

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/228624466>

Unstructured Grid Finite-Volume Algorithm for Shallow-Water Flow and Scalar Transport with Wetting and Drying

Article in *Journal of Hydraulic Engineering* · April 2006

DOI: 10.1061/(ASCE)0733-9429(2006)132:4(371)

CITATIONS

231

READS

2,179

2 authors:



Lorenzo Begnudelli

FM Global

17 PUBLICATIONS 973 CITATIONS

[SEE PROFILE](#)



Brett Sanders

University of California, Irvine

141 PUBLICATIONS 9,092 CITATIONS

[SEE PROFILE](#)

Unstructured Grid Finite-Volume Algorithm for Shallow-Water Flow and Scalar Transport with Wetting and Drying

Lorenzo Begnudelli¹ and Brett F. Sanders²

Abstract: A high-resolution, unstructured grid, finite-volume algorithm is developed for unsteady, two-dimensional, shallow-water flow and scalar transport over arbitrary topography with wetting and drying. The algorithm uses a grid of triangular cells to facilitate grid generation and localized refinement when modeling natural waterways. The algorithm uses Roe's approximate Riemann solver to compute fluxes, a multidimensional limiter for second-order spatial accuracy, and predictor-corrector time stepping for second-order temporal accuracy. The novel aspect of the algorithm is a robust and efficient procedure to consistently track fluid volume and the free surface elevation in partially submerged cells. This leads to perfect conservation of both fluid and dissolved mass, preservation of stationarity, and near elimination of artificial concentration and dilution of scalars at stationary or moving wet/dry interfaces. Multi-dimensional slope limiters, variable reconstruction, and flux evaluation schemes are optimized in the algorithm on the basis of accuracy per computational effort.

DOI: 10.1061/(ASCE)0733-9429(2006)132:4(371)

CE Database subject headings: Algorithms; Shallow water; Water flow; Grid systems; Computation.

Introduction

One of the central challenges to modeling shallow-water flow with fixed grid finite-volume schemes is simulating wetting and drying. Bradford and Sanders (2002) varied the free surface reconstruction scheme near the wet/dry interface to prevent spurious wave generation at boundaries. Brufau et al. (2004) adopted a temporary modification of the bed elevation that renders the cell completely wet, in conjunction with a fairly complicated permutation of the source term upwinding technique first advocated by Bermudez and Vazquez (1994). Both Bradford and Sanders (2002) and Brufau et al. (2004) reported that during wave recession, overdraft of cells can sometimes occur (i.e., a negative depth is predicted), and adopted a correction step involving the recovery of overdrafted fluid from neighboring cells.

Another challenge is accurately simulating scalars as cells transition from wet to dry and vice versa. Finite volume schemes

update the solution in terms of conservative variables and therefore compute scalar concentration, ϕ , as the scalar mass per unit surface area ϕh divided by the fluid volume per unit surface area, h , or depth. Unrealistically large scalar concentrations are predicted when h vanishes from a cell at a faster rate than ϕh . Therefore, the key to accurately modeling both flow and scalar transport with finite volume schemes, while upholding conservation properties, is synchronizing the filling and draining of computational cells with the supply and removal of scalar mass, respectively. Arega and Sanders (2004) coupled the flow model of Bradford and Sanders (2002) with scalar transport equations to give good predictions of advection and dispersion in a tidal wetland. However, artificial dissipation of scalars at the wet/dry interface was necessary to suppress the prediction of unphysically large scalar concentrations in partially wetted cells.

In this paper, we present a novel and elegant procedure to accurately and efficiently model wetting and drying with an unstructured grid finite volume schemes based on triangular cells. Models based on unstructured grids of triangular cells are of particular interest for routine modeling purposes, for meshing of complex waterways characterized by irregular boundaries, islands, branching channels, etc., can be easily accomplished by Delaunay triangulation. At the heart of the wetting and drying procedure are algebraic equations that relate fluid volume to the free surface elevation in partially wetted cells. These volume/free-surface relationships (VFRs) greatly facilitate reconstruction of the free surface in partially wetted cells, which is necessary to compute fluxes. We describe the incorporation of VFRs into a finite volume scheme for unstructured grids that is similar to the model by Bradford and Sanders (2002), but could be easily applied in other models as well. We also carry out tests to identify slope limiting and variable reconstruction methods that optimize the model for accuracy relative to computational effort.

¹Visiting Researcher, Dept. of Civil and Environmental Engineering, Univ. of California, Irvine, CA 92697; Ph.D. Student, Dept. of Engineering, Univ. of Ferrara, Ferrara, Italy.

²Associate Professor, Dept. of Civil and Environmental Engineering, Univ. of California, Irvine, CA 92697 (corresponding author). E-mail: bsanders@uci.edu

Note. Discussion open until September 1, 2006. Separate discussions must be submitted for individual papers. To extend the closing date by one month, a written request must be filed with the ASCE Managing Editor. The manuscript for this paper was submitted for review and possible publication on October 25, 2004; approved on March 25, 2005. This paper is part of the *Journal of Hydraulic Engineering*, Vol. 132, No. 4, April 1, 2006. ©ASCE, ISSN 0733-9429/2006/4-371-384/\$25.00.

Governing Equations

The flow model is based on two-dimensional (2D) shallow-water equations, which are depth-integrated equations of motion consisting of a fluid continuity equation and two momentum equations, one for each horizontal coordinate direction. For consistency with the finite-volume method, which involves the direct discretization of the governing equations in integral form (Anderson 1995), the shallow-water equations are written as follows:

$$\frac{\partial}{\partial t} \int_{\Omega} \mathbf{U} d\Omega + \oint_{\partial\Omega} (\mathbf{F} dx - \mathbf{G} dy) = \int_{\Omega} \mathbf{S} d\Omega \quad (1)$$

where Ω represents a 2D domain (e.g., a computational cell) with a boundary $\partial\Omega$; and $\mathbf{U} = (h \ u \ v)^T$ = vector of flow variables. The flux terms \mathbf{F} , \mathbf{G} , and the source term \mathbf{S} are defined as

$$\mathbf{F} = \begin{pmatrix} uh \\ u^2h + \frac{1}{2}gh^2 \\ uvh \end{pmatrix}$$

$$\mathbf{G} = \begin{pmatrix} vh \\ uvh \\ v^2h + \frac{1}{2}gh^2 \end{pmatrix}$$

$$\mathbf{S} = \begin{pmatrix} 0 \\ -gh \frac{\partial z_b}{\partial x} - c_D u \sqrt{u^2 + v^2} \\ -gh \frac{\partial z_b}{\partial y} - c_D v \sqrt{u^2 + v^2} \end{pmatrix} \quad (2)$$

where u, v = depth-averaged flow velocities in the x, y directions, respectively; z_b = bed elevation; and c_D = bed drag coefficient that could be treated as a constant, computed from Manning's coefficient n_M as $c_D = gn_M^2 h^{-1/3}$, or computed locally based on the bed roughness and Reynolds number as follows (Haaland 1983):

$$c_D = \begin{cases} 0.2044 / \{\log[1.72/R + (k_s/14.8h)^{1.11}]\}^2 & \text{if } k_s/h < 0.2 \\ 1.56 \times 10^{-2} (k_s/h)^{1/3} & \text{if } k_s/h \geq 0.2 \end{cases} \quad (3)$$

where k_s/h = relative roughness and $R = \sqrt{u^2 + v^2}h/\nu$ = Reynolds number, where ν = kinematic viscosity.

Eqs. (2) may also be written in a differential form, in terms of the primitive variables, as follows:

$$\frac{\partial \mathbf{W}}{\partial t} + \mathbf{A}_W \frac{\partial \mathbf{W}}{\partial x} + \mathbf{B}_W \frac{\partial \mathbf{W}}{\partial y} = \mathbf{S}_W \quad (4)$$

where $\mathbf{W} = (h \ u \ v)^T$

$$\mathbf{A}_W = \begin{pmatrix} u & h & 0 \\ g & u & 0 \\ 0 & 0 & u \end{pmatrix}$$

$$\mathbf{B}_W = \begin{pmatrix} v & 0 & h \\ 0 & v & 0 \\ g & 0 & v \end{pmatrix}$$

$$\mathbf{S}_W = \begin{pmatrix} 0 \\ -g \frac{\partial z_b}{\partial x} - c_D h^{-1} u \sqrt{u^2 + v^2} \\ -g \frac{\partial z_b}{\partial y} - c_D h^{-1} v \sqrt{u^2 + v^2} \end{pmatrix} \quad (5)$$

This formulation ignores lateral stresses due to turbulence which may be significant when modeling recirculating flows; however, these stresses are typically ignored when modeling dam-break flows.

Scalars are modeled using depth-integrated transport equations that include advection, turbulent diffusion, dispersion, and source/sink terms that account for nonconservative processes. In integral form, transport equations for N arbitrary scalars, $\phi_i, i=1, \dots, N$ appear as

$$\frac{\partial}{\partial t} \int_{\Omega} \mathbf{Q} d\Omega + \oint_{\partial\Omega} (\mathbf{F}_Q dx - \mathbf{G}_Q dy) = \int_{\Omega} \mathbf{S}_Q d\Omega \quad (6)$$

where $\mathbf{Q} = (h\phi_1 h\phi_2 \dots h\phi_N)^T$ represents scalar concentration integrated over the depth, or scalar mass, the depth-integrated fluxes are given by

$$\mathbf{F}_Q = \begin{pmatrix} hu\phi_1 - h \left(E_{xx} \frac{\partial \phi_1}{\partial x} + E_{xy} \frac{\partial \phi_1}{\partial y} \right) \\ hu\phi_2 - h \left(E_{xx} \frac{\partial \phi_2}{\partial x} + E_{xy} \frac{\partial \phi_2}{\partial y} \right) \\ \vdots \\ hu\phi_N - h \left(E_{xx} \frac{\partial \phi_N}{\partial x} + E_{xy} \frac{\partial \phi_N}{\partial y} \right) \end{pmatrix}$$

$$\mathbf{G}_Q = \begin{pmatrix} hv\phi_1 - h \left(E_{yx} \frac{\partial \phi_1}{\partial x} + E_{yy} \frac{\partial \phi_1}{\partial y} \right) \\ hv\phi_2 - h \left(E_{yx} \frac{\partial \phi_2}{\partial x} + E_{yy} \frac{\partial \phi_2}{\partial y} \right) \\ \vdots \\ hv\phi_N - h \left(E_{yx} \frac{\partial \phi_N}{\partial x} + E_{yy} \frac{\partial \phi_N}{\partial y} \right) \end{pmatrix} \quad (7)$$

E_{xx}, E_{xy} , and E_{yy} account for turbulent diffusion and shear flow dispersion, and $\mathbf{S}_Q = (s_1 s_2 \dots s_N)^T$ = generalized source term vector.

Computational Grid

The governing equations are solved on an unstructured grid of triangular cells. The mesh is characterized by N_n nodes (or vertices), N_c cells, and N_f faces. Topography is defined by the bed elevation, $(z_b)_i$, at nodes with coordinates (x_i, y_i) , $i=1, \dots, N_n$. Organizing the neighborhood of cells, nodes, and faces within the mesh is critical to the success of unstructured grid models, so a neighborhood description is provided to clarify details of the numerical method presented later.

Integer mapping arrays define the neighborhood of nodes, faces, and cells surrounding each cell, as sketched in Fig. 1. Nodes surrounding each cell are mapped by $n_{i,k}^c$, where $i=1, \dots, N_c$ and $k=1, 2$, or 3 . This and all other maps reference with a counter-clockwise convention. Faces are mapped by $f_{i,k}^c$ and cells are mapped by $c_{i,k}^c$, where $i=1, \dots, N_c$ and $k=1, 2$, or 3 . Vectors pointing from the center to the midpoint of each face are

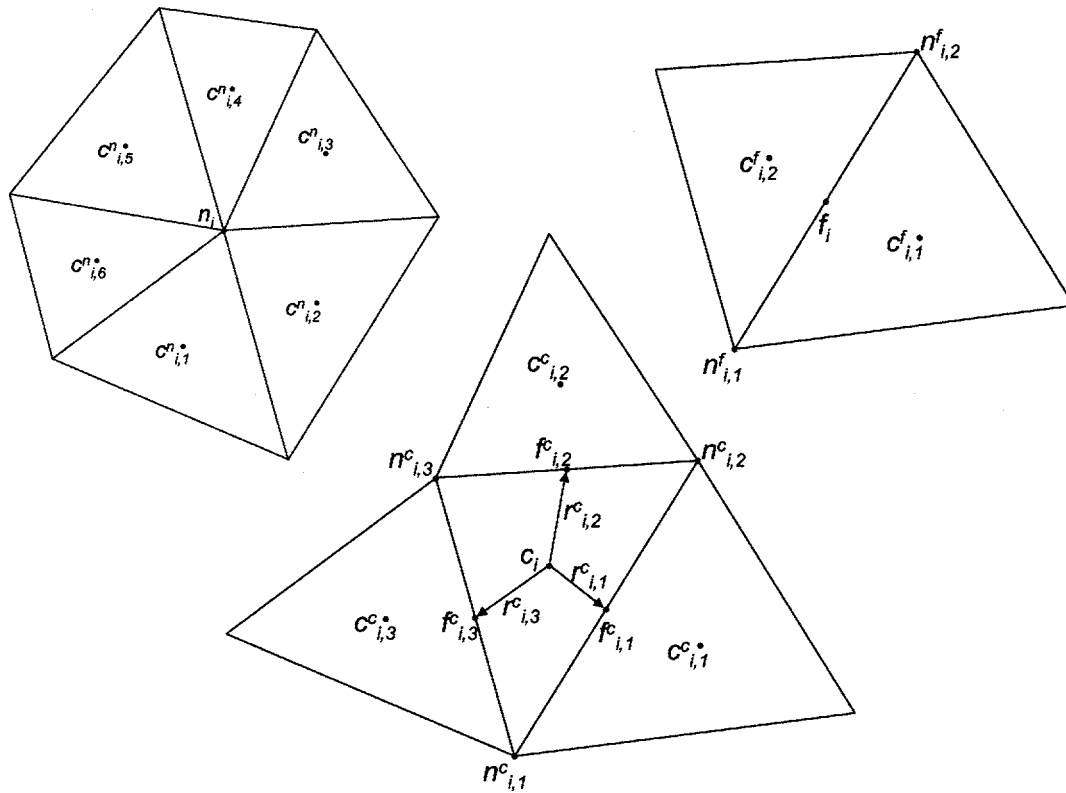


Fig. 1. Sketch of i th computational cell and neighborhood mapping functions

denoted by $\mathbf{r}_{i,k}^c$. Here, the superscript c implies that the map is relative to the cell (as opposed to the node or the face).

Maps that function relative to the cell face (Fig. 1) include $c_{i,m}^f$, $f_{i,m}^f$, and $n_{i,m}^f$, where $i=1, \dots, N_f$ and $m=1, 2$ and coordinate the cells to the left and right of the face, the nodes defining the end-points of each face, and the position of the face relative to the local coordinate system of each neighboring cell. Here, the superscript f implies that the map is relative to the face. Taking $n_{i,m}^f$ to define the node numbers at the top ($m=1$) and bottom ($m=2$) of each face, $c_{i,m}^f$ defines the cell number to the left ($m=1$) and right ($m=2$) of each face. $f_{i,m}^f$ is numbered 1, 2, or 3 depending upon the face's position in the local coordinate system of the left and right cells, and $n_{i,m}^f$ defines the node numbers associated with the top ($m=1$) and bottom ($m=2$) of each face. Unit vectors normal to faces, pointing left to right, are denoted by \mathbf{r}_i^f .

Considering the i th cell, the normal vectors for the three surrounding faces may be directed either inward or outward. The function $p_{i,k}^c$, where $i=1, \dots, N_c$ and $k=1, 3$ assumes a value of 1 or -1 depending on whether the unit vector normal to the k th face of the cell points outward or inward, respectively

$$p_{i,k}^c = \text{sign}(\mathbf{r}_{i,k}^c \cdot \mathbf{r}_{f_{i,k}^c}^f)$$

$$\text{sign}(a) = \begin{cases} +1 & \text{if } a > 0 \\ -1 & \text{if } a \leq 0 \end{cases} \quad (8)$$

The cells surrounding each node are mapped by $c_{i,m}^n$, $i=1, \dots, N_n$ and $m=1 \dots M_i$, as shown in the upper left part of Fig. 1. The number of cells that surround each node, M_i , typically varies in the range 5–10 across the mesh.

Model for Partially Submerged Cells

Finite-volume schemes associate the average depth in a cell, an indicator of fluid volume, with the cell centroid. However, in partially wetted cells, i.e., cells with enough fluid to submerge at least one vertex but not all, the average depth is badly represented by the depth at the centroid. For example, a cell may contain water while the free surface elevation is below the bed elevation of the centroid, z_c . Here, we make a distinction between the free surface elevation at the centroid and the depth at the centroid to improve the modeling of partially wetted cells. The flow depth h of each cell is defined to be the ratio of the fluid volume \mathcal{V} to the cell area \mathcal{A} , while η is defined to be the free surface elevation in the wetted portion of the cell. In wet cells $\eta = h + z_c$, but this equality does not hold in partially wetted cells. For these cases, the relationship between η and h is determined using VFRs.

Let the vertex coordinates of the i th cell be: (x_1, y_1, z_1) , (x_2, y_2, z_2) , (x_3, y_3, z_3) . In the equations of this section it will be assumed that $z_1 \leq z_2 \leq z_3$. If this is not the case, it can easily be achieved by a temporary permutation of the local indexes of the nodes. Furthermore, it is assumed that the free surface elevation is uniform in partially wetted cells. The VFR linking h , η , and the vertex elevations depends on whether zero, one, two, or three vertices are submerged as is illustrated in Fig. 2. If $\eta \leq z_1$ no fluid is in the cell, so $h=0$. If $z_1 < \eta \leq z_2$, the VFR is given by

$$h = \frac{\mathcal{V}}{\mathcal{A}} = \frac{(\eta - z_1)^3}{3(z_2 - z_1)(z_3 - z_1)} \quad (9)$$

or

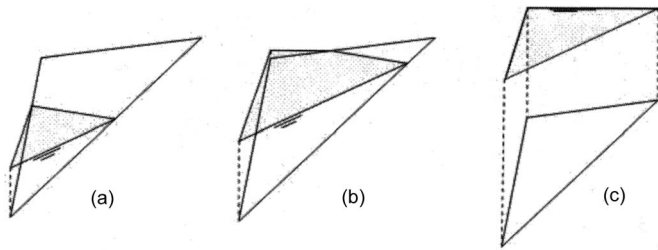


Fig. 2. Water volume in computational cell for: (a) $z_1 < \eta \leq z_2$; (b) $z_2 < \eta \leq z_3$; and (c) $z_3 < \eta$

$$\eta = z_1 + \sqrt[3]{3h(z_2 - z_1)(z_3 - z_1)} \quad (10)$$

If $z_2 < \eta \leq z_3$, the VFR is given by

$$h = \frac{\mathcal{V}}{\mathcal{A}} = \frac{\eta^2 + \eta z_3 - 3\eta z_1 - z_3 z_2 + z_1 z_2 + z_1^2}{3(z_3 - z_1)} \quad (11)$$

or

$$\eta = (1/2) \times (-\gamma_1 + \sqrt{\gamma_1^2 - 4\gamma_2}) \quad (12)$$

where

$$\gamma_1 = z_3 - 3z_1$$

and

$$\gamma_2 = 3hz_1 - 3hz_3 - z_3 z_2 + z_1 z_2 + z_1^2 \quad (13)$$

Another root to the quadratic equation for η given by Eq. (11) is $\eta = (1/2) \times (-\gamma_1 - \sqrt{\gamma_1^2 - 4\gamma_2})$, but it is always outside the correct interval $[z_2, z_3]$ so it is not physically meaningful. Finally, if $z_3 < \eta$ the VFR is given by

$$\eta = h + z_c \quad (14)$$

where

$$z_c = \frac{z_1 + z_2 + z_3}{3} \quad (15)$$

The VFRs above are applied in both the forward (compute h given η) and inverse (compute η given h) directions in the model, making it straightforward to toggle the solution state between the conservative form, which tracks storage within a computational cell, and the primitive form, which tracks the solution at discrete points. The forward VFRs are used to compute the initial condition in terms of h , given η . The inverse VFRs are used each time step to compute η after h is updated based on conservation equations. The model algorithm is presented in the following section to illustrate how VFRs are incorporated into the model and how the use of VFRs streamlines the overall algorithm.

Model Algorithm

The finite volume scheme used in this study is similar to that reported by Bradford and Sanders (2002). The model applies Roe's approximate Riemann solver to compute fluxes, and achieves second-order spatial accuracy using the monotone upwind scheme for conservation laws (MUSCL) approach of variable reconstruction (van Leer 1979). Hancock's predictor-corrector (van Albada et al. 1982) is adopted for efficient time stepping because fluxes need only be evaluated once per time step (Bradford and Sanders 2005). The adaptation of this approach for triangular cells, including slope limiting, reconstruction, and the source term discretization schemes, is described in the sections below in addition to the incorporation of VFRs. The overall design of the algorithm is shown in Fig. 3, where each box corresponds to a subroutine.

Wet/Dry Classification

The model adopts a wet/dry cell classification system to identify fully submerged cells, where both continuity and momentum

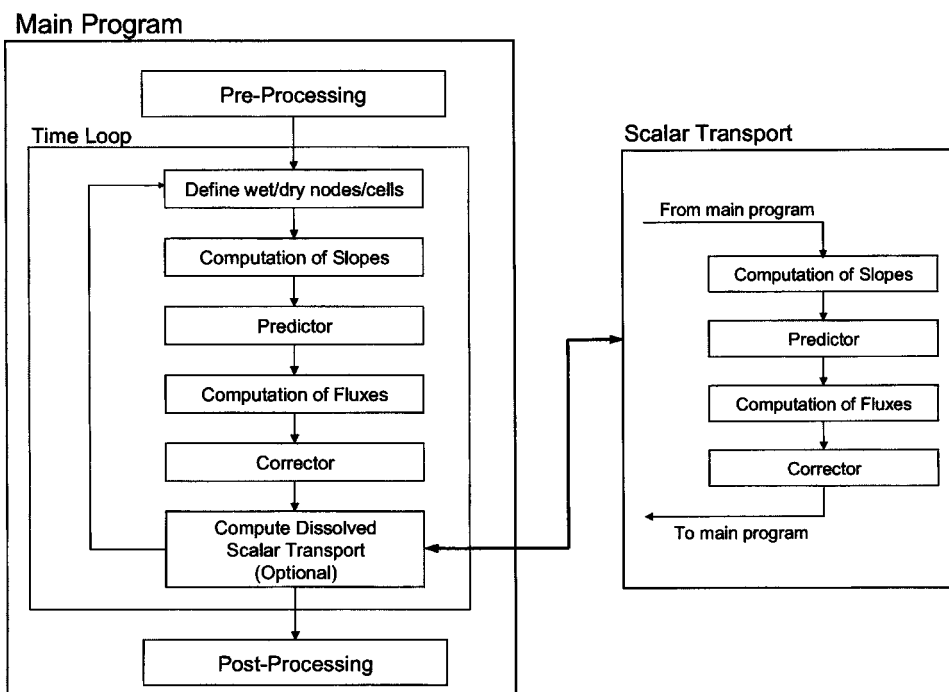


Fig. 3. Algorithm flowchart

equations are solved to update the solution. Cells are classified as wet when all three nodes are wet, and otherwise are classified as dry. Wet nodes are defined by a nodal depth $h_n > \delta^w$ where δ^w = small number set here to 10^{-6} m. Nodal depths are estimated by averaging the free surface elevation in wetted (though not necessarily “wet”) cells surrounding each node, and subtracting the bed elevation at the node. Wetted cells are characterized by an average depth $h > \delta^h$, where δ^h = small number set here to 10^{-3} m. Hence, on a sloping bed a cell containing a small volume of water may be wetted though not necessarily wet. The free surface elevation at the i th node is computed by

$$\eta_i^n = \frac{1}{M_i^n} \sum_{C_{i,k}^n \in C_i^n} \eta_{i,k}^n \quad (16)$$

where C_i^n represents the set of wetted cells surrounding the i th node, and M_i^n represents the number of wetted cells in the set. The nodal depth h_i^n follows as

$$h_i^n = \max(\eta_i^n - z_{bi}, 0) \quad (17)$$

Slope Evaluation

Slopes or gradients in the primitive variables η , u , and v are computed for each cell without regard for its wet/dry status, which streamlines the algorithm. Slopes are evaluated using limited central differencing (LCD) which involves two steps. Unlimited slopes are first computed, and then these are limited to prevent over- and undershoots.

The unlimited slope of a generic dependent variable q is based on its value q_1 , q_2 , and q_3 at three neighboring points, (x_1, y_1) , (x_2, y_2) , and (x_3, y_3) , respectively. The plane defined by these coordinates gives slopes as follows:

$$\begin{pmatrix} \delta_x q \\ \delta_y q \end{pmatrix} = \mathbf{J} \begin{pmatrix} q_2 - q_1 \\ q_3 - q_1 \end{pmatrix} \quad (18)$$

where

$$\mathbf{J} = \frac{1}{(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)} \begin{pmatrix} y_3 - y_1 & -y_2 + y_1 \\ -x_3 + x_1 & x_2 - x_1 \end{pmatrix} \quad (19)$$

Because the Jacobian matrix \mathbf{J} is only a function of grid coordinates, it is computed in a preprocessing step for each cell to minimize computational effort. The limited slopes $\delta_x q$, $\delta_y q$ are then computed from the unlimited slopes as follows:

$$(\overline{\delta_x q}, \overline{\delta_y q}) = \Phi(\delta_x q, \delta_y q) \quad (20)$$

where Φ = scalar limiting function.

The neighborhood around each cell includes many points including the centroid, three nodes, and the centroids of three surrounding cells. Therefore, many combinations of points are possible to compute the unlimited slopes, and there are several alternatives for computing the limiter Φ . The MLG method calls for four slopes from four different sets of points, limiting each for monotonicity purposes, and selecting the one with the largest magnitude. Anastasiou and Chan (1997) used a weighted average of four different slopes, as suggested by Pan and Cheng (1993). Hubbard (1999) concluded that the accuracy gained by using more than a simple three-point slope evaluation justified the computational effort for the case of scalar transport, but not for the case of flow prediction. For this model, we examined two different three-point options: the three centroids surrounding each cell

and the three vertices surrounding each cell. Since the solution is stored at cell centers, interpolation is necessary to compute the solution at vertices. We also considered two different strategies for limiting. One involves constraining the solution at the midpoint of faces, and the other involves constraining the solution at vertices. When the solution is constrained at face midpoints, Φ is computed as

$$\Phi = \min_{k=1,2,3}(\alpha_k) \quad (21)$$

where α_k are computed for each face as (Hubbard 1999)

$$\alpha_k = \begin{cases} \max(q_{i,k}^c - q_i, 0)/(q_{i,k} - q_i) & \text{if } q_{i,k} - q_i > 0 \\ \min(q_{i,k}^c - q_i, 0)/(q_{i,k} - q_i) & \text{if } q_{i,k} - q_i < 0 \\ 1 & \text{otherwise} \end{cases} \quad (22)$$

where $q_{i,k} = \mathbf{r}_{i,k}^c \cdot \nabla q$ represents the reconstructed value of q at the midpoint of the k th face of the i th cell, and $\nabla q = (\delta_x q, \delta_y q)$. This approach is actually quite similar to double minmod limiting in one dimension, since the limited slope is set to the centered difference slope so long as over- and undershoots do not occur, while the steepest possible monotone slope is selected otherwise. Sanders and Bradford (2006) reported that double minmod is the most accurate limiter for general purpose shallow-water flow and transport modeling with quadrilateral-cell based finite-volume schemes. The limiting approach of Barth and Jespersen (1989), which was later adopted by Anastasiou and Chan (1997) for shallow-water flow modeling, was also considered. However, oscillations in flow and scalar transport predictions were observed. For example, in a wet bed dam-break problem with $\phi=0$ on the downstream side and $\phi>0$ on the upstream side, negative concentrations were predicted at the leading edge of the contact discontinuity.

Constraining the solution at faces does not prohibit over- and undershoots at vertices, so constraining the solution at vertices was also examined. In this case, the unlimited solution is reconstructed at vertices, two values of α are computed for each vertex using Eq. (22), and Φ is given by the smallest of the six α values. Limiting at vertices instead of face midpoints can be expected to yield smaller values of Φ due to the added constraints.

Predictor Step

The solution is advanced from time t to $t+\Delta t/2$ in the predictor step. In cells classified as dry, the predictor is simply given by the base time level solution, $\eta^p = \eta$, $u^p = u$, and $v^p = v$, so the scheme reverts to first order accuracy. In wet cells, the solution is updated by solving the primitive equations [Eqs. (4)], which is computationally efficient compared to the conservative form because fluxes need not be computed (Bradford and Sanders 2005). The update equations for the i th cell are given by

$$\eta_i^p = \eta_i - \frac{\Delta t}{2} (u \overline{\delta_x h} + h \overline{\delta_x u} + v \overline{\delta_y h} + h \overline{\delta_y v})_i \quad (23)$$

$$u_i^p = u_i - \frac{\Delta t}{2} (u \overline{\delta_x u} + g \overline{\delta_x \eta} + v \overline{\delta_y u} + c_D h^{-1} u^p \sqrt{u^2 + v^2})_i \quad (24)$$

$$v_i^p = v_i - \frac{\Delta t}{2} (v \overline{\delta_x v} + g \overline{\delta_y \eta} + u \overline{\delta_x v} + c_D h^{-1} v^p \sqrt{u^2 + v^2})_i \quad (25)$$

where p indicates the half-time level solution. Discretization of the bed friction source term follows the approach of Wylie and

Streeter (1993), and is adopted for stability and computational efficiency.

The predictor updates the continuity equation in terms of η not h , which is consistent with use of the primitive, differential form of the shallow-water equations. Note that the predictor preserves stationary solutions defined by $u=v=0$ and $\nabla\eta=0$. The key is ensuring that the limited free-surface slope is zero in wet cells near the wet/dry interface, which is guaranteed when η is identical in two neighboring cells because the corresponding $\alpha_k=0$ based on Eq. (22).

Variable Reconstruction and Flux Computation

Fluxes are computed by Roe's method (Roe 1981) following reconstruction of the predictor at cell faces. Taking q to be a generic primitive variable, reconstruction of η^p , u^p , and v^p at the midpoints of the i th face is accomplished as follows:

$$q_L = q_{cf,i,1} + \mathbf{r}_{cf,i,1}^f \cdot (\nabla q)_{cf,i,1}^f \quad (26)$$

$$q_R = q_{cf,i,2} + \mathbf{r}_{cf,i,2}^f \cdot (\nabla q)_{cf,i,2}^f \quad (27)$$

where q_L and q_R correspond to the reconstructed variable on the left and right side of the cell face, respectively. Reconstruction of the depth on the left and right side of each face is obtained by subtracting z_b at the face midpoint from η_L and η_R , respectively. If either reconstructed depth is computed to be a negative number, it is set to zero. The flux for each face is the sum of the flux computed by Roe's method and a correction

$$\mathbf{F}^\perp = \mathbf{F}_{\text{Roe}}^\perp + \mathbf{F}_{\text{Corr}} \quad (28)$$

where the flux by Roe's method is written as

$$\mathbf{F}_{\text{Roe}}^\perp = \begin{pmatrix} hu^\perp \\ hu u^\perp + \frac{1}{2}gh^2 r_x^f \\ hv u^\perp + \frac{1}{2}gh^2 r_y^f \end{pmatrix} \quad (29)$$

where u^\perp =velocity perpendicular to the cell face; r_x^f and r_y^f =components of the vector \mathbf{r}_i^f along the x , y directions; and the flux correction is given by (Bradford and Sanders 2002)

$$\mathbf{F}_{\text{Corr}} = \begin{pmatrix} 0 \\ \frac{1}{12}g(\Delta h)^2 r_x^f \\ \frac{1}{12}g(\Delta h)^2 r_y^f \end{pmatrix} \quad (30)$$

where Δh represents the difference in h across the length of the face and is computed as

$$\Delta h_i = h_{n,i,1}^f - h_{n,i,2}^f \quad (31)$$

The flux correction is necessary to ensure a hydrostatic balance (stationarity) in the limit of a still water condition (Bradford and Sanders 2002). Another strategy for flux evaluation is to reconstruct the solution at the end points of each cell face, to apply Roe's method at each end point, and to compute the numerical flux by the trapezoidal rule as follows:

$$\mathbf{F}^\perp = \frac{1}{2}(\mathbf{F}_{\text{Roe}}^{\perp,1} + \mathbf{F}_{\text{Roe}}^{\perp,2}) - 2\mathbf{F}_{\text{Corr}} \quad (32)$$

where $\mathbf{F}_{\text{Roe}}^{\perp,1}$ and $\mathbf{F}_{\text{Roe}}^{\perp,2}$ correspond to the fluxes evaluated at the end points of the face.

The model does not consider the wet/dry classification of cells when computing fluxes. However, when the depth on either side of the cell face is less than a tolerance δ^w , the reconstructed ve-

locity is set to zero on that side before Roe's method is applied. Furthermore, when the depth on both sides is less than this tolerance, the fluxes are set to zero.

When these two methods of reconstruction are combined with the methods of slope computation and limiting presented earlier, eight different combinations of slope computation, limiting, and reconstruction schemes are possible. Using a three letter naming convention where the first indicates unlimited differencing based on centroid (C) or vertex (V) data, the second indicates limiting at face midpoints (M) or vertices (V), and the third indicates reconstruction either to face midpoints (M) or vertices (V), the eight schemes are given by CMM , CMV , CVM , CVV , VMM , VMV , VVM , and VVV . Of these, two were found to yield very poor results, CMV and VMV , presumably because the model does not prevent the introduction of new maxima or minima at vertices where fluxes are computed. Test problems presented later highlight the relative performance of the remaining alternatives.

Corrector Step

The solution is advanced from the base time level to the new time level in the corrector step by solving the conservative, integral form of the shallow-water equations [Eqs. (1)] for conservation purposes. Whereas the continuity equation is updated in all cells to advance the solution from t to $t+\Delta t$, the momentum equations are only updated in wet cells. In dry cells (which may be wetted), the velocity is set to zero. For more accurate predictions of wetting fronts, the velocity can be extrapolated as suggested by Bradford and Sanders (2002). The update equations appear as follows:

$$\mathbf{U}_i^{t+\Delta t} = \mathbf{U}_i^t + \frac{\Delta t}{\mathcal{A}_i} \left[- \sum_{k=1}^3 \mathbf{F}_{f_{i,k}}^\perp l_{f_{i,k}}^c p_{i,k}^c + \mathbf{S}_i \right]^{t+\Delta t/2} \quad (33)$$

where \mathbf{U}_i =vector of the conservative variables averaged over the i th cell; $\mathbf{F}_{f_{i,k}}^\perp$ =numerical flux normal to the k th face of the i th cell, whose length is $l_{f_{i,k}}^c$; and \mathbf{S}_i =source term vector integrated over the i th cell. Since bed elevation is assumed to vary linearly, the x momentum bed slope term is discretized exactly based on the depth at the centroid and the bed slope of each cell as follows:

$$\int_{\mathcal{A}_i} gh \frac{\partial z_b}{\partial x} d\mathcal{A} = g\mathcal{A}_i h_i \delta_x z_{bi} \quad (34)$$

while a similar approach is used for the y direction. In contrast, a bilinear approximation of bed elevation is typically adopted using quadrilateral cells. For an exact discretization in the limit of a still water condition, it is necessary to discretize the source term by applying 2×2 Gauss quadrature (Bradford and Sanders 2002), or to convert the area integral into a boundary integral (Valiani and Begnudelli 2005).

To preserve stationarity in the corrector step, there must be an exact balance of hydrostatic forces in wetted cells. This is ensured in the present model by incorporating the hydrostatic correction to the fluxes as shown in Eqs. (28) and (32), and discretizing the bed slope terms exactly as shown in Eq. (34). Many other approaches of balancing source terms and flux gradients have been presented in the literature and recently reviewed by Bradford and Sanders (2005), but the present approach is most straightforward. To clarify stationarity in the model, consider two neighboring cells along a sloping bank where the uphill direction is aligned with the positive x coordinate. Assume the cell on the left is classified as wet while the one on the right is classified as dry, though it is

actually partially wetted. If both cells take on the same free surface elevation at time t , η^t , then the limited slope in both cells will be $\bar{\delta}_x \eta = 0$, and based on Eqs. (26) and (27) the reconstructed values of η on both sides of the cell face will also be η^t . Hence, the flux will be computed such that stationarity is preserved. Use of VFRs ensures that η in partially wetted cells can take on any elevation including elevations below the bed at the centroid.

The corrector step may lead to an overdraft of a cell, i.e., removing more water than is stored as has been reported by Bradford and Sanders (2002) and Brufau et al. (2004). To maintain $h \geq 0$ in each cell, the overdraft is recovered from the neighboring cells in proportion to the amount of water taken by each neighbor. The volumetric fluxes are also corrected, in proportion to the storage correction, for consistency with the postcorrection balance of fluid and for later use in scalar transport predictions. The correction procedure involves sweeping over the three cells surrounding the overdrafted one (the i th) and searching for those cells $c_{i,k}^c$ that satisfy the following two properties:

1. Water flows from cell i to cell $c_{i,k}^c$

$$u_{f_{i,k}^c}^\perp p_{i,k}^c > 0 \quad (35)$$

2. The storage in cell $c_{i,k}^c$ exceeds the overdraft

$$\mathcal{A}_{c_{i,k}^c} h_{c_{i,k}^c} > \mathcal{A}_i |h_i| \quad (36)$$

Let \mathcal{C}_i^c be the set of cells surrounding the i th that satisfy the two aforementioned conditions. The flow depth in these cells and the volumetric fluxes are corrected as follows:

$$(h)_{c_{i,k}^c} = (h)_{c_{i,k}^c}^* + w_{i,k} h_i \frac{\mathcal{A}_i}{\mathcal{A}_{c_{i,k}^c}} \quad (37)$$

$$(pu^\perp h)_{f_{i,k}^c} = (pu^\perp h)_{f_{i,k}^c}^* + \frac{w_{i,k} h_i \mathcal{A}_i}{\Delta t} \quad (38)$$

where the weights $w_{i,k}$ are

$$w_{i,k} = \frac{(pu^\perp h)_{f_{i,k}^c}^*}{\sum_{c_{i,k}^c \in \mathcal{C}_i^c} [(pu^\perp h)_{f_{i,k}^c}^*]} \quad (39)$$

The superscript $*$ in Eqs. (37) and (38) refers to the solution based on Eq. (33), before the overdraft correction is applied. After h is corrected in neighboring cells, h_i is set to zero so mass is conserved. The update procedure is complete following the overdraft procedure. Subsequently, the inverse VFRs are applied to compute η at the new time level.

Due to the explicit nature of the scheme, stability is dictated by the Courant-Friedrichs-Levy (CFL) condition. For a triangular grid, this appears as follows (Batten et al. 1996)

$$\text{Cr} = \Delta t \max_{i=1, \dots, N_c} \left\{ \frac{3 \max_{k=1,2,3} |\lambda_{f_{i,k}^c}^\perp l_{f_{i,k}^c}^c|}{\mathcal{A}_i} \right\} \leq 1 \quad i = 1, \dots, N_c \quad (40)$$

where Cr =Courant number and $\lambda_{f_{i,k}^c}^\perp$ =wave speed normal to the k th face of the i th cell. In test problems presented here, the model was run with a constant time step corresponding to $\text{Cr} \approx 0.6$.

Scalar Transport

Scalar transport equations are solved on the same grid, on the same time levels, and applying the same predictor-corrector ap-

proach as for the flow (see Fig. 3). Since the focus of this paper is flow and transport in the presence of wetting and drying, we are concerned with the problem of numerical concentration and/or dilution and therefore we only consider advection. For insight into unstructured grid discretization of diffusive terms, see Anastasiou and Chan (1997).

In the predictor step, the solution is advanced from time t to time $t + \Delta t/2$ by solving the primitive, differential form of the solute transport equations as follows:

$$\phi_i^p = \phi_i - \frac{\Delta t}{2} (u \bar{\delta}_x \phi_i + v \bar{\delta}_y \phi_i) \quad (41)$$

To compute advective fluxes, the volumetric flux at the interface is multiplied by the reconstructed concentration ϕ on the upstream side of the face. Scalar transport equations are updated after the corrector step for the flow equations, as shown in Fig. 3, to incorporate the overdraft corrected volumetric fluxes. The solution \mathbf{Q}_i is then advanced from time t to $t + \Delta t$ in a corrector step where the integral form of the scalar transport equations [Eqs. (6)] is solved

$$\mathbf{Q}_i^{t+\Delta t} = \mathbf{Q}_i^t + \frac{\Delta t}{\mathcal{A}_i} \left[- \sum_{k=1}^m (\mathbf{F}_Q^\perp)_{f_{i,k}^c}^c l_{f_{i,k}^c}^c p_{i,k}^c + \mathbf{S}_{\mathbf{Q}_i} \right]^{t+\Delta t/2} \quad (42)$$

where $(\mathbf{F}_Q^\perp)_{f_{i,k}^c}^c$ represents the advective flux normal to the k th face.

There are many combinations of slope computation, limiting, and reconstruction schemes that can be adopted to solve scalar transport equations, and these need not necessarily mirror the approach used to solve the flow equations. Test problems are presented later to evaluate the performance of *CMM*, *CVM*, *CVV*, *VMM*, *VVM*, and *VVV*.

The corrector step updates ϕh in each cell, requiring division by h to compute ϕ . As h becomes small, $(\phi h)/h$ becomes singular and its computation may cause overflow or significant round-off errors leading to poor estimates of ϕ . Therefore, when h is less than a user specified tolerance $\delta^\phi = 10^{-6}$, ϕ is set equal to ϕ in neighboring cells with $h > \delta^\phi$. If there are multiple wetted neighbors, the cell with the largest volume is used. If there are no neighbors with $h > \delta^\phi$, ϕ is set to a reference value ϕ° . However, ϕh is not changed so that scalar mass is conserved. The model does not consider the wet/dry status of the solution in any other way when updating the scalar transport equations.

Boundary Conditions

Boundary conditions are enforced on the corrector solution by computing fluxes at boundaries as described by Anastasiou and Chan (1997). Boundary conditions are enforced on the predictor vis-à-vis slopes of dependent variables in boundary cells. When slopes are based on vertex data, the solution at boundary vertices is computed by interpolation from two neighboring centroids. When slopes are based on centroid data, the three points used to estimate the slope are the centroid and two neighboring centroids inside the model domain.

Model Tests

Test problems were considered to determine optimal slope computation, limiting, and reconstruction schemes for use in the algorithm, to verify conservation, monotonicity, and stationarity of predictions, to examine the accuracy of flow predictions in wet-

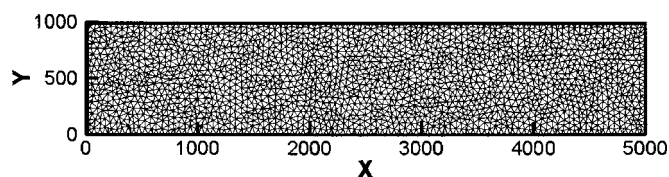


Fig. 4. Unstructured grid generated by Delaunay triangulation

ting and drying applications, and to examine accuracy of scalar transport predictions in the presence of wetting and drying.

All model grids in this study were generated by constrained Delaunay triangulation using software developed by Shewchuk (1996), which can be readily obtained on-line. The software allows angle and cell area constraints to be enforced when triangulation is performed, and for the present study a minimum angle of 30° and a relatively uniform cell size were used. For example, a mesh used for several test problems is shown in Fig. 4. This was generated with an area constraint of $2,000 \text{ m}^2$. We did not consider test cases involving localized grid refinement, even though this is a chief advantage of Delaunay meshing. For example, the mesh could be refined in areas where bottom topography is more irregular, a more accurate prediction is required, or where shocks occur. This topic is left for a future study.

Optimization of Slope Computation/Limiting/Reconstruction Schemes

Steady and unsteady flow problems were considered to identify the slope computation, limiting, and reconstruction methods which lead to the greatest accuracy per computational effort. The most diagnostic of these were found to be: (1) the problem of steady transcritical flow over a bump with a shock; (2) the prob-

lem of a dam break on a dry downstream bed, and (3) the problem of a dam break on a wet downstream bed. 1D problems serve as excellent tests of a Delaunay grid model because the grid cannot be oriented in the direction of flow to improve accuracy, unlike models based on quadrilateral cells.

First, the problem of transcritical flow over a bump with a shock is considered. The spatial domain in this problem is a $25 \text{ m} \times 1 \text{ m}$ rectangular channel, the bed is assumed to be frictionless, and bed elevation is specified as follows:

$$z_b(x) = \begin{cases} 0.2 - 0.05(x - 10)^2 & \text{if } 8 \text{ m} \leq x \leq 12 \text{ m} \\ 0 & \text{otherwise} \end{cases} \quad (43)$$

At the upstream boundary, a unit discharge of $Uh = 0.18 \text{ m}^2 \text{ s}^{-1}$ was specified, while at the downstream boundary, a flow depth $h = 0.33 \text{ m}$ was specified. A constrained Delaunay mesh consisting of 4,004 triangular cells was used. The model was integrated 250,000 time steps to a steady state using a time step $\Delta t = 0.002 \text{ s}$ and either *CMM*, *CVM*, *CVV*, *VMM*, *VVM*, or *VVV*. Fig. 5 shows the cell centroid value of η in all cells versus the x coordinate of the cell centroid. The first two columns of Table 1 present errors in terms of $L_1(\eta) = \{\sum_{i=1, \dots, N_c} |\eta_i - \bar{\eta}|\} / N_c$ and $L_1(Uh) = \{\sum_{i=1, \dots, N_c} |(Uh)_i - U\bar{h}|\} / N_c$, where η and Uh represent the exact solutions. Other error norms were also measured but these offered no additional insight. These results show little difference except for *CMM*, which performs noticeably poorer than the other schemes. In particular, *CMM* diffuses the downstream side of the shock which can be observed both graphically (Fig. 5) and quantitatively (Table 1). These results also show that slope computation based on vertex data requires roughly 9% more computational effort than using cell centroid data, for interpolation is required; limiting at vertices requires roughly 6% more computational effort, since the limiter is applied twice as many times; and

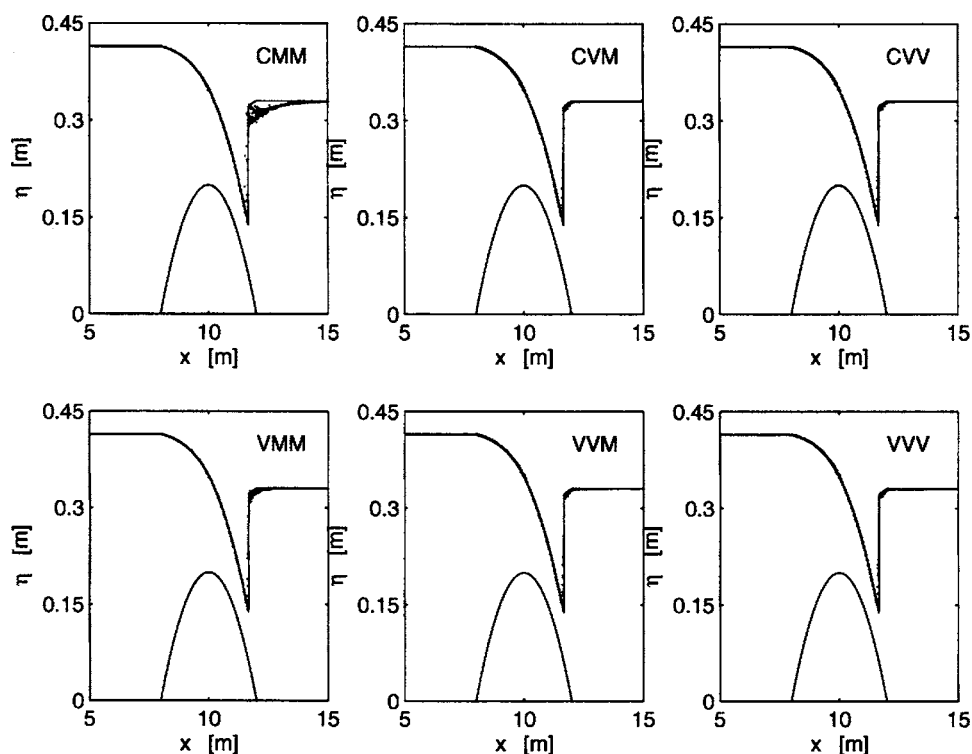


Fig. 5. Water depth in transcritical flow problem: comparison of slope computation/limiting/reconstruction methods. Exact solution and bottom elevation given by solid lines.

Table 1. L_1 Error Norms on Free-Surface Elevation and Unit Discharge, and Relative Run Time (*CMM* Taken as Reference) versus Slope Computation, Limiting, and Reconstruction Methods

	L_1 error norms				Relative run time
	Transcritical flow over a bump with a shock		Dry bed dam-break problem	Wet bed dam-break problem	
	$L_1(\eta)$	$L_1(Uh)$	$L_1(\eta)$	$L_1(\eta)$	
<i>CMM</i>	1.63×10^{-3}	1.38×10^{-2}	1.85×10^{-2}	2.27×10^{-2}	1.00
<i>CVM</i>	7.60×10^{-4}	2.67×10^{-3}	4.01×10^{-2}	4.86×10^{-2}	1.06
<i>CVV</i>	7.57×10^{-4}	2.64×10^{-3}	4.13×10^{-2}	4.94×10^{-2}	1.22
<i>VMM</i>	5.56×10^{-4}	3.75×10^{-3}	1.93×10^{-2}	2.33×10^{-2}	1.09
<i>VVM</i>	7.60×10^{-4}	3.94×10^{-3}	4.01×10^{-2}	4.87×10^{-2}	1.15
<i>VVV</i>	7.58×10^{-4}	2.46×10^{-3}	4.14×10^{-2}	4.94×10^{-2}	1.33

reconstruction at vertices requires roughly 16% more computational effort, since fluxes are computed twice as many times. On the basis of accuracy per computational effort, *VMM* and *CVM* perform best in this problem.

Second, a dry bed dam-break problem is considered. The spatial domain used here consists of a 5,000 m \times 1,000 m rectangular area; the dam is located at $x=2,500$ m, and the bed is horizontal and frictionless. At the initial time, the flow depth is set to 5 m upstream of the dam, whereas downstream the depth is zero. The mesh consists of 3,880 triangular cells, and a time step $\Delta t=0.125$ s is used. Flow predictions were made using *CMM*, *CVM*, *CVV*, *VMM*, *VVM*, and *VVV*, and the results at time $t=150$ s are shown in Fig. 6. The second column of Table 1 presents $L_1(\eta)$. In this problem, *CMM* and *VMM* perform better than the other schemes. Errors primarily arise at the leading edge of the depression wave, which propagates in the negative x direc-

tion, and also at the leading edge of the wetting front. However, *CMM* and *VMM* can also be observed to track the curvature of the expansion wave better than the other methods. Neumann extrapolation of the velocity was also considered for wetted (though not wet) cells instead of setting the velocity to zero (Bradford and Sanders 2002), but this was found to be only slightly more accurate.

Third, a wet bed dam break problem is considered. The problem setup is identical to the dry bed problem presented earlier, except the initial depth downstream of the dam is set to 0.5 m and a larger time step is used, $\Delta t=0.2$ s. Flow predictions were made using *CMM*, *CVM*, *CVV*, *VMM*, *VVM*, and *VVV*, and the results at time $t=250$ s are shown in Fig. 7. The third column of Table 1 presents $L_1(\eta)$. In this problem, *CMM* and *VMM* perform better than the other schemes in terms of error norms, resolving the leading edge of the depression wave, and sharply resolving the shock. All of the schemes exhibit slight oscillations between the dam and the shock. However, the oscillations are most pronounced in *CMM*. Since limiting based on vertices is more restrictive than limiting based on midpoints, the added diffusion of vertex based limiting appears to degrade the prediction at the leading edge of the depression wave but improve it between the dam and the shock.

On the basis of both accuracy and computational efficiency as shown in Table 1, these results point to *VMM* as being the optimal scheme. This approach requires roughly 9% more computational effort than *CMM*, the least time consuming scheme, but its predictions are noticeably better than *CMM* in the transcritical flow problem, and no worse in the other test problems.

The wet bed dam break problem is now used to examine the performance of *CMM*, *CVM*, *VMM*, and *VVM* for scalar transport, while *VMM* is used for the flow equations. Advective fluxes are computed as the concentration on the upstream side of a face

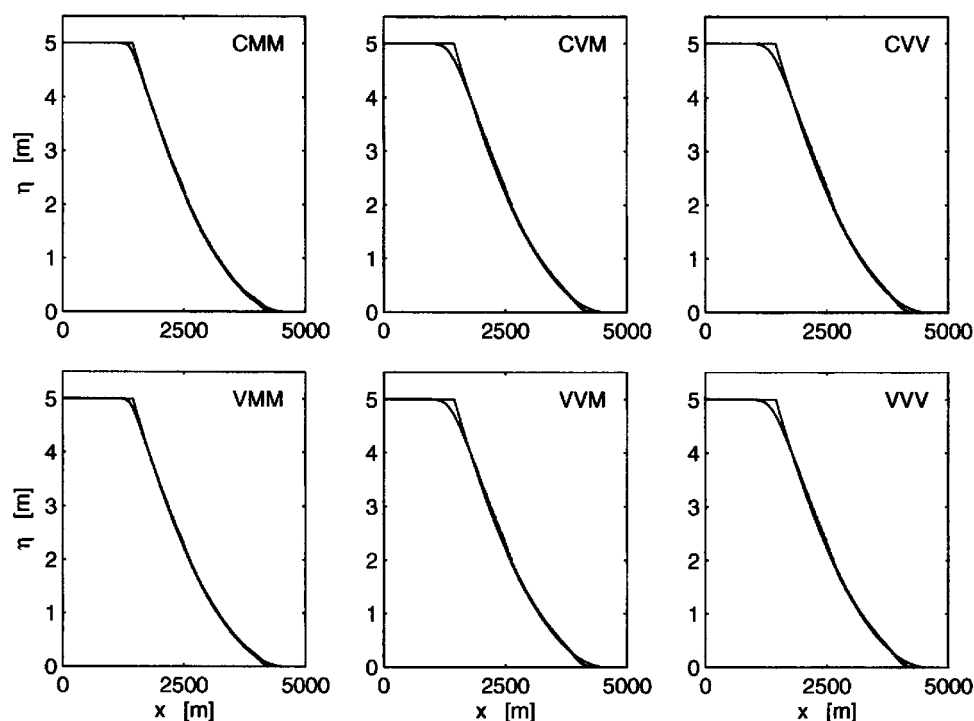


Fig. 6. Water depth in dry bed dam break problem: comparison of slope computation/limiting/reconstruction methods at time $t=150$ s. Exact solution given by solid line.

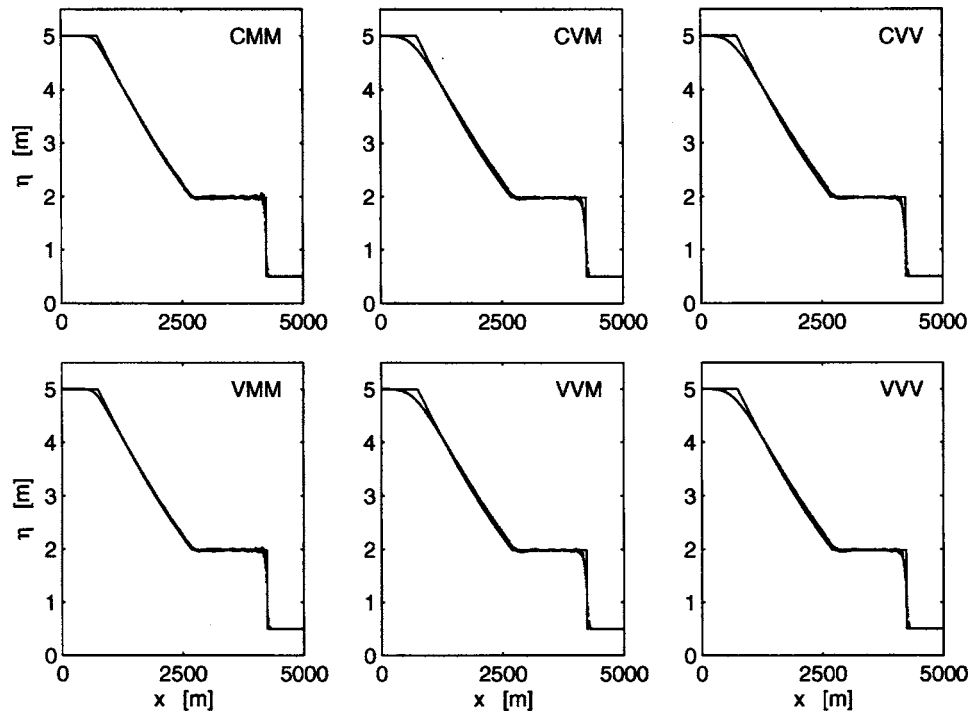


Fig. 7. Water depth in wet bed dam break problem: comparison of slope computation/limiting/reconstruction methods at time $t=250$ s. Exact solution given by solid line.

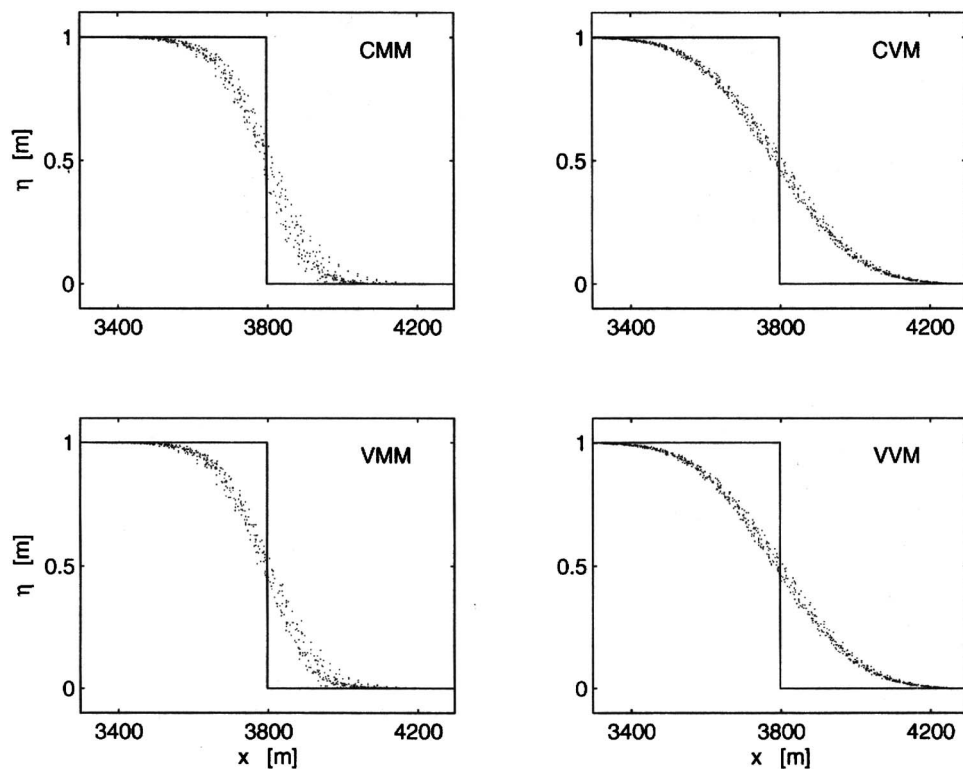


Fig. 8. Scalar concentration near contact discontinuity in wet bed dam break problem: comparison of slope computation/limiting/reconstruction methods at time $t=250$ s. Exact solution given by solid line.

Table 2. $L_1(\phi)$ Error Norms in Wet Bed Dam-Break Problem versus Slope Computation and Limiting Method

	$L_1(\phi)$
<i>CMM</i>	1.59×10^{-2}
<i>CVM</i>	2.69×10^{-2}
<i>VMM</i>	1.66×10^{-2}
<i>VVM</i>	2.70×10^{-2}

multiplied by the volumetric flow rate across the face, so *CVV* and *VVV* are identical to *CVM* and *VVM*, respectively; hence, *CVV* and *VVV* are not considered. The problem setup is as described above, except that $\phi=1$ upstream of the dam and $\phi=0$ downstream. As before, the model is integrated 1,250 time steps using $\Delta t=0.2$ s. Fig. 8 shows the solution near the contact discontinuity predicted by each scheme, and Table 2 presents $L_1(\phi)$. The *CMM* and *VMM* predictions are similar, and so are *CVM* and *VVM*. The former two predict steeper fronts than the latter two, but also predict greater scatter in ϕ . Scatter is due to variations in the position and slope of the front across the width of the channel, which is caused by irregularities in the Delaunay grid. However, it is important to note that the scatter does not include any prediction of $\phi > 1$ or $\phi < 0$. When the limiting method described by Anastasiou and Chan (1997) was tested, the model predicted values of $\phi > 1$ and $\phi < 0$ so this approach is not considered viable.

Based on the results of this test, the optimal limiting approach depends upon whether a smooth prediction or a sharp prediction is more important. If smoothness is paramount, then *CVM* is optimal because the computational effort is less than *VVM*. If sharpness is paramount, then *CMM* is optimal because the computational effort is less than *VMM*.

Model Performance in Wetting and Drying Problems with Scalar Transport

Several tests were utilized to measure the performance of the model in problems involving wetting and drying and scalar transport. Tests were selected to (1) verify that the model preserves stationary solutions; (2) verify that the model is able to predict flow over uneven topography with wetting and drying, and examine whether over- and undershoots in scalar concentration occur

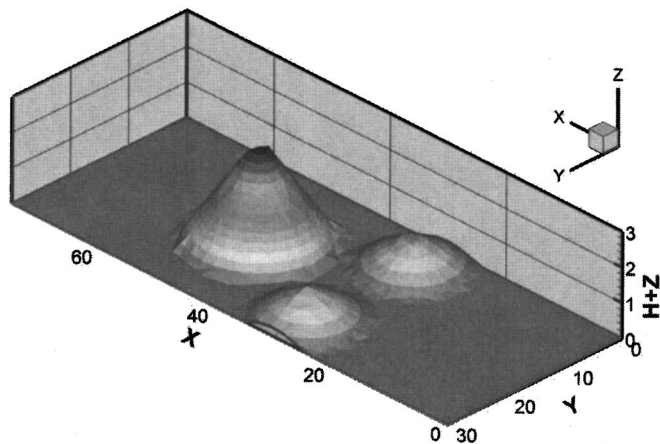


Fig. 9. Topography used for three-mound problem on grid of triangular cells

under such conditions; (3) highlight the benefits of VFRs over a similar scheme that does not use VFRs; and (4) validate the model in a problem involving friction.

To address (1), (2), and (3), flow is examined in a channel with three mounds as shown in Fig. 9. A similar problem has been used by other researchers (Kawahara and Umetsu 1986, Brufau et al. 2002; Brufau and Garcia-Navarro 2003), though the precise geometry has not been made clear. Here, the three mounds are located at $(x,y)=(30,6)$, $(30,24)$, and $(47.5,15)$ m, the heights of the three mounds are set to 1, 1, and 3 m, respectively, and bed elevation is assumed to decrease linearly with radial distance from the centroid at a rate of 1:8, 1:8, and 3:10, respectively. At all four boundaries, a wall condition was imposed.

For item (1), the model was tested for stationarity by specifying the following initial condition:

$$\eta = 0.5 \text{ m}$$

$$U = V = 0 \text{ ms}^{-1}$$

$$\phi = 1 \quad (44)$$

and then integrating the model for 10^6 time steps with $\Delta t=0.01$ s, using *VMM* and *CVM* for flow and scalar transport, respectively. Initially, it was observed that after 10^2 time steps the magnitude of U was roughly 10^{-14} ms^{-1} , but that after 10^5 time steps the magnitude of U had grown to roughly 10^{-3} ms^{-1} . This test was repeated using the model of Bradford and Sanders (2002), and the magnitude of U was roughly 10^{-12} ms^{-1} after 10^6 time steps. Upon further investigation, an accumulation of round-off error was observed in the computation of slopes by Eq. (18). By setting the $\nabla q=0$ when $|\nabla q|$ is computed to be less than 10^{-14} , and repeating this test, the magnitude of U was computed to be roughly 10^{-12} ms^{-1} after 10^6 time steps.

For item (2), the model was applied to simulate flow and scalar transport in a dry-bed dam-break flow over the three mounds, to evaluate how well the model conserves both fluid volume and scalar mass, and to evaluate whether the model predicts artificial concentration and/or dilution of the scalar. For item (3), this test was repeated with VFRs removed from the code. That is, the free surface elevation was assumed to be the sum of the flow depth and the bed elevation at the centroid.

In the model, a dam was positioned at $x=16$ m, and the initial conditions were set as $\eta=1.875$ m and $\phi=1$ upstream of the dam. The model was integrated using a time step $\Delta t=0.01$ s for a total of 8 s; *VMM* and *CVM* were adopted for flow and scalar transport, respectively. The reference concentration was set to zero, $\phi^0=0$. Fig. 10 illustrates the solution in terms of the water surface (A), velocity vectors (B), and scalar concentration (C). The left and right side of Fig. 10 shows model predictions with and without VFRs, respectively. Both numerical predictions appear smooth and free from unrealistically large velocities near the wet/dry interface, but while flow and scalar mass are perfectly conserved using VFRs (errors of roughly 10^{-15} for each), mass conservation errors of about 0.14% result with VFRs are not used. Moreover, when VFRs are used, contours in ϕ show only very small errors near the wet/dry interface (which indicate negligible artificial concentration and/or dilution). When VFRs are not used, the model introduces significant artificial concentration and dilution (roughly 10%). Fig. 11 presents over- and undershoots of ϕ in all cells versus depth, h , using VFRs. This figure shows that over- and undershoots always remain smaller than 2×10^{-3} , and in most cases are smaller than 10^{-5} . Where flow depth is larger, over- and undershoots are generally smaller and slightly less fre-

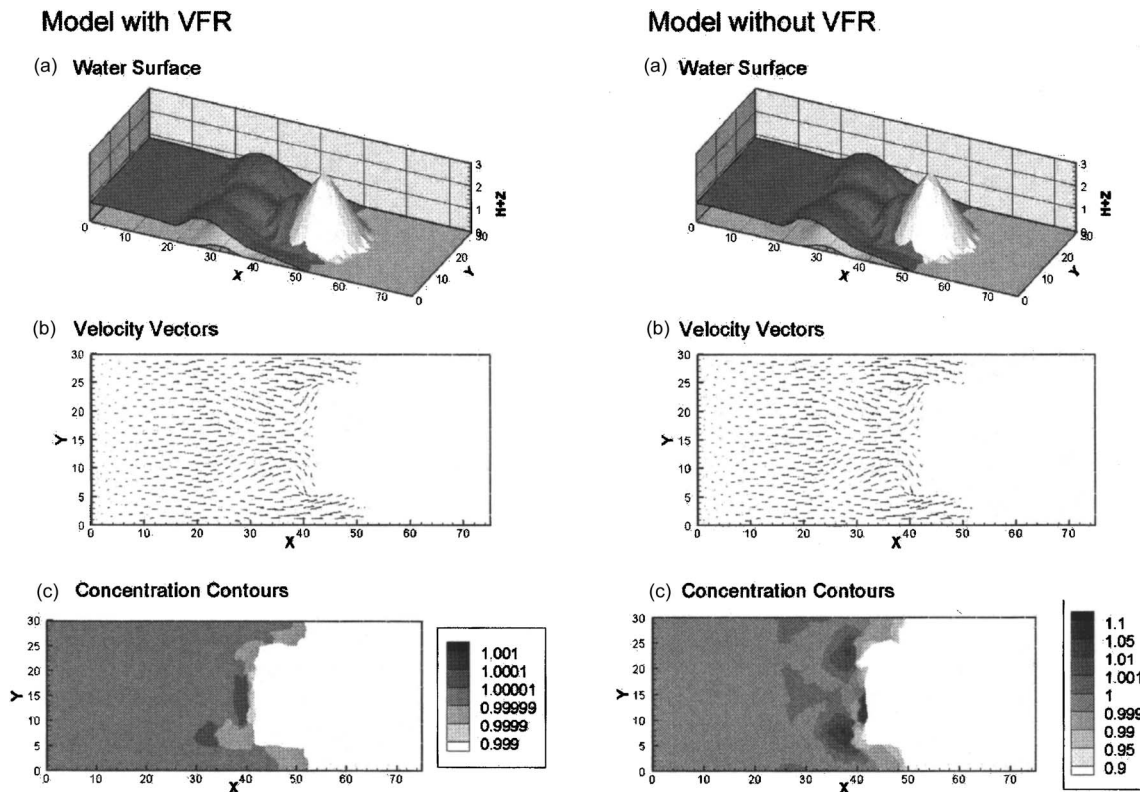


Fig. 10. Dam break over dry bed with three mounds: water surface (a), velocity vectors (b), and scalar concentration (c) at $t=8$ s, using VFRs (left) and without using VFRs (right)

quent. This test was repeated using $\phi^0=1$, but no change in the over- and undershoots was observed. By measuring CPU time of runs with and without VFRs incorporated into the code, it was found that VFRs required 5.6% more CPU time.

Using VFRs, the model conserves both fluid volume and scalar mass perfectly, despite the over- and undershoots. For a point of comparison, Arega and Sanders (2004) reported that while the Bradford and Sanders (2002) model conserves fluid volume perfectly, scalar mass could not be conserved without artificially concentrating scalars at the wet/dry interface. To prevent this, Arega and Sanders (2004) set the scalar mass to zero when the fluid volume in a cell drops below a small number.

Another test problem was utilized to examine model performance in the context of drainage, where wetted cells slowly become dry as water migrates down a sloping bed. This is a classic problem in hydrology, often addressed by solving a kinematic wave equation (Bedient and Huber 2002), and it serves as a useful limiting condition for a shallow-water model. The problem considered here involves a $5,000 \text{ m} \times 1,000 \text{ m}$ rectangular cross section channel sloping downhill in the x direction. It is assumed that a reservoir is initially present at the high end of the channel, a breach occurs instantaneously at $t=0$ s, an avalanche of water flows down hill, and water pools at the downstream end of the channel. The volume of water in the downstream pool should eventually match the initial volume of water, which is the basis of the present test. Moreover, we assume that the reservoir initially contains a scalar $\phi=1$. Scalar mass should also be conserved, and ϕ should not deviate from unity. With wetting and drying, however, a measureable amount of concentration/dilution is expected. For this test, two different channel slopes were considered: $\partial z_b / \partial x = -10^{-3}$ and -10^{-2} , the bed was assumed to be frictionless, the dam was placed at $x=1,000 \text{ m}$, and the initial water level was

set such that $5,000 \text{ m}^3$ /(per meter width) was initially present in the reservoir. The reference concentration was set to zero, $\phi^0=0$. The model was integrated 5×10^4 time steps using *VMM* and *CVM* for flow and scalar transport, respectively. A time step $\Delta t=0.2$ and 0.1 s was used for $\partial z_b / \partial x = -10^{-3}$ and -10^{-2} , respectively. After this period of time, a small volume of water remained on the bank. It was observed that this volume slowly decreased with time, but remained finite. This residual fluid was denoted “artificial depression storage” (ADS) borrowing an expression

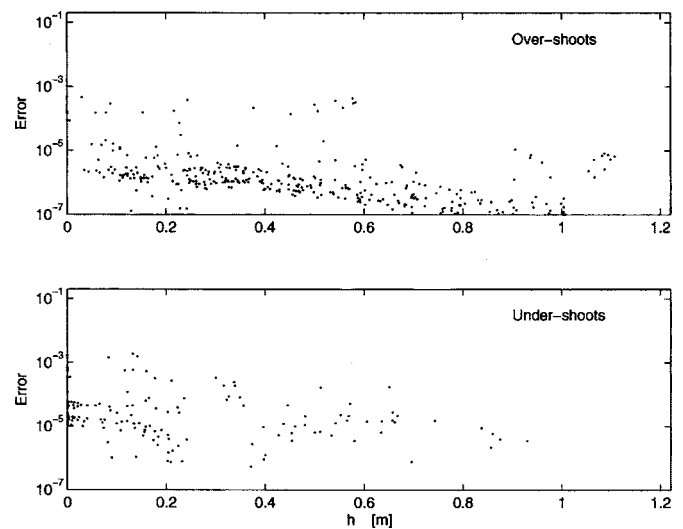


Fig. 11. Dam break over dry bed with three mounds: under- and overshoots of ϕ versus h at $t=8$

Table 3. Sloping Bed Dam Break Problem: Artificial Depression Storage and Overshoots (Superscript +) and Undershoots (Superscript -) in Prediction of Scalar Concentration Sorted by Whether They Occur on Draining Bank (Subscript *b*) Or in Collection Pool (Subscript *p*)

	$\partial z / \partial x = -10^{-2}$	$\partial z / \partial x = -10^{-3}$
h_{ads}	7.2×10^{-3} m	7.1×10^{-4} m
ε_p^+	$< 10^{-15}$	$< 10^{-15}$
ε_p^-	$< 10^{-15}$	$< 10^{-15}$
ε_b^+	$< 10^{-15}$	2.8×10^{-13}
ε_b^-	$< 10^{-15}$	4.1×10^{-13}

from hydrology that describes water trapped in small depressions in the ground surface. This residual was quantified as the average depth on the bank, h_{ads} (volume on bank divided by bank surface area), and is shown in Table 3. h_{ads} was found to be roughly 10^{-3} m on the 10^{-3} slope and 10^{-2} m on the 10^{-2} slope. Using a finer mesh, these errors became smaller indicating a sensitivity to grid resolution. Errors in scalar predictions were also quantified in terms of overshoots, ε^+ , and undershoots, ε^- , and these were found to be negligible as is shown in Table 3. However, larger over- and undershoots resulted when using larger values of δ^ϕ , while no over- or undershoots could be detected when the reference concentration was wet to unity, $\phi_0 = 1$. Larger ADS resulted from using the *CMM* scheme for flow, while similar values resulted from using schemes that compute fluxes at vertices (e.g., *CVV*). Hence, the *VMM/CVM* approach for flow and transport, respectively, was found to perform as well or better than the other slope computation/limiting/reconstruction combinations in drainage applications. When this problem was tested using the model of Bradford and Sanders (2002), the ADS was considerably smaller so we attribute ADS to ponding in the lowest corner of cells, where the free surface elevation may be below the elevation of the nearest face midpoints.

Model Performance in Problem with Friction

Finally, to validate the model in an application involving friction, the model was applied to predict the flow conditions observed in a partial dam-break experiment reported by Betcheler et al. (1999) and previously simulated by Brufau and Garcia-Navarro (2003). The experimental configuration consists of a reservoir 7 m wide and 2 m long separated by a dam from a platform 7 m wide and 2 m long. The dam included a trapezoidal slot that was suddenly opened, producing a partial dam break, and propagating a flood wave across the platform and over a spillway. The initial water depth in the reservoir was reported to be 0.2 m, and the Manning coefficient was reported to be $n_M = 0.01 \text{ m}^{-1/3} \text{ s}$. Using an unstructured grid of 5,205 cells, flow was simulated using a time step $\Delta t = 10^{-3} \text{ s}$ and the wetted region downstream of the dam was measured at $t = 0.66, 1.02, 1.37, 1.71, 2.08, 2.44$, and 2.78 s for comparison with the observations by Betcheler et al. (1999) and predictions by Brufau and Garcia-Navarro (2003). The results are presented in Table 4, and illustrate that model predictions compare well with experimental data, which validates the modeling approach relative to bed resistance. The test is not conclusive relative to the superiority of one method over the other. Errors are smaller than those reported by Brufau and Garcia-Navarro (2003) over the first half of the simulation, and the same or slightly larger over the second half. These differences are undoubtedly influenced by the structure and resolution of grids used by these models, in addition to the numerical schemes.

Table 4. Dam Break Through Trapezoidal Breach: Numerical Prediction of Area Submerged by Advancing Front Compared with Numerical Results of Brufau and Garcia-Navarro (2003) and Experimental Results of Betcheler et al. (1999)

Time (s)	Wetted area (m^2)			Relative error	
	Experimental data	Present model	Brufau and Garcia-Navarro	Present model (%)	Brufau and Garcia-Navarro (%)
0.66	0.98	0.99	0.94	1.0	4.1
1.02	2.10	2.12	2.18	1.0	3.8
1.37	3.71	3.60	3.86	3.0	4.0
1.71	5.58	5.43	6.18	2.7	10.8
2.08	8.25	7.71	8.70	6.5	5.5
2.44	10.70	10.21	11.15	4.6	4.2
2.78	13.05	12.54	13.52	3.9	3.6

Additional information on the ability of shallow-water models to simulate dam-break floods is reported by Morris (2000).

Summary and Conclusions

Volume/free-surface relationships, or VFRs, were introduced to consistently account for fluid volume and the free surface elevation in partially wetted cells in the context of an unstructured grid finite-volume scheme. The resulting algorithm was found to accurately predict steady and unsteady shallow-water flow and scalar transport over uneven topography with wetting and drying. The model was shown to preserve stationarity, perfectly conserve fluid volume and scalar mass, and predict scalars with only negligible amounts of artificial concentration/dilution ($< 0.1\%$ at most, but typically $< 0.001\%$). The success of the model is attributed to VFRs, which build upon the excellent local and global conservation properties of the finite-volume method but also facilitate free surface reconstruction in cells that are not fully submerged with water. The cost of the VFRs is roughly 6% added run time. It is expected that VFRs can be incorporated into many other unstructured grid finite volume schemes to improve performance in wetting and drying applications.

A number of strategies to compute and limit slopes of dependent variables using a three point stencil, reconstruct variables at cell faces, and estimate fluxes were considered to optimize model performance. On the basis of accuracy per computational effort, *VMM* was found to be optimal for the flow equations and *CVM* for scalar transport equations. *VMM* involves interpolating the cell centered solution to vertices by the inverse distance method, computing unlimited slopes of dependent variables based on vertex data, limiting to prevent new maxima or minima at face midpoints, and reconstructing the solution and computing fluxes at midpoints. *CVM* involves unlimited slopes of dependent variables based on data at neighboring centroids, limiting to prevent new maxima or minima at vertices, and reconstructing the solution and computing fluxes at face midpoints.

Several tolerances were introduced to define wet vertices (δ^w), wetted cells ($h > \delta^h$), and cells where ϕ can be computed as $\phi h / h$, ($h > \delta^h$). In the present study, these tolerances were set as follows: $\delta^w = 10^{-6} \text{ m}$, $\delta^h = 10^{-3} \text{ m}$, and $\delta^\phi = 10^{-6} \text{ m}$. Larger values of δ^ϕ resulted in greater over- and undershoots of scalar concentration near wet/dry interfaces. One deficiency identified in the model is that sloping beds retain a thin film of fluid as water levels recede. This is attributed to ponding in the lowest corner of

cells, which occurs when the free surface elevation is below the elevation of the two nearest face midpoints. The thickness of this film was found to increase with bed slope, decrease with grid resolution, but it was not found to be sensitive to δ^w or δ^h . The thickness of the film using VMM was roughly the same or smaller than when using other limiting/reconstruction schemes.

The algorithm presented here is attractive for general purpose flow modeling for several reasons including: (1) it uses triangulated grids that can be easily generated for waterways of complex geometries; (2) all shallow-water flow regimes (subcritical, supercritical, etc.) are accurately predicted; (3) scalar transport is accurately predicted; and (4) the model is applicable to arbitrary topography and problems with wetting and drying.

Acknowledgment

This work was supported by a grant from the National Science Foundation (Grant No. CMS-9984579), whose support is gratefully acknowledged.

References

- Anastasiou, K., and Chan, C. T. (1997). "Solution of the 2D shallow water equations using the finite volume method on unstructured triangular meshes." *Int. J. Numer. Methods Fluids*, 24, 1225–1245.
- Anderson, J. D. (1995). *Computational fluid dynamics*, McGraw-Hill, New York.
- Arega, F., and Sanders, B. F. (2004). "Dispersion model for tidal wetlands." *J. Hydraul. Eng.*, 130(8), 739–754.
- Barth, T. J., and Jespersen, D. C. (1989). "The design and application of upwind schemes on unstructured meshes." *AIAA Pap.*, 89–0366.
- Batten, P., Lambert, C., and Causon, M. D. (1996). "Positively conservative high resolution convection schemes for unstructured elements." *Int. J. Numer. Methods Eng.*, 39, 1821.
- Bedient, P. B., and Huber, W. C. (2002). *Hydrology and floodplain analysis*, Prentice-Hall, Upper Saddle River, N.J.
- Bermudez, A., and Vazquez, M. E. (1994). "Upwind methods for hyperbolic conservation laws with source terms." *Comput. Fluids*, 23, 1049–1071.
- Betcheler, W., Kulish, H., and Nujic, M. (1999). "2-D dam-break flooding waves comparison between experimental and calculated results." *Int. Rep. Inst. of Hydromechanics and Hydraulics*, Univ. of the Federal Armed Forces, Munich, Germany.
- Bradford, S. F., and Sanders, B. F. (2002). "Finite-volume model for shallow-water flooding of arbitrary topography." *J. Hydraul. Eng.*, 128(3), 289–298.
- Bradford, S. F., and Sanders, B. F. (2005). "Performance of high-resolution, non-level bed, shallow water models." *J. Eng. Mech.*, 131(10), 1073–1081.
- Brufau, P., and García-Navarro, P. (2003). "Unsteady free-surface flow simulation over complex topography with a multidimensional upwind technique." *J. Comput. Phys.*, 186, 503–526.
- Brufau, P., García-Navarro, P., and Vázquez-Cendón, M. E. (2002). "A numerical model for the flooding and drying of irregular domains." *Int. J. Numer. Methods Fluids*, 39, 247–275.
- Brufau, P., García-Navarro, P., and Vázquez-Cendón, M. E. (2004). "Zero mass error using unsteady wetting-drying conditions in shallow flows over dry irregular topography." *Int. J. Numer. Methods Fluids*, 45, 1047–1082.
- Haaland, S. E. (1983). "Simple and explicit formulas for the friction factor in turbulent pipe flow." *ASME J. Fluids Eng.*, 105, 89–90.
- Hubbard, (1999). "Multidimensional slope limiters for MUSCL-type finite volume schemes on unstructured grids." *J. Comput. Phys.*, 155, 54–74.
- Kawahara, M., and Umetsu, T. (1986). "Finite element method for moving boundary problems in river flow." *Int. J. Numer. Methods Fluids*, 6, 365–386.
- Morris, M. W. (2000). "CADAM concerted action on dambreak modeling." *Rep. No. SR 571*, HR Wallingford.
- Pan, D., and Cheng, J. C. (1993). "Upwind finite-volume Navier-Stokes computation unstructured triangular meshes." *AIAA J.*, 31(9), 1618.
- Roe, P. L. (1981). "Approximate Riemann solvers, parameter vectors, and difference schemes." *J. Comput. Phys.*, 43, 357–372.
- Sanders, B. F., and Bradford, S. F. (2006). "Impact of limiters on accuracy of high-resolution flow and transport models." *J. Eng. Mech.*, 132(1), 87–98.
- Shewchuk, J. R. (1996). "Triangle: engineering a 2D quality mesh generator and Delaunay triangulator." *Lect. Notes Comput. Sci.*, 1148, 203–222 (<http://www-2.cs.cmu.edu/~quake/triangle.html>).
- Valiani, A., and Begnudelli, L. (2005). "Divergence form for bed slope source term (DFB) in shallow water equations." *J. Hydraul. Eng.*, in press.
- van Albada, G. D., van Leer, B., and Roberts, W. W. (1982). "A comparative study of computational methods in cosmic gas dynamics." *Astron. Astrophys.*, 108, 76–84.
- van Leer, B. (1979). "Towards the ultimate conservation difference scheme. V. A second-order sequel to Godunov's method." *J. Comput. Phys.*, 32(1), 101–136.
- Wylie, E. B., and Streeter, V. L. (1993). *Fluid transients in systems*, Prentice-Hall, Upper Saddle River, N.J.