

An object-oriented overland flow solver for watershed flood inundation predictions: case study of Ulus basin, Turkey

Burak Turan^{1,2}, Keh-Han Wang^{1*}

¹ Department of Civil and Environmental Engineering, University of Houston, Houston, TX, 77204-4003, USA.

² Current address: Emek 4, Cad. No:158/4, Cankaya, Ankara, Turkey. E-mail: burak.turan@nfbproje.com

* Corresponding author. Tel.: 1-713-743-4277. Fax: 1-713-743-4260. E-mail: khwang@uh.edu

Abstract: This paper presents an object-oriented two-dimensional (2-D) overland flow model and its application in simulating flood flows over Ulus basin, located in the north of Turkey adjacent to the Black Sea. A new coding implementation according to the class environment created in object oriented C++ programming language is carried out in structuring and building the solver. The model is based on the Godunov type finite volume scheme on unstructured triangular meshes. A mass balance preserving wet/dry boundary solution algorithm is integrated in the numerical scheme to satisfy the positive-depth condition and minimize the numerical instability when treating the propagation of wave front in regions of dry bed. The balance between bed slope and flux terms is also preserved for still water conditions on irregular topography. The 2-D solver is verified by simulating selected dam break cases, where good agreement with measured data is achieved. For the simulation of flood flows in the Ulus basin, in general, the simulated outflow hydrograph is found to compare well with the recorded data. A selected inundation map that is extracted from the model results is also presented to show the water surface level in the Floodplain.

Keywords: Flood inundation; Dam break; Finite volume; Wet/dry; Hydrograph; Object oriented programming.

INTRODUCTION

The movements of flood waves over watersheds or dam-break induced shock waves in channels are commonly modeled by solving the one-dimensional (1-D) continuity and momentum equations, which form the basis of various hydrological or hydraulic routing methods. However, the flood flows in an improved representation move along two horizontal directions. The extended two-dimensional (2-D) shallow water equations (SWE) can then be applied to model more reasonably the fluid flows over the land surface. Due to the non-linear character of the equations, analytical solutions are limited to only few idealized simple cases. Therefore, to accomplish the practical application, the approach was mostly based on solving the equations numerically.

The finite volume (FV) method can be applied to solve the integral form of the SWE on domains with either structured or unstructured grids. It is known that the FV method has the combined advantages of finite element (FE) method for geometric flexibility and finite difference (FD) method for simple discretization procedure. Zoppou and Roberts (2003) compared several explicit schemes for 1D dam break problems and concluded that the Riemann solvers are more accurate and robust than the finite difference schemes to handle the flow discontinuity.

Godunov (1959) presented an upwind type FV method to solve a Riemann problem at the interfaces of neighboring control volumes. Due to the difficulty of solving exactly the Riemann problem, use of the approximate Riemann solvers, like the one developed by Roe (1981), has been widely adopted. Zhao et al. (1996) mentioned that the 1st order accurate FV scheme cannot model the discontinuities accurately and advised use of limiters to obtain results with higher order accuracy. Anastasiou and Chan (1997) incorporated slope limiters with the Roe type Riemann solver to have a higher-order accurate scheme for the prevention of spurious oscillations.

In order to simulate flood flows through non-flat bottom, the source terms with bed slope are included in the SWE. For an

irregular topography, the computation of bed slope within cells becomes very important while solving SWE as it is a numerical concern to preserve the balance between flux and source terms for still water equilibrium conditions. LeVeque (2002) proposed a wave propagation method; however, that is only valid for quasi-steady cases and cannot be applied on unstructured meshes. Ying and Wang (2008) treated the pressure term as a whole without separating hydrostatic and bottom slope terms. Their approach adopted the assumption that the water surface level is constant and the gradient in surface level is zero at quiescent flow conditions. Other methods have also been employed to resolve this issue (Audusse and Bristeau, 2005; Bermudez and Vazquez, 1994; Bradford and Sanders, 2002; Brufau et al., 2004; Rogers et al., 2003; Valiani and Begnudelli, 2006).

The method using the upwind scheme to the source terms and the eigenvector projection has also been applied to solve the equilibrium condition problem (Bermudez and Vazquez, 1994; Brufau et al., 2004; Hubbard and Garcia-Navarro, 2000). This method allows the total slope terms to be calculated by summing the bed slope source terms at each edge in terms of the projected eigenvectors. This is done with the first-order accuracy. When extended to higher order schemes, e.g. second-order scheme (Cea et al., 2006; Hubbard and Garcia-Navarro, 2000), the higher order corrections should be included to satisfy the balanced condition. A hybrid scheme that uses first order in water depth and second order in discharge with the upwind forms of the bed slope source terms was applied by Cea et al. (2006). According to their findings, similar results to those from fully second-order scheme were obtained. While modeling the flow traveling into dry riverbeds or floodplains, the cells changing from dry to wet or vice versa may cause numerical errors in the velocity calculations. To prevent these numerical problems, a wet/dry tracking and treating algorithm or a so-called check program is needed in the flow solvers. Considering the processes of rainfall interception, evapotranspiration, moisture dynamics, and runoff routing, Ivanov et al. (2004) proposed a physics-based, distributed hydrological model to simulate overland

flows of mid- to large-sized watersheds. Kim et al. (2013) investigated rainfall and overland flow induced soil erosions at selected watersheds by solving the combined 2-D SWE and sediment bed evolution and transport equations.

Most of the numerical codes were developed using procedural type languages such as FORTRAN or C programming language (Malan and Lewis, 2004). However, the powerful and rich class libraries and object-oriented (OO) framework from C++ language may provide the coding advantage for numerical programs applied in irregular domains with large number of grid points. Also, a well-managed object-oriented framework with inheritance, polymorphism and automatic memory allocation was found to be able to improve the productivity and software flexibility according to Kettani and Ouazar (1994). Solutions of SWE on unstructured elements require the formation of the topology and spatial relationship between each cell. This task can be achieved by creating class environment involving data structure for the components of a mesh element such as nodes, lines, centers and the triangles. The object-oriented framework used in FE meshing is given in the studies by Mackie (1992) and Simphson (2003). Also, in those studies, it was stated that efficient implementation of object-oriented framework assures code reusability. In the meantime, the use of the C++ programming language in Microsoft Net environment with assigned object-oriented components, the development of effective and user friendly numerical codes can be achieved.

In this study, a new cell-centered FV overland flow model coded with an improved approach of using object-oriented C++ language and integrated wet/dry tracking algorithm is developed to simulate the movement of dam-break induced wave fronts and real time flood flows over a watershed. Unstructured triangular meshes are used to represent the spatial domain in the model. Under the object-oriented framework, one can increase the functionality of the program and easily overcome the programming difficulty caused by the unstructured triangular elements. The present model with defined topological relationships for elements and associated variables has the advantage of improving the effectiveness of the computational procedure. Both the second-order and hybrid type finite volume schemes are adopted for spatial integration while a second-order Runge-Kutta procedure is used for time integration. To limit the spurious oscillations in the solution, slope limiters are also coded in the solver. The verification of the model is performed by testing two different dam break cases. A practical application of the model is also carried out by simulating a flood event occurred in Ulus basin, Turkey. The data of irregular topography are available to define the bottom elevations for the numerical simulations. The computed outflow hydrograph from the present 2-D model is compared with the measured data for the flood event at the outlet of the Ulus basin. Flood maps showing the maximum extent of the water level are important for flood risk assessment. For the Ulus basin study, selected flood maps are generated from extracted model results using GIS software and presented in this paper.

GOVERNING EQUATIONS AND FINITE VOLUME FORMULATIONS

The usual 2-D shallow water equations in conservative form can be written as follows

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} = \mathbf{S} \quad (1)$$

where $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$, \mathbf{U} = vector of conserved variables, \mathbf{F} = vector of flux functions = (F_x, F_y) , and \mathbf{S} = vector of source terms. The variables of \mathbf{U} , F_x , F_y , and \mathbf{S} can be expressed as

$$\mathbf{U} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \quad \mathbf{F}_x = \begin{bmatrix} hu \\ hu^2 + 0.5gh^2 \\ huv \end{bmatrix}, \quad \mathbf{F}_y = \begin{bmatrix} hv \\ huv \\ hv^2 + 0.5gh^2 \end{bmatrix},$$

$$\mathbf{S} = \begin{bmatrix} 0 \\ gh(S_{0x} - S_{fx}) \\ gh(S_{0y} - S_{fy}) \end{bmatrix} \quad (2)$$

where u = vertically-averaged velocity in x direction, v = vertically-averaged velocity in y direction, h = water depth, g = gravitational acceleration, S_{0x} = bed slope in x direction, S_{0y} = bed slope in y direction, S_{fx} = friction slope in x direction = $\hat{n}^2 u \sqrt{u^2 + v^2} / h^{4/3}$ (in SI unit), S_{fy} = friction slope in y direction = $\hat{n}^2 v \sqrt{u^2 + v^2} / h^{4/3}$ (in SI unit), and \hat{n} = Manning's roughness coefficient. Integration of Eq. (1) over a triangular element i (see Fig. 1) yields

$$\frac{\partial \mathbf{U}_i}{\partial t} A_i + \oint_{B_i} \mathbf{F} \cdot \mathbf{n} dl = \oint_{A_i} \mathbf{S} dA \quad (3)$$

where i is the index of a triangular cell that is calculated, A_i is the area of the cell i , and B_i is the boundary of cell i .

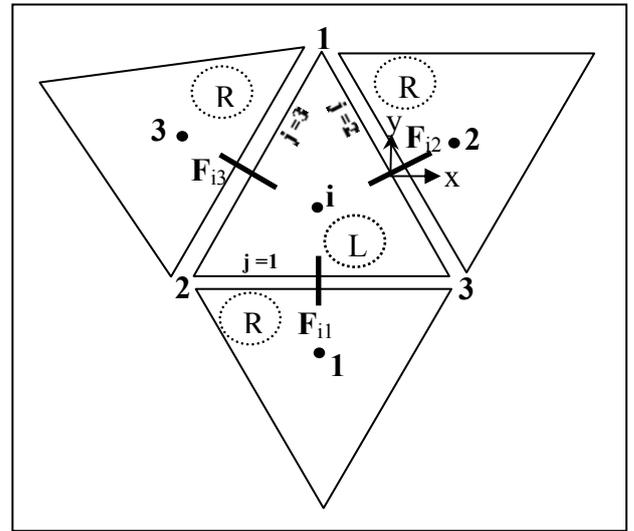


Fig. 1. Topology of a triangular cell with the neighboring cells.

Time scheme

Applying the second-order Runge-Kutta scheme for time marching, Eq. (3) can be solved by the following formulations

$$\mathbf{U}_i^* = \mathbf{U}_i^k + \frac{0.5\Delta t}{A_i} \left(-\sum_{j=1}^3 (\mathbf{F}_{ij}^k \cdot \mathbf{n}_{ij}) \Delta l_{ij} + \mathbf{S}_{b_i}^k + A_i \mathbf{S}_{f_i}^k \right) \quad (4)$$

$$\mathbf{U}_i^{k+1} = \mathbf{U}_i^k + \frac{\Delta t}{A_i} \left(-\sum_{j=1}^3 (\mathbf{F}_{ij}^* \cdot \mathbf{n}_{ij}) \Delta l_{ij} + \mathbf{S}_{b_i}^* + A_i \mathbf{S}_{f_i}^* \right) \quad (5)$$

where \mathbf{U}_i = cell averaged values at the center of element i ,

Δt = time step, superscript k , $*$, and $k+1$ represent the current time level, the intermediate time level, and the next time level, respectively, and $\mathbf{S}_f = [0, -g\bar{h}S_{fx}, -g\bar{h}S_{fy}]^T$. It should be noted that the above described Runge-Kutta scheme is an explicit scheme. Therefore, the well-known Courant-Fredrichs-Lewy (CFL) criterion is used to choose the time step.

Space scheme

For the terms related to the spatial variations in (4) and (5), especially the boundary integral terms, the up to second-order scheme is described in this section. Adopting the formulations of Roe type approximate Riemann solver (Roe, 1981) the boundary integral term $\oint_{B_i} \mathbf{F} \cdot \mathbf{n} dl$ can be determined as

$$\begin{aligned} \oint_{B_i} \mathbf{F} \cdot \mathbf{n} dl &= \sum_{j=1}^3 (\mathbf{F}_{ij} \cdot \mathbf{n}_{ij}) \Delta l_{ij} \\ &= \sum_{j=1}^3 \frac{1}{2} \left[\mathbf{F}_{ij}(U_R) \cdot \mathbf{n}_{ij} + \mathbf{F}_{ij}(U_L) \cdot \mathbf{n}_{ij} - \sum_{m=1}^3 (\alpha_{ij})_m |(\bar{\lambda}_{ij})_m| |(\bar{e}_{ij})_m| \right] \Delta l_{ij} \end{aligned} \quad (6)$$

where j is the index for the neighboring cells; \mathbf{n}_{ij} and Δl_{ij} ($j = 1, 2, 3$) are respectively the outward unit normal vectors and the edge length of cell i for edge j . U_L and U_R are respectively the vectors of physical variables constructed at the left and right sides of the edge. As shown in Fig. 1, the i^{th} cell with variables calculated is called the Left cell while the neighboring cells are called the Right cells for that cell. $(\bar{\lambda}_{ij})_m$ and $(\bar{e}_{ij})_m$ ($m = 1, 2, 3$) are respectively the eigenvalues and eigenvectors of the Jacobian matrix after applying the Roe's average method. $(\alpha_{ij})_m$ ($m = 1, 2, 3$) given in Brufau et al. (2004) are the coefficients calculated at the j^{th} edge for cell i .

The average eigenvalues for cell i , representing the characteristic wave speeds, can be expressed as

$$\bar{\lambda}_{i1} = \bar{u} n_x + \bar{v} n_y + \bar{c} \quad (7)$$

$$\bar{\lambda}_{i2} = \bar{u} n_x + \bar{v} n_y \quad (8)$$

$$\bar{\lambda}_{i3} = \bar{u} n_x + \bar{v} n_y - \bar{c} \quad (9)$$

where \bar{u} , \bar{v} , and \bar{c} are the Roe's (Roe, 1981) averaged values evaluated at the edge between cell i and a neighboring cell. For the computation of the bed slope term, \mathbf{S}_b , in Eqs. (4) and (5), this study utilizes the source term discretization method proposed by Brufau et al. (2004), which gives

$$\mathbf{S}_{b_i} = \int_{A_i} \mathbf{S}_0 dA = \sum_{j=1}^3 S_{0_{ij}} \Delta l_{ij} \quad (10)$$

where

$$S_{ij} = \sum_{m=1}^3 (\beta_{ij})_m (\bar{e}_{ij})_m \quad (11)$$

$$(\beta_{ij})_{1,3} = \pm \left\{ \frac{1}{4\bar{c}} \left(1 - \text{sign}(\bar{\lambda}_{1,3}) \right) [S_{0_2} n_x + S_{0_3} n_y] l_c \right\}_{ij} \quad (12a)$$

$$(\beta_{ij})_2 = \left\{ \frac{1}{2\bar{c}} \left(1 - \text{sign}(\bar{\lambda}_2) \right) [-S_{0_2} n_y + S_{0_3} n_x] l_c \right\}_{ij} \quad (12b)$$

and

$$S_0 = [S_{0_1}, S_{0_2}, S_{0_3}]^T = [0, -g\bar{h}\Delta z_x, -g\bar{h}\Delta z_y]^T \quad (13)$$

In Eqs. (12) and (13), $n_{x_{ij}}$ and $n_{y_{ij}}$ are the outward unit normal vectors for the element i and the j^{th} edge along x and y directions, respectively (Fig. 1), Δz_x and Δz_y are defined for each edge as $\Delta z_x = [z_R - z_L] n_x / l_c$ and $\Delta z_y = [z_R - z_L] n_y / l_c$, and l_c is the distance between the centers of the left cell (i) and right cell ($j = 1, 2, 3$). Here, z_L and z_R are the bottom elevation values at the Left and Right cell centers.

To have more accurate solutions, the second-order schemes can be utilized. Different from the first-order scheme using directly the values at cell centers, the conserved variables are first reconstructed at the left and right sides of cell edges using the cell gradient method. For example, the variables at the left or right side of the edge can be computed following

$$\mathbf{U}_{edge(L \text{ or } R)} = \mathbf{U}_{c(L \text{ or } R)} + \phi \nabla \mathbf{U}_{c(L \text{ or } R)} \cdot \mathbf{r} \quad (14)$$

where $\mathbf{U}_{c(L \text{ or } R)}$ is the matrix of variables at the left or right cell (i or j) center, \mathbf{r} is the distance vector from left (or right) cell center to the midpoint of the edge of the left (or right) cell, $\nabla \mathbf{U}$ is the gradient of variable vector \mathbf{U} , and ϕ is the slope limiter. Without including the effect of slope limiter (e.g. let $= 0$), Eq. (14) is recovered as a first-order accurate reconstruction. According to Pan and Cheng (1993), $\nabla \mathbf{U} = (1/A_\Omega) \oint_{d\Omega} \mathbf{U} \cdot \mathbf{n}_\Omega dl_\Omega$, where A_Ω is the area of a triangle (Ω) that is formed by connecting the three neighboring centers, $d\Omega$ is the boundary of triangle (Ω), and \mathbf{n}_Ω and dl_Ω are respectively the outward unit normal vector and length of any of the edges of triangle (Ω). Those reconstructed values from Eq. (14) are then input to compute \bar{u} , \bar{v} , and \bar{c} using Roe average formula for the calculation of the flux functions at the cell faces.

When modeling flows, reconstructed variables at the cell edges using the second-order scheme have been suggested to be limited with slope type limiters to preserve monotonicity and limit oscillations. Here, the limiter as proposed by Barth and Jespersen (1989) is incorporated in the model development. The approach using the first-order scheme in water depth and the second-order scheme in discharges (hu and hv terms) is named as a hybrid approach. In this study, full second-order and hybrid FV computations are applied to the dam-break problems. Results from selected cases are compared. The hybrid approach, considering the accuracy of the numerical method and computational efficiency, however, is used for simulating flood flows over Ulus basin in Turkey.

WET-DRY ANALYSIS

Numerical modeling of flow over regions with dry lands or small depths may create high velocities and mass errors. Wet/dry boundaries in the computational meshes should be treated carefully to avoid numerical errors and instabilities. Propagation of flow at a wet/dry interface with a downward gradient can be handled straightforwardly by setting a tolerance value for water depth and solving it as the Riemann problem. However, flooding and drying process over an adverse slope can cause spontaneous movement of water to adjacent dry cells and damage the balance between flux and source terms (Brufau et al., 2002, 2004; Cea et al., 2006).

A tolerance constraint (10^{-4} m) is defined for the water depth in this study to separate the wet cells from dry cells. The tolerance

value is sensitive on surfaces with friction, as in the Manning equation the friction slope is obtained by dividing terms as functions of water depth. The following four checking procedures similar to those given by Brufau et al. (2004) are applied at wet/dry interfaces for Roe scheme with upwind slope terms.

(1) Calculate h_L and h_R at the edge.

(2) If h_L or h_R at an edge is smaller than the tolerance value, 10^{-4} m, then that edge is assumed as dry with u_L or u_R and v_L or v_R values being set to equal to zero at *left* or *right* cell. The values of \bar{h} , \bar{u} , and \bar{v} at the edges are estimated by using the formulations given below.

$$\text{If } h_R \geq 10^{-4} \text{ m and } h_L < 10^{-4} \text{ m, then} \\ \bar{u} = u_R, \bar{v} = v_R, \bar{h} = 0.5(h_L + h_R) \quad (15)$$

$$\text{If } h_R < 10^{-4} \text{ m and } h_L \geq 10^{-4} \text{ m, then} \\ \bar{u} = u_L, \bar{v} = v_L, \bar{h} = 0.5(h_L + h_R) \quad (16)$$

(3) If two cells sharing the same edge are both dry, the equations are not solved for that edge.

(4) The bed elevation has to be redefined to preserve the flux and source term balance at wet/dry interfaces (Brufau et al., 2004). If the cell (A) indicated in Fig. 2 is a left and wet cell and the cell (B) is a right and dry cell, the unit discharge at that edge is assumed to be zero and Δz_{ij} between these cells can be calculated using the following equation to preserve the balance with the flux term by assuming stationary flow condition.

$$\text{If } z_L + h_L < z_R, \text{ then} \\ q_{edge} = 0 \text{ and } \Delta z_{ij} = h_L - h_R, \quad (17)$$

where q_{edge} is the unit discharge at the edge and Δz_{ij} = the bed elevation change between right and left cells.

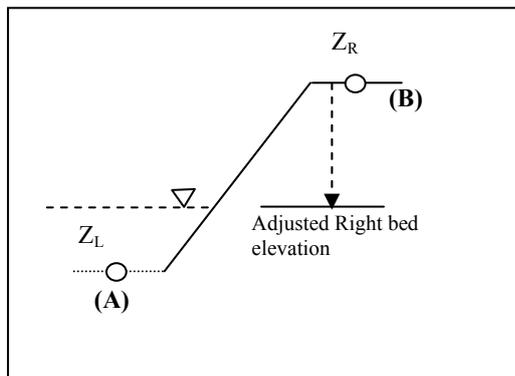


Fig. 2. Wet/dry boundary check for adverse slope condition.

DATA STRUCTURE

Although the unstructured meshes can best represent the geometrically complicated domain, coding the 2-D overland flow model using unstructured elements can be rather challenging as the cells are not following a defined pattern. With the integration of the object-oriented framework, one can increase the functionality of the program. In addition, using the geometric elements of the mesh as objects can overcome the programming difficulty. Forming a topology with element mapping in a pre-processing level provides fast, convenient and user-friendly way to complete the numerical code.

Data structure library and capability of C++ is well-known among the researchers. In this study, the computational domain is divided into triangular elements. The object-oriented programming, which uses the geometric elements of the mesh as objects, is utilized to integrate the spatial information with the code and solve the SWE between an element and its surrounding three neighbors to calculate the conserved variables at the midpoints of the edges, three corner points and centers of the cells.

The classes created in C++ are the triangle, center, line and point. Each triangle is composed of three corner nodes, three lines (edges) and three neighboring triangles. The nodes are numbered in counterclockwise direction as 1, 2 and 3. The edges and neighboring triangles are numbered with the same index as the node opposite to them (Fig. 1). A node can be a common element of numerous triangle cells while line segment is an element of two neighboring triangles at the interior regions and one triangle at the boundaries. By indexing the elements of the unstructured cells, the topological relationship is created and managed for each cell entity and any component easily.

The SWE are solved to compute the conserved variables water depth (h) and velocities in x and y directions (u and v), respectively, either at the nodes or cell centers. To access the variables stored at any entity inside the domain, pointers which are called handles are defined to be used with the topological relationship. Each class has a handle on the other classes so that the other classes' methods and objects can also be used by that class.

In Fig. 3, the dashed arrows represent the class that has a handle and arrows are pointing towards the class that the handle works on. The two sided arrows are between classes that have handle on each other. Triangle class has a handle on point class to know the three corner points and use their methods and attributes while point has a handle on triangle class to identify the triangles that are surrounding that point and reach objects of triangle class. Triangle class also holds a pointer to line class to access the three edges. To access the three neighboring triangles, pointer is used where each of the array elements represent one of the three neighbors. Line class uses a handle to point class to locate the endpoints and use the coordinates of them to build the vectors. Line class also holds a pointer to triangle class to identify the neighboring triangles to the left and right of an edge.

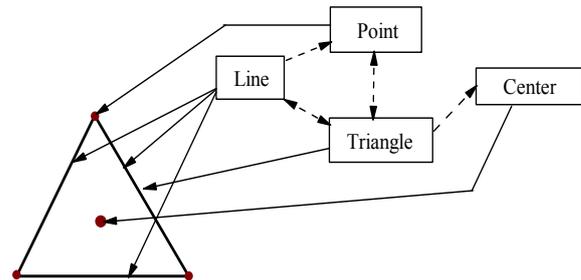


Fig. 3. Classes and their relationship in the data structure model.

BOUNDARY CONDITIONS

Boundary cells are the cells at the interior side of the boundary domain and have at least one of their edges on the boundary. The boundary conditions are assigned at the ghost cells created as neighbors to the boundary cells. In this study, two different boundary conditions are defined, closed boundary (wall boundary) and open boundary at which depth or discharge could be imposed.

At wall boundary condition, the water depths (h) at the ghost cells are kept the same while the velocities (u and v) are taken as the negative values of those at boundary cells to ensure zero velocities on the wall surface. The open boundary at the outlet regions of all test cases are used as transmissive boundaries in this study. It is important that the waves leaving the domain are not reflected back. The third type is the inlet boundary which is used for the flood simulation in Ulus basin, Turkey. Riemann invariants and characteristic curves are used to compute the corresponding values of the velocity and water depth at the inlet cells.

EXAMPLE CALCULATIONS AND MODEL TESTING

Dam-break wave propagating on downward sloping bottom with dry bed

To examine the performance the developed 2-D overland flow model and its capability of handling flows on sloping surface with wet and dry interface, a case related to Bellos et al. (1992) dam break experiments is simulated. The experimental domain with an asymmetric setup of converging-diverging channel and gauging stations are shown in Fig. 4. A dam face is located at $x = 0$ m. A series of dam-break conditions were tested during the experiments (Bellos et al., 1992). The case with an upstream water depth of 0.3 m and downstream river bed dry was selected to be simulated in this study. The downward bottom slope is 0.006 and the Manning's roughness coefficient $n = 0.012$. The domain was discretized into 10,500 triangular elements.

The time variation of water depth computed at gauging stations D1 ($x = -8.5$ m), D2 ($x = -4$ m), and D3 ($x = 2.5$ m) are compared with the experimental observations (Bellos et al., 1992) in Figs. 5(a)–5(c). The results with the second-order scheme and hybrid scheme are presented for comparison. Bellos et al. (1991) simulated this case with their finite difference (FD) numerical model. Their numerical results at station D3 are also shown in Fig. 5(c) to compare to the present model solutions and experimental data. As indicated in Figs. 5(a) to 5(c), the results obtained from the present model agree well with the observed data. The model reproduces the experimental observations with high accuracy at stations D1 and D2. The comparisons of water depth at station D3 (Fig. 5(c)) also reveal that the present model improves the predictions on the peak of the dam-break wave and time to peak when comparing to those obtained from the FD numerical model. It should be noted that both the hybrid and second-order schemes perform equally well in predicting the water depth. The difference is nearly indistinguishable. The factor of converging-diverging channel geometry tends to smooth out the propagation of downstream wave front and the upstream depression wave and results in the better predictions provided by the present FV models.

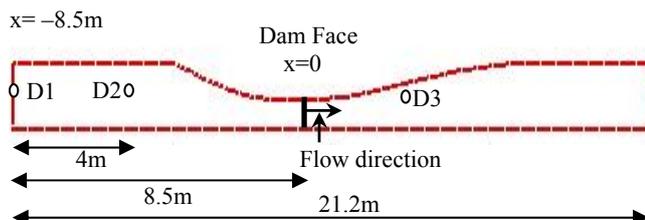


Fig. 4. Geometry of the experimental setup for dam-break wave propagation in a converging-diverging channel given in Bellos et al. (1992).

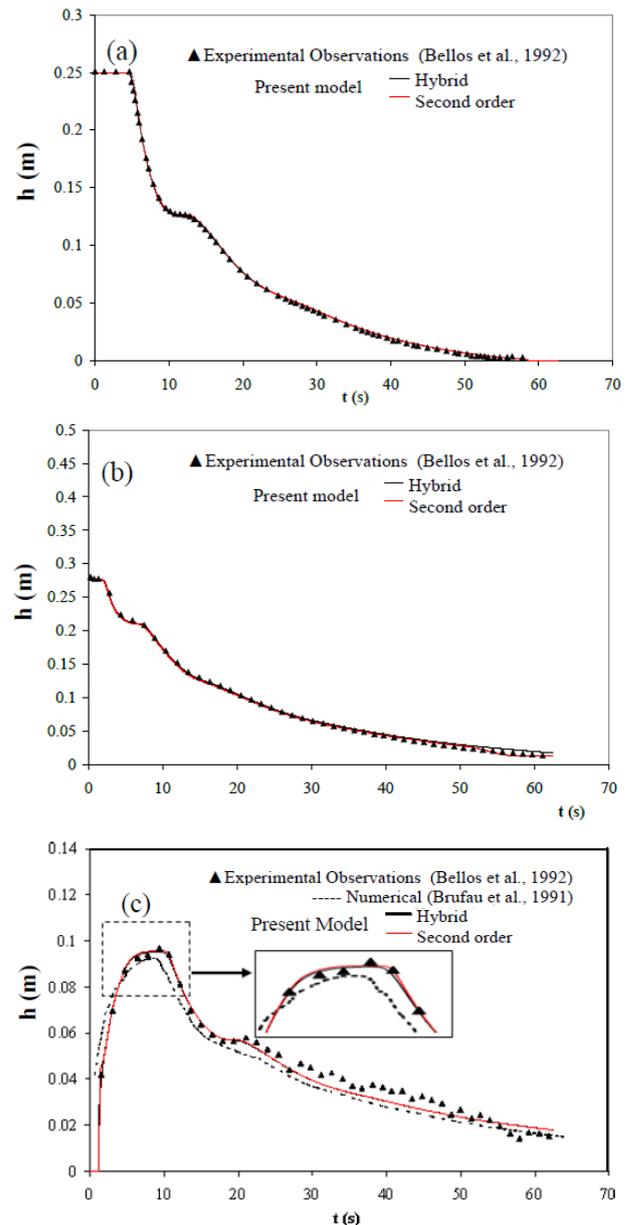


Fig. 5. Comparisons of simulated water depth from the present models (Hybrid and Second-order schemes) with measured data given in Bellos et al. (1992) at gauging stations: (a) D1, (b) D2, and (c) D3, for dam-break wave propagating in a converging-diverging channel.

Wave propagating onto a triangular obstacle

In this test case, the experiments performed by the J.M. Hivier at the Laboratoire de Recherches Hydrauliques, Université de Bruxelles, was simulated to demonstrate the present solver's capability of preserving stability when modeling wetting and drying on adverse sloping bottom. The sketch of the problem statement is shown in Fig. 6 (Brufau et al., 2002). The locations of gauging station with G4 at $x = 4$ m, G10 at 10 m, G11 at 11 m, G13 at 13 m, and G20 at $x = 20$ m are also illustrated in Fig. 6. The length of the entire domain is 38 m and the dam face is located at $x = 15.5$ m. The triangular obstacle (6-m long and 0.4-m high) is situated at $x = 28.5$ m. The initial water depth is $h_0 = 0.75$ m in the reservoir and dry bed is assumed at the downstream of the dam. Manning roughness coefficient is 0.0125 for the bed and 0.011 for the walls.

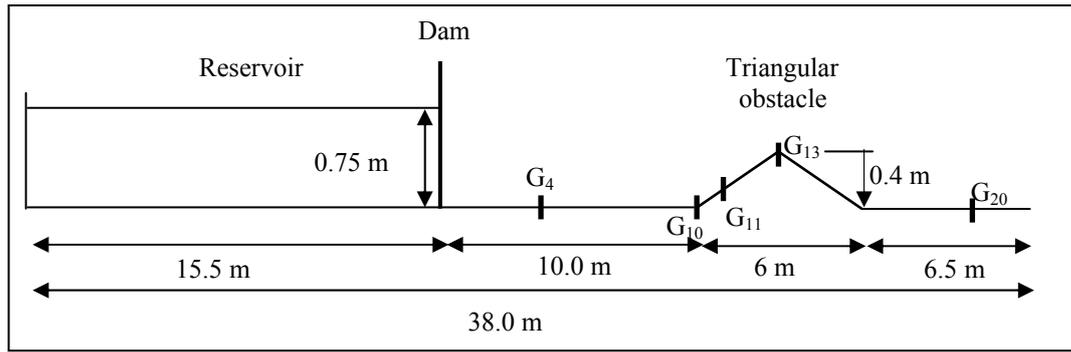


Fig. 6. Sketch of the domain and gauging stations for experimental study as described by Brufau et al. (2002) of dam-break flows over a triangular obstacle.

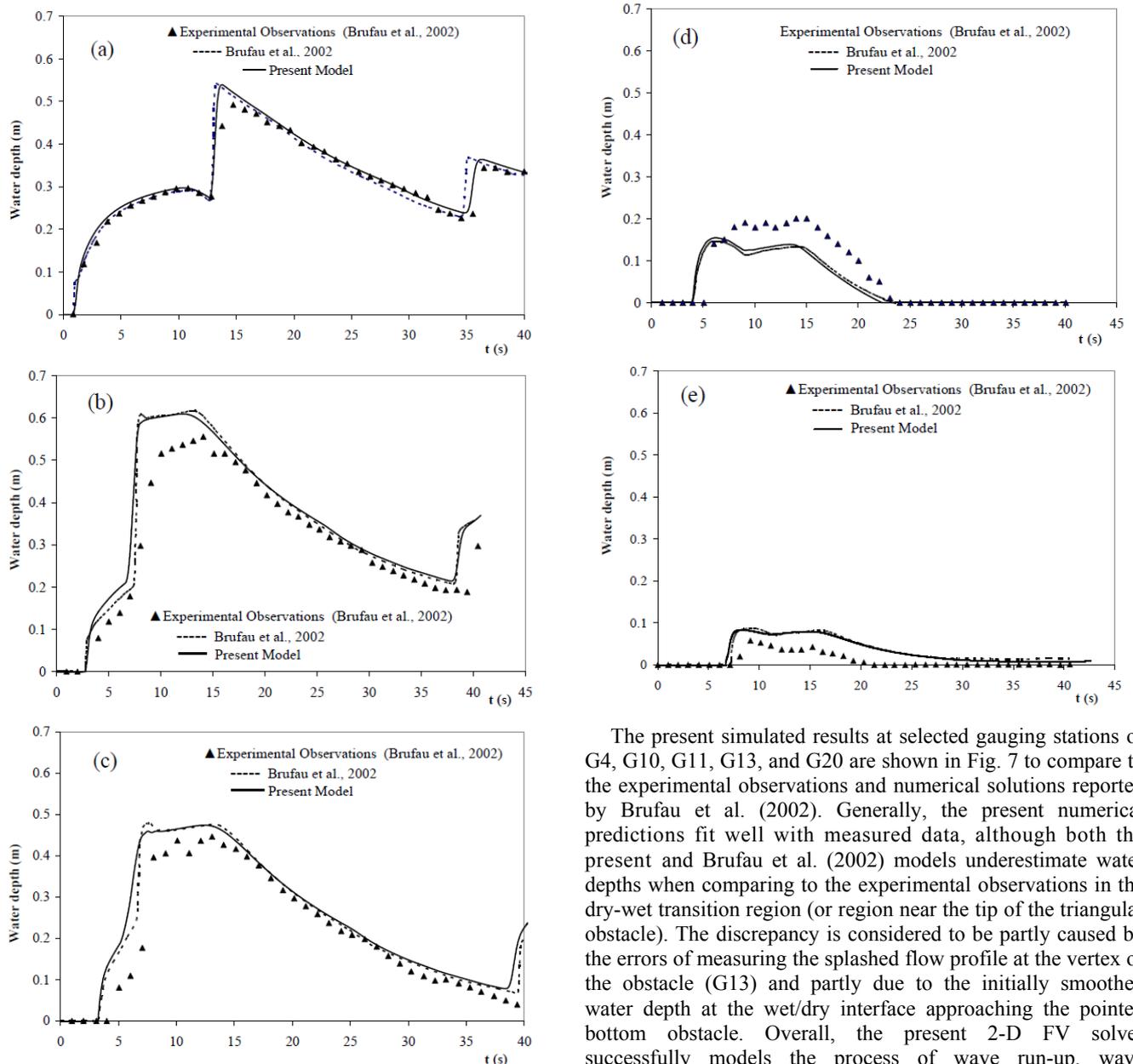


Fig. 7. Comparisons of present simulated water depth with Brufau et al. (2002) numerical results and reported experimental data at gauging stations: (a) G4, (b) G10, (c) G11, (d) G13, and (e) G20.

The present simulated results at selected gauging stations of G4, G10, G11, G13, and G20 are shown in Fig. 7 to compare to the experimental observations and numerical solutions reported by Brufau et al. (2002). Generally, the present numerical predictions fit well with measured data, although both the present and Brufau et al. (2002) models underestimate water depths when comparing to the experimental observations in the dry-wet transition region (or region near the tip of the triangular obstacle). The discrepancy is considered to be partly caused by the errors of measuring the splashed flow profile at the vertex of the obstacle (G13) and partly due to the initially smoothed water depth at the wet/dry interface approaching the pointed bottom obstacle. Overall, the present 2-D FV solver successfully models the process of wave run-up, wave overtopping, and formation of reflected wave without any deformation and instability.

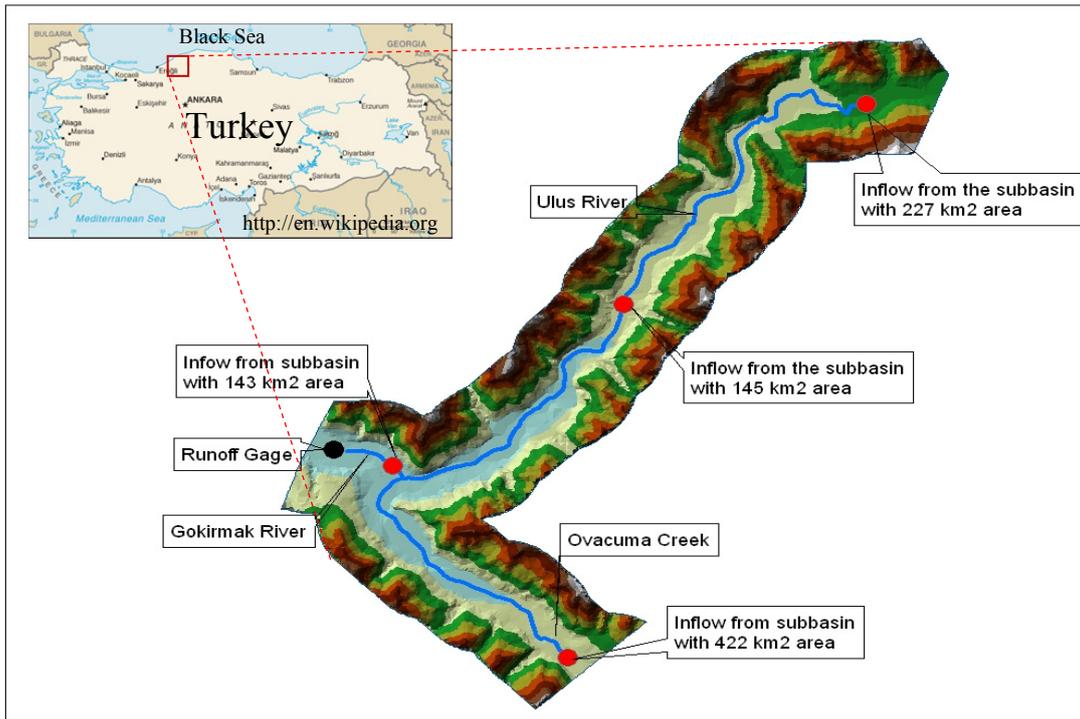


Fig. 8. Geographic location of Ulus basin, its river network, and the locations of inflow hydrographs inputted for flood flow simulation.

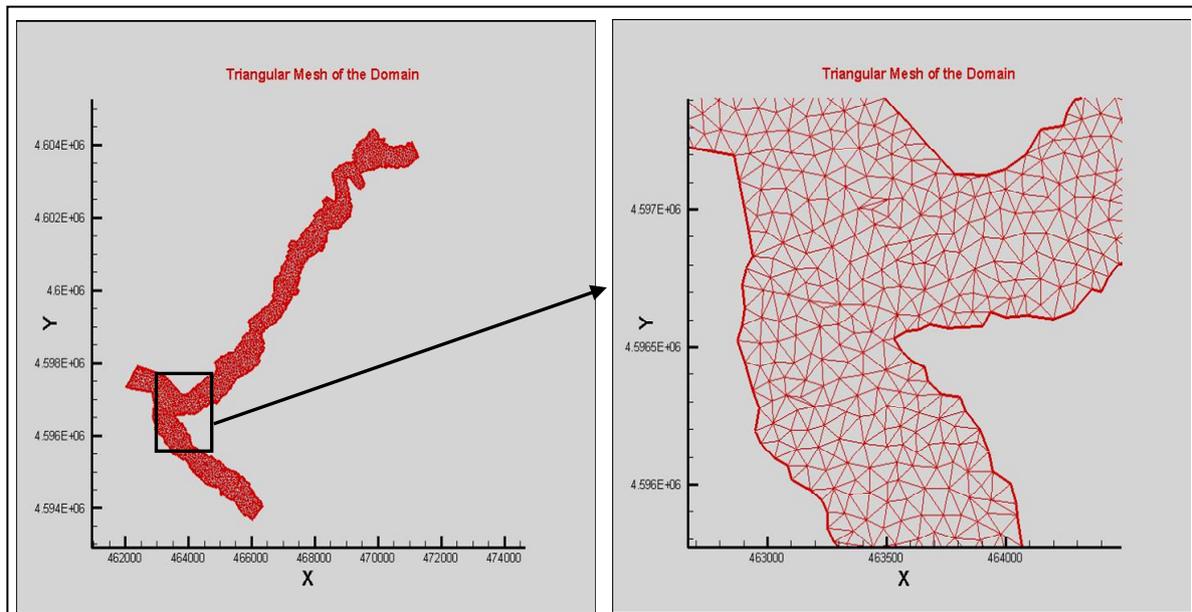


Fig. 9. Triangular mesh system of the rivers and floodplain located on Ulus Basin.

Flood flow modeling on Ulus basin, Turkey

To extend the model's capability of simulating flows on a more complicated topography and catching wet/dry front with conserving mass, the present 2-D overland flow model was applied to a case study of watershed flood flow modeling. The study is to simulate a 1991 flood event on Ulus basin located in the north of Turkey adjacent to the Black Sea (Fig. 8), where floods have been the major threat to the region. The Ulus basin is a sub-basin of Bartın basin, which receives relatively high-

er rainfall amount than the mean rainfall of Turkey (Usul and Turan, 2006). Two major rivers, namely Ulus River and Ovacuma River, flow through the basin and join together to form the Gokirmak River (Fig. 8). The region has some steep sloped mountains and scattered agricultural lands that narrow the riverbed and the flood plains. The flow modeling domain of the Ulus basin as indicated in Fig. 8 includes the inflow and outflow sections and the downstream gauge location. The Manning's roughness coefficients were obtained through the calibration procedure. The values for the river beds of Ulus and

Ovacuma Rivers are respectively 0.03 and 0.04 and for the floodplain the values are 0.04 and 0.065. The recorded outflow hydrograph is available for comparison with computed flow rates at the outflow location.

The hydrographs from the 227 km² upstream subbasin of the Ulus River and 422 km² of the Ovacuma River can be found in Usul and Turan (2006) for the upstream inflow boundary conditions. In addition, inputs of side flow hydrographs (entering to the Ulus and Gokirmak rivers) for the subbasins with the areas 145km² and 143 km² are also included in the simulation. The peak flow of the upstream hydrograph entering the Ulus River is 152 m³/s whereas it is 265 m³/s for the inflows entering the Ovacuma River. The peak flows for the hydrographs entering as the side flows to the Ulus and Gokirmak Rivers are 92 m³/s and 90 m³/s, respectively. As shown in Fig. 9, the computational domain covering the two-river system is divided into 10,000 triangular cells. The bottom elevations at the cell centers and corner points were interpolated from topographic data in a Geographic Information System (GIS).

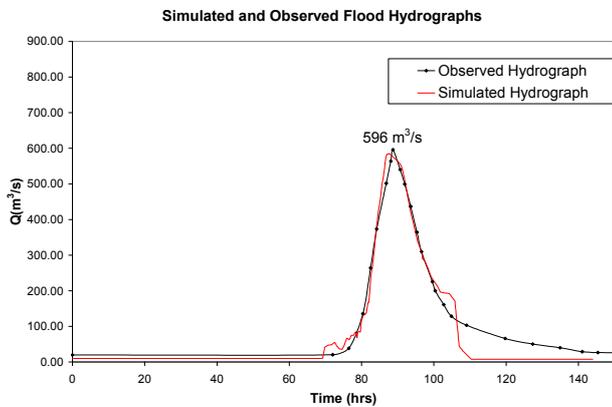


Fig. 10. Comparisons of simulated and observed flood hydrographs at the outlet of Ulus Basin for 1991 flood event.

The present solver was run for the whole flood duration which is 144 hours (6 days) from July 4, 1991 to July 10, 1991. The comparisons between simulated and observed flood hydrographs at the outlet of the basin are presented in Fig. 10. Simulated peak discharge is 585 m³/s whereas the observed one is 596 m³/s. As seen from Fig. 10, reasonable match between the simulated and observed hydrographs can be noticed. Defining the L_1 -norm error as

$$\|L_1\| = \sqrt{\sum_{i=1}^N |\alpha_{num,i} - \alpha_{exp,i}| / \sum_{i=1}^N |\alpha_{exp,i}|},$$

where α_{num} is a value of physical variable generated from numerical simulation, α_{exp} is a measured value of the physical variable, and N represents the total number of data points. For the error analysis of the two most important characteristics of the flow hydrograph; peak flow rate and time to peak, the L_1 -norm errors of the predicted values equal to 0.018 and 0.012, respectively.

The time variation of the rising and decreasing stage of the simulated flow rates is similar to those indicated by the observed hydrograph, while at the later stage of the flow recession the simulated outflows tend to decrease at a fast rate than those of the observed data. This is a result potentially caused by the

concentrated side inflows entering the river system and small amount of groundwater flows that gradually enter the outlet at the time close to the end of the storm event, while this effect is not included in the model simulation. With the computed time variation of water depth in the floodplain, 3-D perspective view plots of the free-surface elevation and inundation maps can be generated. As an example, an inundation map at $t = 94$ hour for this flood event is extracted using GIS software and shown on TIN (Triangulated Irregular Network), an elevation model, in Fig. 11. The present model can serve as an engineering tool to combine the computed flow depths with velocity vectors into flood risk maps, especially considering the events of different return periods, for detailed flood analysis and enhancement of flood warning.

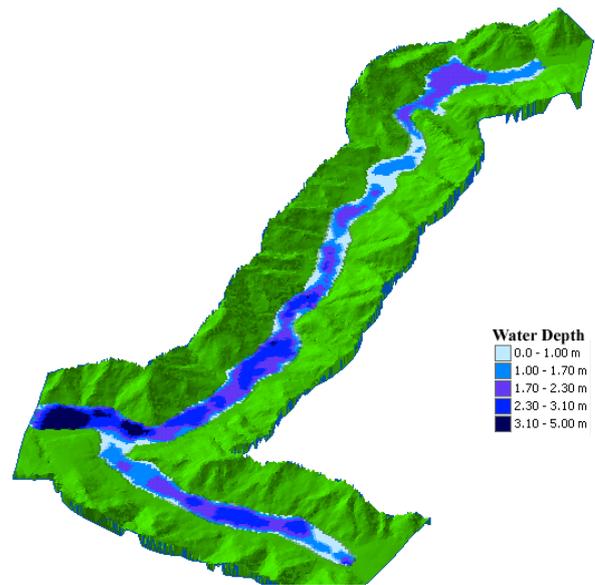


Fig. 11. Example plot showing an inundation map on TIN for 1991 flood event occurred on Ulus Basin at $t = 94$ hour.

CONCLUSIONS

A new coding implementation for the development of a cell-centered 2D flow solver with hybrid finite volume scheme on unstructured meshes is presented in this paper. Detailed description on self-structured class environment is provided to show the methodology of constructing a flow model. The developed solver with the use of triangular meshes and inclusion of wet/dry and shock wave capturing algorithm is capable of modeling complicated flow problems on domains with regular or irregular topography. Certainly, the present model is limited to the horizontally dominated two-dimensional flow problems. Also, the effects of vertical flow component and turbulence are considered to be negligible. In this study, selected test cases are simulated to verify and validate the developed flow model. The solver achieves good agreement with measured data for the cases of modeling propagation of dam-break waves on dry bed or over a triangular obstacle. The stability and convergence of the hybrid scheme are found to be satisfactory. A unique set of geographical and hydrological data is integrated in the solver to model a flood event which occurred in 1991 in Ulus basin located in the north of Turkey. The predicted outflow hydrograph, in general, agrees well with recorded data. Moreover, flood maps were extracted using GIS software to show the water surface level in the floodplain.

REFERENCES

- Anastasiou, K., Chan, C.T., 1997. Solution of the 2D shallow water equations using the finite volume method on unstructured triangular meshes. *Int. J. Numer. Meth. Fluids*, 24, 1225–1245.
- Audusse, E., Bristeau, M.O., 2005. A well-balanced positivity preserving ‘second-order’ scheme for shallow water flows on unstructured meshes. *J. Comp. Phys.*, 206, 1, 311–333.
- Bellos, C.V., Soulis, J.V., Sakkas, J.G., 1991. Computation of two-dimensional dam break induced flows. *Adv. Water Resour.*, 14, 1, 31–41.
- Barth, T.J., Jespersen, D.C., 1989. The design and application of upwind schemes on unstructured meshes. The 27th Aerospace Science Meeting, Reno, Nevada, AIAA-89-0366.
- Bellos, C.V., Soulis, J.V., Sakkas, J.G., 1992. Experimental investigation of two-dimensional dam-break induced flows. *J. Hydr. Res.*, 30, 1, 47–63.
- Bermudez, A., Vazquez, M.E., 1994. Upwind methods for hyperbolic conservation laws with source terms. *Computers and Fluids*, 23, 1049–1071.
- Bradford, S.F., Sanders, B.F., 2002. Finite-volume model for shallow-water flooding of arbitrary topography. *J. Hydr. Eng.*, 128, 3, 289–298.
- Brufau, P., Vazquez-Cendon, M.E., Garcia-Navarro, P., 2002. A numerical model for the flooding and drying of irregular domains. *Int. J. Numer. Meth. Fluids*, 39, 247–275.
- Brufau, P., Garcia-Navarro, P., Vazquez-Cendon, M.E., 2004. Zero mass error using unsteady wetting-drying conditions in shallow flows over dry irregular topography. *Int. J. Numer. Meth. Fluids*, 45, 1047–1082.
- Cea, L., French, J.R., Vazquez-Cendon, M.E., 2006. Numerical modeling of tidal flows in complex estuaries including turbulence: an unstructured finite volume solver and experimental validation. *Int. J. Numer. Meth. Fluids*, 67, 1909–1932.
- Godunov, S., 1959. A divergence scheme for numerical computation of discontinuous solution of hydrodynamic equations. *Math. Sbornik*, 43, 271–306.
- Hubbard, M.F., Garcia-Navarro, P., 2000. Flux difference splitting and the balancing of source terms and flux gradients. *J. Comp. Phys.*, 165, 89–125.
- Ivanov, V.Y., Vivoni, E.R., Bras, R.L., Entekhabi, D., 2004. Catchment hydrologic response with a fully distributed triangulated irregular network model. *Water Resour. Res.*, 40, W11102, 1–23.
- Kettani, E.E.-C.B., Ouazar, D., 1994. Object-oriented finite volume dam-break model. In: *Proceedings of Modeling of Flood Propagation over Initially Dry Areas*, Milan, Italy, pp. 186–196.
- Kim, J., Ivanov, V.Y., Katopodes, N.D., 2013. Modeling erosion and sedimentation coupled with hydrological and overland flow processes at the watershed scale. *Water Resour. Res.*, 49, 5134–5154.
- LeVeque, R.J., 2002. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, Cambridge.
- Mackie, R., 1992. Object-oriented programming of the finite element method. *Int. J. Numer. Meth. Fluids*, 35, 2, 425–436.
- Malan, A.G., Lewis, R.W., 2004. On the development of high-performance C++ object-oriented code with application to an explicit edge-based fluid dynamics scheme. *Computers and Fluids*, 33, 1291–1304.
- Pan, D., Cheng, J.C., 1993. Upwind finite-volume Navier-Stokes computations on unstructured triangular meshes. *AIAA J.*, 31, 9, 1618–1625.
- Roe, P.L., 1981. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comp. Phys.*, 43, 357–372.
- Rogers, B.D., Borthwick, A.G.L., Taylor, P.H., 2003. Mathematical balancing of flux gradient and source terms prior to use Roe’s approximate Riemann solver. *J. Comp. Phys.*, 192, 422–451.
- Simpson, B.R., 2003. C++ Classes for 2-D Unstructured Mesh Programming. INRIA Scientific Report, No. 3592, ISSN 0249-6399. <ftp://ftp.inria.fr/INRIA/publication/public-pdf/RR/RR-3592.pdf>
- Uslu, N., Turan, B., 2006. Flood Analysis within the Ulus Basin, Turkey, using geographic information systems. *Natural Hazards*, 39, 2, 213–229.
- Valiani, A., Begnudelli, L., 2006. Divergence form for bed slope source term (DFB) in shallow water equations. *J. Hydr. Eng.*, 132, 7, 652–665.
- Ying, X., Wang, S.S.Y., 2008. Improved implementation of the HLL approximate Riemann solver for one-dimensional open channel flows. *J. Hydr. Res.*, 46, 1, 21–34.
- Zhao, D.H., Shen, H.W., Lai, J.S., Tabios, G.Q., 1996. Approximate Riemann solvers in FVM for 2D hydraulic shock wave modeling. *J. Hydr. Eng.*, 122, 12, 692–702.
- Zoppou, C., Roberts, S., 2003. Explicit schemes for dam-break simulations. *J. Hydr. Eng.*, 129, 1, 11–34.

Received 14 February 2014

Accepted 26 June 2014