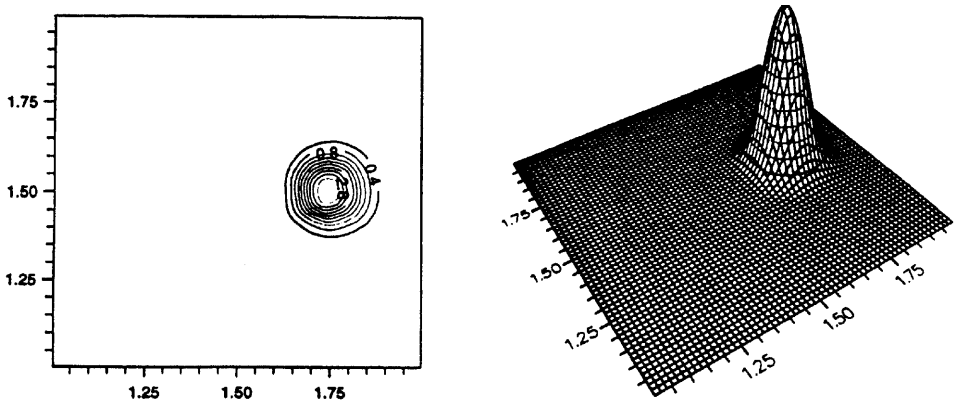

Figure 10(c). 3D view and contour plot of advected cone scalar field: test with minmod limiter

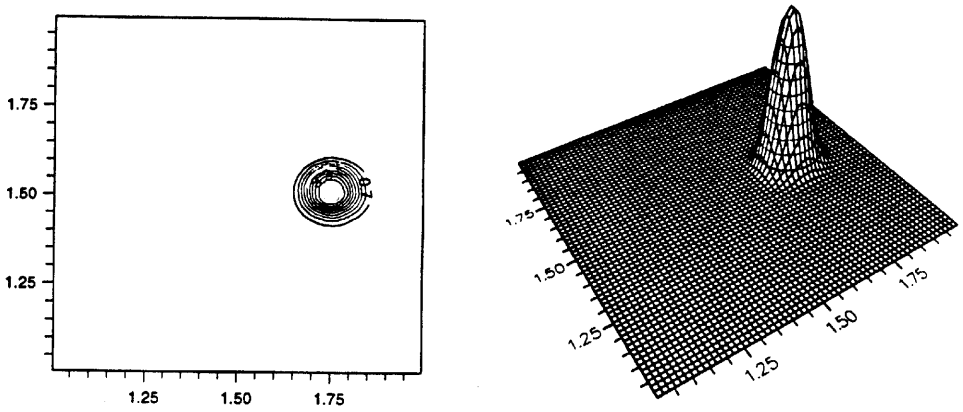

Figure 10(d). 3D view and contour plot of advected cone scalar field: test with $\beta = 1{\cdot}5$ limiter

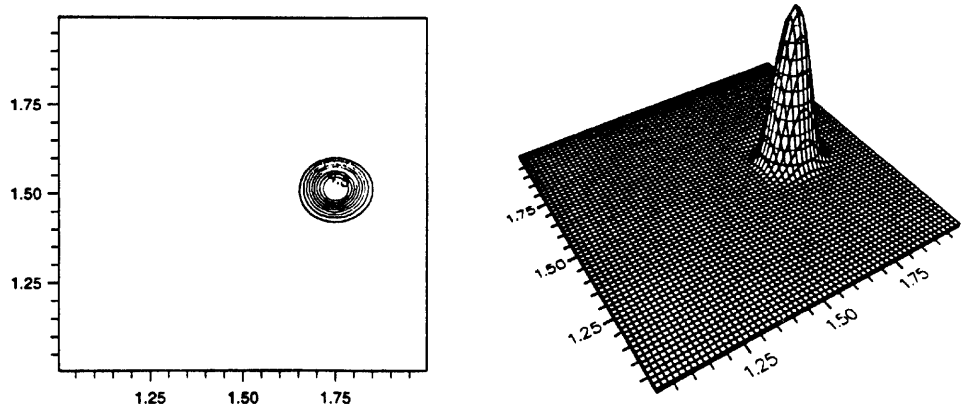

Figure 10(e). 3D view and contour plot of advected cone scalar field: test with Superbee limiter

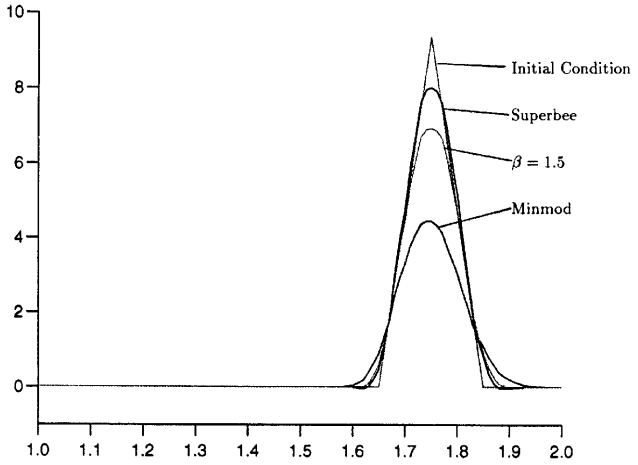

Figure 10(f). Cross-sectional view of advected cone-shaped scalar field using different limiters compared with initial condition

Table II.

	q_{max}	q_{min}	RMS
Minmod	4·487	0	0·313
$\beta = 1{\cdot}5$	6·942	−0·041	0·191
Superbee	8·038	−0·261	0·185

Table III.

	q_{max}	q_{min}	RMS
Minmod	4·785	0	0·292
$\beta = 1{\cdot}5$	7·233	−0·034	0·162
Superbee	8·315	−0·207	0·148

The second linear convection test case involves a rotating cone-shaped scalar field which has also been used by Tamamidis and Assanis[21] in their evaluation of the numerical accuracy of various high-order schemes. In this example a scalar cone-shaped field is advected around by a rotating velocity field in a 1×1 square computational domain. The angular frequency ω of the velocity field is set to be 2 units and the base radius of the cone is set to be 0·1 unit. The maximum and minimum values of the field are $q_{max} = 10$ and $q_{min} = 0$. The definition of the problem domain and the initial condition of the scalar field are shown in Figures 10(a) and 10(b) respectively. For this test case the computational domain was triangulated into 12,800 cells and three runs were carried out using the minmod limiter, the β limiter with β set to 1·5 and the Superbee limiter. The maximum and minimum values and the RMS values of the advected cone after one revolution are recorded in Table II. The 3D views and the corresponding contour plots of the advected fields are depicted in Figures 10(c)–10(e). Figure 10(f) shows the cross-sectional view of the cone-shaped field, advected using different limiters, compared with the initial condition. The plots show that the Superbee limiter performs very satisfactorily.

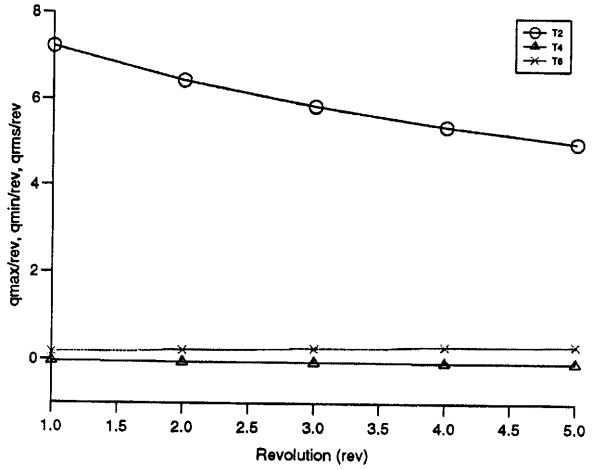

Figure 11. Revolution history of advected cone-shaped scalar field: T2, q_{max}/rev; T4, q_{min}/rev; T6, q_{rms}/rev

An additional test concerning the advection of the cone-shaped field has been carried out using a more refined solution domain with a mesh density of 16,200 cells. The results of the experiment after one full revolution are given in Table III. Inspection of these results reveals that they are superior to those derived from the coarser mesh. Figure 11 shows the revolution history of the cone-shaped field for five full revolutions, as advected over the fine mesh domain, using the β limiter with β set to 1·5. It is clear from these results that the rate of numerical diffusion is a function of the mesh density used and the gradient of the scalar field in the computational domain. Also, the numerical diffusion is seen to be suppressed more the stronger the applied limiter is. On the other hand, reduction in numerical diffusion is accompanied by numerical oscillations, which increase with increasing limiter values. It is concluded that this trade-off between numerical diffusion and numerical oscillations is a fundamental characteristic of the present solver.

7. CONCLUSIONS

A versatile, second-order-accurate, upwind finite volume scheme for the depth-integrated shallow water equations has been developed for 2D unstructured meshes and tested. To the best of the authors's knowledge the finite volume method has been used for the first time in conjunction with Roe's scheme in order to produce a second-order-accurate scheme for solving the full set of 2D shallow water equations as related to environmental hydraulics. An essential feature of the present methodology is the construction of Roe's flux function through each side of a triangular cell by considering a 1D local Riemann problem, and the eigensystem required for the construction of Roe's matrix for the shallow water equations has been derived and presented. The robustness of the algorithm stems from the high flexibility of triangular cells in covering a computational domain and its ability to handle a wide range of flow regimes, including gradually varied and discontinuous flows, steady and unsteady flows, subcritical and supercritical flows. The application of the developed solver to linear convection problems with acute gradients shows that it is highly accurate. A range of limiters of numerical oscillations have also been tested for 2D problems on unstructured meshes. It has been established that while the stronger limiters produce less numerical diffusion, they suppress to a smaller degree numerical oscillations compared with weaker limiters. It is concluded that while the Superbee limiter is seen to perform most satisfactorily in preserving extreme values, optimum performance can be achieved using a medium-strength limiter, noting that experimentation is always required.

REFERENCES

1. A. Jameson and D. Mavriplis, 'Finite volume solution of the two-dimensional Euler equations on a regular triangular mesh', *AIAA Paper 85-0435*, 1985.
2. D. Mavriplis and A. Jameson, 'Multigrid solution of the Euler equations on unstructured and adaptive meshes', *AIAA Paper 87-0353*, 1987.
3. J. A. Desiseri and A. Dervieux, 'Compressible flow solvers using unstructured grids', *VKI Lecture Series 1988-05, 7-11*, 1988, pp. 1–115.
4. T. J. Barth and D. C. Jespersen, 'The design and application of upwind schemes on unstructured meshes', *AIAA Paper 89-0366*, 1989.
5. D. Pan, C. Lu and J. Cheng, 'Incompressible flow solution on unstructured triangular meshes', *Numer. Heat Transfer B*, **26**, 207–224 (1994).
6. D. Pan and J. Cheng, 'A second-order upwind finite-volume method for the Euler solution on unstructured triangular meshes', *Int. j. numer. methods fluids*, **16**, 1079–1098 (1993).
7. D. H. Zhao, H. W. Shen, G. Q. Tabious III, J. S. Lai and W. Y. Tan, 'Finite-volume two-dimensional unsteady-flow model for river basins', *ASCE J. Hydraul. Eng.*, **120**, 864–883 (1994).
8. S. Osher and F. Solomone, 'Upwind difference schemes for hyperbolic systems of conservation laws', *Math. Comput.*, **38**, 339–374 (1982).

9. S. P. Spekreijse, 'Multigrid solution of steady Euler equations', *CWI Tract 46*, Centre for Mathematics and Computer Science, Amsterdam, 1988.
10. F. Alcrudo and P. Garcia-Navarro, 'A high-resolution Godunov-type scheme in finite volumes for the 2D shallow-water equations', *Int. j. numer. methods fluids*, **16**, 489–505 (1993).
11. P. L. Roe, 'Approximate Riemann solvers, parameter vectors, and difference schemes', *J. Comput. Phys.*, **43**, (1981).
12. P. L. Roe, 'A basis for upwind differencing of the two-dimensional unsteady Euler equations', in K. W. Morton and M. J. Baines (eds), *Numerical Methods for Fluid Dynamics II*, 1985, pp. 55–80.
13. P. L. Roe, 'Characteristic based schemes for the Euler equations', *Ann. Rev. Fluid Mech.*, 337–365 (1986).
14. D. Pan and S. R. Chakravarthy, 'Unified formulation for incompressible flows', *AIAA Paper 89-0122*, 1989.
15. C. Hirsch, *Numerical Computation of Internal and External Flows*, Vol. 2, *Computational Methods for Inviscid and Viscous Flows*, Wiley, Chichester, 1990.
16. D. Pan and H. Lomax, 'A new approximate LU factorization scheme for the Navier–Stokes equations', *AIAA J.*, **26**, 163–171 (1988).
17. R. J. Fennema and M. H. Chaudhry, 'Explicit methods for 2D transient free-surface flows', *J. Hydraul. Eng. ASCE*, **116**, 1013–1034 (1990).
18. G. S. Stelling, 'On the construction of computational methods for shallow water flow problems', *Ph.D. Thesis*, Delft University of Technology, 1983.
19. M. K. Denham and M. A. Patrick, 'Laminar flow over a downstream-facing step in a two-dimensional flow channel', *Trans. Inst. Chem. Eng.*, **52**, 361–367 (1974).
20. G. A. Akponasa, 'Solution of the contravariant shallow water equations using boundary-fitted coordinate system', *Ph.D. Thesis*, University of Oxford, 1992.
21. P. Tamamidis and D. N. Assanis, 'Evaluation of various high-order-accuracy schemes with and without flux limiters', *Int. j. numer. methods fluids*, **16**, 931–948 (1993).

