Journal of Computational Physics 228 (2009) 7452–7472

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

# A primitive-variable Riemann method for solution of the shallow water equations with wetting and drying

P. Sivakumar, D.G. Hyams, L.K. Taylor, W.R. Briley *

*SimCenter: National Center for Computational Engineering, University of Tennessee at Chattanooga, Chattanooga, TN 37403, USA*

ARTICLE INFO

*Article history:*
Received 14 November 2008
Received in revised form 1 July 2009
Accepted 2 July 2009
Available online 10 July 2009

*PACS:*
47.11.Df
47.85.Dh
47.35.Bb

*Keywords:*
Shallow water equations
Wet/dry front
Riemann flux
Primitive variables
Boundary conditions
Parallel implicit algorithm
Unstructured grid
Gulf of Mexico

ABSTRACT

A Riemann flux that uses primitive variables rather than conserved variables is developed for the shallow water equations with nonuniform bathymetry. This primitive-variable flux is both conservative and well behaved at zero depth. The unstructured finite-volume discretization used is suitable for highly nonuniform grids that provide resolution of complex geometries and localized flow structures. A source-term discretization is derived for nonuniform bottom that balances the discrete flux integral both for still water and in dry regions. This primitive-variable formulation is uniformly valid in wet and dry regions with embedded wetting and drying fronts. A fully nonlinear implicit scheme and both nonlinear and time-linearized explicit schemes are developed for the time integration. The implicit scheme is solved by a parallel Newton-iterative algorithm with numerically computed flux Jacobians. A concise treatment of characteristic-variable boundary conditions with source terms is also given. Computed results obtained for the one-dimensional dam break on wet and dry beds and for normal-mode oscillations in a circular parabolic basin are in very close agreement with the analytical solutions. Other results for a forced breaking wave with friction interacting with a sloped bottom demonstrate a complex wave motion with wetting, drying and multiple interacting wave fronts. Finally, a highly nonuniform, coastline-conforming unstructured grid is used to demonstrate an unsteady simulation that models an artificial coastal flooding due to a forced wave entering the Gulf of Mexico.


## 1. Introduction

Computational methods for solving the shallow water equations (SWE) are useful to model and simulate water flowing with a free surface under the influence of gravity. Some relevant applications include tidal flows, flows in rivers, streams and estuaries, and open-channel flows. Of particular interest here is the use of Riemann methods for simulation of wave motions in irregular geometries having nonuniform bathymetry and moving wet/dry fronts.

Toro has given thorough accounts of Riemann solvers for both the Euler [1] and SWE [2] equations. The approximate Riemann solver of Roe [3] is widely used and was applied to the one-dimensional SWE by Glaister [4]. The SWE with Roe fluxes have also been solved by Whitfield [5] using curvilinear coordinates and an iterative implicit algorithm, by Anastaciou and Chan [6] using an unstructured triangular mesh and implicit time integration, and by Fujihara and Borthwick [7] using a four-stage Runge–Kutta method. Ambrosi [8] and Nujic [9] noted that a nonuniform bottom causes a source-term imbalance and consequent numerical difficulty using Roe's scheme, even with still water. Numerical techniques for treating this still-water source-term balancing problem have been considered by Leveque [10], Vázquez-Cendón [11], Hubbard and García-Navarro [12], García-Navarro and Vázquez-Cendón [13], Zhou et al. [14], Gascón and Corberán [15], and Burguete

* Corresponding author. Tel.: +1 423 425 5487; fax: +1 423 425 5517.
*E-mail address:* roger-briley@utc.edu (W.R. Briley).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.07.002

and García-Navarro [16]. A different approach that reformulates the equations to incorporate the still-water source-term balance was given by Rogers et al. [17] and applied to coastal current interactions by Rogers et al. [18].

Wetting and drying of the bottom surface are also of interest in many practical applications. Toro [2] gives a detailed treatment of the Riemann problem for a wet/dry interface. The solution is a contact discontinuity with discontinuous velocity front whose propagation speed is faster than system eigenvalues. Toro notes that numerical computation of wet/dry interfaces is very difficult. Computational difficulties arise because the Roe flux is indeterminate at zero average depth, and the use of conserved variables $(h, hu, hv)$ requires a division by depth to compute velocity at cell centers. This behavior can cause computational inaccuracy for small depth, and special numerical techniques are normally required.

Sleigh et al. [19] used an unstructured finite-volume scheme and a modified Roe flux that was limited near wet/dry interfaces using two minimum depth tolerances that identify cells as wet, dry and partially dry, and cell-faces as wet or dry. Dodd [20] employed a modified Roe flux and a fixed-grid tracking scheme at wet/dry boundaries. Bradford and Sanders [21] employed an explicit predictor/corrector scheme and used Roe fluxes with a flux correction term and a bilinear/Gauss quadrature of the bottom-slope source term to maintain still-water equilibrium. The velocities were not updated in cells whose depth is below a specified threshold value, and a velocity extrapolation technique replaced the momentum equation in partially filled cells. Yoon and Kang [22] used an HHL approximate Riemann solver with an explicit Runge–Kutta scheme and implicit treatment of frictional source terms to alleviate numerical instability at small depths. Murillo et al. [23–25] noted that a discrete mass imbalance can arise using the Roe flux when a subcritical wetting front moves over an adverse bed slope. They developed a wetting–drying condition, equivalent to a local redefinition of bed slopes, which modifies Roe's scheme to overcome the generation of negative depth and concentration when fronts advance over variable bed levels. Another modification mitigates a restrictive explicit stability condition for new dry areas, to allow stability with a maximum time step for minimum computational cost. Implicit diffusion terms were added by a splitting technique, and application examples indicate that the method is effective.

The present study includes the development of a primitive-variable Riemann flux for use in conservative finite-volume discretizations. Previous conservative primitive-variable Riemann flux formulations were developed for the compressible Euler and Navier–Stokes equations (e.g., Turkel [26], Briley et al. [27]), primarily in connection with preconditioned methods for low Mach-number flows. Toro and Siviglia [28] have developed several primitive-variable centered schemes for hyperbolic systems that are written in non-conservative form. Their schemes were constructed to be non-oscillatory and are applicable to solutions that include smooth regions, contact discontinuities, and weak shocks, as demonstrated in tests for the one-dimensional shallow water equations. Another method using primitive variables for non-conservative equations is the three-dimensional hydrostatic shallow water model that reduces to the shallow water equations for a single vertical layer. Casulli et al. [29,30] developed an efficient semi-implicit finite-difference method applicable to wetting and drying and unstructured finite-volume discretizations. Their semi-implicit momentum equations treat surface elevation derivatives and horizontal viscous terms implicitly and all other terms explicitly. The free-surface continuity equation becomes a two-dimensional linear, implicit, second-order, symmetric, positive-definite equation, solved by conjugate gradient iteration. Velocity components are then obtained by two scalar-tridiagonal solutions at each vertical column of grid points.

The present computational method employs a finite-volume discretization and introduces a conservative primitive-variable Riemann flux that is well behaved at zero average depth. A primary attraction of Riemann flux approximations is their effectiveness for solutions with discontinuities. A discretization for nonuniform bottom is also introduced that balances the discrete flux integral and source term for both still water and completely dry regions. These features provide straightforward treatment of embedded moving wetting and drying fronts. Unstructured grids are used to conform to irregular geometries or coastlines and to provide resolution of localized flow structures. Both Euler explicit and backward-differentiation implicit solution algorithms are developed for the primitive-variable formulation. The implicit algorithm has accuracy up to second-order in time and third order in space. Characteristic-based boundary conditions with source terms and the influence of friction on wet/dry fronts are also addressed.

In the remainder of this paper, a nondimensional formulation of the governing equations is given in Section 2, and the primitive-variable Riemann flux is developed in Section 3. Section 4 develops the finite-volume and source-term balancing discretizations and boundary conditions, and the time-integration schemes are given in Section 5. Section 6 gives computed results for the dam-break problem on wet and dry beds, for an oscillating wave in a parabolic basin, and for a forced breaking wave interacting with a sloped beach. Section 7 tests the functionality of the method for an irregular Gulf of Mexico bathymetry, and conclusions are given in Section 8.

## 2. Governing equations

Derivations of the shallow water equations have been given by Weiyan [31], and Toro [2]. The notation used here is given in Fig. 1, where the free-surface elevation is expressed as the sum of water depth $h(x, y, t)$ and bottom elevation $b(x, y)$. The still-water equilibrium condition has constant surface elevation, but the depth $h_s(x, y)$ is not constant if the bottom is nonuniform. The equations in conservation form contain source terms such as $gh\partial b/\partial x$ for nonuniform bottom, and this term is analytically balanced by a flux derivative term $\partial\left(\frac{1}{2}gh^2\right)/\partial x$. However, the still-water equilibrium condition is not satisfied in discrete form when using the Roe flux. This has led several researchers [10–16] to develop special numerical techniques to maintain a discrete source-term balance. Rogers et al. [17,18] have written the Roe fluxes and source term in a special form

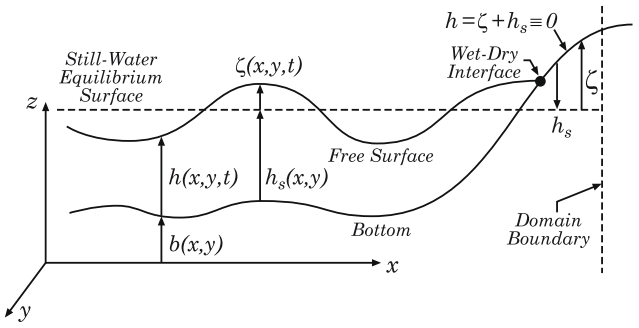


**Fig. 1.** Definition of free-surface variables with wetted surface.

that maintains still-water equilibrium scheme for problems with nonuniform bottom. Their formulation describes the free-surface elevation as a difference $\zeta(x,y,t)$ from the still-water depth $h_s(x,y)$ above the bottom, as shown in Fig. 1. It follows that $h = \zeta + h_s$ and $\zeta = 0$ at equilibrium, for which $h = h_s$. The present development allows for possible wet and dry regions within the solution domain, and the primitive-variable scheme and boundary conditions are constructed so that the solution reduces to $h = 0$ and $\zeta = -h_s$ on any dry surface (cf. Fig. 1).

The equations are written in nondimensional form as follows:

$$\frac{\partial \boldsymbol{Q}}{\partial t} + \frac{\partial \boldsymbol{F}(\boldsymbol{Q})}{\partial x} + \frac{\partial \boldsymbol{G}(\boldsymbol{Q})}{\partial y} + \boldsymbol{S}(\boldsymbol{Q}) = 0 \tag{1}$$

Here, $\boldsymbol{Q} = (h, hu, hv)^T$ is the vector of dependent variables, $\boldsymbol{F}(\boldsymbol{Q})$ and $\boldsymbol{G}(\boldsymbol{Q})$ are the flux functions, and $\boldsymbol{S}$ is a source term. These are given by

$$\boldsymbol{F} = \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}(h^2 - h_s^2) \\ huv \end{pmatrix} \quad \boldsymbol{G} = \begin{pmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}(h^2 - h_s^2) \end{pmatrix} \tag{2}$$

$$\boldsymbol{S} = \begin{pmatrix} 0 \\ -(h - h_s)\frac{\partial h_s}{\partial x} + \tau_{bx} \\ -(h - h_s)\frac{\partial h_s}{\partial y} + \tau_{by} \end{pmatrix} \tag{3}$$

where $(u, v)$ are the depth-averaged $(x, y)$ velocity components, $t$ is time, and $\tau_{bx}$ and $\tau_{by}$ are frictional stresses on the bottom. Terms representing Coriolis forces, frictional stresses on the free surface, atmospheric pressure variations, and viscous forces acting in the horizontal plane are omitted.

The nondimensional formulation given here incorporates dimensional similarity to simplify the equations by eliminating dimensional parameters for gravity and density. All equations and variables in this paper are nondimensional unless otherwise noted. Nondimensional variables are defined using dimensional reference quantities denoted by a prime. All lengths are normalized by a representative depth $d'$, and velocity components are normalized by $c' = \sqrt{g'd'}$, the celerity of a gravity wave, where $g'$ is gravitational acceleration. The nondimensional variables are defined by

$$x = \frac{x'}{d'}, \quad y = \frac{y'}{d'}, \quad h = \frac{h'}{d'}, \quad h_s = \frac{h'_s}{d'}, \quad u = \frac{u'}{c'}, \quad v = \frac{v'}{c'}, \quad t = \frac{c't'}{d'}, \quad \tau = \frac{\tau'}{\rho' c'^2}$$

where $\rho'$ is the fluid density. The nondimensional celerity is $c = \sqrt{h}$, so that a gravity wave propagates at unit speed when $h = 1$. The local Froude numbers are $u/\sqrt{h}$ and $v/\sqrt{h}$.

Solutions of (1)–(3) using Roe's scheme will satisfy the still-water equilibrium condition, for which $u = v = 0$, and $\zeta = h - h_s = 0$. For example, setting $u = v = 0$ and omitting bottom friction reduces (1)–(3) to

$$0 = 0, \quad \frac{\partial}{\partial x}\frac{(h^2 - h_s^2)}{2} = (h - h_s)\frac{\partial h_s}{\partial x}, \quad \frac{\partial}{\partial y}\frac{(h^2 - h_s^2)}{2} = (h - h_s)\frac{\partial h_s}{\partial y} \tag{4}$$

As discussed in detail by Rogers et al. [18], these flux gradient and source terms are in balance (both vanish) at the still-water depth $h = h_s$. Note that the flux terms in Eqs. (1)–(3) are equivalent to those in [18] because of the identity $h^2 - h_s^2 \equiv \zeta^2 + 2\zeta h_s$.

Note that in a dry region with $h = 0$, the flux gradient and source terms in (4) further reduce to

$$\frac{\partial}{\partial x}\frac{(-h_s^2)}{2} = -h_s\frac{\partial h_s}{\partial x} \tag{5}$$

These terms are analytically equivalent but do not generally balance in discrete form for a nonuniform bottom. For rectangular grids, a centered spatial approximation will produce a balance for a linear bottom of constant slope. For arbitrary

bottom surfaces, however, some means for ensuring a discrete balance is needed to preserve the zero-depth equilibrium state. A discretization that provides an exact balance for both still water and dry regions is given in Section 4.3.

## 3. Primitive-variable Riemann flux

It is convenient to develop the primitive-variable Riemann flux in a one-dimensional example for which

$$\frac{\partial \mathbf{Q}_i}{\partial t} + \frac{1}{\Delta x_i}\left(\mathbf{F}_{i+\frac{1}{2}} - \mathbf{F}_{i-\frac{1}{2}}\right) + \mathbf{S}_i = 0 \tag{6}$$

Here, $\Delta x$ is the grid spacing, $i$ is the nodal index, and $\mathbf{F}_{i+\frac{1}{2}}(\mathbf{Q}_L, \mathbf{Q}_R)$ denotes an interfacial Riemann flux approximation constructed from left and right state variables $\mathbf{Q}_L$ and $\mathbf{Q}_R$. Characteristic-based numerical flux approximations maintain discrete flux conservation in integral form and are designed to perform well in the presence of either smooth or discontinuous solutions. Roe's flux approximation [3] is probably the most widely used and leads to a flux-difference formula based on the conserved variables $\mathbf{Q} = (h, hu, hv)^T$. A difficulty with the Roe flux in the present context is that it is not well behaved at zero average depth and therefore near wetting and drying fronts. In the present paper, a primitive-variable Riemann flux is developed that is well behaved at zero average depth, while maintaining discrete conservation and the discontinuity capturing features of Roe's scheme.

Consider a change of variables in the one-dimensional form of (1) from conserved variables $\mathbf{Q}$ to a set of primitive variables $\boldsymbol{q} = (h, u, v)^T$, as in

$$\frac{\partial \mathbf{Q}(\boldsymbol{q})}{\partial t} + \boldsymbol{A}\frac{\partial \mathbf{Q}(\boldsymbol{q})}{\partial x} + \boldsymbol{S}[\mathbf{Q}(\boldsymbol{q})] = \boldsymbol{M}\frac{\partial \boldsymbol{q}}{\partial t} + \boldsymbol{A}\boldsymbol{M}\frac{\partial \boldsymbol{q}}{\partial x} + \boldsymbol{S}(\boldsymbol{q}) = 0 \tag{7}$$

Here, $\boldsymbol{M} = \partial \mathbf{Q}/\partial \boldsymbol{q}$ is the change-of-variable matrix, the flux Jacobian $\boldsymbol{A} = \partial \boldsymbol{F}/\partial \mathbf{Q}$ is the system matrix for $\mathbf{Q}$, and $\boldsymbol{a} = \boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{M}$ is the system matrix for $\boldsymbol{q}$. An eigensystem for $\boldsymbol{a}$ is needed to develop the primitive-variable interfacial flux. The eigenvalues $\lambda_\kappa$ of both $\boldsymbol{a}$ and $\boldsymbol{A}$ are $\lambda_\kappa = (u, u+\sqrt{h}, u-\sqrt{h})$. If $\boldsymbol{R}_q$ denotes the matrix of right eigenvectors of $\boldsymbol{a}$, then $\boldsymbol{a} = \boldsymbol{R}_q \boldsymbol{\Lambda} \boldsymbol{R}_q^{-1}$, where $\boldsymbol{\Lambda} = diag\,(\lambda_1, \lambda_2, \lambda_3)$. The relevant matrices are given by

$$\boldsymbol{M} = \begin{pmatrix} 1 & 0 & 0 \\ u & h & 0 \\ v & 0 & h \end{pmatrix} \quad \boldsymbol{a} = \begin{pmatrix} u & h & 0 \\ 1 & u & 0 \\ 0 & 0 & u \end{pmatrix} \quad \boldsymbol{R}_q = \begin{pmatrix} 0 & 1 & 1 \\ 0 & h^{-\frac{1}{2}} & -h^{-\frac{1}{2}} \\ h^{-1} & 0 & 0 \end{pmatrix} \tag{8}$$

A conservative primitive-variable interfacial flux $\boldsymbol{F}_{i+\frac{1}{2}}(\boldsymbol{q}_L, \boldsymbol{q}_R)$ that approximates the conservative flux function $\boldsymbol{F}(\boldsymbol{q})$ of (2) can be expressed as

$$\boldsymbol{F}_{i+\frac{1}{2}}(\boldsymbol{q}_L, \boldsymbol{q}_R) = \frac{1}{2}\left[(\boldsymbol{F}_L + \boldsymbol{F}_R) - \widetilde{\boldsymbol{M}}|\tilde{\boldsymbol{a}}|(\boldsymbol{q}_R - \boldsymbol{q}_L)\right] \tag{9}$$

Here, $\boldsymbol{q}_L$ and $\boldsymbol{q}_R$ are left and right state variables to be defined for each cell interface, and $\boldsymbol{F}_L, \boldsymbol{F}_R$ denote $\boldsymbol{F}(\boldsymbol{q}_L), \boldsymbol{F}(\boldsymbol{q}_R)$, respectively. Also, $|\boldsymbol{a}| = \boldsymbol{R}_q|\boldsymbol{\Lambda}|\boldsymbol{R}_q^{-1}$, where $|\boldsymbol{\Lambda}| = \boldsymbol{\Lambda}^+ - \boldsymbol{\Lambda}^-$, and $\boldsymbol{\Lambda}^\pm$ denotes the diagonal matrix of positive or negative eigenvalues. Some form of averaging $\tilde{\boldsymbol{q}}(\boldsymbol{q}_L, \boldsymbol{q}_R)$ is used to evaluate $\tilde{\boldsymbol{M}}|\tilde{\boldsymbol{a}}|$. The first-order flux is obtained by choosing $\boldsymbol{q}_L = \boldsymbol{q}_i$ and $\boldsymbol{q}_R = \boldsymbol{q}_{i+1}$. Note that $\boldsymbol{M}|\boldsymbol{a}| = |\boldsymbol{A}|\boldsymbol{M}$.

### 3.1. Primitive-variable and Roe flux behaviors at zero average depth

It is shown here that the primitive-variable flux (9) is well behaved at zero average depth and is therefore attractive for capturing moving wetting and drying fronts. The interfacial flux computation requires an evaluation of the flux dissipation matrix $\widetilde{\boldsymbol{M}}|\tilde{\boldsymbol{a}}|$ using averaged variables $\tilde{\boldsymbol{q}}$ but does not require evaluation of the right eigenvectors $\boldsymbol{R}_q$ themselves, which are singular at $h = 0$. The flux dissipation matrix is given by

$$\boldsymbol{M}|\boldsymbol{a}| = \frac{1}{2}\begin{pmatrix} \beta^+ & \sqrt{h}\beta^- & 0 \\ u\beta^+ + \sqrt{h}\beta^- & h\beta^+ + u\sqrt{h}\beta^- & 0 \\ v\beta^+ & v\sqrt{h}\beta^- & 2h|\lambda_1| \end{pmatrix} \tag{10}$$

where $\beta^\pm = |\lambda_2| \pm |\lambda_3|$. At a dry state ($h = u = v = 0$), the eigenvalues $\lambda_\kappa$ are all zero, and since each term in (10) is a multiple of an eigenvalue, the dissipation matrix becomes a null matrix. If algebraic averages $\bar{\boldsymbol{q}} = (\boldsymbol{q}_L + \boldsymbol{q}_R)/2$ are used, then $\bar{\boldsymbol{q}} = 0$ and $\boldsymbol{F}_{i+\frac{1}{2}}(\boldsymbol{q}_L, \boldsymbol{q}_R) = 0$ at a dry/dry interface for which $\boldsymbol{q}_L = \boldsymbol{q}_R = 0$. Consequently, the interfacial flux computation (9) is well behaved at cell interfaces having zero average depth and in completely dry regions. This discrete interfacial flux will respond appropriately to wetting from either direction ($\boldsymbol{q}_L$ or $\boldsymbol{q}_R \neq 0$).

This behavior at zero depth differs from that of the Roe flux, which can be written as

$$\boldsymbol{F}_{i+\frac{1}{2}}(\mathbf{Q}_R, \mathbf{Q}_L) = \frac{1}{2}\left[(\boldsymbol{F}_L + \boldsymbol{F}_R) - |\widetilde{\boldsymbol{A}}|(\mathbf{Q}_R - \mathbf{Q}_L)\right] \tag{11}$$

Here, $|\boldsymbol{A}| = \boldsymbol{R}_Q|\Lambda|\boldsymbol{R}_Q^{-1}$, and $\boldsymbol{R}_Q = \boldsymbol{M}\boldsymbol{R}_q$ is the corresponding matrix of right eigenvectors of $\boldsymbol{A}$. The Roe dissipation matrix is given by

$$|\boldsymbol{A}| = \frac{1}{2\sqrt{h}}\begin{pmatrix} \sqrt{h}\beta^+ - u\beta^- & \beta^- & 0 \\ (h-u^2)\beta^- & \sqrt{h}\beta^+ + u\beta^- & 0 \\ v\sqrt{h}(\beta^+ - 2\lambda_1) - uv\beta^- & v\beta^- & 2\sqrt{h}|\lambda_1| \end{pmatrix} \tag{12}$$

The dissipation matrix (12) and hence the Roe flux contains terms such as $u/\sqrt{h}$ that are indeterminate as $h, u, v \to 0$. Consequently, the numerical evaluation of the Roe flux (11) is not well behaved at cell interfaces having zero average depth. The Roe-averaged variables are also indeterminate if $h_L = h_R = 0$.

## 4. Finite-volume discretization for unstructured grids

The following development is applicable to general unstructured grids with mixed element shapes. Eq. (1) is written as an integral conservation equation for a planar area $\Omega$ enclosed by a curve $\partial\Omega$:

$$\frac{\partial}{\partial t}\int_\Omega \boldsymbol{Q}\, d\mathcal{A} + \oint_{\partial\Omega} \vec{\boldsymbol{E}}\cdot\vec{\boldsymbol{n}}\, d\mathcal{S} + \int_\Omega \boldsymbol{S}\, d\mathcal{A} = 0 \tag{13}$$

Here, $\vec{\boldsymbol{E}} = \vec{\boldsymbol{i}}\boldsymbol{F} + \vec{\boldsymbol{j}}\boldsymbol{G}$ is the resultant vector of flux components, where $\vec{\boldsymbol{i}}$ and $\vec{\boldsymbol{j}}$ are Cartesian unit vectors. Also, $d\mathcal{A}$ denotes a differential area of $\Omega$, and $d\mathcal{S}$ denotes a differential arc length of the enclosing curve $\partial\Omega$, whose outward-facing normal unit vector is $\vec{\boldsymbol{n}}$.

The spatial discretization for a node-centered finite-volume scheme is constructed from the median dual of a two-dimensional triangulated unstructured grid as illustrated in Fig. 2. A control volume in this context implies a two-dimensional area extruded to unit thickness. The overall solution domain $\Omega$ is thereby divided into $I$ non-overlapping control volumes or cells $\Omega_i, i = 1, 2\cdots I$, each associated with a nodal point $i$. Each cell $\Omega_i$ has an area $\mathcal{A}_i = \int_{\Omega_i} d\mathcal{A}$. At each node, $\boldsymbol{q}_i$ denotes an area-averaged value defined by $\boldsymbol{q}_i = (\int_{\Omega_i} \boldsymbol{q}\, d\mathcal{A})/\mathcal{A}_i$. Then $\boldsymbol{Q}_i = \boldsymbol{Q}(\boldsymbol{q}_i)$ and $\boldsymbol{S}_i = \boldsymbol{S}(\boldsymbol{q}_i)$. Each node $i$ is surrounded by a set $\mathcal{N}(i)$ of neighboring nodes denoted $j$, such that $j$ takes on each value of $i$ in $\mathcal{N}(i)$. Each cell $i$ is enclosed by a set of cell interfaces $\partial\Omega_{ij}$ with $j \in \mathcal{N}(i)$. Each cell interface is formed by two distinct line segments that connect the midpoint of the $\{ij\}$ line to the centroids of the two adjacent triangles.

### 4.1. Finite-volume discretization of the flux integral

The surface flux integral in (13) is approximated by midpoint integration of the fluxes over each of the $\partial\Omega_{ij}$ cell interfaces that enclose the cell. A simple and efficient technique for the midpoint integration is to evaluate the fluxes at the midpoint of the $\{ij\}$ line and to represent the two-segment cell interface $\partial\Omega_{ij}$ by a single resultant area vector, obtained by summing the area vectors for these two segments. The arc length of this $\{ij\}$ cell interface is denoted $\delta\mathcal{S}_{ij}$, and its unit normal vector is denoted $\vec{\boldsymbol{n}}_{ij}$. The surface flux integral is written as

$$\oint_{\partial\Omega} \vec{\boldsymbol{E}}\cdot\vec{\boldsymbol{n}}\, d\mathcal{S} \approx \sum_{j\in\mathcal{N}(i)} \boldsymbol{H}_{ij} \tag{14}$$

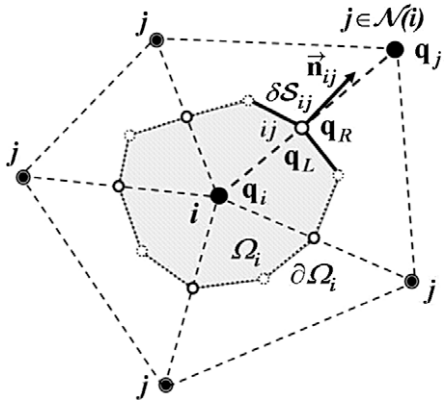


**Fig. 2.** Illustration of a control volume $\Omega_i$ at point $i$ surrounded by a set $\mathcal{N}(i)$ of neighboring points $j$. A median dual control volume is constructed by connecting the midpoint of each $\{ij\}$ line to the centroids of the two adjacent triangular elements.

where the unit normal vector is $\vec{\boldsymbol{n}} = \vec{\boldsymbol{i}}\hat{n}_x + \vec{\boldsymbol{j}}\hat{n}_y$. The $x$ and $y$ projections of $\vec{\boldsymbol{n}}\delta\mathcal{S}$ are denoted $s_x$ and $s_y$, so that $\hat{n}_x = s_x/\delta\mathcal{S}$ and $\hat{n}_y = s_y/\delta\mathcal{S}$. The arc-length weighted flux function $\boldsymbol{H}$ projected in the $\vec{\boldsymbol{n}}$ direction is then given by

$$\boldsymbol{H} = \vec{\boldsymbol{E}}\cdot\vec{\boldsymbol{n}}\,d\mathcal{S} = \boldsymbol{F}s_x + \boldsymbol{G}s_y = \begin{pmatrix} h\Theta \\ hu\Theta + s_x(h^2 - h_s^2)/2 \\ hv\Theta + s_y(h^2 - h_s^2)/2 \end{pmatrix} \tag{15}$$

where $\Theta = s_x u + s_y v$. The numerical fluxes at each cell interface are evaluated using the discrete primitive-variable flux as in (9) but written for the projected flux $\boldsymbol{H}$ as

$$\boldsymbol{H}_{ij}(\boldsymbol{q}_L, \boldsymbol{q}_R) = \frac{1}{2}[(\boldsymbol{H}_L + \boldsymbol{H}_R) - \widetilde{\boldsymbol{M}}|\tilde{\boldsymbol{a}}|(\boldsymbol{q}_R - \boldsymbol{q}_L)] \tag{16}$$

In the projected flux (16), $\boldsymbol{a}$ is given by $\boldsymbol{M}^{-1}\boldsymbol{A}_H\boldsymbol{M}$, where $\boldsymbol{A}_H = \partial\boldsymbol{H}/\partial\boldsymbol{Q}$. An eigensystem of $\boldsymbol{a}$ is needed to construct these fluxes. The eigenvalues $\lambda_\kappa$ of $\boldsymbol{a}$ are $(\Theta, \Theta + \sqrt{H}, \Theta - \sqrt{H})$, where $H = (s_x^2 + s_y^2)h$. A set of right and left eigenvectors of $\boldsymbol{a}$ are given by

$$\boldsymbol{R}_q = \frac{1}{\sqrt{h}}\begin{pmatrix} 0 & \sqrt{h} & \sqrt{h} \\ -\hat{n}_y h^{-\frac{1}{2}} & \hat{n}_x & -\hat{n}_x \\ \hat{n}_x h^{-\frac{1}{2}} & \hat{n}_y & -\hat{n}_y \end{pmatrix} \quad \boldsymbol{R}_q^{-1} = \frac{1}{2}\begin{pmatrix} 0 & -2\hat{n}_y h & 2\hat{n}_x h \\ 1 & \hat{n}_x\sqrt{h} & \hat{n}_y\sqrt{h} \\ 1 & -\hat{n}_x\sqrt{h} & -\hat{n}_y\sqrt{h} \end{pmatrix} \tag{17}$$

Finally, the primitive-variable flux dissipation matrix can be written as

$$\boldsymbol{M}|\boldsymbol{a}| = \frac{1}{2}\begin{pmatrix} \beta^+ & \hat{\eta}_x\beta^- & \hat{\eta}_y\beta^- \\ u\beta^+ + \hat{\eta}_x\beta^- & u\hat{\eta}_x\beta^- + \Psi_1 & u\hat{\eta}_y\beta^- + \Psi_2 \\ v\beta^+ + \hat{\eta}_y\beta^- & v\hat{\eta}_x\beta^- + \Psi_2 & v\hat{\eta}_y\beta^- + \Psi_3 \end{pmatrix} \tag{18}$$

where

$$\beta^\pm = |\lambda_2| \pm |\lambda_3|, \quad \hat{\eta}_x = \hat{n}_x\sqrt{h}, \quad \hat{\eta}_y = \hat{n}_y\sqrt{h}$$
$$\Psi_1 = \hat{\eta}_x^2\beta^+ + 2\hat{\eta}_y^2|\lambda_1|, \quad \Psi_2 = \hat{\eta}_x\hat{\eta}_y(\beta^+ - 2|\lambda_1|), \quad \Psi_3 = \hat{\eta}_y^2\beta^+ + 2\hat{\eta}_x^2|\lambda_1|$$

Since all eigenvalues vanish for $h, u, v = 0$, the dissipation matrix (18) again becomes a null matrix. Note that the one-dimensional case with uniform grid spacing implies that $s_x = \hat{n}_x = 1, s_y = \hat{n}_y = 0$. In this case, Eq. (18) reduces to Eq. (10).

The left and right state variables $\boldsymbol{q}_L$ and $\boldsymbol{q}_R$ at each $\{ij\}$ cell interface are needed to evaluate the fluxes. The flux for the unstructured scheme uses either linear (second-order) or quadratic (third-order) reconstruction to obtain values of $\boldsymbol{q}_L$ (or $\boldsymbol{q}_R$) from nodal values $\boldsymbol{q}_i, \boldsymbol{q}_j$, and the gradient $\bigtriangledown\boldsymbol{q}$ at point $i$ (or $j$). For example, $\boldsymbol{q}_L$ is evaluated by

$$(\boldsymbol{q}_L)_i = \boldsymbol{q}_i + \phi\left[(1-\psi)\bigtriangledown\boldsymbol{q}_i\cdot\vec{\boldsymbol{r}}_{ij} + \psi^2(\boldsymbol{q}_j - \boldsymbol{q}_i)\right] \tag{19}$$

Here, $\vec{\boldsymbol{r}}_{ij}$ is the position vector from point $i$ to the $\{i\,j\}$ cell interface (the quadrature point), $\psi = 0$ for linear reconstruction, $\psi = \frac{1}{2}$ for quadratic reconstruction, and $\phi$ is a slope limiter.

The gradient $\bigtriangledown\boldsymbol{q}_i$ in (19) is evaluated from an unweighted linear least-squares fit to solution values at point $i$ and its neighboring points $j \in \mathcal{N}(i)$. This technique is discussed in detail by Anderson and Bonhaus [32] and Hyams [33], who also give formulas for the least-squares coefficients obtained by $QR$ factorization. The Barth–Jesperson [34] limiter is used to evaluate $\phi$. Values of $\boldsymbol{q}_R$ for this $\{i\,j\}$ cell interface are obtained by applying this same technique in the adjacent cell centered at point $j$. The primitive-variable flux is not sensitive to the type of averaging used, and an algebraic average $\bar{\boldsymbol{q}}$ is used unless otherwise noted.

### 4.2. Wetting and drying fronts

Problems with wetting and drying fronts are treated by computing the solution for all cells, wet or dry, and letting the balance of flux integral and source term govern the propagation of wetting and drying fronts throughout the solution domain. A common technique to treat zero depth using conserved variables and the Roe flux is to enforce a small minimum depth $h_{min}$ during the solution process. Toro [2] shows that setting a minimum depth can cause significant error in front propagation speed at practical values of $h_{min}$, due to the fundamentally different structure of the Riemann solution at a wet/dry interface. Other techniques have included local modification of velocity or bottom slope. Since the present primitive-variable formulation is well behaved at $h = 0$, it does not require a minimum depth or other local treatment of wetting and drying fronts. The present time integration does occasionally produce a small negative depth at some cells. These are of order $10^{-6}$ to $10^{-10}$ times the reference depth and are not a source of significant error. The depth at these cells is reset to zero instead of a minimum depth, to avoid a negative square root when computing celerity. The velocity is also set to zero in cells with zero depth.

### 4.3. Exact source-term balancing at zero depth

If the bottom slope is calculated using the least-squares gradient technique used for fluxes in Eq. (19), the dry-bottom source terms will balance only for a flat bottom. An exact dry-bottom balance for unstructured grids and arbitrary bottom surfaces can be obtained by approximating the source term as follows:

$$\int_{\Omega}(h-h_s)\bigtriangledown h_s\,d\mathcal{A}\approx(h-h_s)_i\int_{\Omega}\bigtriangledown h_s\,d\mathcal{A}=(h-h_s)_i\oint_{\partial\Omega}h_s\vec{\boldsymbol{n}}\,d\mathcal{S} \tag{20}$$

This equation uses the divergence theorem to express the area-averaged gradient of $h_s$ as a contour integral. Combining the contour integral in (20) with the flux integral in 14, 15 and summing over cell interfaces gives the following discretization of the $x$ momentum equation for $u=v=0$:

$$\sum_{j\in\mathcal{N}(i)}\left[s_x\frac{1}{2}(h^2-h_s^2)\right]_{ij}-(h-h_s)_i\sum_{j\in\mathcal{N}(i)}(s_x\,h_s)_{ij}=0 \tag{21}$$

The corresponding discretization for the $y$ direction is obtained by replacing $s_x$ by $s_y$ in (21). To implement this discretization, the nodal data for bottom elevation $(h_s)_i$ is first transferred to the $\{i\,j\}$ cell interfaces throughout the grid using algebraic averages $(\bar{h}_s)_{ij}=[(h_s)_i+(h_s)_j]/2$. With $h_s$ thereby redefined as $\{ij\}$ cell-interface data instead of nodal data, new nodal values of $(h_s)_i$ can be calculated from (21) to ensure that this equation is balanced exactly for $h=0$. Separate values of $(h_s)_i$ are calculated for the $x$ and $y$ directions to ensure an exact balance for each direction. The calculation of $(h_s)_{ij}$ and nodal balancing values $(h_s)_i$ is performed only once, prior to starting the solution. In this way, the flux integral and source terms are balanced for both still water and in dry regions.

### 4.4. Characteristic-variable boundary conditions with source terms

Boundary conditions are developed from the non-conservative form of the equations as in Eq. (7). This development extends that of Whitfield [5] to include source terms and gives an explicit solution for boundary values of $\boldsymbol{q}$ that applies to all characteristic orientations. Recalling that $\boldsymbol{a}=\boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{M}=\boldsymbol{R}_q\Lambda\boldsymbol{R}_q^{-1}$, the equation for the direction normal to a boundary surface can be written as

$$\boldsymbol{R}_q^{-1}\frac{\partial\boldsymbol{q}}{\partial t}+\Lambda\boldsymbol{R}_q^{-1}\frac{\partial\boldsymbol{q}}{\partial n}+\boldsymbol{R}_q^{-1}\boldsymbol{M}^{-1}\boldsymbol{S}(\boldsymbol{q})=0 \tag{22}$$

where $\boldsymbol{R}_q^{-1}$ was given in Eq. (17). Approximate relationships governing characteristic variables are developed by assuming that the coefficient matrices $\boldsymbol{R}_q^{-1}$ and $\boldsymbol{M}^{-1}$ are constant matrices that are evaluated from a chosen constant local state, denoted $\boldsymbol{q}_o$. This allows (22) to be written as

$$\frac{\partial\boldsymbol{W}_o}{\partial t}+\boldsymbol{\Lambda}\frac{\partial\boldsymbol{W}_o}{\partial n}+\boldsymbol{R}_{q,o}^{-1}\boldsymbol{M}_{q,o}^{-1}\boldsymbol{S}(\boldsymbol{q}_o)=0 \tag{23}$$

where the characteristic-variable vector is given by $\boldsymbol{W}_o(\boldsymbol{q})=\boldsymbol{R}_{q,o}^{-1}\boldsymbol{q}$. Because $\boldsymbol{\Lambda}$ is a diagonal matrix, (23) can be written as three uncoupled equations governing the characteristic variables $w_{o,\kappa}$ $(\kappa=1,2,3)$:

$$\frac{\partial w_{o,\kappa}}{\partial t}+\lambda_\kappa\frac{\partial w_{o,\kappa}}{\partial n}=\widetilde{S}_{o,\kappa} \tag{24}$$

Here, $\widetilde{S}_{o,\kappa}$ is defined to be the $\kappa$th component of the vector $-\boldsymbol{R}_{q,o}^{-1}\boldsymbol{M}_{q,o}^{-1}\boldsymbol{S}_o$. The characteristic variables corresponding to $\lambda_\kappa$ are given by

$$w_{o,1}=h_o(\hat{n}_x v-\hat{n}_y u) \tag{25a}$$

$$w_{o,2}=\left(h+\sqrt{h_o}\,\widehat{\Theta}\right)/2 \tag{25b}$$

$$w_{o,3}=\left(h-\sqrt{h_o}\,\widehat{\Theta}\right)/2 \tag{25c}$$

where $\widehat{\Theta}=\hat{n}_x u+\hat{n}_y v$.

Eq. (24) can also be written as a total derivative along a characteristic line whose slope $dn/dt$ is $\lambda_\kappa$:

$$\frac{dw_{o,\kappa}}{dt}=\widetilde{S}_{o,\kappa} \tag{26}$$

According to (26), the rate of change of $w_{o,\kappa}$ along the $\kappa$th characteristic line is equal to the source term. If the source term is zero, then $w_{o,\kappa}$ is constant along this characteristic line.

The boundary conditions are obtained by writing (26) as an upwind difference scheme for each characteristic line, $\kappa$. The upwind direction relative to the boundary point is the direction from which the $\kappa$th characteristic line approaches the boundary point. The $\kappa$th upwind formula can be written as

$$(w_{o,\kappa})_B = (w_{o,\kappa})_{A_\kappa} + \Delta t \widetilde{S}_{o,\kappa} \tag{27}$$

The subscripts $B$ and $A_\kappa$ in (27) are explained with reference to Fig. 3, which shows a control volume adjacent to a boundary point designated by the subscript $B$. The subscript $A_\kappa$ selects the appropriate upwind point for each value of $\kappa$. If the $\lambda_\kappa$ characteristic approaches $B$ from outside the boundary, then $A_\kappa$ refers to the exterior point $bc$, and $\boldsymbol{q}_{bc}$ is evaluated from specified boundary conditions. If the $\lambda_\kappa$ characteristic approaches $B$ from inside the boundary, then $A_\kappa$ refers to the interior point $i$, and $\boldsymbol{q}_i$ is obtained from recently computed solution values. Note that the boundary conditions impose characteristic-variable data from the correct side of the boundary but do not rigorously account for propagation of characteristic variables along characteristic directions.

The three equations represented by (27) can be solved explicitly for the boundary values $\boldsymbol{q}_B$. It is convenient to define the right-hand side of Eq. (27) as $\Phi_{A_\kappa} = (w_{o,\kappa})_{A_\kappa} + \Delta t \widetilde{S}_{o,\kappa}$. The solution for $\boldsymbol{q}_B$ can then be written as

$$h_B = \Phi_{A_2} + \Phi_{A_3} \tag{28a}$$

$$u_B = \frac{-\hat{n}_y}{h_o}\Phi_{A_1} + \frac{\hat{n}_x}{\sqrt{h_o}}(\Phi_{A_2} - \Phi_{A_3}) \tag{28b}$$

$$v_B = \frac{\hat{n}_x}{h_o}\Phi_{A_1} + \frac{\hat{n}_y}{\sqrt{h_o}}(\Phi_{A_2} - \Phi_{A_3}) \tag{28c}$$

The local reference state chosen here is $\boldsymbol{q}_o = (\boldsymbol{q}_B^n + \boldsymbol{q}_i^n)/2$, and the implicit algorithm applies the boundary conditions by updating boundary points following each innermost iteration in the time-iterative solution algorithm. This technique is effective because the characteristic-variable boundary conditions do not introduce spurious, nonphysical wave reflections at boundaries, which might otherwise occur if boundary conditions are in conflict with the characteristic theory for the governing equations. The use of characteristic-variable boundary conditions is also discussed in Refs. [35,5].

### 4.5. Bottom friction model

The influence of bottom friction on an advancing wet/dry front is examined using the Chezy model (e.g., Weiyan [31]), written here in nondimensional form using the Darcy–Weisbach friction factor $f$:

$$\tau_{bx} = fu\sqrt{u^2 + v^2}/8 \quad \tau_{by} = fv\sqrt{u^2 + v^2}/8 \tag{29}$$

The friction factor is related to the (dimensional) Chezy coefficient by $f = 8g/C^2$, where $C$ is the Chezy coefficient in units of $\sqrt{\text{m}}/\text{s}$, and $g = 9.81\ \text{m}/\text{s}^2$ is gravitational acceleration. The value of $C$ must be determined from experience and can range from 20 for flood plains to 70 for deep channels [31]. More generally, $f$ depends on surface roughness and Reynolds number based on hydraulic radius. It can be related to fully developed flow in ducts using the Moody diagram.

## 5. Explicit and implicit time-integration schemes

The primitive-variable flux and finite-volume formulation can be implemented with different time-integration schemes. The present study gives two Euler explicit schemes and an iterative implicit backward-differentiation scheme for the conservation Eq. (13). The implicit scheme is a fully coupled, nonlinear algorithm that is solved by a parallel iterative scheme adapted from established and effective methods for solving the Navier–Stokes equations [33,27,36]. The enhanced stability of the implicit algorithm is considered attractive for highly nonuniform cell sizes that provide definition of complex coastlines and resolution of localized flow features, and for steady solutions in which large local time steps can accelerate convergence. An Euler explicit scheme using conserved variables and a time-linearized explicit scheme using primitive variables are also developed; these are less stable but are simpler and require significantly less computational effort per time step.

A first-order Euler explicit scheme in conserved variables is constructed by combining a forward time difference with the spatial discretization as in

$$\frac{\mathcal{A}_i}{\Delta t}(\Delta \boldsymbol{Q}_i^n) + \sum_{j \in \mathcal{N}(i)} \boldsymbol{H}_{ij}^n \delta \mathcal{S}_{ij} + \mathcal{A}_i \boldsymbol{S}^n = 0 \tag{30}$$

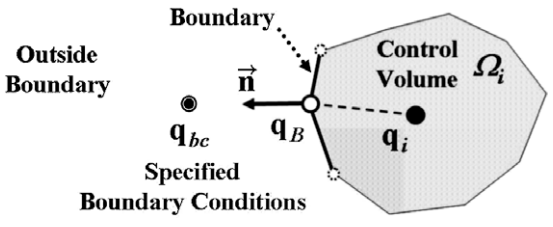


**Fig. 3.** Notation for points associated with specification of characteristic-value boundary conditions.

Here, $\boldsymbol{Q}^n$ denotes $\boldsymbol{Q}(t^n)$, and $\Delta\boldsymbol{Q}^n = (\boldsymbol{Q}^{n+1} - \boldsymbol{Q}^n)$. Eq. (30) is explicit for the conserved variables $\boldsymbol{Q}^{n+1}$, and the velocity is computed by solving $\boldsymbol{Q}(\boldsymbol{q}^{n+1}) = \boldsymbol{Q}^{n+1}$ to obtain $u^{n+1} = (hu)^{n+1}/h^{n+1}$ and $v^{n+1} = (hv)^{n+1}/h^{n+1}$. If $h = 0$, the division by zero can be avoided by setting velocity to zero, but the velocity computation may be inaccurate at small but nonzero depth if a minimum depth criterion is imposed (cf. Toro [2]).

A second explicit scheme in primitive variables is developed using a time linearization as in

$$\Delta t\, \partial\boldsymbol{Q}/\partial t \approx \Delta\boldsymbol{Q}^n = \boldsymbol{M}^n \Delta\boldsymbol{q}^n + O(\Delta t^2) \tag{31}$$

which preserves the first-order time accuracy in (30). The matrix $\boldsymbol{M}$ is singular for $h = 0$, however, and it is approximated by $\breve{\boldsymbol{M}} = \boldsymbol{M} + \varepsilon diag(0, 1, 1)$, where $\varepsilon$ is a small positive number. The parameter $\varepsilon$ provides a minimum value for $h$ in the linearized time derivative. The matrix $\breve{\boldsymbol{M}}$ has eigenvalues $(1, h + \varepsilon, h + \varepsilon)$ and is nonsingular at zero depth. The time-linearized explicit scheme becomes

$$\frac{\mathcal{A}_i}{\Delta t}(\breve{\boldsymbol{M}}^n \Delta\boldsymbol{q}^n) + \sum_{j\in\mathcal{N}(i)} \boldsymbol{H}_{ij}^n \delta\mathcal{S}_{ij} + \mathcal{A}_i \boldsymbol{S}^n = 0 \tag{32}$$

Eq. (32) defines a $3 \times 3$ linear system at each node that is solved for the primitive variables $\boldsymbol{q}^{n+1}$.

The implicit scheme employs a two-step backward differentiation formula and is given by

$$\frac{\mathcal{A}_i}{\Delta t}[(1+\theta)\Delta\boldsymbol{Q}_i^n - \theta\Delta\boldsymbol{Q}_i^{n-1}] + \sum_{j\in\mathcal{N}(i)} \boldsymbol{H}_{ij}^{n+1}\, \delta\mathcal{S}_{ij} + \mathcal{A}_i \boldsymbol{S}^{n+\gamma} = 0 \tag{33}$$

In Eq. (33), the parameter $\theta$ selects first or second-order time accuracy ($\theta = 0$ or $\theta = 1/2$), and the parameter $\gamma$ selects implicit ($\gamma = 1$) or explicit ($\gamma = 0$) evaluation of the source term.

### 5.1. Iterative solution of implicit scheme

The nonlinear system (33) is solved at each time step using a discrete Newton-linearized iteration scheme devised for the primitive-variable flux formulation, and given by

$$\frac{\mathcal{A}_i}{\Delta t}\left[(1+\theta)\breve{\boldsymbol{M}} + \gamma\left(\frac{\partial\boldsymbol{S}}{\partial\boldsymbol{q}}\right)\right]_i^{n+1,m} \Delta_m\boldsymbol{q}_i^{n+1} + \sum_{j\in\mathcal{N}(i)} \delta\mathcal{S}_{ij}\left[\left(\frac{\partial\boldsymbol{H}_{ij}}{\partial\boldsymbol{q}_{L_j}}\right)^{n+1,m} \Delta_m\boldsymbol{q}_i^{n+1} + \left(\frac{\partial\boldsymbol{H}_{ij}}{\partial\boldsymbol{q}_{R_j}}\right)^{n+1,m} \Delta_m\boldsymbol{q}_j^{n+1}\right] = -(\mathbb{R}_U)_i^{n+1,m} \tag{34}$$

This equation defines an iterative sequence ($m = 0, 1, 2, \ldots$). The unsteady residual $(\mathbb{R}_U)_i^{n+1}$ in (34) is defined by equality with the left-hand side of Eq. (33). The left-hand side of (34) is an approximate Newton linearization of $(\mathbb{R}_U)_i^{n+1}$ with respect to the primitive variable $\boldsymbol{q}$ at point $i$ and at each neighboring point $j$. It replaces $\boldsymbol{M}$ by the nonsingular matrix $\breve{\boldsymbol{M}}$. Eq. (34) defines a multidimensional linear system that is solved for the $m$th Newton increment

$$\Delta_m\boldsymbol{q}^{n+1} = (\boldsymbol{q}^{n+1,m+1} - \boldsymbol{q}^{n+1,m}) \tag{35}$$

at all $i$ points. An updated value $\boldsymbol{q}^{n+1,m+1} = \boldsymbol{q}^{n+1,m} + \Delta_m\boldsymbol{q}^{n+1}$ is then computed for the next Newton iteration. The iteration is initialized using $\boldsymbol{q}^{n+1,m=0} \leftarrow \boldsymbol{q}^n$. At convergence, $\Delta_m\boldsymbol{q}^{n+1} \to 0$, and Eq. (34) satisfies $(\mathbb{R}_U)_i^{n+1} = 0$. Therefore, $\boldsymbol{q}^{n+1}$ satisfies the fully implicit and nonlinear unsteady approximation (33).

This Newton iteration is generally very reliable since $\boldsymbol{q}^n$ usually provides a good initialization, especially with smaller time steps. Although the nonsingular matrix $\breve{\boldsymbol{M}}$ secures an expectation of convergence for small values of $h$ and sufficiently small $\Delta t$, the Newton iteration is somewhat sensitive to $\varepsilon$ during the initial start-up of dry-bed cases, as will be discussed in the results section. However, the converged solution still satisfies the unsteady residual Eq. (33) without any special treatment for zero depth.

The discrete flux-Jacobian linearizations in (34) are defined and computed numerically as proposed by Whitfield and Taylor [37,38]. The $\alpha$-column of each matrix is evaluated as in

$$\frac{\partial\boldsymbol{H}_{ij}}{\partial\boldsymbol{q}_{L_j}} \approx \frac{1}{\sigma}[\boldsymbol{H}_{ij}(\boldsymbol{q}_{L_j} + \sigma e_\alpha, \boldsymbol{q}_{R_j}) - \boldsymbol{H}_{ij}(\boldsymbol{q}_{L_j}, \boldsymbol{q}_{R_j})] \tag{36}$$

$$\frac{\partial\boldsymbol{H}_{ij}}{\partial\boldsymbol{q}_{R_j}} \approx \frac{1}{\sigma}[\boldsymbol{H}_{ij}(\boldsymbol{q}_{L_j}, \boldsymbol{q}_{R_j} + \sigma e_\alpha) - \boldsymbol{H}_{ij}(\boldsymbol{q}_{L_j}, \boldsymbol{q}_{R_j})] \tag{37}$$

where $e_\alpha$ is the unit vector for the $\alpha$-component $q_\alpha$ of $\boldsymbol{q}$, and $\sigma$ is a small number of the order of the square root of the machine epsilon.

Each Newton iteration (34) is multidimensional and must itself be solved iteratively. The following Lower/Upper Symmetric Gauss–Seidel (LU/SGS) scheme is used for this linear subiteration:

$$(\mathcal{D} + \mathcal{L})\Delta_m\boldsymbol{q}^{k+\frac{1}{2}} + (\mathcal{U})\Delta_m\boldsymbol{q}^k = \mathbb{R}_U^{n+1,m} \tag{38a}$$

$$(\mathcal{D} + \mathcal{U})\Delta_m\boldsymbol{q}^{k+1} + (\mathcal{L})\Delta_m\boldsymbol{q}^{k+\frac{1}{2}} = \mathbb{R}_U^{n+1,m} \tag{38b}$$

For completeness, definitions of the diagonal $\mathcal{D}$, lower $\mathcal{L}$, and upper $\mathcal{U}$ coefficients are given as

$$\mathcal{D} = \frac{\mathcal{A}_i}{\Delta t}\left[(1+\theta)\breve{\boldsymbol{M}} + \gamma\left(\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{q}}\right)\right]_i^{n+1,m} + \sum_{j\in\mathcal{N}(i)} \delta\mathcal{S}_{ij}\left(\frac{\partial \boldsymbol{H}_{ij}}{\partial \boldsymbol{q}_{L_j}}\right)^{n+1,m}$$

$$\mathcal{L} = \sum_{j\in\mathcal{N}_L(i)} \delta\mathcal{S}_{ij}\left(\frac{\partial \boldsymbol{H}_{ij}}{\partial \boldsymbol{q}_{R_j}}\right)^{n+1,m} \qquad \mathcal{U} = \sum_{j\in\mathcal{N}_U(i)} \delta\mathcal{S}_{ij}\left(\frac{\partial \boldsymbol{H}_{ij}}{\partial \boldsymbol{q}_{R_j}}\right)^{n+1,m} \tag{39}$$

In these summations, $\mathcal{N}_L(i)$ denotes the subset of $\mathcal{N}(i)$ for which $j < i$, and $\mathcal{N}_U(i)$ denotes the subset of $\mathcal{N}(i)$ for which $j > i$. The iteration is initialized using $\Delta_m \boldsymbol{q}^{k=0} = 0$. Further details of the LU/SGS scheme used are given by Hyams [33].

### 5.2. Implementation as a parallel algorithm

The iterative implicit algorithm is implemented in a scalable parallel code based on coarse-grained domain decomposition for concurrent solution within subdomains assigned to multiple distributed-memory processors that communicate using MPI message passing. The parallel algorithm itself requires only a minor modification to remove sequential dependencies in the SGS iteration algorithm (38). In this modification, each lower/upper SGS sweep is started concurrently for each subdomain processor using the most recent lower or upper iteration values for the solution. Solution values at each subdomain interface are synchronously updated following each lower and upper SGS sweep. This modification has no effect on the converged Newton iteration, and since a coarse-grained decomposition is used, it has an almost imperceptible effect on the iterative convergence rate of the linear SGS iteration. In effect, the parallel algorithm becomes a block-Jacobi LU/SGS iteration by subdomains, instead of a global LU/SGS iteration.

The parallel software implementation uses the METIS software [39] to partition the unstructured grid into subdomains. Aside from code initialization and I/O operations, the code must update the solution data at domain interfaces within the hierarchical iteration procedure at each time step. Subdomain-interface values of $\Delta\boldsymbol{q}$ are updated after each LU/SGS iteration sweep. Once this linear iteration is completed, interface values of $\boldsymbol{q}$ are updated for the next Newton iteration. Finally, interface values of the solution gradient $\bigtriangledown\boldsymbol{q}$ are communicated across subdomains following their calculation during each Newton iteration. The present parallel implementation for the two-dimensional SWE closely follows that in [33,36] for the three-dimensional Navier–Stokes equations.

All of the present results were computed on the UTC SimCenter's Dell 1300-core 325-node diskless Linux cluster, which is configured with dual-core Intel EM64T 3.0 GHz Xeon processors, 4 GB RAM per node, and GigE interconnect with 576-port Force 10 E1200 switch. The runtime performance of the SWE code for a problem with one million grid points and one thousand time steps is about 264 s, using 32 cores. This timing assumes that 7 Newton and 7 SGS iterations are performed at each time step. The runtime for a problem of this size scales linearly below 32 cores, but the performance gradually falls below linear scalability with decomposition in the range of 32–64 cores, due to an increasing communication/computation ratio. This two-dimensional problem requires a granularity on the order of 30,000 points per core for negligible message-passing overhead and linear scalability.

## 6. Validation results in one and two dimensions

### 6.1. Dam-break problem

The one-dimensional dam-break problem consists of a region $0 \leqslant x \leqslant 1$ having uniform left and right water levels, $h_L$ and $h_R$, and separated by a wall located at $x = 0.5$. Impulsive removal of the wall at $t = 0$ produces a left-running rarefaction wave that reduces the initial depth, and a right-running discontinuous wave front (a shock or bore) that increases the initial depth. Exact solutions for this Riemann problem are given by Toro [2] for both wet-bed and computationally more challenging dry-bed problems. Cases of both types are considered here.

The first test case is a dam break on a wet bed with $h_L = 1.0$ and $h_R = 0.05$, using a high-resolution grid of 1001 points with $\Delta x = 0.001$. The Roe flux and the primitive-variable flux (19) were both tested using the implicit scheme, with Roe-averaging for both fluxes. The primitive-variable flux does not satisfy Roe's flux-difference property using either Roe or algebraic averaging, but both fluxes are insensitive to the choice of averaging variables for this problem. Second-order linear reconstruction ($\psi = 0$) was used for both fluxes. The second-order implicit scheme (33) was run with $\theta = \frac{1}{2}$, constant $\varepsilon = 0.001$, $\gamma = 0$, and with 5 Newton and 5 SGS iterations. The initial discontinuity at $x = 0.5$ is located at $x = 0.75$ after 1183 steps with $\Delta t = 0.0002$. The *CFL* number in one dimension is defined by $CFL = \max_i(|u_i| + \sqrt{h_i})\Delta t/\mathcal{A}_i$, and the *CFL* for this case was about 0.28.

The primitive-variable and Roe-flux solutions are shown at full scale in Fig. 4, and both are in very close agreement with the exact solution. Fig. 5 shows details of these same solutions near the rarefaction-wave and shock fronts. Both solutions agree with the exact solution except for very small overshoots at the trailing edge of the rarefaction wave. These overshoots are about 60% larger with the Roe flux than with the primitive-variable flux. These comparisons show that the primitive-variable flux is effective in representing solution discontinuities and maintaining the correct propagation speed.

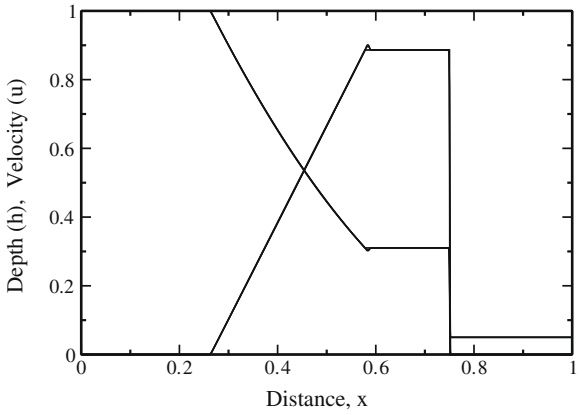


**Fig. 4.** Dam break: implicit solutions for the primitive-variable and Roe fluxes compared with the exact solution.

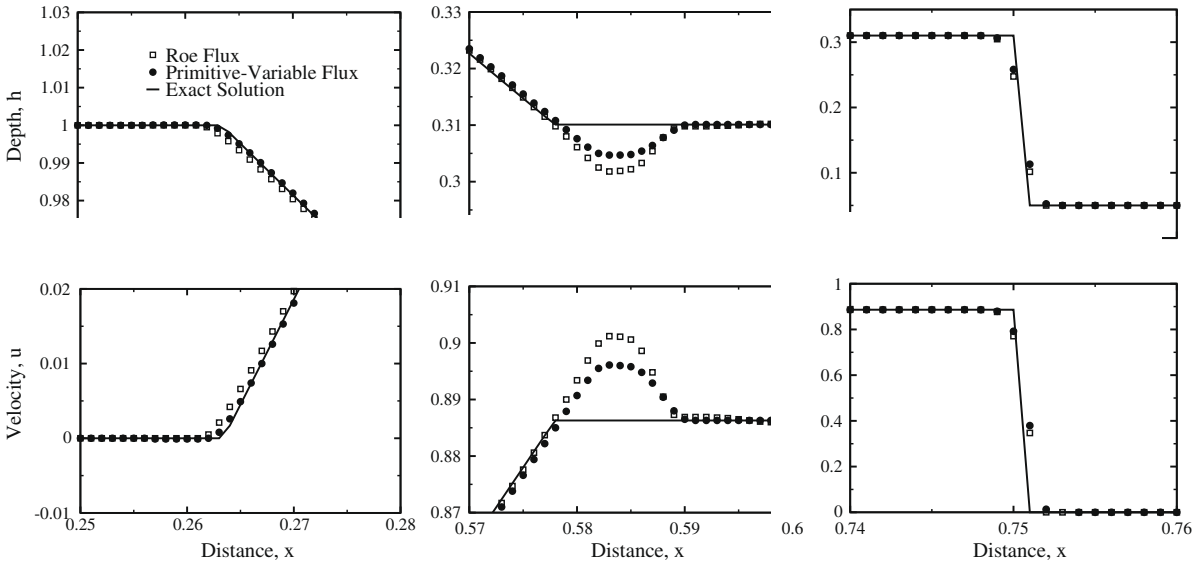


**Fig. 5.** Dam break: details of solutions near the rarefaction wave ends and shock front for primitive-variable and Roe fluxes.

### 6.2. Dam break on a dry bed

The test problem for a dam break on a dry bed has initial depths of $h_L = 1$ and $h_R = 0$, with initial discontinuity at $x = 0.5$ and $\Delta x = 0.001$. Solutions with 1001 points were computed using the explicit scheme (30), the time-linearized explicit scheme (32), and the implicit scheme (33), each with first, second, and third-order fluxes. In these one-dimensional solutions, the following MUSCL reconstruction of $\boldsymbol{q}_L$ and $\boldsymbol{q}_R$ was used with a van Leer limiter, as implemented by Anderson, Thomas, and van Leer [40]:

$$(\boldsymbol{q}_L)_i = \boldsymbol{q}_i + \frac{\phi}{4}[(1-\chi)(\boldsymbol{q}_i - \boldsymbol{q}_{i-1}) + (1+\chi)(\boldsymbol{q}_{i+1} - \boldsymbol{q}_i)] \tag{40}$$

This formula gives first-order ($\phi = 0$), second-order ($\chi = -1$), and third-order ($\chi = 1/3$) fluxes. The explicit solutions are first-order in time, and the implicit solutions are second order ($\theta = \frac{1}{2}$).

In cases with a wet/dry front, the Newton iteration for the implicit scheme is sensitive to the combination of $\varepsilon$ and $\Delta t$ during the initial start-up period, and a practical method for choosing $\varepsilon$ in (34) is needed to secure Newton convergence. Here, the Newton convergence was studied empirically by running a series of start-up calculations for one or more time steps using different values for $\varepsilon$ and time step. It is reasonable to expect that the unsteady residual $\mathbb{R}_U$ should be reduced by a factor of at least $10^{-2}$ to $10^{-3}$ below its initial value to ensure time accuracy. An initial value of $\varepsilon = 10^{-2}$ or $10^{-3}$ worked well in most cases. Once the initial pattern of Newton convergence is established, $\varepsilon$ can be reduced gradually to very small values, and the iteration then converges very rapidly. For this dam-break problem, both implicit and time-linearized explicit

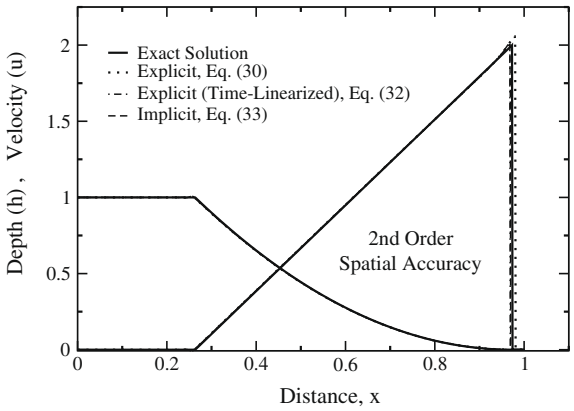


**Fig. 6.** Dam break on a dry bed: solutions for explicit and implicit schemes with second-order flux are compared with the exact solution at time $t = 0.2366$.

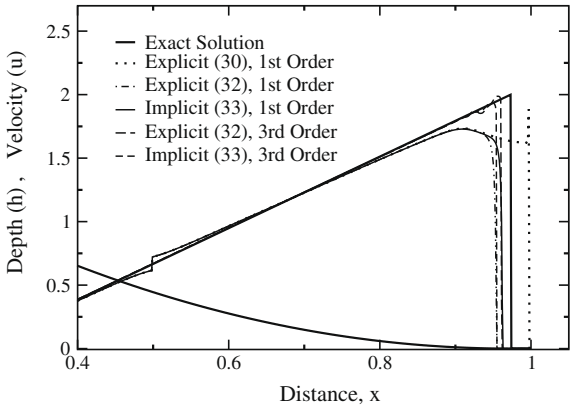


**Fig. 7.** Dam break on a dry bed: solutions for explicit and implicit schemes with first and third-order fluxes are compared with the exact solution at time $t = 0.2366$.

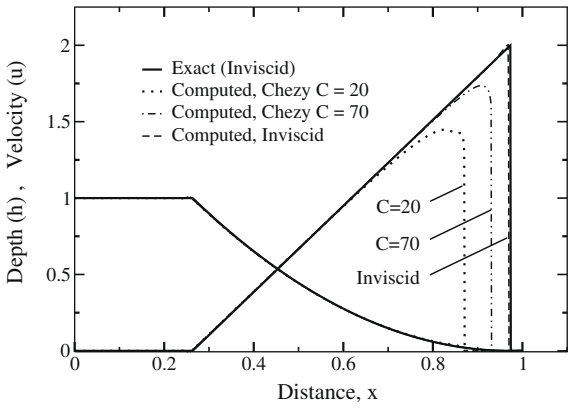


**Fig. 8.** Dam break on a dry bed: solutions computed with friction using the implicit scheme (33) with second-order flux are compared with inviscid solutions at time $t = 0.2366$.

schemes were run by reducing $\varepsilon$ from $10^{-2}$ to $10^{-7}$ over the first 1800 steps. The implicit scheme used 4 Newton and 5 SGS iterations.

Computed results at time $t = 0.2366$ are given in Fig. 6 for the second-order fluxes and in Fig. 7 for the first and third-order fluxes. Sensitivity to time step was tested by running test cases for step sizes in the range $(5 \times 10^{-6} \leqslant \Delta t \leqslant 2 \times 10^{-4})$. All of the solutions in Fig. 6 and 7 were run with $\Delta t = 2.5 \times 10^{-5}$ $(CFL = 0.07)$ and are essentially time-step independent. The differences between solutions are therefore due to the different fluxes and time-integration schemes used.

The second-order schemes were stable for $\Delta t$ up to $10^{-4}(CFL = 0.14)$ for the explicit schemes and up to $2 \times 10^{-4}(CFL = 0.28)$ for the implicit scheme. The depth distribution for all of these solutions agrees well with the exact solution, but there are differences in how accurately the velocity front is predicted.

The implicit and explicit solutions with the second-order flux are all in very close agreement with the exact solution, as is evident in Fig. 6. The conserved-variable explicit scheme (30) gives a very slight over-prediction of the exact peak-velocity value of 2.0 at the front. The solutions with first and third-order fluxes are shown in Fig. 7. The solutions with first-order flux have greater dissipative smoothing of the velocity front and a small rarefaction shock at the initial discontinuity location. The conserved-variable explicit scheme (30) has a somewhat erratic behavior at the velocity front, and it was unstable for all third-order cases tested. The two solutions with third-order flux agree with the velocity front almost as well as the second-order flux, except for slight oscillations near the front.

Since frictional effects will become increasingly important as depth is reduced by the advancing wet/dry front, it is of interest to examine the degree of influence of friction on the wet/dry front. Computed results with added friction are shown in Fig. 8 for the Chezy model given in Eq. (29). For smooth surfaces, the Chezy coefficient $C = 20$ corresponds to $Re_{Rh} \approx 160$, and $C = 70$ corresponds to $Re_{Rh} \approx 10^5$. The solutions in Fig. 8 agree with the inviscid Riemann solution except in the region near the wet/dry front where frictional effects are significant. With friction, the velocity front still propagates as a discontinuity but at a reduced speed, and the depth and velocity distributions are modified. Although the exact solution and error with frictional effects are not known, these solutions are essentially independent of the time step, and they are believed to be time-accurate solutions, subject of course to the modeling error for friction.

### 6.3. Free oscillation in a closed parabolic basin

Exact solutions for normal-mode oscillations of an axisymmetric long wave in a circular parabolic basin were given by Thacker [41]. These have been compared previously with numerical solutions by Murillo, et. al. [24] and Casulli and Zanolli [42]. The solution is given here in nondimensional variables, using the dimensional reference quantities given in Section 2. The free-surface elevation $\zeta$ is given by

$$\zeta(r,t) = \frac{\sqrt{1-A^2}}{1-A\cos(\omega t)} - \frac{r^2}{a^2}\left[\frac{1-A^2}{[1-A\cos(\omega t)]^2} - 1\right] - 1 \tag{41}$$

Here, $r$ is radial distance, $t$ is time, $A = (a^4 - r_0^4)/(a^4 + r_0^4)$, and $\omega = \sqrt{8}/a$. The still-water surface elevation is defined by $\zeta = 0$, and the parabolic bottom elevation $b(r) = r^2/a^2 - 1$ is measured from the still-water surface. The still-water surface has unit depth at the centerline $r = 0$, and its circular shoreline is located at $r = a$. The initial water surface has a parabolic cross section that intersects the bottom at $r = r_0$.

This axisymmetric inviscid solution was run as a two-dimensional solution in a square region of nondimensional size $(-100 \leqslant x, y \leqslant 100)$ using a uniform rectangular grid of $257 \times 257$ (65,536) points. The test case has a still-water shoreline $a = 75$, initial shoreline $r_0 = 60$, and period of oscillation $T = 2\pi/\omega = 166.608$. These conditions correspond to the test case of [24].

The first solution was computed using the implicit scheme (33), with second-order time accuracy $(\theta = \frac{1}{2}, \gamma = 0)$, and using the second-order flux $(\psi = 0)$ with no limiter and exact source-term balancing as in (21). Sixteen periods of oscillation were computed using 16 processors and a time step of $\Delta t = 0.017$, with constant $\varepsilon = 0.001$, and using 4 Newton and 7 LU/SGS iterations. This time step corresponds to 9800 steps per oscillation period.

Computed results are given in Fig. 9 for surface elevation $\zeta$ vs. time $t/T$ at selected radial distances ($x$ at $y = 0$). Fig. 10 gives contours of depth $h$ at times corresponding to maximum and minimum depth, and of velocity components $u$ and $v$

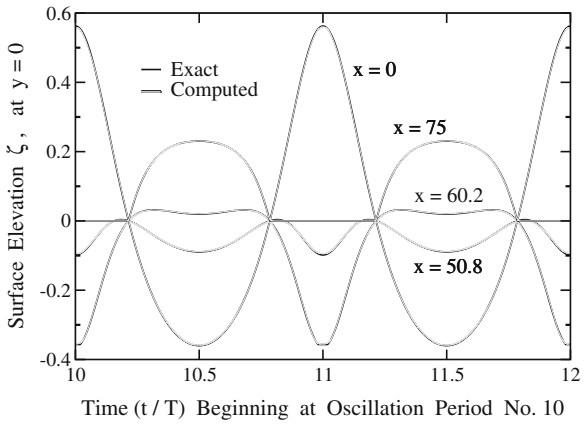


**Fig. 9.** Surface oscillation in a parabolic basin: computed surface elevation $\zeta$ vs. time at selected points along the $x$ axis.

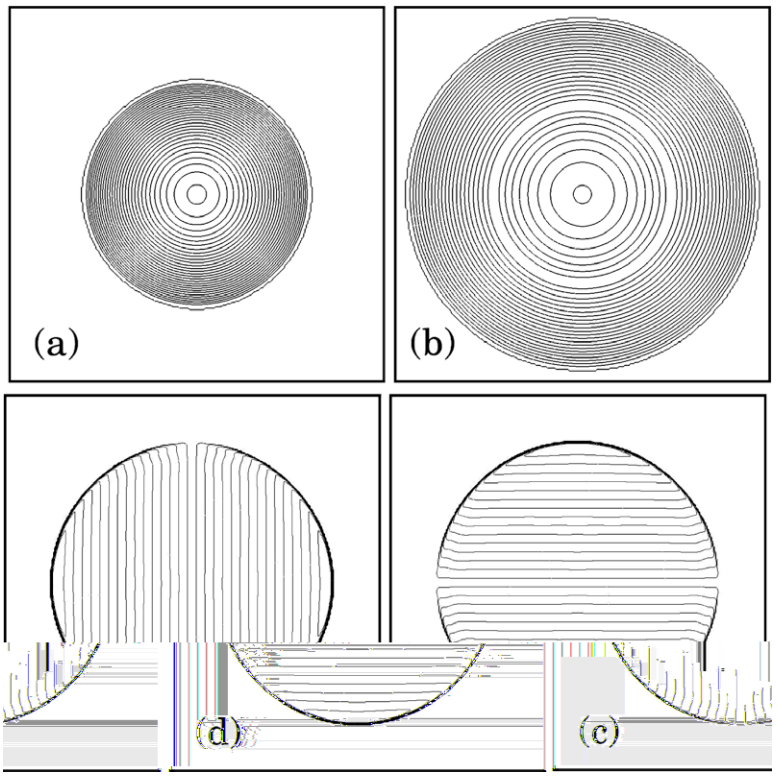


**Fig. 10.** Contours of depth $h$ and velocity $u$ and $v$ in the parabolic basin, shown during the 10th oscillation period: (a) maximum depth at $t/T = 10.0$, (b) minimum depth at $t/T = 10.5$, and (c) $u$, and (d) $v$ at $t/T = 10.214$.

at $t/T = 10.214$. Plots of surface elevation $\zeta$ and velocity $u$ are shown in Figs. 11 and 12 at selected times during the 10th period of oscillation. All of these computed results are in very close agreement with the exact analytical solution.

Some additional tests for different time steps and algorithm variants were computed to quantify the error for this test case, and these results are given in Table 1. The CFL number for the solutions in Table 1 is defined as

$$CFL = \max_i \left[ \sum_{j \in \mathcal{N}_L(i)} \max_m |\lambda_{ij}^{(m)}| \, \frac{\Delta t}{\mathcal{A}_i} \right] \tag{42}$$

The *Max*|*Error*| column in Table 1 gives the absolute value of the maximum difference between the computed and exact solutions, sampled over all four of the $x$ locations shown in Fig. 9 and at each time step.

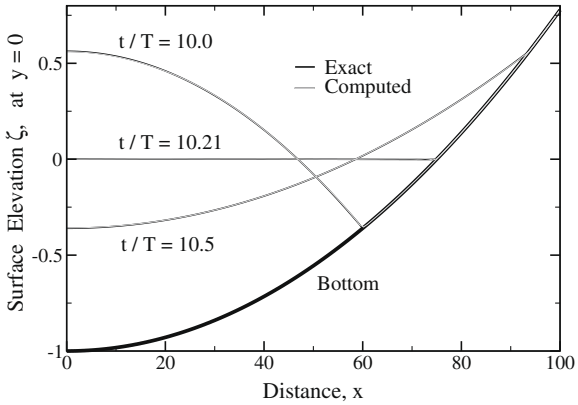


**Fig. 11.** Surface oscillation in a parabolic basin: computed surface elevation $\zeta$ vs. $x$ at $y = 0$ during the 10th period of oscillation.

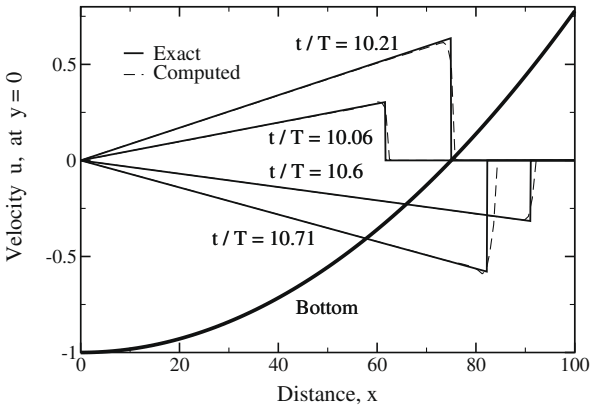


**Fig. 12.** Surface oscillation in a parabolic basin: computed velocity $u$ vs. $x$ at $y = 0$ during the 10th period of oscillation.

**Table 1**

| Case | Algorithm | Flux | $\Delta t$ | *CFL* | Periods | *Max*\|*Error*\| |
|---|---|---|---|---|---|---|
| 1a | Implicit, $O(\Delta t^2)$ | 2nd order | 0.2 | 1.5 | 4 | $1.00 \times 10^{-2}$ |
| 1b | | | 0.1 | 0.75 | 4 | $7.28 \times 10^{-3}$ |
| 1c | | | 0.05 | 0.375 | 4 | $6.30 \times 10^{-3}$ |
| 1d | | | 0.017 | 0.13 | 4 | $5.38 \times 10^{-3}$ |
| 1e | | | 0.017 | 0.13 | 16 | $7.70 \times 10^{-3}$ |
| 2a | Implicit, $O(\Delta t^2)$ | 3rd order | 0.017 | 0.13 | 16 | $2.65 \times 10^{-3}$ |
| 2b | | | 0.2 | 1.5 | 4 | $7.49 \times 10^{-2}$ |
| 3a | Implicit, $O(\Delta t)$ | 2nd order | 0.017 | 0.13 | 4 | $3.93 \times 10^{-2}$ |
| 3b | | | 0.017 | 0.13 | 12 | $1.24 \times 10^{-1}$ |
| 4a | Explicit, $O(\Delta t)$ | 2nd order | 0.017 | 0.13 | 4 | $8.16 \times 10^{-2}$ |
| 4b | | | 0.017 | 0.13 | 12 | $1.88 \times 10^{-1}$ |

In Cases 1a–d, the error after four oscillation periods increases only slightly with time step up to $\Delta t = 0.2$. In Case 1e, the error using $\Delta t = 0.017$ remains small after 16 periods. The solution for Case 2a with third-order flux $\left(\psi = \frac{1}{2}\right)$ has a smaller error than Case 1e, but comparing Cases 1a and 2b, this error increased faster with time step size, and on balance the second-order flux is deemed preferable. Both algorithms were unstable for $\Delta t = 0.3 (CFL = 2.25)$. In Cases 3 and 4, the implicit scheme with $O(\Delta t)$ accuracy $(\theta = 0)$ and second-order flux is compared with the corresponding time-linearized explicit scheme (30) for four and twelve periods. The implicit scheme is slightly more accurate, and in both schemes, the increased error after twelve periods is due to a small phase shift rather than significant amplitude damping. The stability limit for the explicit scheme was approximately $\Delta t = 0.09 (CFL = 0.69)$.

### 6.4. Forced breaking wave with friction interacting with a sloped bottom

This test problem consists of a forced breaking wave with friction interacting with a sloped bottom, which demonstrates a more complex wave motion that includes wetting and drying as well as multiple discontinuous interacting wave fronts. The single-depth shallow-water model captures breaking waves as discontinuities. The computational domain for this case is $(0 \leqslant x \leqslant 100)$, and the bottom surface is defined by

$$b(x) = \begin{cases} 0 & \text{if } 0 \leqslant x \leqslant 30 \\ (x - 30)/60 & \text{if } 30 \leqslant x \leqslant 100 \end{cases} \tag{43}$$

The initial condition is a still-water surface of $h_s = 1$, with $h$ set to zero on the dry portion of the bottom surface. A forced wave motion with $h(t) = 1 + 0.1 \sin(2\pi t/40)$ is imposed at the left boundary at the start of the calculation. The case was run with $\Delta t = 0.1$ and $\Delta x = 0.1$, giving a resolution of 400 steps per cycle and 400 points per wavelength for the forced wave. The Chezy friction model with $C = 70 (Re_{Rh} \approx 10^5)$ was used. The wave motion is shown in Fig. 13 as a sequence of surface wave patterns at selected points in time. Although the error due to grid resolution and friction model is not known, these results appear reasonable. The degree of time resolution was tested further by running this same case with a five-fold in-

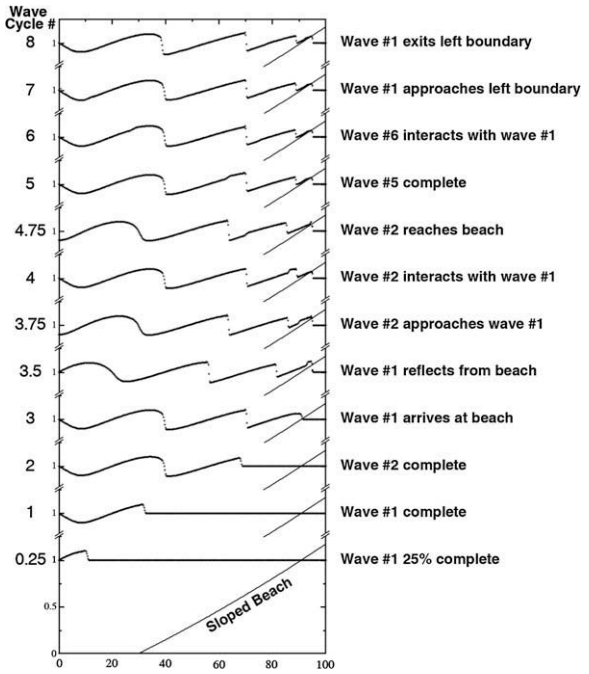


**Fig. 13.** Time sequence of surface wave patterns for a forced wave interacting with a sloped beach ($C = 70, \Delta x = 0.1, \Delta t = 0.1, \varepsilon = 0.05$, 1001 points). The maximum *CFL* is about 1.0.

crease in time step ($\Delta t = 0.5$), giving only 80 steps per cycle. Although a slight amount of dispersive error was evident, the overall wave patterns and propagation speeds were unchanged.

## 7. Demonstration for artificial coastal flooding by a propagating wave front using Gulf of Mexico bathymetry

This section summarizes computed results for a test problem that demonstrates wetting and drying of coastal areas in the Gulf of Mexico, caused by an elevated wave front that propagates into still water. A coastline-conforming unstructured grid was constructed for the Gulf of Mexico, bounded by its coastline and inflow/outflow boundaries as shown in Figs. 14 and 15. To create some floodable dry ground within the gridded solution domain, the average Gulf depth of 1570 m was reduced by 30 m to a new artificial still-water initial condition with average depth of 1540 m. This exposes significant areas of floodable coastal dry ground within the gridded area, without the need to add new onshore land areas beyond the coastline. The unsteady simulation is initiated by an increase of 30 m in water elevation between the Yucatan peninsula and Cuba, which initiates a wave propagating across the Gulf that eventually floods the dry coastal areas. The purpose of this solution is to test the functionality of the present method for large-scale complex nonuniform grids with highly irregular bathymetry data. These results do not account for physical processes such as Coriolis forces, atmospheric pressure variations, wind shear, thermohaline distributions, or river inflows, and are not intended to be accurate predictions of specific physical observations or measurements in the Gulf of Mexico.

The coastal geometry for the Gulf of Mexico was obtained from the USCG Coastline Extractor of Signell [43], and the bathymetry was obtained from the Coastal Relief Model of Divens and Metzger [44], available from databases at the NOAA National Geophysical Data Center. The bathymetry data was non-dimensionalized using the average depth of the Gulf of Mexico, $d' = 1570$ m. The grid was generated using the hierarchical unstructured grid generator of Karman [45], which employs a hierarchical tree structure and recursive cell division to discretize regions surrounded by complex geometries. Fig. 14 shows the grid of 846,885 points, with a shaded overlay of bathymetry data. Close-up views of the grid near the coast of Louisiana are shown in Fig. 15 to illustrate the hierarchical grid structure, degree of nonuniformity, and resolution near the coastlines. The maximum cell size is about 6 km in the central Gulf, and the smallest cells along the coast are about 80 m, giving a cell length-scale ratio of 75:1.

Although the Gulf of Mexico coastline data is very detailed and well fitted by the locally refined unstructured grid, the bathymetry data is much coarser than the coastline data and the computational grid. It was therefore necessary to interpolate the coarse-grid bathymetry data onto a computational grid that has much higher local resolution than the data along coastlines and islands. This interpolation process artificially introduced some localized but highly irregular and steep bottom slopes, especially near small coastal islands. In addition, many of these irregularities were exposed as dry ground when the still-water depth was artificially reduced. To avoid difficulties with abrupt changes in bottom topography on dry areas, the

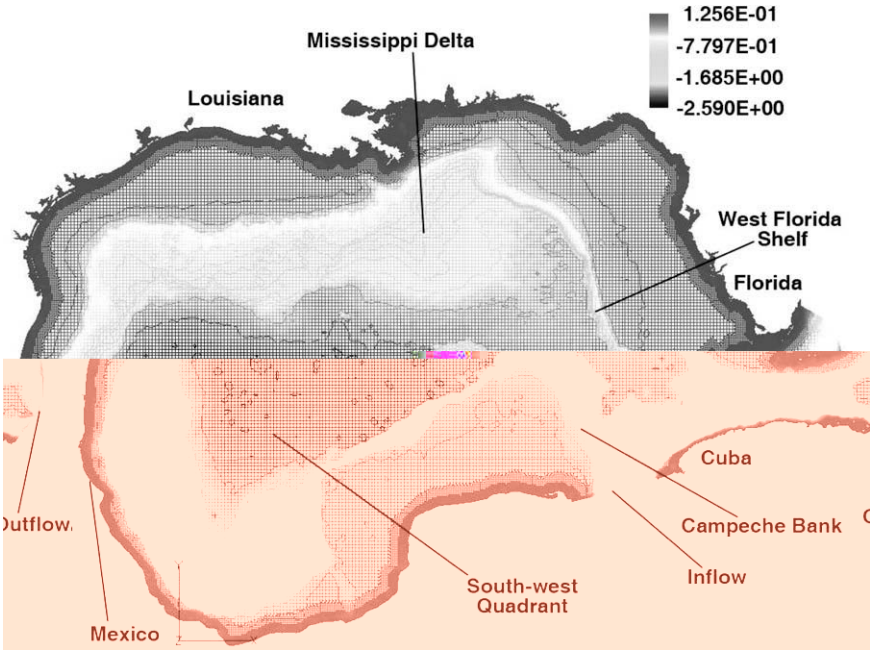


**Fig. 14.** The Gulf of Mexico grid with shaded overlay of bathymetry data.

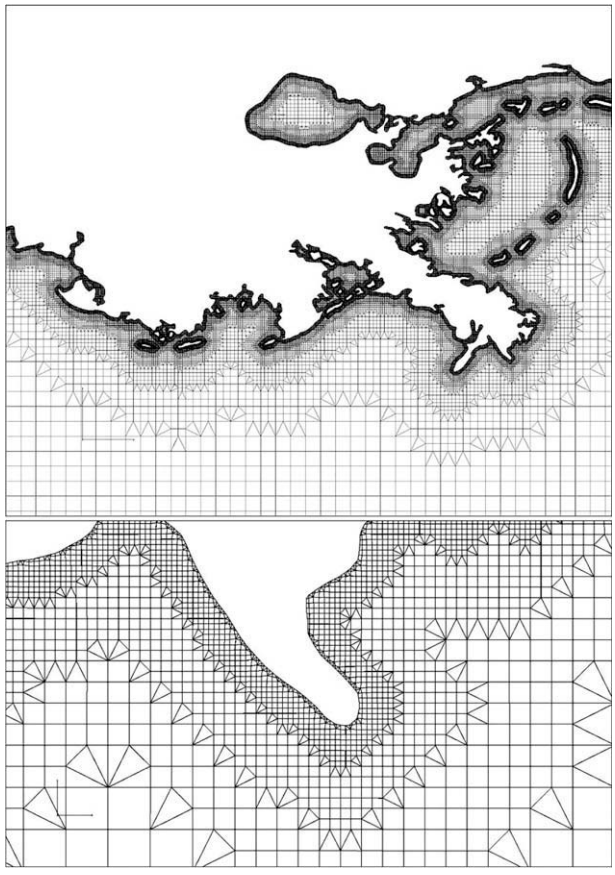

**Fig. 15.** Close-up views of the grid near the southern coast of Louisiana.

interpolated bathymetry was modified so that all bottom surfaces with elevation higher than the reduced still-water level were redefined to be flat surfaces at the still-water level. This simplification for the floodable dry ground also gave a simpler flat bottom, and the dry-bed source balancing in (21) was not used. This test case was run using a constant $\varepsilon = 0.001$ and with 7 Newton and 7 LU/SGS iterations at each time step. With these parameters, the Newton iteration did not converge in some localized regions of irregular and steep bottom. To secure Newton convergence, an artificial minimum depth of $h_{min} = 1 \times 10^{-08}$ (about $1.6 \times 10^{-5}$ m) was imposed on the solution for this case.

Fig. 16 shows the initial still-water surface (black) and the coastal dry-ground region (gray). The surface height was specified at the inflow boundary, and a flow direction normal to the boundary was specified by enforcing $v_{bc} = u_{bc}\hat{n}_y$ at the boundary. A similar flow-angle condition was used at the outflow boundary. Coastlines and islands were treated as stationary impermeable boundaries with zero normal velocity.

Computed results are shown in Figs. 17–20. This solution was computed for a duration of 44 h of physical time, using a time step $\Delta t = 1$ (12.65 s). The wave front appropriately travels faster in deep water, and the wave front initially propagates northward between the Campeche Bank and the West Florida Shelf, and eastward along the north coast of Cuba. The wave travels reaches a location half way to the Louisiana coast in about 15 min and first reaches the Louisiana coast in about 49 min. The wave then proceeds westward toward the deepest portion of the Gulf basin in the Southwest Quadrant. Friction impedes the flow as water floods initially dry land, slowing down the wave. Barrier islands also impede the progress of the wavefront. The wave almost completely fills the coastal regions after 44 h, except where small islands obstruct its path near Louisiana and behind the Florida keys. Fig. 17 shows the surface distribution after about 2.5 h, when the wave front has first propagated throughout the gulf except for shallow coastal areas. Figs. 18–20 give detailed views of the Louisiana coast at 2.5, 10, and 33 h.

**Fig. 16.** Initial bathymetry and coastal elevation (black indicates initial water surface; gray indicates dry areas).

**Fig. 17.** Flooding in the Gulf of Mexico, $T \approx 2.5$ h (gray shading indicates elevation above original water surface).

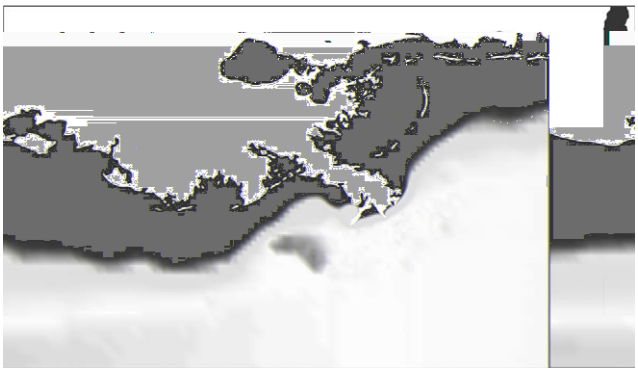

**Fig. 18.** Flooding near the Louisiana coast, $T \approx 2.5$ h (gray shading indicates elevation above original water surface).

**Fig. 19.** Flooding near the Louisiana coast, $T \approx 10$ h (gray shading indicates elevation above original water surface).

**Fig. 20.** Flooding near the Louisiana coast, $T \approx 33$ h (gray shading indicates elevation above original water surface).

## 8. Conclusions

A conservative primitive-variable Riemann flux was developed for the shallow water equations and implemented with a finite-volume discretization for unstructured grids. The source term discretization balances the discrete flux integal for both still water and dry regions. Characteristic-variable boundary conditions that include the influence of source terms were developed. Conserved and primitive-variable explicit schemes and a backward-differentiation implicit scheme were developed for the time integration.

The primitive-variable and Roe fluxes were equally effective in representing the discontinuity and maintaining the correct propagation speed in solutions for the one-dimensional dam break on a wet bed. For the dam break on a dry bed, solutions for depth and velocity were in close agreement with the exact solution for both explicit and implicit schemes using the sec-

ond-order primitive-variable flux. This case was computed with and without friction and did not require special treatment of the propagating wet/dry front. Computed solutions for long-wave oscillations in a circular parabolic basin maintained very close agreement with the exact analytical solution for 16 oscillation periods. The implicit scheme with second-order time and flux approximations had good accuracy for *CFL* up to 1.5 for this problem. Another test case demonstrated a forced breaking wave with friction interacting with a sloped bottom, with multiple interacting wave fronts. Finally, the method was demonstrated for a highly nonuniform, coastline-conforming unstructured grid using Gulf of Mexico bathymetry. Although a very small minimum depth of $h_{min} = 1 \times 10^{-08}$ was imposed to obtain Newton convergence for this difficult case, the solution was successful in demonstrating forced-wave coastal flooding.

## Acknowledgments

The authors gratefully acknowledge support by the Center of Excellence for Applied Computational Science and Engineering at the University of Tennessee at Chattanooga. The authors also wish to thank their colleagues Dr. Steve Karman for his assistance in generating unstructured grids for the Gulf of Mexico, and Dr. Kidambi Sreenivas for his suggestions on improving Newton convergence.

## References

[1] E. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, second ed., Springer, Berlin, 1999.
[2] E. Toro, Shock-Capturing Methods for Free-Surface Shallow Flows, John Wiley & Sons, Ltd., New York, 2001.
[3] P. Roe, Approximate Riemann solvers, parameter, vector and difference schemes, Journal of Computational Physics 43 (1981) 357.
[4] P. Glaister, Approximate Riemann solvers of the shallow water equations, Journal of Hydraulic Research 26 (3) (1988) 293–306.
[5] D. Whitfield, Numerical solution of the shallow water equations, Tech. Rep. MSSU-EIRS-ERC-96-4, Engineering Research Center, Mississippi State University, 1996.
[6] K. Anastaciou, C. Chan, Solution of the 2D shallow water equations using the finite volume method on unstructured triangular meshes, International Journal for Numerical Methods in Fluids 24 (1997) 1225–1245.
[7] M. Fujihara, A. Borthwick, Godunov-type solution of curvilinear shallow water equations, Journal of Hydraulic Engineering 126 (11) (2000) 827–836.
[8] D. Ambrosi, Approximation of shallow water equations by Roe's Riemann solver, International Journal for Numerical Methods in Fluids 20 (2) (1995) 157–168.
[9] M. Nujic, Efficient implementation of non-oscillatory schemes for the computation of free surface flows, Journal of Hydraulic Research 33 (1) (1995) 101–111.
[10] R. LeVeque, Balancing source terms and flux gradients in high resolution Godunov methods: the quasi-steady wave-propagation algorithm, Journal of Computational Physics 146 (1998) 346–365.
[11] M.E. Vázquez-Cendón, Improved treatment of source terms in upwind schemes for the shallow water equations in channels with irregular geometry, Journal of Computational Physics 148 (1999) 497–526.
[12] M. Hubbard, P. García-Navarro, Flux difference splitting and the balancing of source terms and flux gradients, Journal of Computational Physics 165 (2000) 89–125.
[13] P. García-Navarro, M. Vázquez-Cendón, On numerical treatment of the source terms in the shallow water equations, Computers and Fluids 29 (8) (2000) 951–979.
[14] J. Zhou, D. Causon, C. Mingham, D. Ingram, The surface gradient method for the treatment of source terms in the shallow water equations, Journal of Computational Physics 168 (2001) 1–25.
[15] L. Gascón, J. Corberán, Construction of second-order TVD schemes for non-homogeneous hyperbolic conservation laws, Journal of Computational Physics 172 (2001) 261–297.
[16] J. Burguete, P. García-Navarro, Efficient construction of high-resolution TVD conservative schemes for equations with source terms: Application to shallow water flows, International Journal for Numerical Methods in Fluids 37 (2) (2001) 209–248.
[17] B. Rogers, M. Fujihara, A. Borthwick, Adaptive Q-tree Godunov-type scheme for shallow water equations, International Journal for Numerical Methods in Fluids 35 (2001) 247–280.
[18] B. Rogers, A. Borthwick, P. Taylor, Mathematical balancing of flux gradient and source terms prior to using Roe's approximate Riemann solver, Journal of Computational Physics 192 (2003) 422–451.
[19] P. Sleigh, P. Gaskell, M. Berzins, N. Wright, An unstructured finite volume algorithm for predicting flow in rivers and estuaries, Computers and Fluids 27 (4) (1998) 479–508.
[20] N. Dodd, Numerical model of wave run-up, overtopping, and regeneration, Journal of Waterways Port, Coastal, and Ocean Engineering 124 (2) (1998) 73–81.
[21] S. Bradford, B. Sanders, Finite-volume model for shallow-water flooding of arbitrary topography, Journal of Hydraulic Engineering 128 (3) (2002) 289–298.
[22] T. Yoon, S.-K. Kang, Finite volume model for two-dimensional shallow water flows on unstructured grids, Journal of Hydraulic Engineering 130 (7) (2004) 678–688.
[23] J. Murillo, P. García-Navarro, J. Burguete, Analysis of a second-order upwind method for the simulation of solute transport in 2D shallow water flow, International Journal for Numerical Methods in Fluids 56 (2008) 661–686.
[24] J. Murillo, P. García-Navarro, J. Burguete, P. Brufau, A conservative 2D model of inundation flow with solute transport over dry bed, International Journal for Numerical Methods in Fluids 52 (10) (2006) 1059–1092.
[25] P. Brufau, P. García-Navarro, M.E. Vázquez-Cendón, Zero mass error using unsteady wetting–drying conditions in shallow flows over dry irregular topography, International Journal for Numerical Methods in Fluids 45 (2004) 1047–1082.
[26] E. Turkel, Preconditioning techniques in computational fluid dynamics, Annual Reviews in Fluid Mechanics (1999) 385–416.
[27] W. Briley, L. Taylor, D. Whitfield, High-resolution viscous flow simulations at arbitrary Mach number, Journal of Computational Physics 184 (2003) 79–105.
[28] E.F. Toro, A. Siviglia, Price: primitive centred schemes for hyperbolic systems, International Journal for Numerical Methods in Fluids 42 (12) (2003) 1263–1291.
[29] V. Casulli, E. Cattani, Stability, accuracy and efficiency of a semi-implicit method for three-dimensional shallow water flow, Computers and Mathematical Applications 27 (4) (1994) 99–112.
[30] V. Casulli, R.A. Walters, An unstructured grid, three-dimensional model based on the shallow water equations, International Journal for Numerical Methods in Fluids 32 (2000) 331–348.
[31] T. Weiyan, Shallow Water Hydrodynamics: Mathematical Theory and Numerical Solution for a Two-dimensional System of Shallow Water Equations, Elsevier, Amsterdam, 1992.

[32] W. Anderson, D. Bonhaus, An implicit upwind algorithm for computing turbulent flows on unstructured grids, Computers and Fluids 23 (1) (1994) 1–21.
[33] D. Hyams, An investigation of parallel implicit solution algorithms for incompressible flows on unstructured topologies, Ph.D. Thesis, Mississippi State University, May 2000.
[34] T. Barth, D. Jesperson, The design and application of upwind schemes on unstructured meshes, AIAA Paper 93-0192, 1993.
[35] D. Whitfield, J. Janus, Discretized Newton-relaxation solution of high resolution flux-difference split schemes, AIAA Paper 841552, 17th Fluid Dynamics, Plasma Dynamics and Lasers Conference, Snowmass, CO, 1984.
[36] K. Sreenivas, D. Hyams, D. Nichols, B. Mitchell, L. Taylor, W. Briley, D. Whitfield, Development of an unstructured parallel flow solver for arbitrary Mach numbers, AIAA Paper 2005-0325, 43rd Aerospace Sciences Meeting and Exhibit, Reno, NV, 2005.
[37] D. Whitfield, L. Taylor, Discretized Newton-relaxation solution of high resolution flux-difference split schemes, AIAA Paper 911539, 1991.
[38] L. Taylor, D. Whitfield, Unsteady three-dimensional incompressible Euler and Navier–Stokes solver for stationary and dynamic grids, AIAA Paper 911650, 1991.
[39] G. Karypis, V. Kumar, METIS: A Family of Multilevel Partitioning Algorithms. <http://glaros.dtc.umn.edu/gkhome/views/metis>, 1998.
[40] W. Anderson, J. Thomas, B. van Leer, Comparison of finite volume flux vector splittings for the Euler equations, AIAA Journal 24 (1986) 1453.
[41] W.C. Thacker, Some exact solutions to the nonlinear shallow-water wave equations, Journal of Fluid Mechanics 107 (1981) 499–508.
[42] V. Casulli, P. Zanolli, Comparing analytical and numerical solution of nonlinear two and three-dimensional hydrostatic flows, International Journal for Numerical Methods in Fluids 53 (2007) 1049–1062.
[43] R. Signell, Coastline Extractor, National Geophysical Data Center. <http://rimmer.ngdc.noaa.gov/mgg/coast/getcoast.html>, 2006.
[44] D. Divins, D. Metzger, NGDC Coastal Relief Model. <http://www.ngdc.noaa.gov/mgg/coastal/coastal.html>, 2006.
[45] S. Karman Jr., Hierarchical unstructured mesh generation, AIAA Paper 2004-0613, 2004.