Available online at www.sciencedirect.com

SCIENCE DIRECT®

Advances in Water Resources 26 (2003) 861–873

Advances in Water Resources

www.elsevier.com/locate/advwatres

Finite-volume component-wise TVD schemes for 2D shallow water equations

Gwo-Fong Lin [a,*], Jihn-Sung Lai [b], Wen-Dar Guo [a]

[a] Department of Civil Engineering, National Taiwan University, Taipei 10617, Taiwan
[b] Hydrotech Research Institute, National Taiwan University, Taipei 10617, Taiwan

Received 21 November 2002; received in revised form 31 March 2003; accepted 29 April 2003

Abstract

Four finite-volume component-wise total variation diminishing (TVD) schemes are proposed for solving the two-dimensional shallow water equations. In the framework of the finite volume method, a proposed algorithm using the flux-splitting technique is established by modifying the MacCormack scheme to preserve second-order accuracy in both space and time. Based on this algorithm, four component-wise TVD schemes, including the Liou–Steffen splitting (LSS), van Leer splitting, Steger–Warming splitting and local Lax–Friedrichs splitting schemes, are developed. These schemes are verified through the simulations of the 1D dam-break, the oblique hydraulic jump, the partial dam-break and circular dam-break problems. It is demonstrated that the proposed schemes are accurate, efficient and robust to capture the discontinuous shock waves without any spurious oscillations in the complex flow domains with dry-bed situation, bottom slope or friction. The simulated results also show that the LSS scheme has the best numerical accuracy among the schemes tested.

Keywords: Finite-volume component-wise TVD scheme; Shallow water equations; Flux-vector splitting; MacCormack scheme

1. Introduction

The two-dimensional (2D) shallow water equations (SWE) describe the free surface flows, such as dam-break flows and oblique hydraulic jumps in open channels. The system of SWE is a time-dependent set of non-linear partial differential equations of hyperbolic type. The numerical simulation of the SWE is rather difficult because it requires special considerations for achieving conservative property as well as solving discontinuity accurately without spurious oscillation [1,2]. Based on the important concept of the total variation diminishing (TVD), several high-resolution TVD schemes for the hyperbolic conservation laws have been proposed in the literature [3–7]. These schemes resolve sharp discontinuities without spurious oscillations in the vicinity of large gradients and do not introduce too much numerical dissipation.

In recent years, the high-resolution TVD schemes have been gradually adopted to solve the 2D SWE. For instance, Alcrudo and Garcia-Navarro [8] developed a high-resolution Godunov-type scheme in finite volume method to achieve second-order accuracy. Louaked and Hanich [9] employed the TVD Lax–Wendroff scheme and used the artificial compression method to deal with the abrupt changes in the open channel flows. Using the finite volume method, Tseng [10] constructed some high-resolution non-oscillatory shock-capturing schemes. Wang et al. [11] developed a finite-difference TVD scheme by combining first-order Roe scheme and second-order Lax–Wendroff scheme. Tseng and Chu [12] proposed a modified TVD-MacCormack scheme based on the finite volume method. The TVD schemes described above often make use of the first-order Roe scheme as a basis. Hence, these TVD schemes require the set of the eigenvalues of the Jacobian matrix. In addition, the Roe scheme may generate a non-physical expansion shock due to the violation of the entropy condition; therefore, when the Roe scheme is employed, the entropy fix is required for practical computations [1,2]. On the other hand, when the TVD-MacCormack

* Corresponding author. Tel.: +886-2-23629023; fax: +886-2-23631558.
E-mail address: gflin@ntu.edu.tw (G.-F. Lin).

0309-1708/03/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/S0309-1708(03)00075-7

scheme is applied to 2D problems, four possible combinations of the predictor–corrector steps should be cycled during the computation to avoid a bias provided by an eventual accumulation of errors [6,12].

More recently, Yu and Liu [13] proposed a two-step, component-wise TVD scheme for hyperbolic conservation laws. Compared with other TVD schemes mentioned above, this component-wise TVD scheme is simpler to construct because it does not require the characteristic decompositions. It does not also require additional treatments of the directional dependence in the predictor–corrector steps, while the TVD-MacCormack scheme [12] does. Furthermore, this component-wise TVD scheme essentially is second-order-accurate in both time and space, and it may be applied to SWE without any difficulty. In the present study, the two-step, component-wise TVD scheme will be adopted in a proposed algorithm for solving SWE.

In this paper, an algorithm is established by modifying the well-known MacCormack scheme and using the flux-splitting technique to estimate the numerical flux. Based on this algorithm, four first-order flux-vector splittings are used as a basis to construct four component-wise TVD schemes in the framework of the finite volume method. These four schemes are evaluated based on the simulations of the 1D dam-break, the oblique hydraulic jump, the partial dam-break and circular dam-break problems. The proposed schemes are also verified with a laboratory dam-break experiment to show their further applications.

2. Two-step component-wise TVD scheme

Consider the following one-dimensional (1D) scalar hyperbolic conservation law

$$\frac{\partial q}{\partial t} + \frac{\partial f(q)}{\partial x} = 0 \tag{1}$$

where q is a variable of scalar hyperbolic conservation law, and x and t represent the space and time, respectively. Eq. (1) can be approximated by a higher-order conservative scheme given as [4]

$$q_i^{n+1} = q_i^n - \tau(b_{i+1/2} - b_{i-1/2}) \tag{2}$$

where n is time index, i is space index, $\tau = \Delta t/\Delta x$ is the uniform mesh ratio, and $b_{i+1/2}$ is a high-order numerical flux. The $b_{i+1/2}$ can be regarded as the first-order upwind numerical flux ($f_{i+1/2}$) with an additional second-order antidiffusive flux term ($w_{i+1/2}$), i.e.,

$$b_{i+1/2} = f_{i+1/2} + \frac{1}{2} w_{i+1/2} \tag{3}$$

According to the technique of flux splitting, the flux $f(q)$ is decomposed into negative and positive fluxes as [6,7]

$$f(q) = f^+(q) + f^-(q) \tag{4}$$

Hence, the first-order upwind numerical flux at the cell interface $i + 1/2$ can be expressed as

$$f_{i+1/2} = f^+(q_i) + f^-(q_{i+1}) = f_i^+ + f_{i+1}^- \tag{5}$$

Based on the direction of wave propagation, the well-known MacCormack scheme was modified to establish a component-wise TVD scheme [13]. For right-moving waves (i.e., $f^- \equiv 0$ in Eq. (4)), the left-upwind predictor in the MacCormack scheme can be expressed as

$$\hat{q}_i^{n+1} = q_i^n - \tau(f_i - f_{i-1}) \tag{6}$$

where f_i is the numerical flux for the center of the cell i at time n, and the corrector has the following expression

$$q_i^{n+1} = \frac{1}{2}(q_i^n + \hat{q}_i^{n+1}) - \frac{\tau}{2}(\hat{f}_{i+1} - \hat{f}_i) \tag{7}$$

where $\hat{f}_i = f(\hat{q}_i^{n+1})$ as the predictor flux.

Likewise, for left-moving waves (i.e., $f^+ \equiv 0$ in Eq. (4)), the right-upwind predictor in the MacCormack scheme can be expressed as

$$\hat{q}_i^{n+1} = q_i^n - \tau(f_{i+1} - f_i) \tag{8}$$

and similarly the corrector has the following form

$$q_i^{n+1} = \frac{1}{2}(q_i^n + \hat{q}_i^{n+1}) - \frac{\tau}{2}(\hat{f}_i - \hat{f}_{i-1}) \tag{9}$$

The flux splitting expressed in Eq. (4) is needed for the coexistence of left-moving and right-moving waves. Substituting Eq. (6) into Eq. (7) and Eq. (8) into Eq. (9) and then rewriting the MacCormack scheme in the conservative form of Eq. (2), one can obtain the predictor–corrector steps as

$$\hat{q}_i^{n+1} = q_i^n - \tau(f_{i+1/2} - f_{i-1/2}) \tag{10}$$

$$q_i^{n+1} = q_i^n - \tau(b_{i+1/2} - b_{i-1/2}) \tag{11}$$

where $f_{i+1/2}$ is defined by Eq. (5) at time n and

$$b_{i+1/2} = f_{i+1/2} + \frac{1}{2}(w_{i+1/2}^+ - w_{i+1/2}^-) \tag{12a}$$

$$= f_i^+ + f_{i+1}^- + \frac{1}{2}(\hat{f}_{i+1}^+ - f_i^+ - f_{i+1}^- + \hat{f}_i^-) \tag{12b}$$

To ensure TVD constraints, a flux-limiter function can be added to the second-order antidiffusive term. Hence, the second-order numerical flux $b_{i+1/2}$ in Eq. (12) can be rewritten as

$$b_{i+1/2} = f_{i+1/2} + \frac{1}{2}[\phi(r_{i+1/2}^+)w_{i+1/2}^+ - \phi(r_{i+1/2}^-)w_{i+1/2}^-] \tag{13a}$$

$$w_{i+1/2}^+ = \hat{f}_{i+1}^+ - f_i^+, \quad w_{i+1/2}^- = f_{i+1}^- - \hat{f}_i^- \tag{13b}$$

where ϕ is the non-linear flux-limiter function that plays an important role to avoid spurious oscillations, and slope ratio is defined by

$$r^{+}_{i+1/2} = \frac{w^{+}_{i-1/2}}{w^{+}_{i+1/2}}, \quad r^{-}_{i+1/2} = \frac{w^{-}_{i+3/2}}{w^{-}_{i+1/2}} \tag{13c}$$

Apparently, when $\phi(r) = 1$ and $\phi(r) = r$, the component-wise TVD scheme expressed in Eqs. (10)–(13) can be switched to the MacCormack scheme and the Warming–Beam scheme, respectively [13]. The component-wise TVD scheme coupled with the flux-splitting technique does not require the Jacobian matrix for system of conservation laws; hence its programming is rather simple. The applications of the component-wise TVD scheme to SWE should not cause any additional difficulty.

3. Solving 2D shallow water equations

3.1. Governing equations

The 2D SWE expressed in the differential conservative form are [1,2]

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = \mathbf{S} \tag{14}$$

in which

$$\mathbf{Q} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}; \quad \mathbf{F} = \begin{bmatrix} hu \\ hu^2 + \frac{gh^2}{2} \\ huv \end{bmatrix};$$
$$\mathbf{G} = \begin{bmatrix} hv \\ huv \\ hv^2 + \frac{gh^2}{2} \end{bmatrix}; \quad \mathbf{S} = \begin{bmatrix} 0 \\ gh(s_{0x} - s_{\mathrm{f}x}) \\ gh(s_{0y} - s_{\mathrm{f}y}) \end{bmatrix} \tag{15}$$

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; 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; $\mathbf{S}$ is a source term containing the bed friction and slope; $s_{\mathrm{f}x}$ 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.

3.2. Numerical discretization by finite volume method

A cell-centred finite volume method is adopted in this paper. Applying the divergence theorem, Eq. (14) can be integrated over the control volume Ω and expressed as

$$\iint_{\Omega} \frac{\partial \mathbf{Q}}{\partial t}\,\mathrm{d}W + \int_{\partial\Omega} \mathbf{E}\cdot\mathbf{n}\,\mathrm{d}l = \iint_{\Omega} \mathbf{S}\,\mathrm{d}W \tag{16}$$

in which $\mathbf{n}$ is the outward unit vector normal to the boundary $\partial\Omega$; $\mathrm{d}W$ and $\mathrm{d}l$ are the area and arc elements, respectively; and the integrand $\mathbf{E}\cdot\mathbf{n}$ is the normal flux vector in which $\mathbf{E} = [\mathbf{F}, \mathbf{G}]^{\mathrm{T}}$.

Assuming that the vector quantity $\mathbf{Q}$ is constant over a cell and can represent the average at the center of the cell, Eq. (16) can be discretized into the basic equation of the FVM as follows:

$$A\frac{\mathrm{d}\mathbf{Q}}{\mathrm{d}t} + \sum_{m=1}^{M} \mathbf{E}^{m}_{\mathrm{n}} L^{m} = A\mathbf{S} \tag{17}$$

where A is the area of a cell, m is the index for the side of a cell, M is the total number of the sides for a cell, $\mathbf{E}^{m}_{\mathrm{n}}$ is the intercell flux normal to each side m, and L^{m} is the length of side m. Applying the rotational invariance of the governing equations [14], the intercell flux normal to each side is defined as

$$\begin{aligned}\mathbf{E}_{\mathrm{n}}(\mathbf{Q}) &= \mathbf{F}\cos\Phi + \mathbf{G}\sin\Phi = \mathbf{T}(\Phi)^{-1}\mathbf{F}[\mathbf{T}(\Phi)\mathbf{Q}] \\ &= \mathbf{T}(\Phi)^{-1}\mathbf{F}(\overline{\mathbf{Q}})\end{aligned} \tag{18}$$

where Φ is the angle measured counterclockwise from x-axis between the outward unit vector $\mathbf{n}$ and x-axis; $\overline{\mathbf{Q}} = \mathbf{T}(\Phi)\mathbf{Q}$ is the vector variables transformed from $\mathbf{Q}$; $\mathbf{F}(\overline{\mathbf{Q}})$ is the transformed normal flux; and $\mathbf{T}(\Phi)$ and $\mathbf{T}(\Phi)^{-1}$ are the transformation and inverse transformation matrices, respectively, which are defined as

$$\mathbf{T}(\Phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Phi & \sin\Phi \\ 0 & -\sin\Phi & \cos\Phi \end{bmatrix},$$
$$\mathbf{T}(\Phi)^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Phi & -\sin\Phi \\ 0 & \sin\Phi & \cos\Phi \end{bmatrix} \tag{19}$$

Using the relation expressed in Eq. (18), Eq. (17) becomes

$$A\frac{\mathrm{d}\mathbf{Q}}{\mathrm{d}t} + \sum_{m=1}^{M} \mathbf{T}(\Phi)^{-1}\mathbf{F}(\overline{\mathbf{Q}})L^{m} = A\mathbf{S} \tag{20}$$

The objective of this study is to develop finite-volume component-wise TVD schemes for estimating $\mathbf{F}(\overline{\mathbf{Q}})$ in Eq. (20).

3.3. Estimation of cell interface flux $\mathbf{F}(\overline{\mathbf{Q}})$

The estimation of the normal flux $\mathbf{F}(\overline{\mathbf{Q}})$ at each cell interface is required for solving Eq. (20). The normal flux $\mathbf{F}(\overline{\mathbf{Q}})$ can be taken as the numerical flux corresponding to the local 1D Riemann problem in the direction normal to the cell interface. The local 1D Riemann problem is an initial-value problem and is written as [15]

$$\frac{\partial \overline{\mathbf{Q}}}{\partial t} + \frac{\partial[\mathbf{F}(\overline{\mathbf{Q}})]}{\partial \bar{x}} = 0 \tag{21a}$$

with

$$\overline{\mathbf{Q}}(\bar{x}, 0) = \begin{cases} \overline{\mathbf{Q}}_{\mathrm{L}}, & \bar{x} < 0 \\ \overline{\mathbf{Q}}_{\mathrm{R}}, & \bar{x} > 0 \end{cases} \tag{21b}$$

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

As illustrated in Fig. 1, the $\bar{x}$ axis with its origin located at the midpoint of interface of two cells is directed to outward normal. The $\mathbf{F}(\overline{\mathbf{Q}})$ is set to be a normal outward flux at the origin of a local axis $\bar{x}$. The quantities, $\overline{\mathbf{Q}}_{\rm L}$ and $\overline{\mathbf{Q}}_{\rm R}$, are the left and right transformed quantities at the center of the cell interface, respectively. In this paper, the flux-vector splitting is used to estimate the numerical fluxes. The normal outward flux $\mathbf{F}(\overline{\mathbf{Q}})$ can be decomposed into positive and negative parts as

$$\mathbf{F}(\overline{\mathbf{Q}}) = \mathbf{F}^{+} + \mathbf{F}^{-} \tag{22}$$

where $\mathbf{F}^{+}$ and $\mathbf{F}^{-}$ are the positive and negative split fluxes, respectively.

Accordingly, the first-order numerical flux $\mathbf{F}_{\rm LR}$ at each cell interface is estimated using the variables of the neighboring cells (i.e., $\overline{\mathbf{Q}}_{\rm L}$ and $\overline{\mathbf{Q}}_{\rm R}$) as

$$\mathbf{F}_{\rm LR} = \mathbf{F}^{+}(\overline{\mathbf{Q}}_{\rm L}) + \mathbf{F}^{-}(\overline{\mathbf{Q}}_{\rm R}) = \mathbf{F}_{\rm L}^{+} + \mathbf{F}_{\rm R}^{-} \tag{23}$$

where $\mathbf{F}_{\rm L}^{+}$ and $\mathbf{F}_{\rm R}^{-}$ are the left and the right split fluxes of the cell interface.

3.4. Second-order component-wise TVD schemes

In this section, higher-order spatial and time accuracy is achieved by employing the two-step component-wise TVD scheme described in Section 2. This second-order component-wise TVD scheme is extended in the framework of the finite volume method.

According to the two-step algorithm expressed in Eqs. (10) and (11), the finite-volume component-wise TVD scheme with source terms is formulated to preserve second-order accuracy in both time and space and it is expressed as

$$\widehat{\mathbf{Q}}_{i,j} = \mathbf{Q}_{i,j}^{n} - \frac{\Delta t}{A}\left[\sum_{m=1}^{M}\mathbf{T}(\Phi)^{-1}\mathbf{F}^{(1)}(\overline{\mathbf{Q}})L^{m}\right]_{i,j} + \Delta t\mathbf{S}_{i,j}^{n} \tag{24}$$

$$\mathbf{Q}_{i,j}^{n+1} = \mathbf{Q}_{i,j}^{n} - \frac{\Delta t}{A}\left[\sum_{m=1}^{M}\mathbf{T}(\Phi)^{-1}\mathbf{F}^{(2)}(\overline{\mathbf{Q}})L^{m}\right]_{i,j} + \Delta t\widehat{\mathbf{S}}_{i,j} \tag{25}$$

where i and j are the space index; n is the time index; Δt is the time increment; $\widehat{\mathbf{Q}}_{i,j}$ and $\widehat{\mathbf{S}}_{i,j}$ are the predicted variables and source terms for the cell (i,j), respectively; $\mathbf{Q}_{i,j}^{n}$ and $\mathbf{S}_{i,j}^{n}$ are the conserved variables and source terms at time index n, respectively; $\mathbf{F}^{(1)}(\overline{\mathbf{Q}})$ is the first-order numerical flux at time index n; $\mathbf{F}^{(2)}(\overline{\mathbf{Q}})$ is the second-order numerical flux taking the predicted variables into account. The computational cells in the x–y coordinate system are illustrated in Fig. 2.

In Eq. (24), the first-order numerical flux though the cell interface $(i+1/2, j)$ is estimated by

$$\mathbf{F}^{(1)}(\overline{\mathbf{Q}}) = \mathbf{F}_{\rm LR} = \mathbf{F}_{\rm L}^{+} + \mathbf{F}_{\rm R}^{-} = \mathbf{F}_{i,j}^{+} + \mathbf{F}_{i+1,j}^{-} \tag{26}$$

In Eq. (25), the second-order numerical flux through the cell interface $(i+1/2, j)$ is estimated by adding the antidiffusive term with the flux limiter function to $\mathbf{F}_{\rm LR}$. According to Eq. (13), it is formulated as

$$\mathbf{F}^{(2)}(\overline{\mathbf{Q}}) \equiv \mathbf{F}_{\rm LR} + \frac{1}{2}[\phi(r_{i+1/2,j}^{+})w_{i+1/2,j}^{+} - \phi(r_{i+1/2,j}^{-})w_{i+1/2,j}^{-}] \tag{27a}$$

$$r_{i+1/2,j}^{+} = \frac{w_{i-1/2,j}^{+}}{w_{i+1/2,j}^{+}}; \quad r_{i+1/2,j}^{-} = \frac{w_{i+3/2,j}^{-}}{w_{i+1/2,j}^{-}} \tag{27b}$$

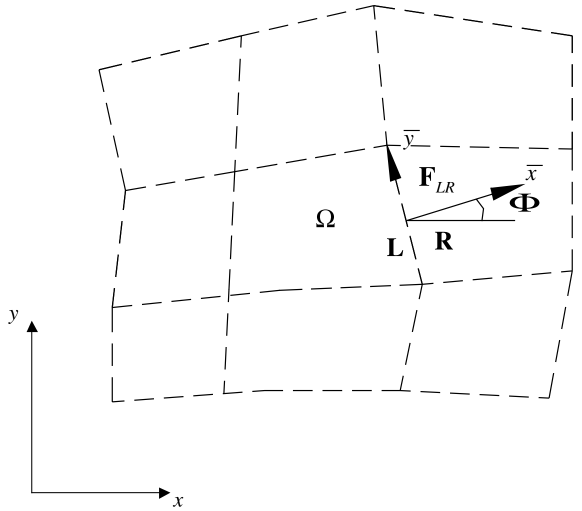

Fig. 1. The finite volume Ω in the x–y coordinates.

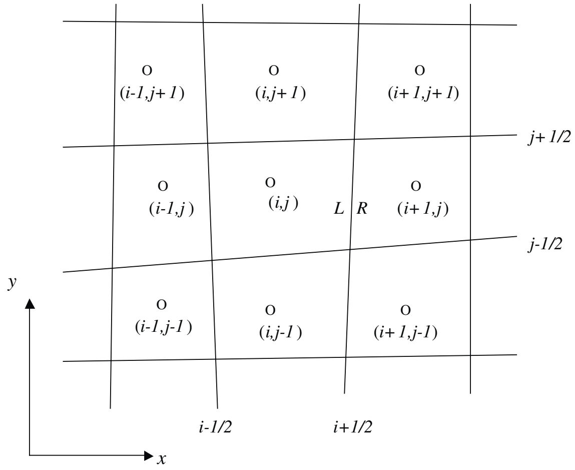

Fig. 2. The computational cells in the x–y coordinates.

$$w^+_{i+1/2,j} = \widehat{\mathbf{F}}^+_{i+1,j} - \mathbf{F}^+_{i,j}, \quad w^-_{i+1/2,j} = \mathbf{F}^-_{i+1,j} - \widehat{\mathbf{F}}^-_{i,j} \tag{27c}$$

where $\widehat{\mathbf{F}}_{i,j} = \mathbf{F}(\widehat{\mathbf{Q}}_{i,j})$ is regarded as the predictor flux. Several forms of limiter function $\phi(r)$ have been proposed [6]. In this paper, the van Leer limiter is adopted and it is defined as

$$\phi(r) = \frac{r + |r|}{1 + |r|} \tag{28}$$

Regarding Eqs. (24) and (25), the predictor is obtained through estimating the first-order numerical flux at each cell interface. Then, in the corrector step the second-order numerical flux is calculated by adding the second-order antidiffusive term to the first-order one. Essentially, the second-order component-wise TVD scheme does not require the set of eigenvalues of the Jacobian matrix. Nevertheless, the technique of the flux splitting is needed for its two-step procedures. In this paper, four flux-vector splittings are employed and described below, including the Liou–Steffen splitting (LSS), van Leer splitting (VLS), Steger–Warming splitting (SWS) and local Lax–Friedrichs splitting (LLFS).

3.4.1. Liou–Steffen splitting

The flux-vector splitting, proposed by Liou and Steffen [7,16] for the system of the Euler equations in gas dynamics, is adopted herein to solve SWE. Splitting the flux vector $\mathbf{F}$ into a convective component $\mathbf{F}^{(c)}$ and a pressure component $\mathbf{F}^{(p)}$, $\mathbf{F}$ can be expressed as

$$\mathbf{F} = \begin{bmatrix} hu_n \\ hu_n^2 + \frac{gh^2}{2} \\ hu_n v_t \end{bmatrix} = \begin{bmatrix} hu_n \\ hu_n^2 \\ hu_n v_t \end{bmatrix} + \begin{bmatrix} 0 \\ p \\ 0 \end{bmatrix} \equiv \mathbf{F}^{(c)} + \mathbf{F}^{(p)} \tag{29}$$

where $p = gh^2/2$ is the hydrostatic pressure. By introducing the Froude number ($F_r = u_n/c$), the convective flux vector is rewritten as

$$\mathbf{F}^{(c)} = F_r \begin{bmatrix} hc \\ hu_n c \\ hv_t c \end{bmatrix} \tag{30}$$

where $c = \sqrt{gh}$. According to the technique of the flux splitting, the first-order numerical flux is expressed as

$$\mathbf{F}_{LR} = \mathbf{F}^{(c)}_{LR} + \mathbf{F}^{(p)}_{LR} \tag{31}$$

$$\begin{aligned} \mathbf{F}^{(c)}_{LR} &= F^+_{rL} \begin{bmatrix} hc \\ hu_n c \\ hv_t c \end{bmatrix}_L + F^-_{rR} \begin{bmatrix} hc \\ hu_n c \\ hv_t c \end{bmatrix}_R, \\ \mathbf{F}^{(p)}_{LR} &= \begin{bmatrix} 0 \\ p^+_L \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ p^-_R \\ 0 \end{bmatrix} \end{aligned} \tag{32}$$

The splittings of the Froude number and hydrostatic pressure flux component are required in Eq. (32). They can be expressed in terms of second-order polynomials $(F_r \pm 1)^2$ as [16]

$$F^\pm_r = \begin{cases} \pm\frac{1}{4}(F_r \pm 1)^2, & \text{if } |F_r| \leqslant 1 \\ \frac{1}{2}(F_r \pm |F_r|), & \text{otherwise} \end{cases} \tag{33}$$

$$p^\pm = \begin{cases} \frac{p}{4}(F_r \pm 1)^2(2 \mp F_r), & \text{if } |F_r| \leqslant 1 \\ \frac{p}{2}(F_r \pm |F_r|)/F_r, & \text{otherwise} \end{cases} \tag{34}$$

The algorithm of Eqs. (24)–(27) coupled with the Liou–Steffen splitting is referred to as the LSS scheme herein.

3.4.2. van Leer splitting

The van Leer splitting (VLS) [7,17] is based on the flow conditions. For subcritical flow, $|F_r| \leqslant 1$, the splitting of flux vector $\mathbf{F}$ is derived as follows [1]

$$\mathbf{F}^\pm = \begin{bmatrix} \mathbf{F}^\pm_1 \\ \mathbf{F}^\pm_1(u_n \pm 2c)/2 \\ \mathbf{F}^\pm_1 v_t \end{bmatrix} \tag{35}$$

where $\mathbf{F}^\pm_1 = \pm hc(F_r \pm 1)^2/4$. For supercritical flow ($F_r > 1$), $\mathbf{F}^+ = \mathbf{F}$ and $\mathbf{F}^- = 0$; whereas if $F_r < -1$, $\mathbf{F}^+ = 0$ and $\mathbf{F}^- = \mathbf{F}$ [1]. The proposed algorithm of Eqs. (24)–(27) coupled with the VLS is denoted as the VLS scheme herein.

3.4.3. Steger–Warming splitting

For the SWE, the split fluxes by Steger–Warming [18] can be obtained according to the flow conditions. For subcritical flow, we have

$$\mathbf{F}^+ = \frac{h}{4}\begin{bmatrix} 3u_n + c \\ 2u_n^2 + (u_n + c)^2 \\ 2u_n v_t + (u_n + c)v_t \end{bmatrix}; \quad \mathbf{F}^- = \frac{h}{4}\begin{bmatrix} u_n - c \\ (u_n - c)^2 \\ (u_n - c)v_t \end{bmatrix} \tag{36}$$

For supercritical flow, we have $\mathbf{F}^+ = \mathbf{F}$ and $\mathbf{F}^- = 0$. The algorithm of Eqs. (24)–(27) coupled with the Steger–Warming splitting is referred to as the SWS scheme herein.

3.4.4. Local Lax–Friedrichs splitting

For the Lax–Friedrichs splitting [2], the estimation of the numerical fluxes simply depends on the maximum (absolute value) wave speed locally. The split fluxes for Eq. (21) can be expressed as [13]

$$\mathbf{F}^\pm(\overline{\mathbf{Q}}) = \frac{1}{2}[\mathbf{F}(\overline{\mathbf{Q}}) \pm \lambda_{\max}\overline{\mathbf{Q}}] \tag{37}$$

where $\lambda_{\max} = \max[|\lambda_k(\overline{\mathbf{Q}}_R)|, |\lambda_k(\overline{\mathbf{Q}}_L)|]$ is the maximum wave speed. The eigenvalues, λ_k ($k = 1$, 2 and 3), are obtained from Eq. (21a), which are $\lambda_1 = u_n$, $\lambda_2 = u_n + c$, and $\lambda_3 = u_n - c$ [7]. The proposed algorithm of Eqs. (24)–(27) coupled with the local Lax–Friedrichs splitting is denoted herein as the LLFS scheme.

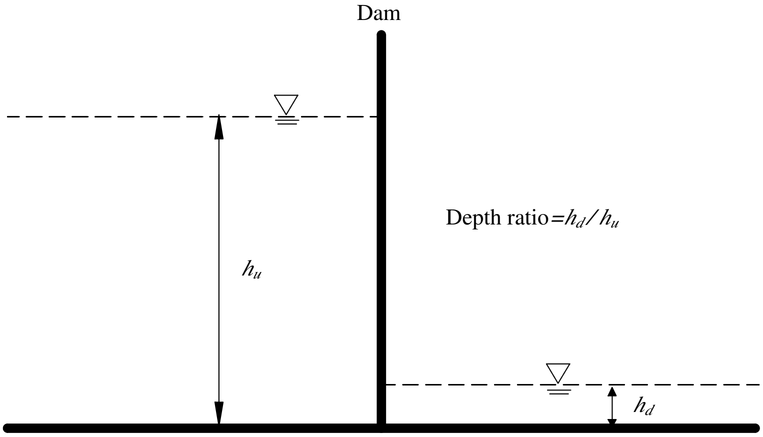

Fig. 3. A schematic representation of 1D dam-break problem.

3.5. Stability and boundary conditions

In order to ensure numerical stability of the proposed schemes, the time step Δt must be restricted by the Courant–Friedrichs–Lewy (CFL) stability condition [19] which is expressed as

$$\mathrm{CFL} = \frac{\Delta t}{\Delta d}\left|\frac{\Delta \mathbf{F}(\overline{\mathbf{Q}})}{\Delta \overline{\mathbf{Q}}}\right|_{\max} \leqslant 1 \tag{38}$$

where Δd is the distance between the centers of two adjacent cells.

The boundary conditions used in this paper are divided into two different types, namely the closed boundary and the open boundary. For the closed boundary condition, the velocity normal to the land is set to zero. For the open boundary condition, the velocity and depth are needed using the Riemann invariants [15].

4. Numerical results and discussions

The numerical accuracy and the computational efficiency are investigated to evaluate the performance of four proposed schemes. Four cases in the open-channel flows with discontinuities are tested, including the 1D dam-break, oblique hydraulic jump, partial dam-break and circular dam-break problems. In all of the test cases, zero bed slope is given and no bed friction is assumed in the simulation domain. Computations for the problems presented below were performed on a Pentium IV PC equipped with a 256 megabyte RAM.

4.1. 1D dam-break flow

The proposed schemes are first applied to 1D dam-break problem. The 1D dam-break problem is chosen to test the shock-capturing ability of the proposed schemes because it serves as a classical example with exact solutions to verify the numerical model. The dam-break problem is illustrated in Fig. 3. A horizontal channel with 2000 m in length and 10 m in width is considered. The dam is located at 1000 m downstream of the channel inlet. The initial upstream and downstream water depths, h_u and h_d, are 10 and 0.05 m respectively. Thus the water depth ratio $h_\mathrm{d}/h_\mathrm{u}$ is 0.005. The exact solutions can be found in the literature [20]. The CFL number is set to be 0.9, and the flow domain is discretized into 100 uniform cells.

Comparisons of exact solutions with simulated depths as well as velocities at $t = 50$ s using the four schemes are presented in Fig. 4. As shown in Fig. 4(a) with the close-up of the shock wave, all schemes can predict the shock wave and the rarefaction wave without spurious oscillation. Obviously, the resolutions of the shock wave using the LSS and VLS schemes are identical, and they are better than the other two schemes. Among four proposed schemes, the LLFS scheme presents relatively dissipative results. Similarly, Fig. 4(b) shows the same tendency for the velocity of shock wave.

To evaluate the numerical performance quantitatively, Table 1 summarizes the relative error in L_2 norm and the computational (CPU) time. The L_2 norm is defined as [21]

$$L_2 = \sqrt{\frac{\sum\left(Y_{i,j}^{\mathrm{sim}} - Y_{i,j}^{\mathrm{exact}}\right)^2}{\sum\left(Y_{i,j}^{\mathrm{exact}}\right)^2}} \tag{39}$$

where, $Y_{i,j}^{\mathrm{sim}}$ and $Y_{i,j}^{\mathrm{exact}}$ are the simulated solution and the exact solution at cell (i,j), respectively. Table 1 shows that the LSS scheme yields the smallest L_2 norm among the four schemes, whereas the LLFS gives the largest one. Table 1 also shows that all schemes consume almost the same CPU time. From the simulated results presented above, we can conclude that the LSS scheme has the best numerical accuracy for solving the 1D dam-break problem.

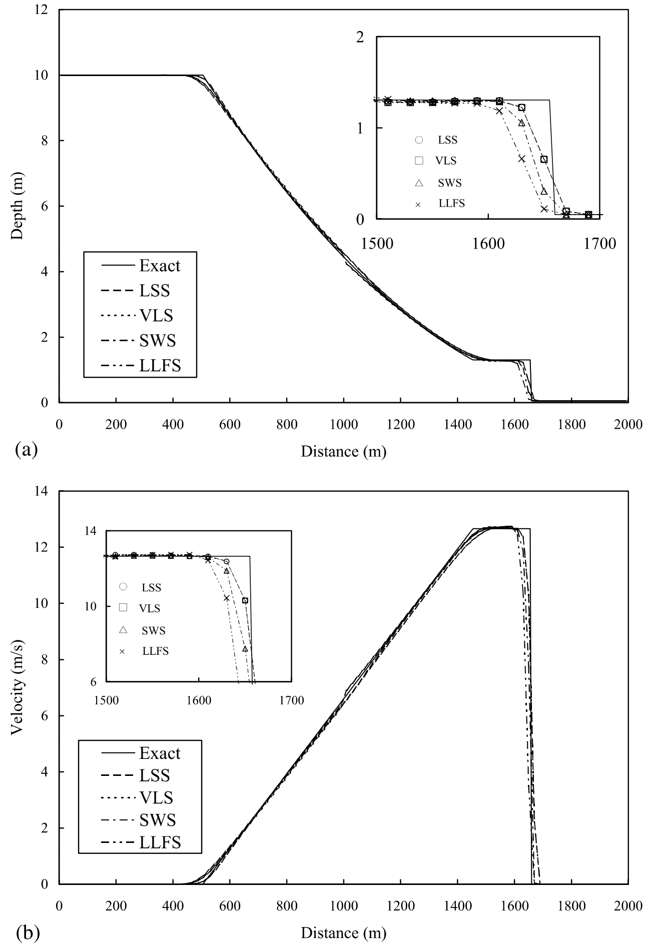

Fig. 4. Comparisons of exact solutions with simulated (a) water depths and (b) velocities using the four schemes for a water depth ratio h_d/h_u of 0.005.

Table 1
The L_2 norm and CPU time for 1D dam-break problem

Scheme	L_2 norm of depth	L_2 norm of velocity	CPU time (s)
LSS	0.011	0.050	0.99
VLS	0.014	0.052	0.98
SWS	0.018	0.077	1.01
LLFS	0.024	0.147	0.98

Due to the fact that the MacCormack scheme incorporates forward–backward differences in separated predictor–corrector steps, various combinations for directional difference steps can be found for 1D or 2D problems [6]. Different combinations can lead to results with quite distinctive differences [6]. In the case study of 1D dam-break flow, Tseng and Chu [22] switched forward and backward differences in the cyclic manner as an attempt to avoid the accumulation of errors. However, they found the accumulation of initial errors caused by the directional dependence could not be eliminated simply by the cyclic manners.

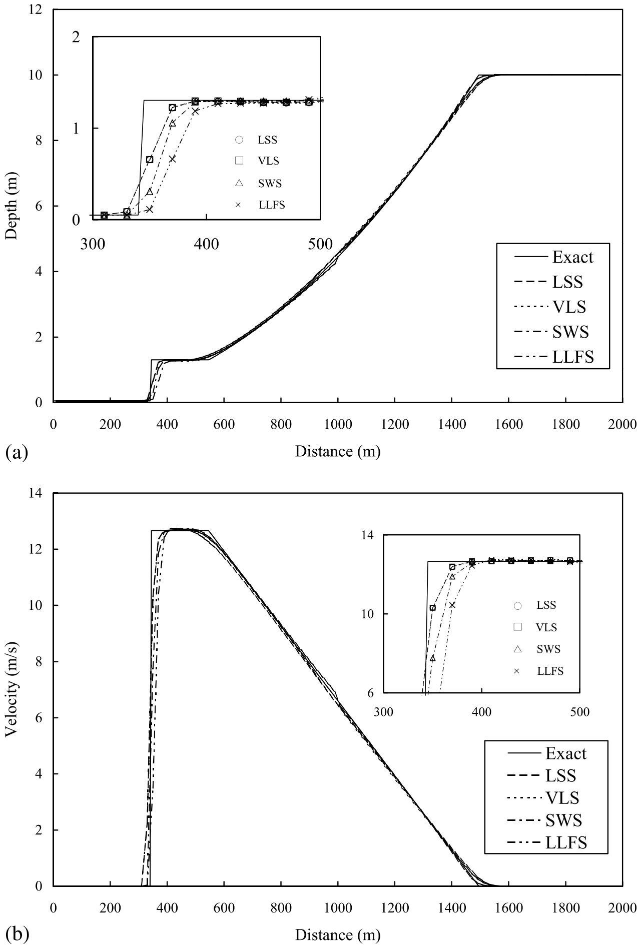

Fig. 5. Comparisons of exact solutions with simulated (a) water depths and (b) velocities using the four schemes for a water depth ratio h_d/h_u of 200.

Fortunately, the proposed schemes can avoid this accumulation of errors due to the use of the flux-splitting technique. To demonstrate the robustness of the proposed schemes, the 1D dam-break flow with a depth ratio of 200 is simulated. This test is similar to the previous case but with reversed shock-wave direction. As shown in Fig. 5, the shock wave propagates from right to left, and apparently the results are exactly the same as those shown in Fig. 4 except the propagating direction.

To show the effect of computational grid size on the numerical accuracy, the 400 computational cells is considered herein. Fig. 6 depicts the comparisons of exact solutions with simulated depths at $t = 50$ s. From the simulated results shown in Fig. 6, the proposed schemes using 400 cells capture the shock fronts steeper and less dissipative than those shown in Fig. 4(a). Moreover, the commonly used second-order Roe scheme with MUSCL approach [2,6–8,23] is applied herein to compare with the proposed schemes. As shown in Fig. 6, the simulated shock front profile by the Roe-MUSCL scheme is very close to that by LSS or VLS schemes.

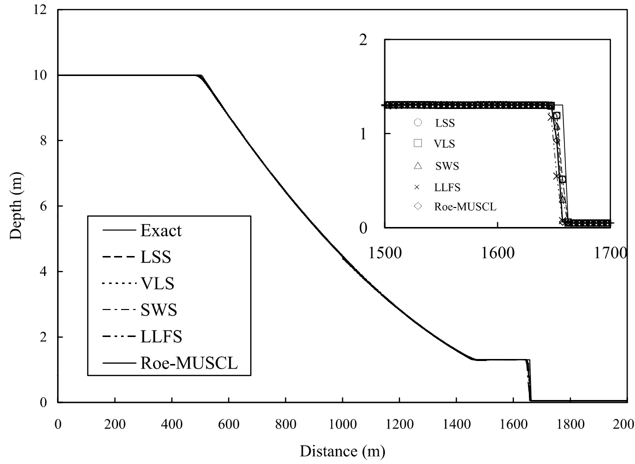

Fig. 6. Comparisons of exact solutions with simulated water depths using the 400 computational cells.

To demonstrate the proposed schemes are capable of simulating flow over a dry bed, the dry bed dam-break problem is tested herein. The initial upstream and downstream water depths, h_u and h_d, are 10 and 0 m, respectively. The CFL number is set to be 0.8, and the flow domain is discretized into 400 uniform cells. Accordingly, an absolute zero depth in the dry bed can cause mathematical problems [2]. To avoid this, an almost negligible water depth of 0.00001 m is assumed at the downstream of the dam. Comparisons of exact solution with simulated depth at $t = 30$ s are presented in Fig. 7. All schemes can predict the rarefaction wave quite well. Form the close-up of the dry/wet fronts shown in Fig. 7, the LSS and VLS schemes give better results than those by SWS and LLFS schemes.

4.2. Oblique hydraulic jump

The steady supercritical flow in converging channels, namely the oblique hydraulic jump (also called the oblique shock wave), is simulated to test the proposed schemes. As illustrated in Fig. 8, an oblique hydraulic jump is generated when a converging vertical boundary is deflected through an angle δ inward the supercritical flow. Due to the contraction of channel geometry, the oblique hydraulic jump originating at point O is formed with an angle of β by producing an abrupt increase in water depth. The simulation conditions used herein are the same as those in the literature [8,10,12]. The angle between the converging wall and the flow direction is set to be $\delta = 8.95°$. The initial conditions corresponding to a Froude number of 2.74 are given as: the water depth of 1 m, the velocity component u of 8.57 m/s and v of zero. The supercritical flow boundary conditions of $h = 1$ m, $u = 8.57$ m/s and $v = 0$ m/s are given at the upstream boundary. The transmissive boundary conditions are imposed at the downstream boundary [2]. The computational mesh with the 40×30 non-rectangular cells is illustrated in the Fig. 9. The computational time step is 0.04 s. In order to obtain a steady-state solution, we use a convergence criterion that is defined in terms of the relative error R as

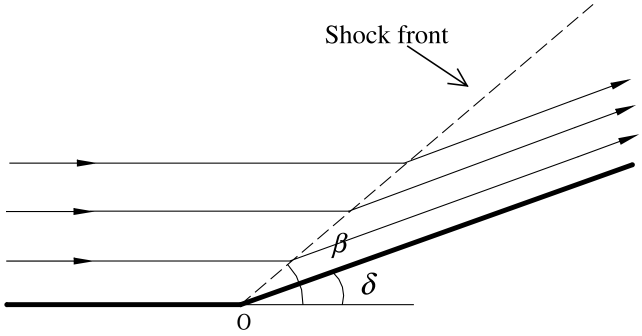

Fig. 8. The plan view of the oblique shock front.

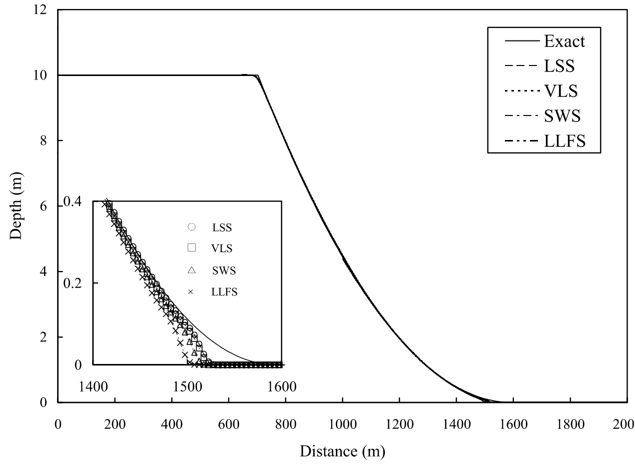

Fig. 7. Comparisons of exact solutions with simulated water depths at $t = 30$ s for a dry bed dam-break problem.

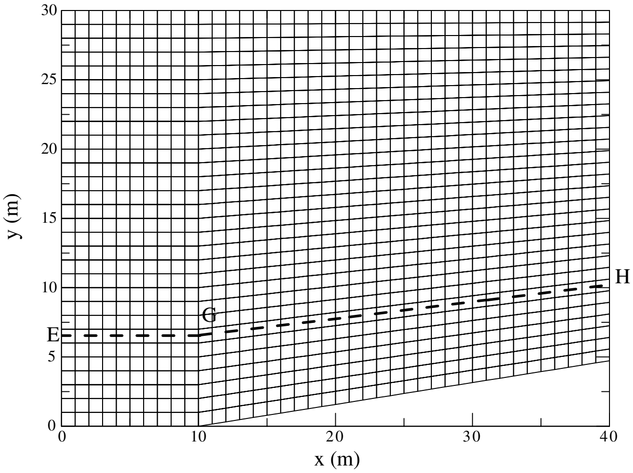

Fig. 9. The geometry and the computational mesh for the 2D oblique hydraulic jump problem.

$$R = \sqrt{\frac{\sum (h_{i,j}^{n+1} - h_{i,j}^{n})^2}{\sum (h_{i,j}^{n})^2}} \leqslant 1.0 \times 10^{-5} \qquad (40)$$

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

The simulated water depth contours of the steady state solutions using the LSS, VLS and SWS schemes are very similar, while the contours by LLFS scheme presents much dissipative solution near the shock front. Fig. 10 shows the simulated water depth contours plots by LSS and LLFS schemes. Fig. 11(a) shows the exact solutions and the simulated results of the water depths along Line EGH illustrated in Fig. 8. The simulated depth profiles by the LSS scheme show the best resolution at the shock front, whereas the LLFS scheme produces much dissipative solution. In Fig. 11(b) the simulated results are compared with the exact solutions for the velocities variations along Line EGH to demonstrate the resolution of the velocity at the shock front. Overall, the LSS scheme presents the best numerical resolution at the shock front among the proposed schemes. Fig. 12 shows the convergence history for each scheme. Except the LLFS scheme, the others produce similar convergence rate.

Table 2 compares the simulated results with the exact solutions in L_2 norm. The CPU time for reaching the convergence criterion is also presented in Table 2. The exact solutions can be found in the literature [24]. Table 2 shows that the LSS scheme produces the smallest relative error in L_2 norms of depth and shock angle, while the LLFS scheme yields the largest relative error in L_2 norm and CPU time consumed. From the simulated results presented above, we may conclude that the LSS scheme has the best numerical accuracy for solving the oblique hydraulic jump problem.

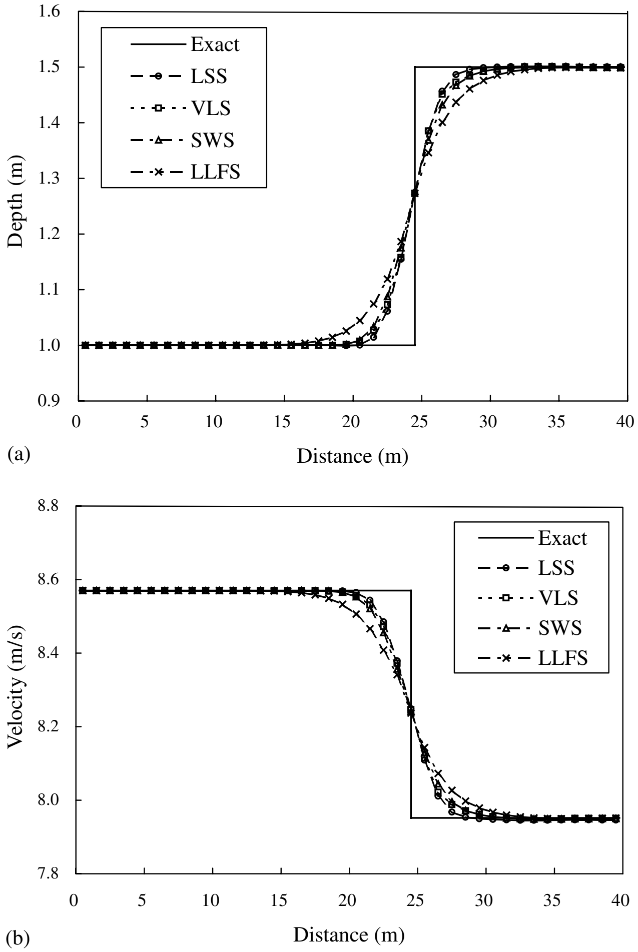

Fig. 11. Comparisons of the exact solutions with the simulated (a) water depths and (b) velocities along Line EGH (see Fig. 9) using the four schemes.

4.3. Partial dam breaking

This hypothetical test problem has been investigated by several researchers [8,9,11,12]. It provides the

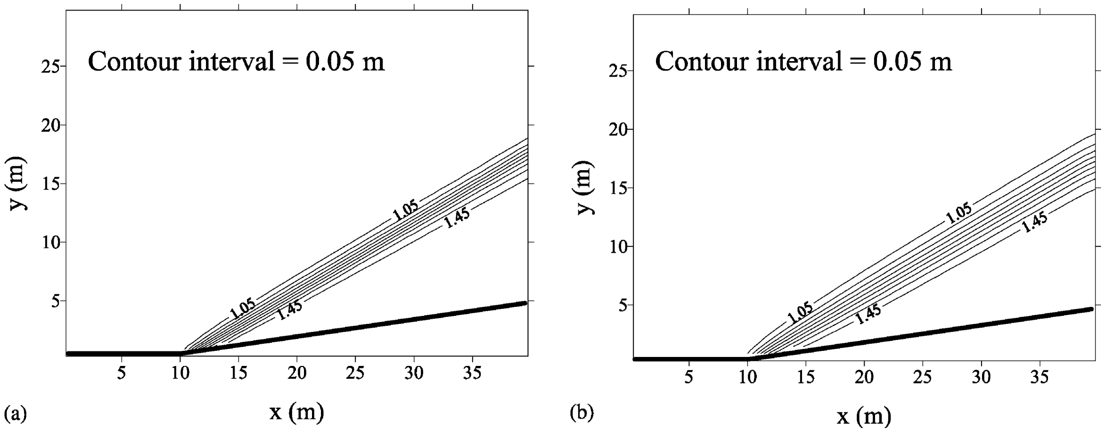

Fig. 10. The contour plots of the 2D oblique hydraulic jump using (a) LSS, and (b) LLFS schemes.

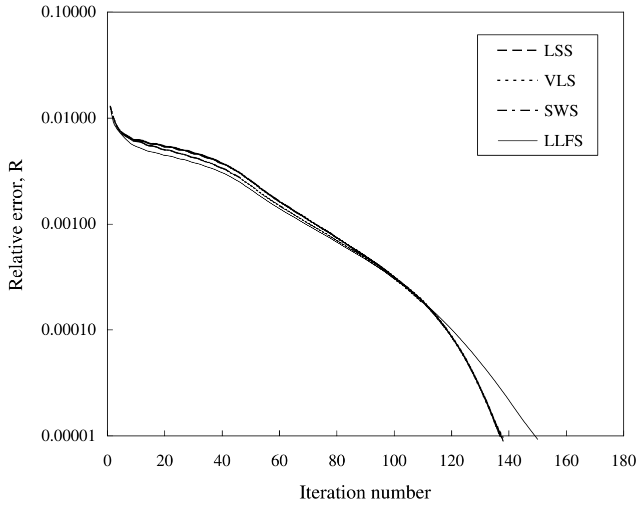

Fig. 12. The convergence history for four schemes.

Table 2
Simulated results for the oblique hydraulic jump problem

Scheme	L_2 norm of depth	L_2 norm of velocity	L_2 norm of shock angle	Number of iteration	CPU time for convergence criterion (s)
LSS	0.035	0.0073	0.0039	138	84.25
VLS	0.036	0.0073	0.0065	138	83.32
SWS	0.037	0.0074	0.0069	137	85.92
LLFS	0.042	0.0085	0.0092	150	90.23

extreme case to examine the shock-capturing capability of the numerical schemes. In this problem the dam is assumed to fail instantaneously. Initially, the water depth upstream of the dam is 10 m and the downstream water depth is 5 m. The geometry of the problem consists of a 200 m × 200 m flow domain as illustrated in Fig. 13. A computational mesh with 40 × 40 rectangular cells is used, and the computational time step is 0.2 s. After dam breaking, through a 75 m wide breach water is released towards downstream to form a shock wave propagating while spreading laterally. For a 7.2 s period of simulation, the CPU time required is 50.92 s for the LSS scheme. The VLS, SWS and LLFS schemes consume respectively 100%, 102% and 97% of the CPU time for the LSS scheme. The simulated 2D contours of the water surface elevations at $t = 7.2$ s obtained using four schemes are very close, and only the result by LSS scheme is shown in Fig. 14. Fig. 14 shows that the proposed schemes give results similar to those found in the literature for this problem [8,9,11,12]. For quantitative comparisons of these four schemes, simulated results along the cross-section of line CD are shown in Fig. 15. As presented in Fig. 15, the LSS, VLS, and SWS give similar simulated solutions.

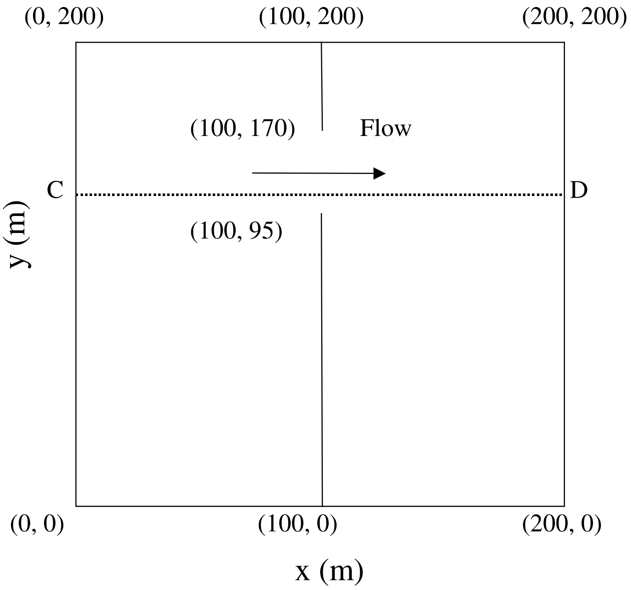

Fig. 13. The definition of problem domain for the 2D partial dam breaking case.

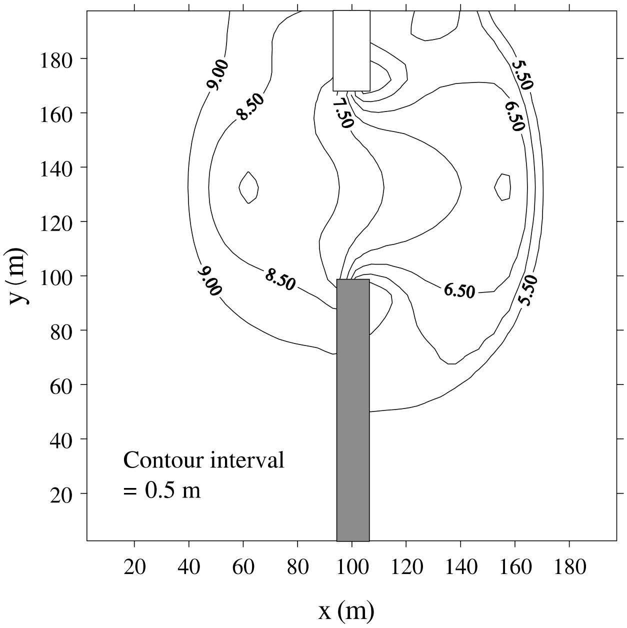

Fig. 14. The contour plots showing the water depth for the 2D partial dam breaking case at $t = 7.2$ s using LSS scheme.

4.4. Circular dam breaking

This problem tests the symmetric shock-capturing capability of the proposed schemes. The simulation condition and geometry used herein are the same as those of Alcrudo and Garcia-Navarro [8], Tseng [10], and Tseng and Chu [12]. The computational mesh using the cylindrical grid system is shown in Fig. 16, in which the circular mesh consisting of 50 cells in the tangential

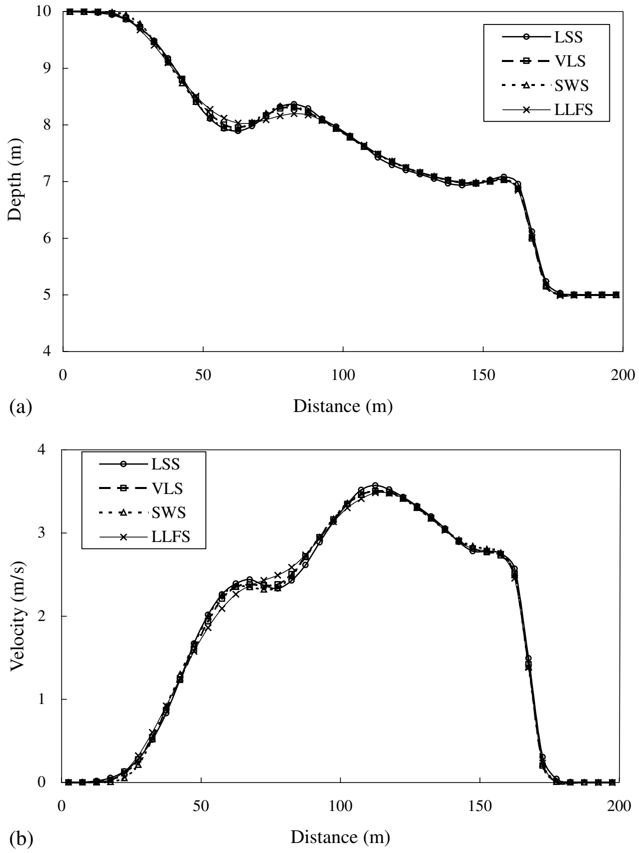

Fig. 15. Comparisons of the simulated (a) water depths and (b) velocities along Line CD (see Fig. 13) using the four schemes.

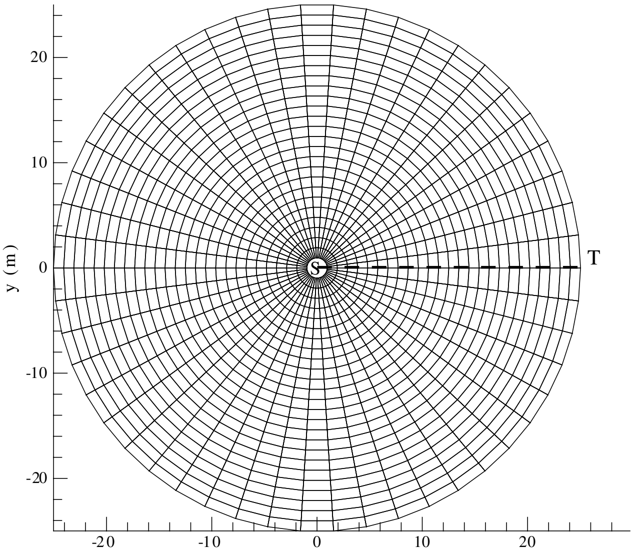

Fig. 16. The geometry and the computational mesh for the circular dam breaking case.

direction and 25 cells of 1 m length along the radial direction is used.

The initial condition comprises two regions of still water separated by a cylindrical dam of radius 11 m. The water depth inside the dam is 10 m and outside the dam is 1 m. The computational time step is 0.02 s for all

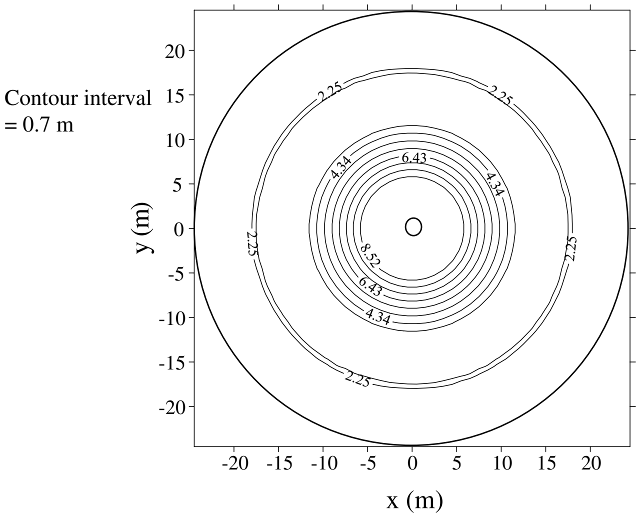

Fig. 17. The contour plots showing the water depth variation for the circular dam breaking case at $t = 0.69$ s using LSS scheme.

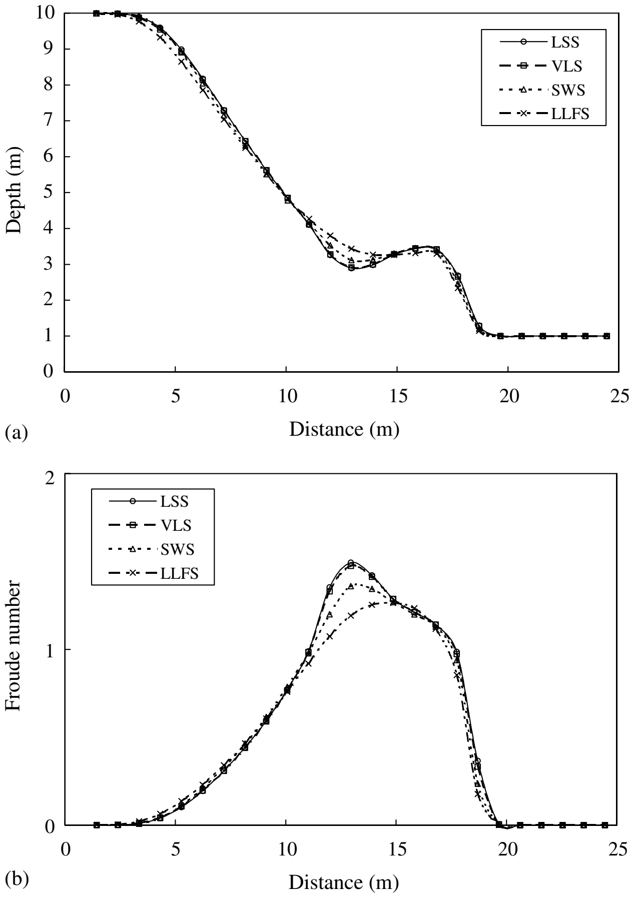

Fig. 18. Comparisons of the simulated (a) water depths and (b) Froude number along Line ST (see Fig. 16) using the four schemes.

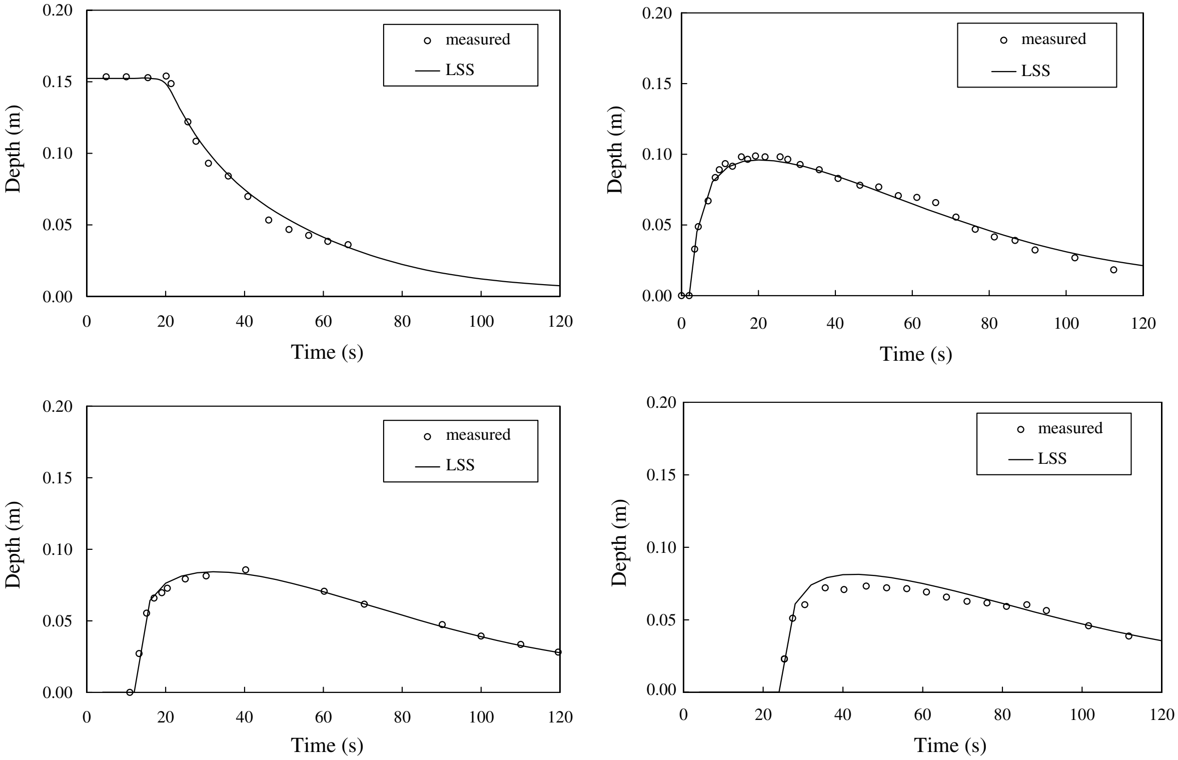

Fig. 19. Experimental and simulated results with sudden dam collapse at (a) $x = 30.5$ m, (b) $x = 68.625$ m, (c) $x = 85.4$ m, and (d) $x = 106.625$ m.

schemes. The CPU time required is 26.02 s for the LSS scheme. The VLS, SWS and LLFS schemes consume respectively 100%, 101% and 96% of the CPU time for the LSS scheme. The 2D contours of the water surface elevations at $t = 0.69$ s for four schemes are very similar, and only the result by LSS scheme is shown in Fig. 17. The 2D contours present perfect symmetric flow behavior, which agrees very well with those found in the literature [8,10,12]. For quantitative comparisons of four schemes, simulated results along the central line are shown in Fig. 18. Fig. 18(a) and (b) show respectively the simulated water depth and Froude number along Line ST. The results show that fluid inside the dam progresses outwards symmetrically, and there is a transition from subcritical to supercritical flows. Visibly, the phenomena observed are an outward-propagating circular shock wave and an inward-propagating circular rarefaction wave. The LSS and VLS schemes produce the similar simulated results.

4.5. Dam-break experiment of WES

In order to demonstrate that the proposed scheme is capable of describing a real dam-break scenario, the data from laboratory dam-break experiment by the Waterway Experiment Station (WES), US Corps of Engineers [25] is adopted. In this section, only the LSS scheme is selected to simulate the shock wave propagation.

The experiment was conducted in a 122 m long and 1.22 m wide rectangular channel with a bottom slope of 0.005. Initially, the water depth on the upstream side of the dam is 0.305 m, and the downstream water depth is zero (dry bed). The flow domain along the channel is discretized into 122 grids with the uniform grid spacing $\Delta x = 1.0$ m. Manning's n is calibrated to be of 0.009. Fig. 19(a)–(d) show the simulated results for the WES experiment in which the dam-break wave propagates downstream on a dry bed. The agreement between simulated and measured water depth against time at various locations is good for the entire duration by LSS scheme.

5. Conclusions

In the framework of the finite volume method, a proposed algorithm using the flux-splitting technique is established by modifying the MacCormack scheme to preserve second-order accuracy in both space and time. Four component-wise TVD schemes based on this algorithm, including the LSS, VLS, SWS and LLFS schemes, have been developed for solving the 2D SWE.

Verifications of the proposed schemes are made by the comparisons of exact solutions and simulated results in the 1D dam-break and 2D steady oblique hydraulic jump problems. Based on the simulated results of 1D dam-break flow with/without dry-bed condition, it is found that the resolutions of shock front using LSS and VLS schemes are identical and better than those of the other two schemes, while the LLFS scheme presents much dissipative results of water depth and velocity profiles. The CPU time consumed by each scheme is about the same. In addition, the proposed schemes can avoid the accumulation of errors caused by the directional dependences in the conventional TVD-MacCormack schemes [6,12,22].

In the 2D steady oblique hydraulic jump problem, the LSS scheme shows the best resolution at the shock front. According to the overall numerical accuracy, the LSS scheme produces the smallest relative error in L_2 norm; therefore, the LSS scheme has the best performance in numerical accuracy among the proposed schemes. The partial dam-break and the circular dam-break problems are simulated and compared to show good agreements with the results reported by other researchers [8,12]. Moreover, a dam-break experiment with bed slope, friction and dry bed conditions is simulated, which also demonstrates the applicability and reliability of the proposed schemes. In conclusion, the proposed finite-volume component-wise TVD schemes are accurate, robust and highly stable for shallow water flows with strong discontinuities.

Acknowledgements

This paper is based on research partially supported by the National Science Council, Taiwan, under Grants NSC 91-2625-Z-002-016 and NSC 91-2625-Z-002-011. The authors would like to thank Professor S.U. Choi of the Department of Civil Engineering, Yonsei University, Korea, for his helpful comments to improve this paper. Constructive comments from an anonymous referee are also much appreciated.

References

[1] Tan WY. Shallow water hydrodynamics. New York: Elsevier; 1992.

[2] Toro E. Shock-capturing methods for free-surface shallow water flows. New York: John Wiley & Sons; 2001.

[3] Harten A. High resolution schemes for hyperbolic conservation laws. J Comput Phys 1983;49:357–93.

[4] Sweby PK. High resolution schemes using flux limiters for hyperbolic conservation laws. SIAM J Numer Anal 1984;21:995–1011.

[5] Yee HC. Construction of explicit and implicit symmetric TVD schemes and their applications. J Comput Phys 1987;68:151–79.

[6] Hirsch C. Numerical computation of internal and external flows, vol. 2. New York: John Wiley & Sons; 1990.

[7] Toro E. Riemann solvers and numerical methods for fluid dynamics. Berlin: Springer-Verlag; 1997.

[8] Alcrudo F, Garcia-Navarro P. A high-resolution Godunove-type scheme in finite volumes for the 2D shallow water equations. Int J Numer Meth Fluids 1993;16:489–505.

[9] Louaked M, Hanich L. TVD scheme for the shallow water equations. J Hydraulic Res 1998;36(3):363–78.

[10] Tseng MH. Explicit finite-volume non-oscillatory schemes for 2D transient free surface flows. Int J Numer Meth Fluids 1999;30: 831–43.

[11] Wang JS, Ni HG, He YS. Finite-difference TVD scheme for computation of dam-break problems. J Hydraulic Eng 2000; 126(4):253–62.

[12] Tseng MH, Chu CR. Two-dimensional shallow water flows simulation using TVD-MacCormack scheme. J Hydraulic Res 2000;38(2):123–31.

[13] Yu H, Liu YP. A second-order accurate, component-wise TVD scheme for nonlinear, hyperbolic conservation laws. J Comput Phys 2001;173:1–16.

[14] Spekreijse SP. Multigrid solution of steady Euler equations, CWI tract 46. The Netherlands: Amsterdam; 1988.

[15] Zhao DH, Shen HW, Lai JS, Tabios GQ. Approximate Riemann solvers in FVM for 2D hydraulic shock wave modeling. J Hydraulic Eng 1996;122(12):692–702.

[16] Liou MS, Christopher J, Steffen JR. A new flux splitting scheme. J Comput Phys 1993;107:23–39.

[17] van Leer B. Flux vector splitting for the Euler equations. In: Proceedings of the 8th International Conference on Numerical Methods in Fluid Dynamics. Springer-Verlag; 1982. p. 507–12.

[18] Steger JL, Warming RF. Flux vector splitting of the inviscid gas dynamic equations with application to finite difference methods. J Comput Phys 1981;40:263–93.

[19] Chaudhry MF. Open-channel flow. Englewood Cliffs: Prentice-Hall; 1993.

[20] Stoker JJ. Water waves: Mathematical theory with applications. Singapore: Wiley-Interscience; 1958.

[21] LeVeque RJ. Numerical methods for conservational laws. Switzerland: Birkhauser Verlagl; 1992.

[22] Tseng MH, Chu CR. The simulation of dam-break flows by an improved predictor–corrector TVD scheme. Adv Water Resour 2000;23:637–43.

[23] Choi SU, Paik J. Performance test of high resolution schemes for 1D dam break problem. KSCE J Civil Eng 2001;5(3):273–80.

[24] Hager WH, Schwalt M, Jimenez O, Chaudry MH. Supercritical flow near an abrupt wall deflection. J Hydraulic Res 1994;32(1): 103–18.

[25] Waterways Experiment Station (WES). Flood resulting from suddenly breached dams, Miscellaneous paper, 2 (374), Report 1. US Army Engineer Waterways Experiment Station, Corps of Engineers, Vicksburg, Mississippi, 1960.