INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2005; **48**:1149–1174
Published online 28 April 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.974

An upstream flux-splitting finite-volume scheme for 2D shallow water equations

Jihn-Sung Lai[1], Gwo-Fong Lin[2,*,†] and Wen-Dar Guo[1]

[1] *Hydrotech Research Institute, National Taiwan University, Taipei 10617, Taiwan*
[2] *Department of Civil Engineering, National Taiwan University, Taipei 10617, Taiwan*

SUMMARY

An upstream flux-splitting finite-volume (UFF) scheme is proposed for the solutions of the 2D shallow water equations. In the framework of the finite-volume method, the artificially upstream flux vector splitting method is employed to establish the numerical flux function for the local Riemann problem. Based on this algorithm, an UFF scheme without Jacobian matrix operation is developed. The proposed scheme satisfying entropy condition is extended to be second-order-accurate using the MUSCL approach. The proposed UFF scheme and its second-order extension are verified through the simulations of four shallow water problems, including the 1D idealized dam breaking, the oblique hydraulic jump, the circular dam breaking, and the dam-break experiment with 45° bend channel. Meanwhile, the numerical performance of the UFF scheme is compared with those of three well-known upwind schemes, namely the Osher, Roe, and HLL schemes. It is demonstrated that the proposed scheme performs remarkably well for shallow water flows. The simulated results also show that the UFF scheme has superior overall numerical performances among the schemes tested. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: shallow water equations; finite-volume method; artificially upstream flux vector splitting method; Riemann problem

1. INTRODUCTION

The two-dimensional (2D) shallow water equations (SWE) are a system of hyperbolic conservation laws. The numerical schemes for solving 2D SWE require special considerations for achieving conservative and shock-capturing properties. Many shock-capturing schemes for hyperbolic conservation laws have been proposed in References [1–5]. These schemes resolve discontinuities without spurious oscillations and perform remarkably well in smooth regions. Most of them are the upwind schemes, which are commonly used to discretize hyperbolic

[*] Correspondence to: Gwo-Fong Lin, Department of Civil Engineering, National Taiwan University, Taipei 10617, Taiwan.
[†] E-mail: gflin@ntu.edu.tw

Received 21 January 2004
Revised 4 January 2005
Accepted 7 February 2005

equations according to the direction of wave propagation. The upwind shock-capturing schemes can be generally categorized into two classes: the flux-difference splitting (FDS) scheme and the flux-vector splitting (FVS) scheme [1, 2]. The FDS-type schemes use an approximate solution of the local Riemann problem, such as the Osher scheme, the Roe scheme, the Harten, Lax and van Leer (HLL) scheme, etc. The FVS-type schemes split the flux vector into positive and negative parts, such as the van Leer splitting (VLS) scheme, the Steger–Warming splitting (SWS) scheme, the local Lax-Friedrichs splitting (LLFS) scheme, etc.

In recent years, these shock-capturing upwind schemes have been applied to the solutions of the 2D SWE based on the finite-volume method (FVM). For instance, the first-order Osher scheme is used by Zhao *et al.* [6] and Wan *et al.* [7]; the second-order Roe scheme is adopted by Alcrudo and Garcia-Navarro [8], Anastasiou and Chan [9], Sleigh *et al.* [10], Tseng [11], Tseng and Chu [12], and Brufau and Garcia-Navarro [13]; the second-order HLL scheme is employed by Mingham and Causon [14], Hu *et al.* [15], Causon *et al.* [16], and Valiani *et al.* [17]; and the first-order HLLC scheme, where C stands for Contact, is applied by Zoppou and Roberts [18]. The comparisons of different first- and second-order-accurate schemes can also be found in several articles. For instance, Zhao *et al.* [19] compared the numerical accuracy, efficiency and stability of three first-order upwind schemes, including the Osher, the Roe, and the SWS schemes. Lin *et al.* [20] compared four second-order FVS schemes, including the Liou–Steffen splitting (LSS), the VLS, the SWS, and the LLFS schemes. Comprehensive comparisons of the performance of finite-volume solutions to SWE by five shock-capturing upwind schemes, namely the Osher, HLL, HLLC, Roe and the SWS schemes, were reported by Erduran *et al.* [21]. Among the five schemes, they found that the Osher scheme is the most accurate but quite complex to implement. According to the above literature review, with higher accuracy comparing to FVS-type schemes the FDS-type schemes such as the Osher, Roe and HLL schemes are popularly employed for solving SWE.

More recently, an artificially upstream FVS method for solving the Euler equations has been proposed by Sun and Takayama [22]. This upwind method splits the flux vector into two simple flux vectors by introducing two artificial wave speeds. One flux vector is discretized using the Steger–Warming approach. The other flux vector is easily solved by one-side upwind differencing. Unlike the well-known Roe finite difference splitting method, this method does not need any matrix operation and avoids the expansion shocks without any additional entropy fix. Moreover, its accuracy is comparable with the exact Riemann solver. The purpose of this present study is to adopt this upwind method to solve the 2D SWE based on the framework of the FVM.

Adopting the FVM, the 2D problem for solving SWE is reduced to a number of local 1D Riemann problems in the direction normal to the cell interface. The artificially upstream FVS method is then employed to formulate the numerical flux function for the solution of the local Riemann problem. Based on this algorithm, a first-order upstream flux-splitting finite-volume (UFF) scheme is proposed. To evaluate the numerical performances for solving SWE, the three FDS-type schemes, namely the Osher, Roe and HLL schemes, are selected to compare with the proposed UFF scheme. The second-order extension of the UFF scheme is achieved based on the monotonic upstream schemes for conservation laws (MUSCL) method [1, 2]. The proposed first-order and second-order schemes are applied to simulate four shallow water flows, including the 1D idealized dam-break problem, the oblique hydraulic jump, the circular dam-break problem, and the dam-break experiment with 45° bend channel.

2. GOVERNING EQUATIONS

The conservative form of the 2D SWE can be written in vector notation as [23, 24]

$$\frac{\partial \mathbf{Q}}{\partial t}+\frac{\partial \mathbf{F}}{\partial x}+\frac{\partial \mathbf{G}}{\partial y}=\mathbf{S} \tag{1}$$

in which

$$\mathbf{Q}=\begin{bmatrix} h \\ hu \\ hv \end{bmatrix};\quad \mathbf{F}=\begin{bmatrix} hu \\ hu^2+\dfrac{gh^2}{2} \\ huv \end{bmatrix};\quad \mathbf{G}=\begin{bmatrix} hv \\ huv \\ hv^2+\dfrac{gh^2}{2} \end{bmatrix};\quad \mathbf{S}=\begin{bmatrix} 0 \\ gh(s_{0x}-s_{\mathrm{f}x}) \\ gh(s_{0y}-s_{\mathrm{f}y}) \end{bmatrix} \tag{2}$$

where $\mathbf{Q}$ is the vector of conserved variable; $\mathbf{F}$ and $\mathbf{G}$ are the flux vectors in the x- and y-directions, respectively; $\mathbf{S}$ is a source term vector; h is the water depth; u and v are the depth-averaged velocity components in the x- and y-directions, respectively; g is the acceleration due to gravity; $s_{\mathrm{f}x}$ and $s_{\mathrm{f}y}$ are the bed friction slopes in the x- and y-directions, respectively; and s_{0x} and s_{0y} are the bed slopes in the x- and y-directions, respectively. The bed friction slopes are estimated using the Manning formula

$$s_{\mathrm{f}x}=\frac{un_{\mathrm{m}}^2\sqrt{u^2+v^2}}{h^{4/3}},\quad s_{\mathrm{f}y}=\frac{vn_{\mathrm{m}}^2\sqrt{u^2+v^2}}{h^{4/3}} \tag{3}$$

in which n_{m} is Manning's roughness coefficient.

3. NUMERICAL SCHEME

3.1. Discretization in finite-volume method

Discretization using the FVM is based on the integral form of the conservation equations. Integrating Equation (1) over an arbitrary control volume Ω gives

$$\iint_{\Omega}\frac{\partial \mathbf{Q}}{\partial t}\,\mathrm{d}W+\iint_{\Omega}\nabla\cdot\mathbf{E}\,\mathrm{d}W=\iint_{\Omega}\mathbf{S}\,\mathrm{d}W \tag{4}$$

in which $\mathrm{d}W$ is the area element and the flux vector $\mathbf{E}=[\mathbf{F},\mathbf{G}]^{\mathrm{T}}$. Using the divergence theorem, one can obtain the basic equation of FVM as follows:

$$\iint_{\Omega}\frac{\partial \mathbf{Q}}{\partial t}\,\mathrm{d}W+\int_{\partial\Omega}\mathbf{E}\cdot\mathbf{n}\,\mathrm{d}l=\iint_{\Omega}\mathbf{S}\,\mathrm{d}W \tag{5}$$

where $\partial\Omega$ is the boundary of the control volume Ω; $\mathbf{n}$ is the outward unit vector normal to the $\partial\Omega$; $\mathrm{d}l$ is the arc element; and the integrand $\mathbf{E}\cdot\mathbf{n}$ is the normal flux across a surface with normal $\mathbf{n}$. The vector of conserved variables $\mathbf{Q}$ is assumed to be constant over each cell. Hence, the basic vector equation of the FVM can be further discretized as

$$A\frac{\mathrm{d}\mathbf{Q}}{\mathrm{d}t}+\sum_{m=1}^{M}\mathbf{E}_{\mathrm{n}}^{m}L^{m}=A\mathbf{S} \tag{6}$$

where A is the area of the cell; m is the index that represents the side of the cell; M is the total number of the sides for the cell; $\mathbf{E}_\mathrm{n}^m$ is the normal flux across each side m separating two neighbouring cells; and L^m is the length of the m side for the cell. For convenience, the superscript m will be omitted hereafter. It will be only used for the length L^m as a reminder.

Based on the rotational invariance property of the governing equations [23], the interface flux normal to each side m is expressed as

$$\mathbf{E}_\mathrm{n}(\mathbf{Q}) = \mathbf{T}(\theta)^{-1}\mathbf{F}(\bar{\mathbf{Q}}) \tag{7}$$

where θ is the angle between the outward unit vector $\mathbf{n}$ and the x-axis; $\bar{\mathbf{Q}} = \mathbf{T}(\theta)\mathbf{Q}$ is the vector variables transformed from $\mathbf{Q}$; $\mathbf{F}(\bar{\mathbf{Q}}) = \mathbf{T}(\theta)\mathbf{E}_\mathrm{n}(\mathbf{Q})$ is the transformed normal flux; $\mathbf{T}(\theta)$ is the transformation matrix which can be obtained by rotating the coordinate axes; and $\mathbf{T}(\theta)^{-1}$ is the inverse transformation matrix. Substituting Equation (7) into Equation (6) leads to

$$A\frac{\mathrm{d}\mathbf{Q}}{\mathrm{d}t} + \sum_{m=1}^{M} \mathbf{T}(\theta)^{-1}\mathbf{F}(\bar{\mathbf{Q}})L^m = A\mathbf{S} \tag{8}$$

where $\bar{\mathbf{Q}} = [h, hu_\mathrm{n}, hv_\mathrm{t}]^\mathrm{T}$; $\mathbf{F}(\bar{\mathbf{Q}}) = [hu_\mathrm{n}, hu_\mathrm{n}^2 + gh^2/2, hu_\mathrm{n}v_\mathrm{t}]^\mathrm{T}$; and u_n and v_t are, respectively, the flow velocity components in $\bar{x}$ (normal) and $\bar{y}$ (tangential) directions, which are given by $u_\mathrm{n} = u\cos\theta + v\sin\theta$ and $v_\mathrm{t} = v\cos\theta - u\sin\theta$.

3.2. Upstream flux-splitting finite-volume (UFF) scheme

In practice, it is not always possible to use the FVM to solve Equation (8) directly when the source terms exist. To deal with the source terms, the fractional splitting technique [24] is employed herein. The splitting can be expressed as

$$\mathbf{Q}^{n+1} = I^{(\Delta t)}H^{(\Delta t)}(\mathbf{Q}^n) \tag{9}$$

where $I^{(\Delta t)}$ and $H^{(\Delta t)}$ are operators corresponding to solutions of the inhomogeneous (source terms) and the homogeneous parts, respectively; n is the time index; and Δt is the time increment.

Based on Equation (9), Equation (8) becomes

$$\frac{\mathrm{d}\mathbf{Q}}{\mathrm{d}t} + \frac{1}{A}\sum_{m=1}^{M} \mathbf{T}(\theta)^{-1}\mathbf{F}(\bar{\mathbf{Q}})L^m = 0 \tag{10a}$$

$$\frac{\mathrm{d}\mathbf{Q}}{\mathrm{d}t} = \mathbf{S} \tag{10b}$$

In this paper, Equation (10a) (homogeneous part) is solved using Euler's method [21]. In addition, Equation (10b) (inhomogeneous part) can be solved by taking the conservative values calculated at the previous time step as initial values [21]. Therefore, the conservative finite-volume scheme is formulated as

$$\hat{\mathbf{Q}}_{i,j} = \mathbf{Q}_{i,j}^n - \frac{\Delta t}{A_{i,j}}\left[\sum_{m=1}^{M} \mathbf{T}(\theta)^{-1}\mathbf{F}^{(1)}(\bar{\mathbf{Q}})L^m\right]_{i,j} \tag{11a}$$

$$\mathbf{Q}_{i,j}^{n+1} = \hat{\mathbf{Q}}_{i,j} + \Delta t\mathbf{S}(\hat{\mathbf{Q}}_{i,j}) \tag{11b}$$

where i and j are the space indexes; $A_{i,j}$ is the area for the cell (i,j); $\mathbf{Q}_{i,j}^{n}$ is the vector of conserved variables for the cell (i,j) at time index n; $\hat{\mathbf{Q}}_{i,j}$ is the vector of predicted variables on the cell centre (i,j); $\mathbf{S}(\hat{\mathbf{Q}}_{i,j})$ is the vector of the source terms based on the predicted variables; $\mathbf{Q}_{i,j}^{n+1}$ is the vector of conserved variables for the cell (i,j) at next time step $n+1$; and $\mathbf{F}^{(1)}(\bar{\mathbf{Q}})$ is the first-order numerical flux.

3.2.1. The local Riemann problem. In Equation (11a), the estimation of the numerical flux is required to obtain the solutions. Because of the rotational invariance property, the 2D problem in Equation (8) can be dealt with as a series of 1D local problems in the direction normal to the cell interface. The Riemann problem is an initial value problem, which can be written as [20]

$$\frac{\partial \bar{\mathbf{Q}}}{\partial t}+\frac{\partial[\mathbf{F}(\bar{\mathbf{Q}})]}{\partial \bar{x}}=0 \tag{12a}$$

with

$$\bar{\mathbf{Q}}(\bar{x},0)=\begin{cases}\bar{\mathbf{Q}}_{\mathrm{L}} & \bar{x}<0\\ \bar{\mathbf{Q}}_{\mathrm{R}} & \bar{x}>0\end{cases} \tag{12b}$$

As illustrated in Figure 1, the origin of the $\bar{x}$-axis is located at the midpoint of the cell interface along outward normal direction. The $\mathbf{F}(\bar{\mathbf{Q}})$ is a normal outward flux at the origin of a local axis $\bar{x}$. The conserved vectors $\bar{\mathbf{Q}}_{\mathrm{L}}$ and $\bar{\mathbf{Q}}_{\mathrm{R}}$ represent the transformed quantities on the left and the right cells of the cell interface, respectively.

Many numerical flux functions of the shock-capturing upwind schemes can be adopted for Equation (12), such as the Osher, the Roe, and the HLL schemes. In this paper, the artificially

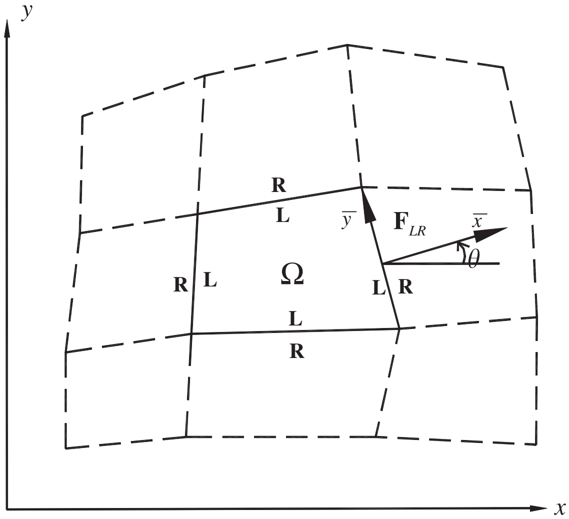


Figure 1. The finite-volume cell Ω and the Riemann interface.

upstream FVS method proposed by Sun and Takayama [22] is employed to estimate the numerical flux $\mathbf{F}_{\mathrm{LR}}(\bar{\mathbf{Q}}_{\mathrm{L}}, \bar{\mathbf{Q}}_{\mathrm{R}})$ through each cell interface as shown in Figure 1.

3.2.2. The formulation of the numerical flux function. According to the algorithm of the artificially upstream FVS method [22], the normal outward flux $\mathbf{F}(\bar{\mathbf{Q}})$ can be decomposed into a convective component $u_{\mathrm{n}}\bar{\mathbf{Q}}$ and a pressure component $\mathbf{P}$ as

$$\mathbf{F}(\bar{\mathbf{Q}}) = \begin{bmatrix} hu_{\mathrm{n}} \\ hu_{\mathrm{n}}^2 + \dfrac{gh^2}{2} \\ hu_{\mathrm{n}}v_{\mathrm{t}} \end{bmatrix} = u_{\mathrm{n}} \begin{bmatrix} h \\ hu_{\mathrm{n}} \\ hv_{\mathrm{t}} \end{bmatrix} + \begin{bmatrix} 0 \\ p \\ 0 \end{bmatrix} = u_{\mathrm{n}}\bar{\mathbf{Q}} + \mathbf{P} \tag{13}$$

where $p = gh^2/2$ is the hydrostatic pressure for shallow water flows. The fundamental idea of the artificially upstream FVS method is to split the flux vector $\mathbf{F}(\bar{\mathbf{Q}})$ in Equation (13) by introducing a weighting parameter K and two wave speeds (s_1 and s_2):

$$\mathbf{F}(\bar{\mathbf{Q}}) = (1-K)[(u_{\mathrm{n}} - s_1)\bar{\mathbf{Q}} + \mathbf{P}] + K[(u_{\mathrm{n}} - s_2)\bar{\mathbf{Q}} + \mathbf{P}] = (1-K)\mathbf{F}_1 + K\mathbf{F}_2 \tag{14}$$

where $\mathbf{F}_1 = (u_{\mathrm{n}} - s_1)\bar{\mathbf{Q}} + \mathbf{P}$; $\mathbf{F}_2 = (u_{\mathrm{n}} - s_2)\bar{\mathbf{Q}} + \mathbf{P}$; and K is defined as

$$K = \frac{s_1}{s_1 - s_2} \tag{15}$$

Therefore, these two flux vectors, $\mathbf{F}_1$ and $\mathbf{F}_2$, are, respectively, different from the original $\mathbf{F}(\bar{\mathbf{Q}})$ because of the auxiliary terms $-s_1\bar{\mathbf{Q}}$ and $-s_2\bar{\mathbf{Q}}$. Their corresponding matrixes of eigenvalues are diagonal $(u_{\mathrm{n}} - s_1 - c, u_{\mathrm{n}} - s_1, u_{\mathrm{n}} - s_1 + c)$ and diagonal $(u_{\mathrm{n}} - s_2 - c,\ u_{\mathrm{n}} - s_2,\ u_{\mathrm{n}} - s_2 + c)$, respectively. The local wave celerity is given by $c = \sqrt{gh}$.

Obviously, the eigenvalues of the flux vectors $\mathbf{F}_1$ and $\mathbf{F}_2$ can be changed by varying the introduced wave speeds s_1 and s_2. Hence, appropriate values of s_1 and s_2 may be adopted to simplify the upwinding discretization of the governing equations. Following the choices of the wave speeds proposed by Sun and Takayama [22], s_1 and s_2 are expressed as

$$s_1 = u_{\mathrm{n}}, \quad s_2 = \begin{cases} u_{\mathrm{n}} - c & \text{if } s_1 > 0 \\ u_{\mathrm{n}} + c & \text{if } s_1 \leqslant 0 \end{cases} \tag{16}$$

and then the two splitted flux vectors become

$$\mathbf{F}_1 = \mathbf{P}, \quad \mathbf{F}_2 = (u_{\mathrm{n}} - s_2)\bar{\mathbf{Q}} + \mathbf{P} \tag{17}$$

Since the eigenvalues of the Jacobian matrix of flux vector $\mathbf{F}_1$ become $(-c, 0, +c)$, the numerical discretization of $\mathbf{F}_1$ can be derived following the Steger–Warming approach [1]. The corresponding numerical flux $\mathbf{F}_{\mathrm{LR},1}$ can be expressed as

$$\mathbf{F}_{\mathrm{LR},1} = \tfrac{1}{2}(\mathbf{P}_{\mathrm{L}} + \mathbf{P}_{\mathrm{R}}) + \Delta\bar{\mathbf{Q}}_{\mathrm{av}} \tag{18}$$

where $\Delta\bar{\mathbf{Q}}_{\mathrm{av}}$ represents the numerical viscosity, which is given by

$$\Delta\bar{\mathbf{Q}}_{\mathrm{av}} = \frac{1}{2c_{\mathrm{L}}}\begin{bmatrix} p \\ pu_{\mathrm{n}} \\ pv_{\mathrm{t}} \end{bmatrix}_{\mathrm{L}} - \frac{1}{2c_{\mathrm{R}}}\begin{bmatrix} p \\ pu_{\mathrm{n}} \\ pv_{\mathrm{t}} \end{bmatrix}_{\mathrm{R}} \tag{19}$$

In Equation (19), the values of c_{L} and c_{R} can be replaced by an intermediate wave celerity c_{LR} for improving the numerical accuracy [25]. Thus, Equation (19) becomes

$$\Delta\bar{\mathbf{Q}}_{\mathrm{av}} = \frac{1}{2c_{\mathrm{LR}}}\begin{bmatrix} p_{\mathrm{L}} - p_{\mathrm{R}} \\ (pu_{\mathrm{n}})_{\mathrm{L}} - (pu_{\mathrm{n}})_{\mathrm{R}} \\ (pv_{\mathrm{t}})_{\mathrm{L}} - (pv_{\mathrm{t}})_{\mathrm{R}} \end{bmatrix} \tag{20}$$

The eigenvalues of the Jacobian matrix for the second flux vector $\mathbf{F}_2$ still maintains the property that is either all non-negative or all non-positive. Therefore, flux vector $\mathbf{F}_2$ can be easily upwinded based on the sign of the wave speed s_1. The corresponding numerical flux $\mathbf{F}_{\mathrm{LR},2}$ is given by

$$\mathbf{F}_{\mathrm{LR},2} = [(u_{\mathrm{n}})_{\mathrm{L/R}} - s_2]\bar{\mathbf{Q}}_{\mathrm{L/R}} + \mathbf{P}_{\mathrm{L/R}} \tag{21}$$

where the subscript L/R is defined as

$$\mathrm{L/R} = \begin{cases} \mathrm{L} & \text{if } s_1 > 0 \\ \mathrm{R} & \text{if } s_1 \leqslant 0 \end{cases} \tag{22}$$

To completely determine the numerical fluxes in Equations (18) and (21), the estimations of c_{LR}, s_1 and s_2 are required. For improving numerical accuracy, the concept of the common wave speed, $c_{\mathrm{LR}} = \max(c_{\mathrm{L}}, c_{\mathrm{R}})$, introduced by Wada and Liou [25] is employed herein to replace the algebraic average, $c_{\mathrm{LR}} = 0.5(c_{\mathrm{L}} + c_{\mathrm{R}})$, proposed by Sun and Takayama [22]. Numerical values of s_1 and s_2 can be computed from

$$s_1 = \tfrac{1}{2}(u_{\mathrm{nL}} + u_{\mathrm{nR}}) \tag{23a}$$

$$s_2 = \begin{cases} \min(0, u_{\mathrm{nL}} - c_{\mathrm{L}}, u_{\mathrm{n}}^* - c^*) & \text{if } s_1 > 0 \\ \max(0, u_{\mathrm{nR}} + c_{\mathrm{R}}, u_{\mathrm{n}}^* + c^*) & \text{if } s_1 \leqslant 0 \end{cases} \tag{23b}$$

in which u_{n}^* and c^* is estimated using the exact solutions given by Toro [24]

$$u_{\mathrm{n}}^* = \tfrac{1}{2}(u_{\mathrm{nL}} + u_{\mathrm{nR}}) + c_{\mathrm{L}} - c_{\mathrm{R}} \tag{24a}$$

$$c^* = \tfrac{1}{2}(c_{\mathrm{L}} + c_{\mathrm{R}}) + \tfrac{1}{4}(u_{\mathrm{nL}} - u_{\mathrm{nR}}) \tag{24b}$$

Accordingly, the interface numerical flux $\mathbf{F}_{\mathrm{LR}}$ is expressed as

$$\mathbf{F}_{\mathrm{LR}} = (1 - K)\mathbf{F}_{\mathrm{LR},1} + K\mathbf{F}_{\mathrm{LR},2} \tag{25}$$

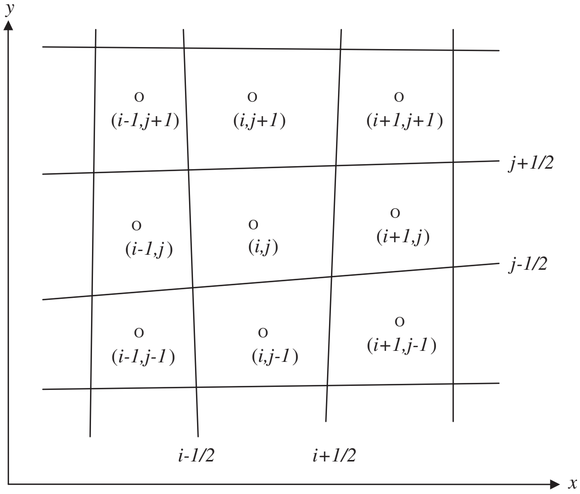


Figure 2. The computational cells in the x–y coordinates.

Substituting Equations (18) and (21) into Equation (25) results in the first-order numerical flux function:

$$\mathbf{F}_{\rm LR}(\bar{\mathbf{Q}}_{\rm L},\bar{\mathbf{Q}}_{\rm R}) = (1-K)[\tfrac{1}{2}(\mathbf{P}_{\rm L}+\mathbf{P}_{\rm R})+\Delta\bar{\mathbf{Q}}_{\rm av}] + K[(u_{\rm n})_{\rm L/R}-s_2]\bar{\mathbf{Q}}_{\rm L/R} + K\mathbf{P}_{\rm L/R} \tag{26}$$

Equation (26) shows that only the variables of the neighbouring cells (i.e. $\bar{\mathbf{Q}}_{\rm L}$ and $\bar{\mathbf{Q}}_{\rm R}$) are used to calculate the numerical flux $\mathbf{F}_{\rm LR}$ through each cell interface. It is also shown that the proposed numerical flux function does not need any Jacobian matrix operation for estimating the cell interface flux.

Based on Equation (26), the first-order numerical flux through the cell interface $(i+1/2,j)$ in Equation (11a) can be expressed as

$$\mathbf{F}^{(1)}(\bar{\mathbf{Q}}) = \mathbf{F}_{\rm LR}(\bar{\mathbf{Q}}_{\rm L},\bar{\mathbf{Q}}_{\rm R}) = \mathbf{F}_{\rm LR}(\bar{\mathbf{Q}}^n_{i,j},\bar{\mathbf{Q}}^n_{i+1,j}) \tag{27}$$

where $\bar{\mathbf{Q}}^n_{i,j}$ is the vector of transformed variables for the cell (i,j) at time index n. The computational cells in the x–y coordinate system are illustrated in Figure 2. By using Equation (27) to obtain the first-order numerical flux, Equation (11) leads to the UFF scheme.

3.3. Second-order extension

To obtain second-order accuracy in space, the MUSCL method [1, 2] is adopted herein. In addition, the predictor–corrector method for time integration [1] is used to achieve second-order accuracy in time. Therefore, Equation (8) can be further discretized as

$$\hat{\mathbf{Q}}_{i,j} = \mathbf{Q}^n_{i,j} - \frac{\Delta t}{2A_{i,j}}\left[\sum_{m=1}^{M}\mathbf{T}(\theta)^{-1}\mathbf{F}^{(1)}(\bar{\mathbf{Q}})L^m\right]_{i,j} + \frac{\Delta t}{2}\mathbf{S}(\mathbf{Q}^n_{i,j}) \tag{28a}$$

$$\mathbf{Q}^{n+1}_{i,j} = \mathbf{Q}^n_{i,j} - \frac{\Delta t}{A_{i,j}}\left[\sum_{m=1}^{M}\mathbf{T}(\theta)^{-1}\mathbf{F}^{(2)}(\bar{\mathbf{Q}})L^m\right]_{i,j} + \Delta t\mathbf{S}(\hat{\mathbf{Q}}_{i,j}) \tag{28b}$$

where the first-order numerical flux has been defined in Equation (27), and $\mathbf{F}^{(2)}(\bar{\mathbf{Q}})$ is the numerical flux with second-order accuracy in space. The second-order numerical flux $\mathbf{F}^{(2)}(\bar{\mathbf{Q}})$ through the cell interface $(i+1/2, j)$ can be obtained from the proposed numerical flux function

$$\mathbf{F}^{(2)}(\bar{\mathbf{Q}}) = \mathbf{F}_{\mathrm{LR}}(\bar{\mathbf{Q}}^{\mathrm{L}}_{i+1/2,j}, \bar{\mathbf{Q}}^{\mathrm{R}}_{i+1/2,j}) \tag{29}$$

where $\bar{\mathbf{Q}}^{\mathrm{L}}_{i+1/2,j}$ and $\bar{\mathbf{Q}}^{\mathrm{R}}_{i+1/2,j}$ are the left and right cell-interface variables, respectively. Based on the MUSCL method, the values of conserved variables on the left and right of the cell interface $(i+1/2, j)$ are

$$\bar{\mathbf{Q}}^{\mathrm{L}}_{i+1/2,j} = \mathbf{T}(\theta)\hat{\mathbf{Q}}_{i,j} + \tfrac{1}{2}\bar{\Delta}_{i,j} \tag{30a}$$

$$\bar{\mathbf{Q}}^{\mathrm{R}}_{i+1/2,j} = \mathbf{T}(\theta)\hat{\mathbf{Q}}_{i+1,j} - \tfrac{1}{2}\bar{\Delta}_{i+1,j} \tag{30b}$$

in which $\bar{\Delta}_{i,j} = \bar{\Delta}_{i,j}(\Delta_{i+1/2,j}, \Delta_{i-1/2,j})$ is a nonlinear slope limiter function for the cell (i, j). The van Leer limiter function is employed herein:

$$\bar{\Delta}_{i,j} = [\mathrm{sgn}(\Delta_{i+1/2,j}) + \mathrm{sgn}(\Delta_{i-1/2,j})]\frac{|\Delta_{i+1/2,j}|\cdot|\Delta_{i-1/2,j}|}{|\Delta_{i+1/2,j}| + |\Delta_{i-1/2,j}|} \tag{31}$$

where $\Delta_{i-1/2,j} = \bar{\mathbf{Q}}^n_{i,j} - \bar{\mathbf{Q}}^n_{i-1,j}$, $\Delta_{i+1/2,j} = \bar{\mathbf{Q}}^n_{i+1,j} - \bar{\mathbf{Q}}^n_{i,j}$, and sgn refers to the sign function [1]. The second-order extension of the UFF scheme is denoted as the UFF-MUSCL scheme in the following sections.

3.4. Stability and boundary conditions

To ensure numerical stability of the proposed schemes, the time step Δt must be restricted by the Courant–Friedrichs–Lewy (CFL) stability condition [17], which is expressed as

$$\mathrm{CFL} = \frac{\Delta t}{\min[d_{i,j}]}\max\left[\left(\sqrt{u^2+v^2} + \sqrt{gh}\right)_{i,j}\right] \leqslant 1 \tag{32}$$

where i, j are the cell indexes and $d_{i,j}$ indicates the whole set of distances between the i, jth centroid and the centroids of the four adjacent cells.

The boundary conditions used herein are divided into two different types: the land boundary and the open boundary [19, 24]. At the land boundary, the velocity normal to the land is set to be zero to represent no flux through the boundary. The land boundary condition is specified as:

$$h_{\mathrm{R}} = h_{\mathrm{L}}, \quad u_{\mathrm{nR}} = -u_{\mathrm{nL}}, \quad v_{\mathrm{tR}} = v_{\mathrm{tL}} \tag{33}$$

where the variables with subscript L and R are the known and the unknown states, respectively. Besides, the subscripts R and L stand for the right and left Riemann states, respectively, at a cell interface boundary. At the open boundary, two different boundary conditions, supercritical inflow at upstream inflow boundary and transmissive at downstream outflow boundary, are used herein. For the supercritical inflow boundary condition, three variables (i.e. h_R, u_{nR} and v_{tR}) are given. The transmissive boundary allows waves to pass through without reflection, and its condition is given as

$$h_R = h_L, \quad u_{nR} = u_{nL}, \quad v_{tR} = v_{tL} \tag{34}$$

4. NUMERICAL RESULTS AND DISCUSSIONS

The numerical performance of the proposed first-order UFF scheme and its second-order extension UFF-MUSCL scheme will be evaluated through modelling some well-known shallow water flow problems, including the 1D idealized dam breaking, the oblique hydraulic jump, the circular dam breaking, and the dam-break experiment with 45° bend channel. Except the test problem of the dam-break experiment, three well-known upwind schemes, including the Osher, Roe, and HLL schemes, are selected to compare with the proposed UFF scheme for the evaluation of the numerical performance. All of the tests were performed on a Pentium IV equipped with a 256 MB RAM.

4.1. 1D idealized dam-break flow

The 1D idealized dam-break problem is applied to test the shock-capturing capabilities of the UFF scheme. The idealized dam-break flow problem in a rectangular, frictionless, and horizontal channel is illustrated in Figure 3, where h_u and h_d are initial water depths upstream and downstream of the dam, respectively. A channel with 2000 m in length and 10 m in width is considered. A dam is located at the middle of the channel. At time $t = 0^+$, the dam is broken instantaneously. A shock wave travelling downstream and a rarefaction wave moving upstream are then created. The initial upstream water depth is 10 m, and the corresponding initial downstream water depths are given as 5, 0.1 and 0 m, respectively, for each test. Thus, three test cases with water depth ratios h_d/h_u of 0.5, 0.01, and 0 (dry bed condition) are considered. In the extreme test case (dry bed condition), an almost negligible water depth of 0.00001 m is assumed at the downstream of the dam to avoid the mathematical difficulty. The CFL number is set to be 0.95, and 100 computational cells are used for all test cases. The

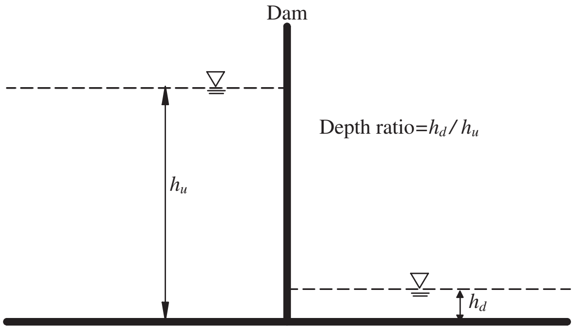


Figure 3. A schematic representation of 1D dam-break problem.

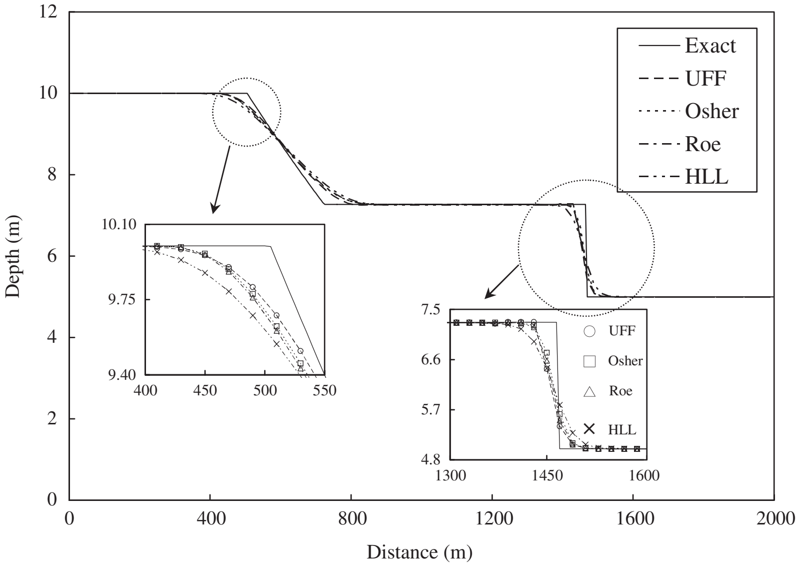


Figure 4. Comparisons of exact solutions with simulated depths using first-order schemes for a water depth ratio h_d/h_u of 0.5.

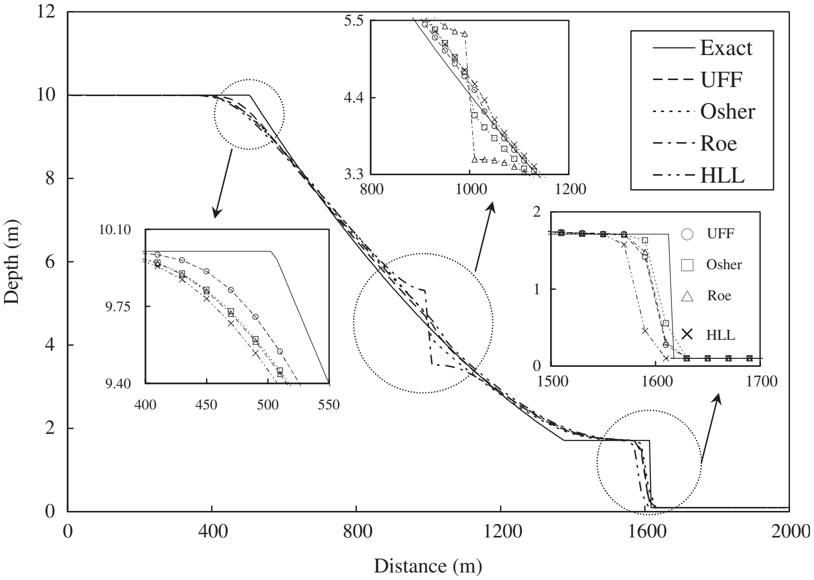


Figure 5. Comparisons of exact solutions with simulated depths using first-order schemes for a water depth ratio h_d/h_u of 0.01.

simulation time is 50 s for test cases with water depth ratios of 0.5 and 0.01. For the test case with dry bed condition, the simulation time is 30 s after dam break. The exact solutions can be found in Reference [26].

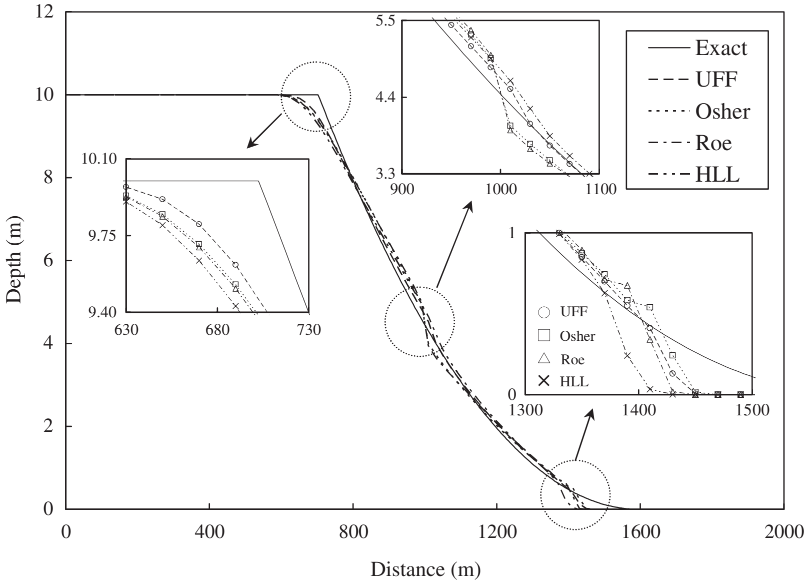


Figure 6. Comparisons of exact solutions with simulated depths using first-order schemes for a water depth ratio h_d/h_u of 0.

Comparison of the exact solutions with the simulated water depths using the four first-order upwind schemes for a water depth ratio of 0.5 is shown in Figure 4. As shown in Figure 4 with the close-up of the shock wave front as well as the head of the rarefaction wave, all schemes provide solutions without any spurious oscillations. The Osher scheme presents the most accurate solution of the shock wave front, whereas the HLL scheme yields more diffusive results. Among the schemes tested, the proposed UFF scheme has the best resolution of the head of the rarefaction wave.

For the test case with a water depth ratio of 0.01, as shown in Figure 5, the Roe scheme produces the expansion shock at the dam site, whereas the UFF, Osher and HLL schemes do not. Hence, the formula of entropy correction given by Harten and Hyman [1] is employed for the Roe scheme afterward. Results of comparisons in this test case are similar to those in the previous test case for the resolution of the shock front and the head of the rarefaction wave. The simulated results at the dam site also show that the UFF scheme produces the best resolution. Unlike the Roe scheme, the UFF scheme satisfying entropy condition can resolve the rarefaction wave smoothly at the dam site. For the test case with dry bed condition, as shown in Figure 6, the UFF scheme gives the best resolutions of the rarefaction wave at the dam site as well as at the head. The simulated wet/dry front by the UFF scheme fits the exact solution much more smoothly and closely than that by other presented schemes.

To show the influence of the CFL number on the simulated results, different CFL numbers are considered herein with fixed cell size of 20 m. Using the first-order UFF scheme, Figure 7 shows the influence of the CFL number on the simulated water depth for the test case with the water depth ratio h_d/h_u of 0.01. It is clear that the proposed scheme do not produce oscillations in the solutions even if the CFL number equals one. In addition, the scheme with a higher CFL can achieve better resolution of the shock front.

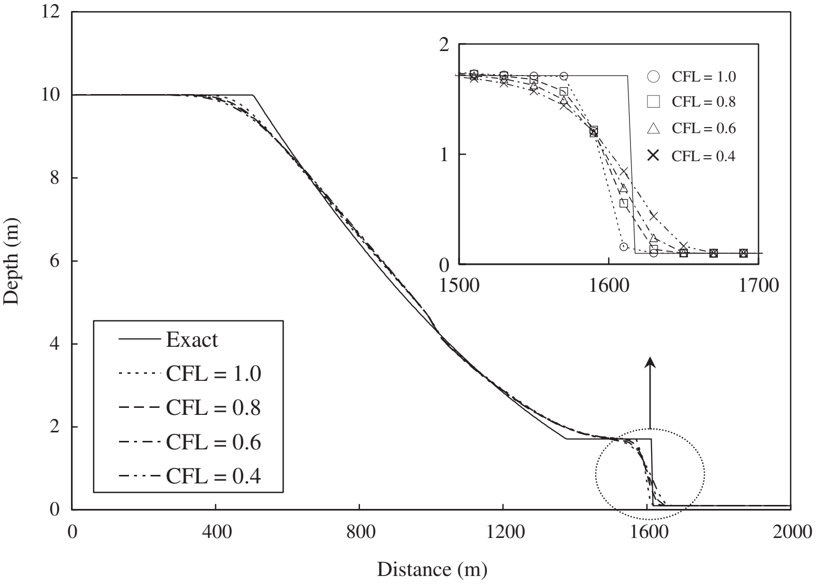


Figure 7. The influence of the CFL number on the simulated water depth using the first-order UFF scheme for the water depth ratio h_d/h_u of 0.01.

To evaluate the numerical accuracy quantitatively, two different error norms, L_2 (overall error norm) and L_∞ (max error norm), are used herein [27].

$$L_2 = \sqrt{\frac{\sum(Y_{i,j}^{\text{sim}} - Y_{i,j}^{\text{exact}})^2}{\sum(Y_{i,j}^{\text{exact}})^2}}, \quad L_\infty = \frac{\max|Y_{i,j}^{\text{sim}} - Y_{i,j}^{\text{exact}}|}{\max|Y_{i,j}^{\text{exact}}|} \tag{35}$$

where $Y_{i,j}^{\text{sim}}$ and $Y_{i,j}^{\text{exact}}$ are the simulated solution and the exact solution at cell (i,j), respectively. Table I summarizes the error norms of the water depth and CPU time for water depth ratios of $0.5, 0.01$ and 0. The results show that the proposed UFF scheme yields the smallest L_2 norm of water depth, whereas the HLL scheme gives the largest those. As listed in Table I, for the test cases with water depth ratios of 0.5 and 0.01, the values of the L_∞ norms indicate that the Osher scheme presents better solutions near the shock front. Nevertheless, for the test case with dry bed condition downstream, the UFF scheme performs the most accurate solutions in the entire simulation domain. Accordingly, the UFF scheme consumes the shortest CPU time, whereas the Roe scheme takes the longest one.

Based on the analyses of the above tested cases, the UFF scheme can avoid the entropy-violating solution and it resolves rarefaction wave more smoothly as well as accurately at the dam site. Although the Osher scheme performs better solutions near the shock front locally referring to the smallest L_∞ in Table I for the test cases with water depth ratios of 0.5 and 0.01, the proposed UFF scheme achieves superior overall numerical accuracy and efficiency among the schemes tested. Furthermore, the UFF scheme can simulate the wet/dry wave front passing over a dry bed condition downstream well.

Table I. Simulated results for the 1D idealized dam-break problem using first-order schemes.

	$h_d/h_u = 0.5$			$h_d/h_u = 0.01$			$h_d/h_u = 0$		
Schemes	L_2	L_∞	CPU (s)	L_2	L_∞	CPU (s)	L_2	L_∞	CPU (s)
UFF	0.0155	0.0695	0.058	0.0319	0.1612	0.076	0.0168	0.0474	0.045
Osher	0.0158	0.0619	0.077	0.0321	0.1392	0.101	0.0233	0.0539	0.061
Roe	0.0169	0.0688	0.102	0.0361	0.1612	0.136	0.0259	0.0561	0.079
HLL	0.0212	0.0825	0.071	0.0406	0.1629	0.091	0.0326	0.0656	0.055

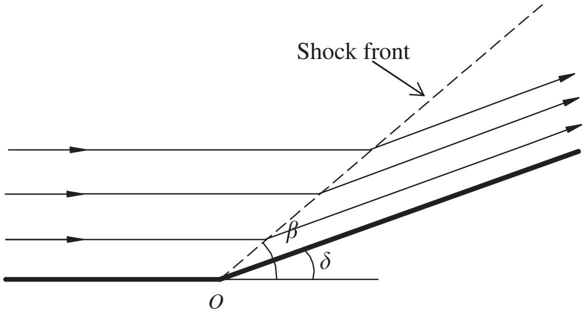


Figure 8. The plan view of the oblique shock front.

4.2. Oblique hydraulic jump

When a converging vertical boundary is deflected along the channel contraction through an angle δ inward the supercritical flow, an oblique hydraulic jump originating at point O can be formed with an angle of β as shown in Figure 8. This test problem in steady supercritical flow has been commonly adopted and simulated by researchers [8, 11, 12, 15, 16, 20, 21]. In the converging channel with zero bed slope, the angle between the converging wall and the flow direction is taken as $\delta = 8.95°$. Figure 9 shows the geometry and the computational mesh, in which 80×60 non-rectangular cells are used. The initial conditions corresponding to a Froude number of 2.74 are given by: the water depth of 1 m, the velocity component u of 8.57 m/s and v of zero. At the upstream inflow boundary, the supercritical inflow boundary conditions of $h = 1$ m, $u = 8.57$ m/s and $v = 0$ m/s are imposed. The transmissive boundary conditions are given using Equation (34) at the downstream outflow boundary. The computational time step is taken as 0.02 s. To obtain a steady-state solution, a convergence criterion in terms of the relative error R is defined as

$$R = \sqrt{\frac{\sum (h_{i,j}^{n+1} - h_{i,j}^{n})^2}{\sum (h_{i,j}^{n})^2}} \leqslant 1.0 \times 10^{-5} \tag{36}$$

where $h_{i,j}^{n}$ and $h_{i,j}^{n+1}$ are the local water depths at the time steps n and $n+1$, respectively.

Figure 10 shows the comparison between the exact solutions and the simulated water depths using the UFF, Osher, Roe and HLL schemes along line EGH illustrated in Figure 9. The

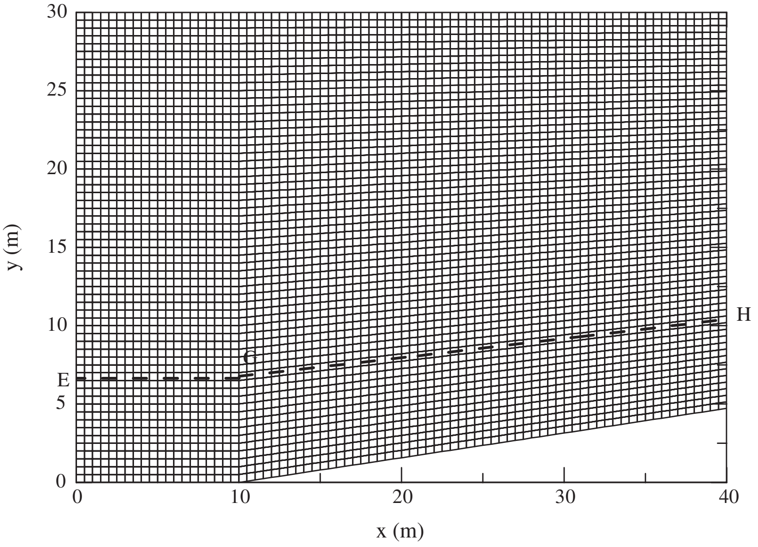


Figure 9. The geometry and the computational mesh for the 2D oblique hydraulic jump problem.

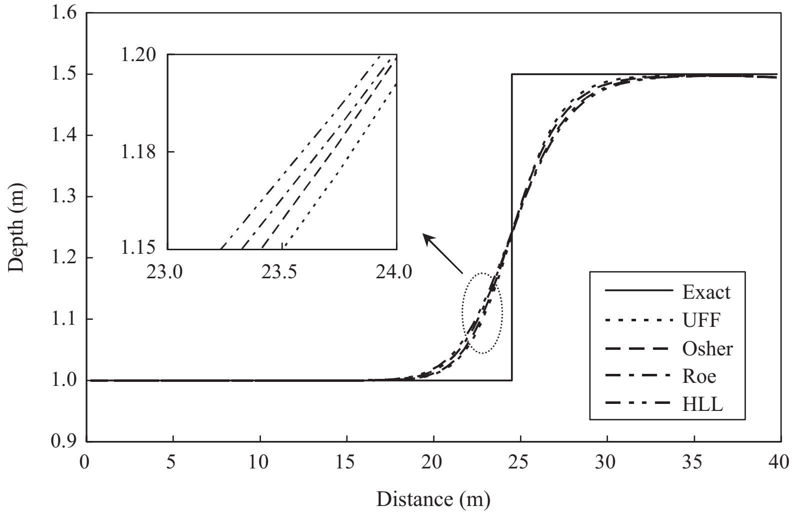


Figure 10. Comparisons of the exact solutions with the simulated water depth profiles along line EGH (see Figure 9) using first-order schemes.

exact solutions can be found in Reference [28]. As shown in Figure 10, the oblique hydraulic jump is captured well by all schemes. However, from the close-up near the shock front shown in Figure 10, the proposed UFF scheme even resolves the jump slightly sharper than the other schemes.

Table II. Simulated results for the oblique hydraulic jump problem using first-order schemes.

Scheme	L_2 norm of depth	L_2 norm of velocity	L_2 norm of shock angle	CPU time for convergence criterion (s)
UFF	0.035	0.0058	0.0030	15.63
Osher	0.037	0.0061	0.0036	21.02
Roe	0.039	0.0063	0.0042	29.05
HLL	0.041	0.0068	0.0044	19.26

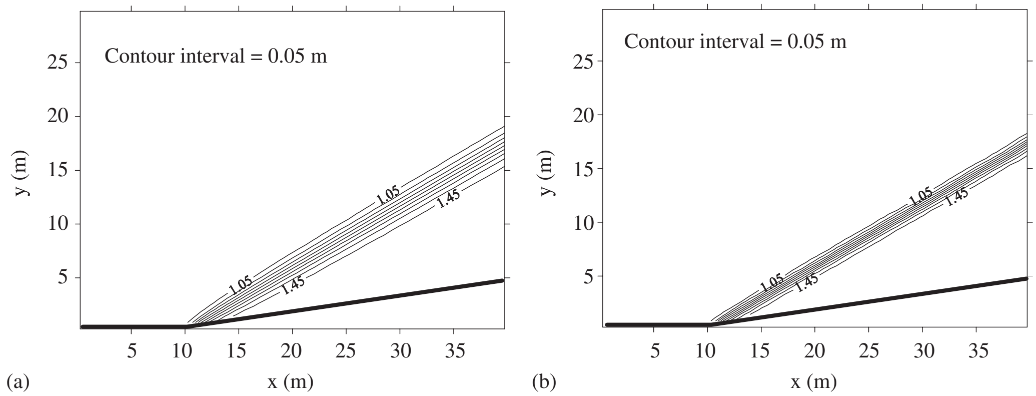


Figure 11. The contour plots of the 2D oblique hydraulic jump using: (a) UFF; and (b) UFF-MUSCL schemes.

Table II lists the simulated results, including the L_2 norms of water depth, velocity, and shock angle. The CPU time for reaching the convergence criterion is also presented in Table II. It is found that the proposed UFF scheme has the best numerical accuracy and efficiency. In addition, Figures 11(a) and 11(b) show the simulated water depth contours of the steady state solutions using the UFF and UFF-MUSCL schemes, respectively. The simulated results show no oscillations produced for both schemes. Certainly, the second-order UFF-MUSCL scheme presents a better resolution than the first-order UFF scheme. From the simulated results presented above, we may conclude that the UFF scheme has the best numerical performance in modelling the 2D oblique hydraulic jump problem.

4.3. *Circular dam-break flow*

This problem is designed to test the capability of the proposed scheme in modelling 2D symmetric discontinuous free surfaces. Numerical results for this test problem can be found in References [8, 11, 14, 21]. Figure 12 shows the geometry, in which a cylindrical dam with radius 11 m is located in the middle of the computational domain. The circular mesh (Figure 13) consisting of 50 cells in the tangential direction and 25 cells of 1 m length along the radial direction is employed herein.

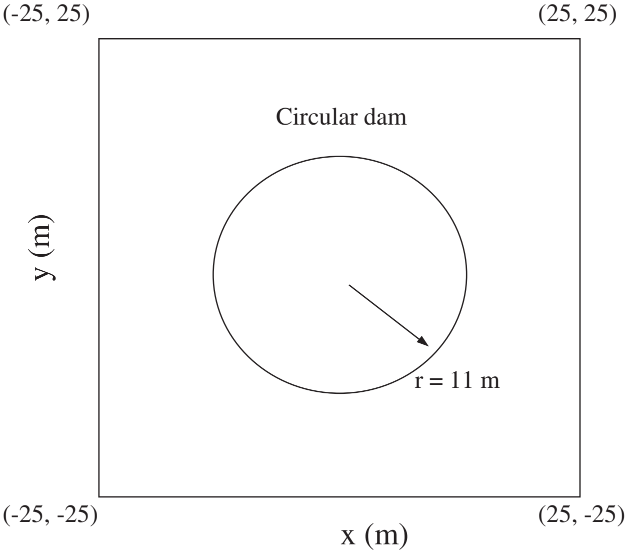


Figure 12. The geometry for the 2D circular dam-break problem.

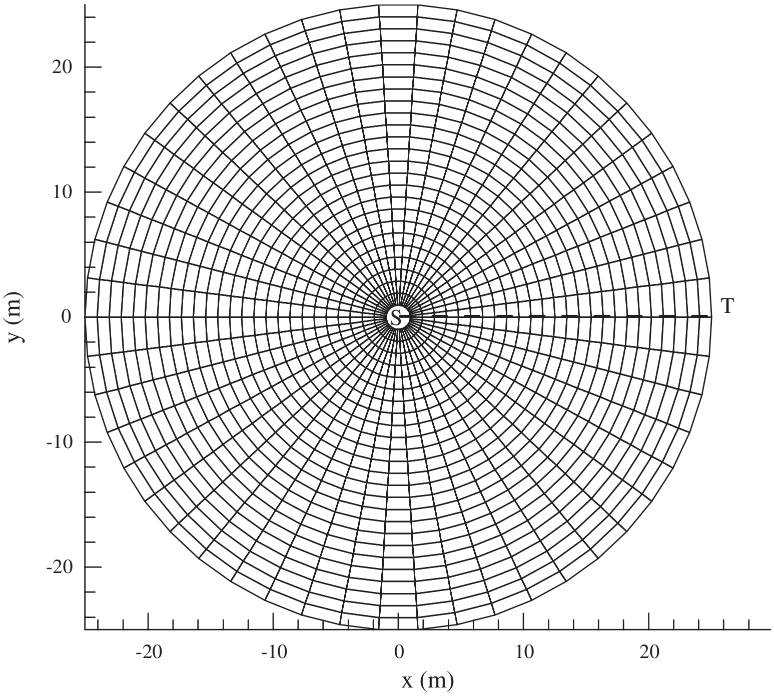


Figure 13. The computational mesh for the 2D circular dam-break problem.

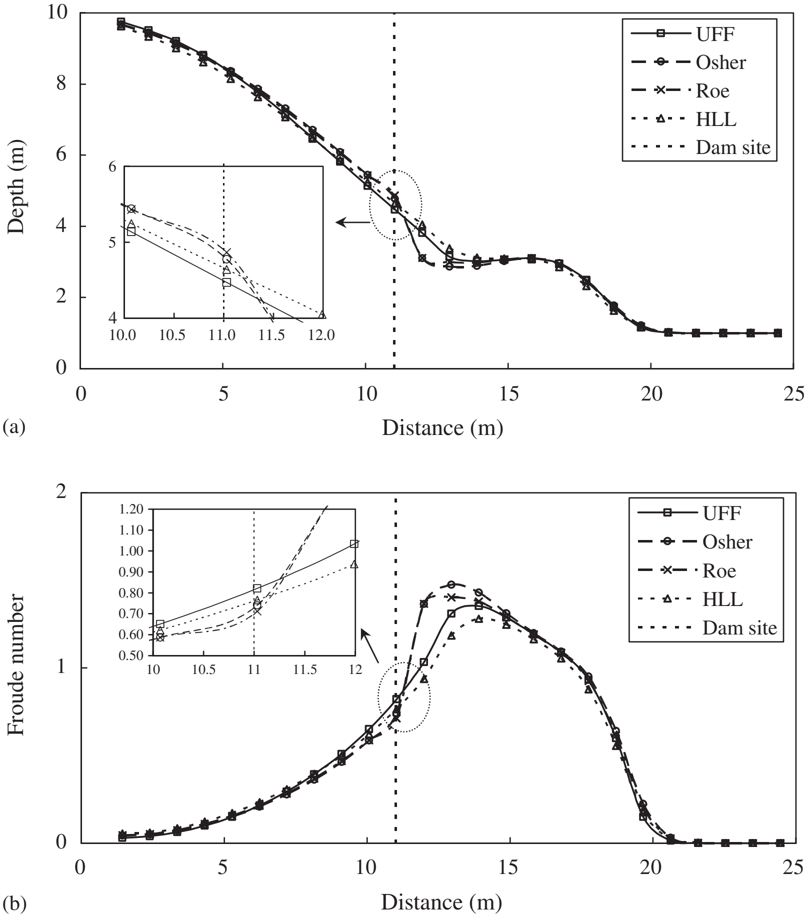


Figure 14. Comparisons of the: (a) simulated water depth profiles; and (b) Froude numbers along line ST (see Figure 13) using first-order schemes.

The initial condition comprises two regions of still water depth separated by the cylindrical dam, at which water depth inside the dam is 10 m and outside the dam is 1 m. The computational time step is 0.02 s. Figures 14(a) and 14(b) show, respectively, the simulated water depths and the Froude numbers using all presented first-order schemes along line ST at $t = 0.69$ s. The results show that the Osher and the Roe schemes produce significant glitches at the dam site whereas the proposed UFF and the HLL schemes do not. The glitch stands for the disadvantage of the Osher and the Roe schemes whereas the solution at the dam site should be a smooth water surface profile for rarefaction wave. Besides, the flow at the dam site is the critical flow [24], i.e. the exact solution of the Froude number (N_{F}) should be equal to one, which is very useful for comparison. From the close-up view of dam site in Figure 14(b), the simulated Froude number by the proposed UFF scheme produces the closest value to the exact solution (i.e. $N_{\mathrm{F}} = 1$). On the other hand, the CPU time required is 10.52 s

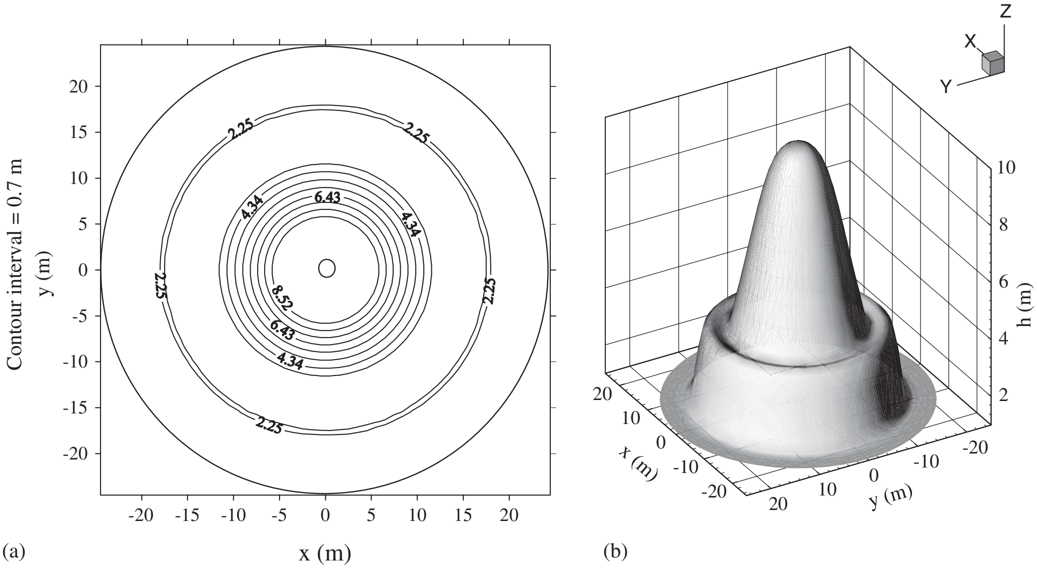


Figure 15. (a) The 2D contour plot; and (b) 3D free-surface view showing water depth variations for the circular dam breaking at $t = 0.69$ s using the UFF-MUSCL scheme.

Table III. Influence of the computational mesh on the simulated Froude number at the dam site for 2D circular dam-break problem.

Mesh	Number of cells in the tangential direction	Number of cells in the radial direction	UFF	Osher	Roe	HLL
M1	50	25	0.815	0.731	0.705	0.762
M2	100	50	0.892	0.841	0.831	0.857
M3	200	100	0.935	0.905	0.905	0.911
M4	300	150	0.965	0.942	0.938	0.956
M5	400	200	0.985	0.972	0.971	0.981

for the UFF scheme. The ratios of the CPU time consumed by the Osher, Roe and HLL schemes to that required by the UFF scheme are 1.25, 1.82, and 1.36, respectively. From the above comparisons, it is found that the UFF scheme is the most accurate and efficient among the schemes presented.

The 2D contour as well as the 3D view of the water surface elevation at $t = 0.69$ s computed by the second-order UFF-MUSCL scheme are shown in Figures 15(a) and 15(b), respectively. The simulated results show that there is an outward-propagating circular shock wave and an inward-propagating circular rarefaction wave. The perfect symmetric flow behaviour agrees very well with those found in References [8, 11, 14, 21].

To further demonstrate the numerical performance of the proposed UFF scheme, a grid convergence study known as the grid refinement study is performed herein [29]. Five different

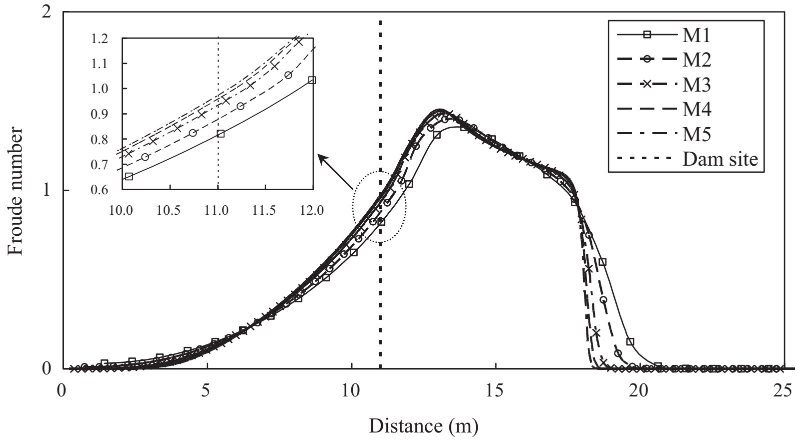


Figure 16. Influence of the computational mesh on simulated Froude number using the UFF scheme.

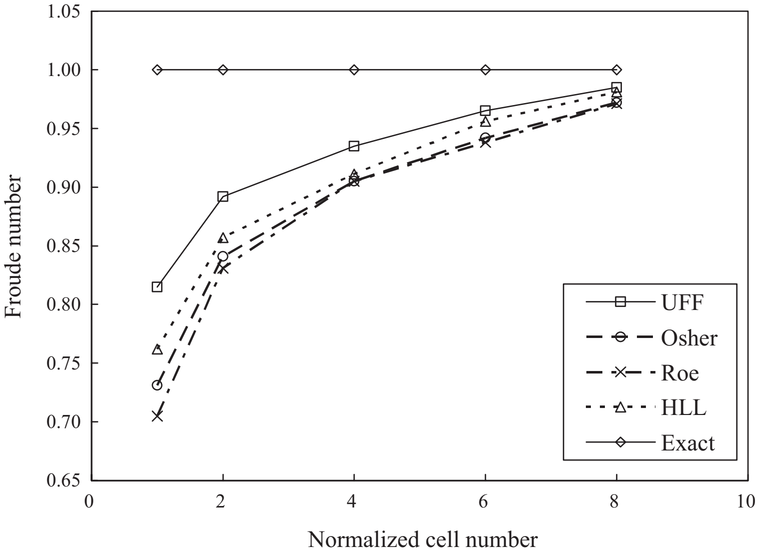


Figure 17. Comparison of the convergence curves using first-order schemes.

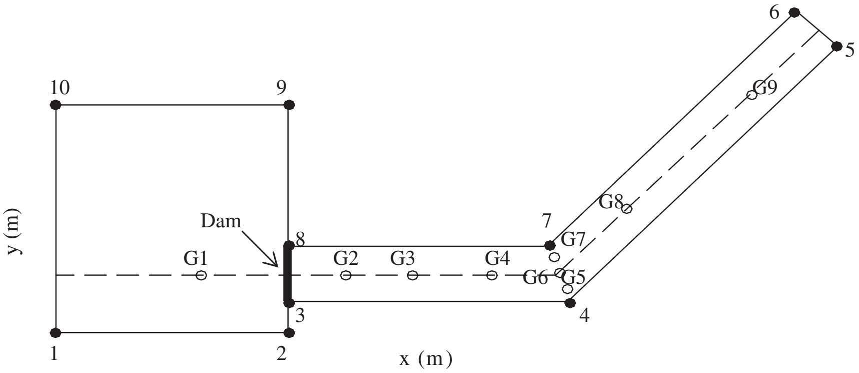
meshes (M1–M5), each with twice the number of cells in the tangential and the radial direction as the previous mesh, are considered. Table III lists the mesh information and the resulting Froude number simulated at the dam site by four first-order upwind schemes. Figure 16 shows the influence of the computational mesh on simulated Froude number using the UFF scheme. As the number of computational cells increases, the simulated Froude number approaches to the exact solution ($N_F = 1$) at the dam site. The results also show the UFF scheme performs the best solution at the dam site. The convergence curve with varying cell number is shows in Figure 17, where the number of cells is normalized by the cell number of the coarsest mesh (M1). Each solution is properly converged with respect to iterations and an almost mesh independent solution (i.e. $N_F = 1$) is achieved using the UFF scheme with the M5 mesh.

4.4. Dam-break experiment with 45° *bend channel*

This section tests the capability of the proposed UFF and its second-order extension UFF-MUSCL scheme in modelling 2D dam-break flow over a channel with a 45° bend. The dam-break experiment with a 45° bend channel was carried out by Sorares *et al.* [30] at the Catholic University of Louvain, Belgium; more details can be found in Toro's book [24]. The experimental domain consists of a rectangular reservoir connected to a channel containing a 45° bend, as shown in Figure 18.

Both the reservoir and the channel are horizontal and connected by a dam. The geometry of the experiment layout and the locations of the observation stations in the physical model are given in Figure 18. The initial water depths are 0.25 m in the reservoir and 0.01 m in the channel. The total simulation time is 45 s after dam break. Manning's roughness coefficient n_m for bed friction is calibrated to be 0.011. The computational mesh with 3950 non-rectangular cells is used.

Using the UFF scheme and its second-order UFF-MUSCL scheme, the comparisons of the simulated water depth hydrographs with the experimental data at different observation



Point number	x (m)	y (m)	Observations	x(m)	y (m)
1	0.0	0.0	G1	1.59	0.69
2	2.39	0.0	G2	2.74	0.69
3	2.39	0.445	G3	4.24	0.69
4	6.64	0.445	G4	5.74	0.69
5	9.575	3.38	G5	6.74	0.72
6	9.225	3.73	G6	6.65	0.80
7	6.435	0.94	G7	6.56	0.89
8	2.39	0.94	G8	7.07	1.22
9	2.39	2.44	G9	8.13	2.28
10	0.0	2.44			

Figure 18. The layout of the dam-break experiment with 45° bend channel.

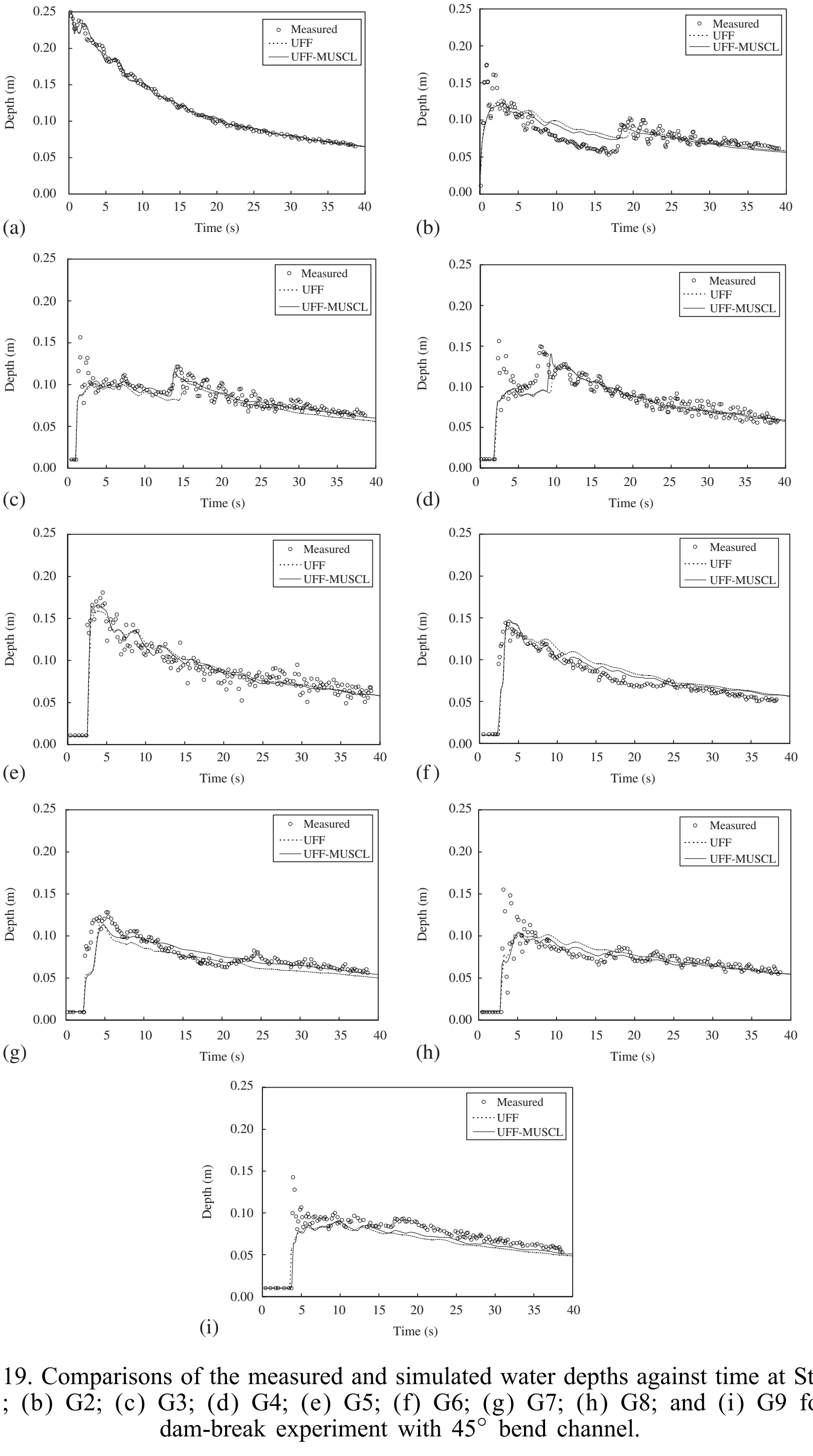


Figure 19. Comparisons of the measured and simulated water depths against time at Station: (a) G1; (b) G2; (c) G3; (d) G4; (e) G5; (f) G6; (g) G7; (h) G8; and (i) G9 for the dam-break experiment with 45° bend channel.

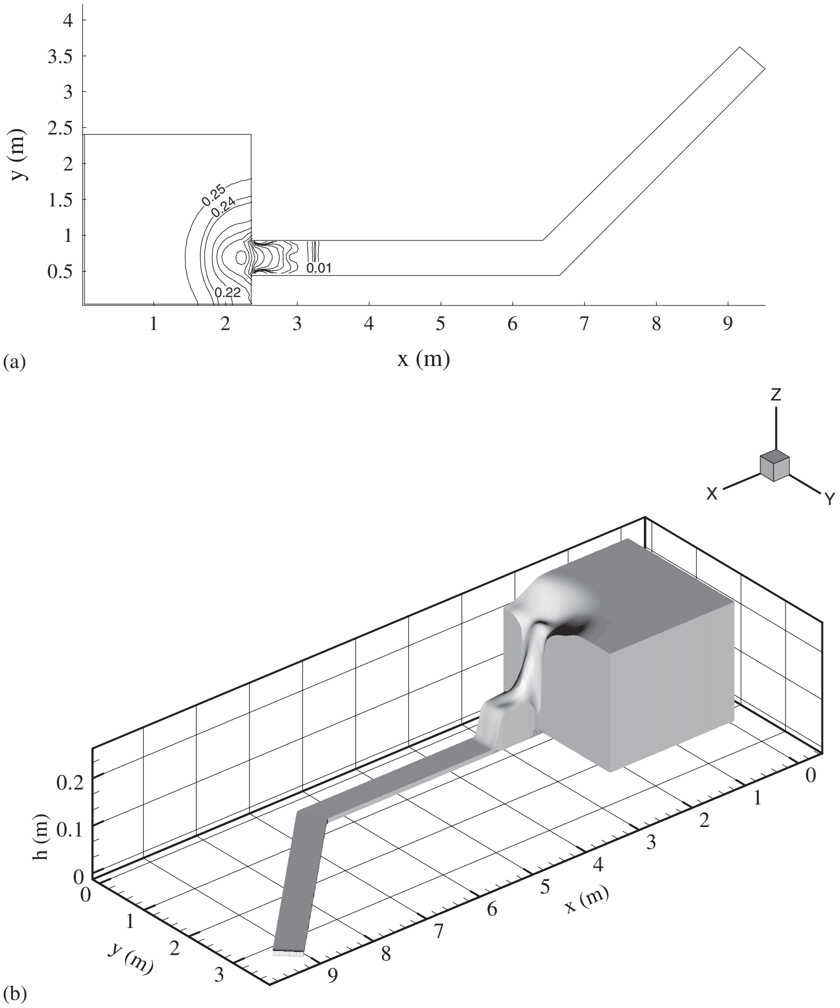


Figure 20. The simulated: (a) 2D contour plot; and (b) 3D free-surface view showing water depth variations for the dam-break experiment at $t = 0.5$ s by the UFF-MUSCL scheme.

stations are made in Figure 19. The results show shock waves propagating toward downstream channel and reflecting on the channel bend boundaries. The rarefaction wave travels into the reservoir. For all observation stations shown in Figure 19, the simulated water depths agree well with the measured. Although experimental data give remarkable oscillatory behaviours, the UFF and its second-order extension UFF-MUSCL schemes present satisfactory overall solutions. Obviously, the simulated results of the reflections from the bend by the UFF-MUSCL scheme are better captured, while the UFF scheme gives smoother solutions, typically shown in Figures 19(b), 19(c) and 19(d) for observation stations G2, G3 and G4, respectively. The simulated 2D contour plot and 3D free-surface view showing water depth variations at $t = 0.5$ s using the UFF-MUSCL scheme are shown in Figures 20(a) and 20(b), respectively.

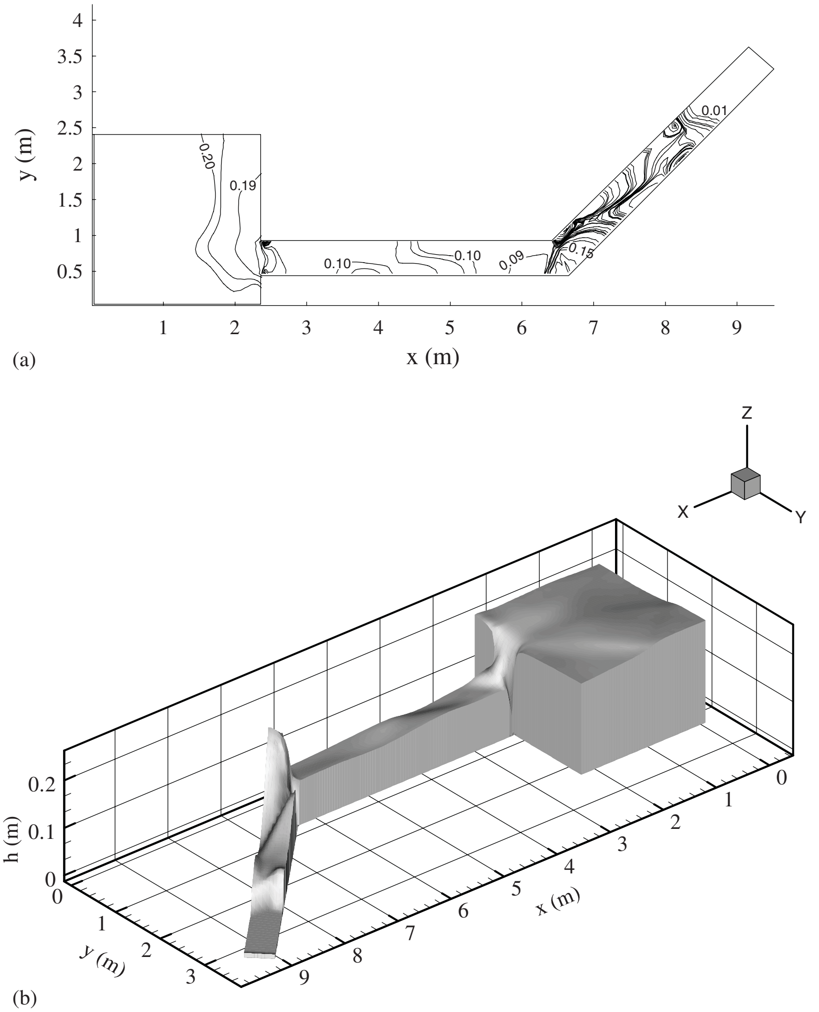


Figure 21. The simulated: (a) 2D contour plot; and (b) 3D free-surface view showing water depth variations for the dam-break experiment at $t = 4.0$ s by the UFF-MUSCL scheme.

Figure 21 shows the simulated water depth variations at $t = 4.0$s. By this moment the propagating shock wave front has reached the bend, and it has reflected and diffracted. It continues to travel along the downstream section after the corner of the bend. The application demonstrates the capability of the proposed scheme in modelling the complicated flow structure.

5. CONCLUSIONS

The artificially upstream flux vector splitting method developed by Sun and Takayama [22] is adopted to estimate the numerical flux through cell interface for the solution of the local

Riemann problem. By applying the corresponding numerical flux function, the upstream flux-splitting finite-volume (UFF) scheme is proposed for solving the 2D SWE in the framework of the FVM. The MUSCL method and the predictor–corrector approach are employed to achieve the second-order-accurate extension in space and in time, respectively.

The numerical performance of the proposed UFF scheme is compared with those of the Osher, Roe and HLL schemes for the 1D idealized dam break, 2D steady oblique hydraulic jump and 2D circular dam-break problems. Based on the simulated results of the 1D idealized dam-break problem, it is found that the UFF scheme presents superior overall numerical accuracy as well as efficiency because it produces the least L_2 error norm and consumes the smallest CPU time. Due to satisfaction of the entropy condition, the UFF scheme can resolve rarefaction wave smoothly and accurately at the dam site. Furthermore, the UFF scheme can simulate the wet/dry wave front passing over a dry bed condition downstream well.

According to the simulated results of the 2D oblique hydraulic jump problem, the UFF scheme also obtains superior overall numerical performances among the schemes tested. From the simulated results of the 2D circular dam-break problem, the UFF scheme presents the best numerical efficiency and provides good agreement with the results reported by other researchers [8, 11, 14, 21]. A grid convergence study is also tested for the 2D circular dam-break problem to conclude that the UFF scheme performs the best solution at the dam site. Moreover, the applications of the UFF and UFF-MUSCL schemes to the dam-break experiment with a 45° bend channel demonstrate the capability and reliability of dam-break flow simulation. From the above analyses, the proposed schemes are simple, accurate, efficient and suitable for modelling shallow water flows containing discontinuities.

REFERENCES

1. Hirsch C. *Numerical Computation of Internal and External Flows*, vol. 2. Wiley: Chichester, 1990.
2. Toro EF. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer: Berlin, 1997.
3. Roe PL. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics* 1981; **43**:357–372.
4. Osher S, Solomone F. Upwind difference schemes for hyperbolic systems of conservation laws. *Mathematics and Computers in Simulation* 1982; **38**:339–374.
5. Steger JL, Warming RF. Flux vector splitting of the inviscid gas dynamic equations with application to finite difference methods. *Journal of Computational Physics* 1981; **40**:263–293.
6. Zhao DH, Shen HW, Tabios GQ, Lai JS, Tan WY. Finite-volume two-dimensional unsteady-flow model flow river basins. *Journal of Hydraulic Engineering* 1994; **120**(12):863–882.
7. Wan Q, Wan H, Zhou C, Wu Y. Simulating the hydraulic characteristics of the lower Yellow River by the finite-volume technique. *Hydrological Processes* 2002; **16**(14):2767–2769.
8. Alcrudo F, Garcia-Navarro P. A high-resolution Godunov-type scheme in finite volumes for the 2D shallow water equations. *International Journal for Numerical Methods in Fluids* 1993; **16**:489–505.
9. Anastasiou K, Chan CT. Solution of the 2D shallow water equations using the finite volume method on unstructured triangular meshes. *International Journal for Numerical Methods in Fluids* 1997; **24**:1225–1245.
10. Sleigh PA, Gaskell PH, Berzins M, Wright NG. An unstructured finite-volume algorithm for predicting flow in rivers and estuaries. *Computers and Fluids* 1998; **27**(4):479–508.
11. Tseng MH. Explicit finite volume non-oscillatory schemes for 2D transient free-surface flows. *International Journal for Numerical Methods in Fluids* 1999; **30**:831–843.
12. Tseng MH, Chu CR. Two-dimensional shallow water flows simulation using TVD-MacCormack scheme. *Journal of Hydraulic Research* 2000; **38**(2):123–131.
13. Brufau P, Garcia-Navarro P. Two-dimensional dam break flow simulation. *International Journal for Numerical Methods in Fluids* 2000; **33**:35–57.
14. Mingham CG, Causon DM. High-resolution finite-volume method for shallow water flows. *Journal of Hydraulic Engineering* 1998; **124**(6):605–614.
15. Hu K, Mingham CG, Causon DM. A bore-capturing finite volume method for open-channel flows. *International Journal for Numerical Methods in Fluids* 1998; **28**:1241–1261.

16. Causon DM, Mingham CG, Ingram DM. Advances in calculation methods for supercritical flow in spillway channels. *Journal of Hydraulic Engineering* 1999; **125**(10):1039–1050.
17. Valiani A, Caleffi V, Zanni A. Case study: Malpasset Dam-break simulation using a two-dimensional finite volume method. *Journal of Hydraulic Engineering* 2002; **28**(5):460–472.
18. Zoppou C, Roberts S. Catastrophic collapse of water supply reservoirs in urban areas. *Journal of Hydraulic Engineering* 1999; **125**(7):686–695.
19. Zhao DH, Shen HW, Lai JS, Tabios GQ. Approximate Riemann solvers in FVM for 2D hydraulic shock wave modeling. *Journal of Hydraulic Engineering* 1996; **122**(12):692–702.
20. Lin GF, Lai JS, Guo WD. Finite-volume component-wise TVD schemes for 2D shallow water equations. *Advances in Water Resources* 2003; **26**(8):861–873.
21. Erduran KS, Kutija V, Hewett CJM. Performance of finite volume solutions to the shallow water equations with shock-capturing schemes. *International Journal for Numerical Methods in Fluids* 2002; **40**:1237–1273.
22. Sun M, Takayama K. An artificially upstream flux vector splitting scheme for the Euler equations. *Journal of Computational Physics* 2003; **189**:305–329.
23. Tan WY. *Shallow Water Hydrodynamics*. Elsevier: New York, 1992.
24. Toro EF. *Shock-capturing Methods for Free-surface Shallow Water Flows*. Wiley: New York, 2001.
25. Wada Y, Liou MS. An accurate and robust flux splitting scheme for shock and contact discontinuities. *SIAM Journal on Scientific Computing* 1997; **18**(3):633–657.
26. Stoker JJ. *Water Waves*: *Mathematical Theory with Applications*. Wiley-Interscience: Singapore, 1958.
27. LeVeque RJ. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press: Cambridge, U.K., 2002.
28. Hager WH, Schwalt M, Jimenez O, Chaudry MH. Supercritical flow near an abrupt wall deflection. *Journal of Hydraulic Research* 1994; **32**(1):103–118.
29. Roache PJ. *Verification and Validation in Computational Science and Engineering*. Hermosa Publishers: Albuquerque, New Mexico, 1998.
30. Sorares S, Sillen X, Zech Y. Dam-break flow through sharp bends. Physical models and 2d boltzmann model validation. European Commission, *IBSN 92-828-7108-8*, 1999.