Lawrence Berkeley National Laboratory

Title
Improved multislope MUSCL reconstruction on unstructured grids for shallow water equations

Permalink
https://escholarship.org/uc/item/27p812b6

Journal
International Journal for Numerical Methods in Fluids, 87(8)

ISSN
0271-2091

Authors
Zhao, J
Özgen, I
Liang, D
et al.

Publication Date
2018-07-20

DOI
10.1002/fld.4499

Peer reviewed

Improved multislope MUSCL reconstruction on unstructured grids for shallow water equations

J. Zhao[1] I. Özgen[1,3] D. Liang[2] R. Hinkelmann[1]

[1]Chair of Water Resources Management and Modeling of Hydrosystems, Technische Universität Berlin, Berlin, Germany [2]Department of Engineering, University of Cambridge, Cambridge, UK [3]Computational Disaster Mitigation and Reduction Research Unit, RIKEN Advanced Institute for Computational Science, Kobe, Japan

Correspondence J. Zhao, Chair of Water Resources Management and Modeling of Hydrosystems, Technische Universität Berlin, Sector TIB1-B14, Gustav-Meyer-Allee 25, 13355 Berlin, Germany. Email: jiaheng.zhao@wahyd.tu-berlin.de

Summary

In shallow water flow and transport modeling, the monotonic upstream-centered scheme for conservation laws (MUSCL) is widely used to extend the original Godunov scheme to second-order accuracy. The most important step in MUSCL-type schemes is MUSCL reconstruction, which calculate-extrapolates the values of independent variables from the cell center to the edge. The monotonicity of the scheme is preserved with the help of slope limiters that prevent the occurrence of new extrema during reconstruction. On structured grids, the calculation of the slope is straightforward and usually based on a 2-point stencil that uses the cell centers of the neighbor cell and the so-called far-neighbor cell of the edge under consideration. On unstructured grids, the correct choice for the upwind slope becomes nontrivial. In this work, 2 novel total variation diminishing schemes are developed based on different techniques for calculating the upwind slope and the downwind slope. An additional treatment that stabilizes the scheme is discussed. The proposed techniques are compared to 2 existing MUSCL reconstruction techniques, and a detailed discussion of the results is given. It is shown that the proposed MUSCL reconstruction schemes obtain more accurate results with less numerical diffusion and higher efficiency.

1 INTRODUCTION

The monotonic upstream-centered scheme for conservation laws (MUSCL)[1] is a well-known approach to achieving higher-order accuracy by data reconstruction. An overview of MUSCL-type high-order methods can be found, for example, in chapter 13.4 in the work of Toro.[2](pp426-440) MUSCL-type schemes are essentially an extension of the original Godunov scheme[3] with the following steps: (1) define the Riemann problems at cell interfaces using cell averages, (2) solve the Riemann problems to get the numerical flux, and (3) update the cell averages by summing up flux and source terms.[4] In contrast, MUSCL-type schemes replace cell averages by piecewise linear functions. Thus, step 1 of the Godunov scheme is replaced by the following:

(1) extrapolate the cell averages (linearly) to cell interfaces in defining the Riemann problems (MUSCL reconstruction step). In order to avoid spurious oscillations, the slope of the extrapolation is limited by so-called slope limiter functions.[5] Many slope limiter functions have been derived in the literature, and an overview can be found in the work of Hou et al[6] and in classical textbooks such as those by Toro[2] and LeVeque.[7]

For high-order schemes to produce physical results, they have to be monotonic; otherwise, spurious oscillations may occur. Monotonicity of a numerical scheme can be deduced from a property called total variation that is defined as the summation of differences between every 2 neighboring states over the whole domain at a fixed time. If the total variation does not increase in time, the scheme is said to be total variation diminishing (TVD), and the monotonicity of the scheme is ensured.[2] Slope limiters limit the slope of the MUSCL reconstruction such that it is ensured that no new extrema are created at the cell interfaces. As will be discussed in the following, the limiter function depends on the slope of the upwind and downwind directions of the cell centers. The calculation of these slopes has significant influence on the accuracy of the scheme. Early TVD schemes were derived on structured grids and, when applied directly to unstructured meshes, give unsatisfactory results. The reason is that on structured grids, the slope calculation on the left and the right side is very straightforward as the direct neighbor cell and the so-called far-neighbor cell (the neighbor of the direct neighbor cell) can be used directly. In addition, the ratio of these slopes is a good indicator of stability because all points that contribute to the ratio are equidistant. In contrast, on unstructured meshes, the calculation of the points that contribute to the calculation of the slopes is not straightforward. Hence, the calculation of the slope itself poses a challenge. TVD MUSCL reconstruction techniques for unstructured meshes can be divided into monoslope and multislope methods.[8] The monoslope method that is initially presented in the work of Venkatakrishnan[9] calculates a single vector of slope for the entire cell based on the 3 direct neighbors of the cell. In contrast, the multislope method calculates a slope for each edge based on a 3-point stencil. The challenge of applying the multislope method to unstructured grids is that the determination of the points of the stencil is nontrivial. In the literature, several multislope methods for unstructured grids can be found. For example, in the work of Darwish and Moukalled,[10] a local slope based on a 2-point stencil is calculated without considering the far-neighbor cell. In the work of Li and Liao,[11] a 3-point stencil is used, but instead of the far-neighbor cell, a "virtual" node is included. Motivated by these approaches, in the work of Hou et al,[6] a multislope method that calculates individual weights for slopes depending on the distance of the cell centers to the cell interface is derived. In the work of Hou et al,[12] slopes are calculated based on points that are located on a line normal to the cell interface. Based on the multislope method by Buffard and Clain,[8] Hou et al[13] proposed a more straightforward

vector-based multislope method, which obtains both robustness and accuracy.

In this work, 2 novel TVD MUSCL reconstruction techniques are investigated based on an additional condition for the TVD scheme, and a treatment for limiting the velocity at wet and dry interfaces is proposed for avoiding the instability caused by MUSCL schemes. The proposed schemes are compared with the multislope methods by Buffard and Clain[8] and Hou et al.[13] The schemes are compared in 5 computational test cases: (i) Thacker's planar rotation benchmark, (ii) a steady-state oblique jump, (iii) a radial dam break, (iv) a 2-dimensional (2D) Riemann problem, and (v) a Tsunami wave around a canonical island. In these benchmarks, the accuracy of the TVD method and their computational cost is compared.

2 GOVERNING EQUATIONS AND NUMERICAL MODEL

2.1 Shallow water equations

The 2D shallow water equations (SWEs) can be written in vector form as

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} = \mathbf{s} \tag{1}$$

with vectors defined as

$$\mathbf{q} = \begin{bmatrix} h \\ q_x \\ q_y \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} q_x \\ uq_x + gh^2/2 \\ uq_y \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} q_y \\ vq_x \\ vq_y + gh^2/2 \end{bmatrix}, \tag{2}$$

$$\mathbf{s} = \begin{bmatrix} 0 \\ -gh\frac{\partial z}{\partial x} - c_f u\sqrt{u^2 + v^2} \\ -gh\frac{\partial z}{\partial y} - c_f v\sqrt{u^2 + v^2} \end{bmatrix}, \tag{3}$$

where t is time, x and y are the Cartesian coordinates, and q represents the variable vector consisting of h, q_x, and q_y that denote water depth, unit-width discharges in the x- and y-directions, respectively. u and v are defined as depth-averaged velocities in the x- and y-directions, respectively; f and g are the flux vectors in the x- and y-directions, respectively; sis the source term that includes bed slope and friction contributions; z is the bed elevation; and c_f is the bed roughness coefficient. Here, viscous, diffusive, and turbulent flux terms are neglected.

2.2 Finite-volume discretization of SWEs on unstructured grids

The SWEs in Equation 1 can be written in integral form as

$$\int_\Omega \frac{\partial \mathbf{q}}{\partial t} d\Omega + \int_\Omega \left(\frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} \right) d\Omega = \int_\Omega \mathbf{s} d\Omega, \tag{4}$$

where Ω is an arbitrary control volume. Applying the Green-Gauß theorem and replacing the boundary integral with a sum over all edges, Equation 4 becomes

$$\int_{\Omega} \frac{\partial \mathbf{q}}{\partial t} d\Omega + \sum_{k=1}^{m} \mathbf{F} \cdot \mathbf{n}_k l_k = \int_{\Omega} s d\Omega, \tag{5}$$

wherein m is the number of edges, and is the unit normal vector pointing in the outward normal direction of the boundary edge, l is the length of the edge, and F·n is the flux vector normal to the boundary and can be written as

$$\mathbf{F} \cdot \mathbf{n} = (\mathbf{f} n_x + \mathbf{g} n_y) = \begin{bmatrix} q_x n_x + q_y n_y \\ (u q_x + g h^2/2) n_x + v q_y n_y \\ u q_x n_x + (v q_y + g h^2/2) n_y \end{bmatrix}. \tag{6}$$

The value of q in cell i is updated using the 2-stage explicit Runge-Kutta scheme,([14-16]) where the value at the next time level in cell i, , is updated by

$$\mathbf{q}_i^{n+1} = \frac{1}{2} \left\{ \mathbf{q}_i^n + \kappa \left[\kappa \left(\mathbf{q}_i^n \right) \right] \right\} \tag{7}$$

with

$$\kappa \left(\mathbf{q}_i^n \right) = \mathbf{q}_i^n + \frac{\Delta t^n}{\Omega} \left[\int_{\Omega} \mathbf{s}^{n+1} d\Omega - \sum_{k=1}^{m} \mathbf{F}(\mathbf{q}_i^n)_k \cdot \mathbf{n}_k l_k \right], \tag{8}$$

where s^{n+1} is the friction source term and discretized in a splitting point implicit way[17]; the slope source term is calculated based on the slope flux calculation method from the work of Hou et al,[18] which is added into the flux term ; and κ is a function to represent the updating process to a new time level in the considered cell. Δt^n is the time step at the nth time level. For this work, the Courant-Friedrichs-Lewy condition is used for maintaining stability, ie,

$$\Delta t = \text{CFL} \min \left(\frac{R_1}{\sqrt{u_1^2 + v_1^2} + \sqrt{g h_1}}, \dots, \frac{R_n}{\sqrt{u_n^2 + v_n^2} + \sqrt{g h_n}} \right), \tag{9}$$

where R_n is the minimum distance from the cell center to the edge, and CFL is the Courant-Friedrichs-Lewy number. For explicit time-marching algorithms, . In this work, CFL=0.5 is adopted.

2.3 MUSCL reconstruction

In order to obtain a second-order accurate numerical scheme, Godunov's theorem[3] can be circumvented by reconstructing the cell-averaged values linearly using MUSCL reconstruction.[1] MUSCL reconstruction is applied successfully for many physical problems (see, eg, other works([2, 8, 15, 16, 19-21])).

The reconstruction from the cell center i to the cell interface $(i,i+1)$, hereinafter also denoted by the subscript $i+1/2$, is calculated as

$$q_{i+1/2} = q_i + \Delta x_{i,i+1/2} \psi(r) \frac{q_{i+1} - q_i}{\Delta x_{i,i+1/2} + \Delta x_{i+1/2,i+1}}, \quad (10)$$

as shown in Figure 1, where $\Delta x_{i,i+1/2}$ and $\Delta x_{i+1/2,i+1}$ are the distances from the cell center i to the edge and the cell center $i+1$ to the edge, respectively, ψ is the limiting coefficient of slope, and r is the slope ratio, which will be discussed in the following section.
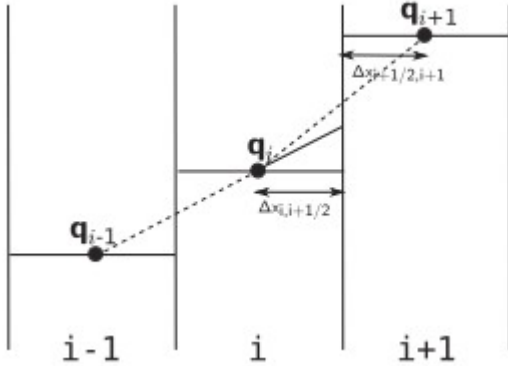
Figure 1

One-dimensional monotonic upstream-centered scheme for conservation laws reconstruction

MUSCL reconstruction gives values at the left and right cell interfaces that can be used to construct a Riemann problem. The solution of the Riemann problem then yields the numerical flux in Equation 6.[2] In this work, a Harten, Lax, and van Leer Riemann solver with the contact wave restored (HLLC)[22] is used. The positivity-preserving hydrostatic reconstruction by Audusse et al[23] is used to maintain a nonnegative water depth and correct reconstruction of the Riemann states, and the C-property–preserving divergence form of the bed slope source term proposed by Hou et al[18] is used; the source term treatment does not influence the well-balanced property of any of the MUSCL schemes.

3 MULTISLOPE MUSCL RECONSTRUCTION METHODS

In multislope MUSCL reconstruction methods, the slope for the MUSCL reconstruction in Equation 10 is calculated at each edge individually.

In the original multislope scheme (derived in the work of Roe[24] for uniform grids), the edge value is calculated with a diffusive first-order upwind value and an antidiffusive term as

$$q_{CD} = q_C + \frac{1}{2} \psi(r)(q_D - q_C). \quad (11)$$

Here, r is the ratio of consecutive slopes,[25] which can be calculated with a 3-point stencil that consists of 2 adjacent cells C and D and the far-neighbor cell U located in the upstream direction. The ratio r becomes

$$r = \frac{\nabla q_{\text{up}}}{\nabla q_{\text{down}}}. \quad (12)$$

It is noted that due to the uniform grid assumption, the ratio $\frac{\Delta x_{i,j+1/2}}{\Delta x_{i,j+1/2}+\Delta x_{i+1/2,j+1}}$ in Equation 10 is simplified to in Equation 11.

In the literature, 2D MUSCL schemes on unstructured grids are mainly separated into 1-dimensional (1D) gradient methods,([6, 10-12]) 2D nodal evaluating methods,([26, 27]) and vector manipulation methods.([8, 13]) In this paper, 2D nodal evaluating methods and vector manipulation methods are compared with regard to efficiency, accuracy, and ease of implementation.

3.1 2D nodal evaluation methods

A straightforward approach to calculating the MUSCL slope on unstructured grids is to directly apply the classical TVD methods derived for 1D structured grids. Then, the upstream node U of the stencil can be calculated by extrapolating along a certain distance in the upstream direction. Darwish and Moukalled[10] noted that the difficulty of implementing MUSCL reconstructions on unstructured grids is in determining U. Based on prior work by Bruner and Walters,[28] Darwish and Moukalled[10] proposed a MUSCL reconstruction method (Darwish's scheme) where r is defined as

$$r = \frac{2(\nabla q)_C \cdot \vec{d}_{C,D}}{q_D - q_C} - 1. \quad (13)$$

Here, $\vec{d}_{C,D}$ is the distance vector from the cell center of C to the cell center of D. $(\nabla q)_C$ is the cell value gradient of cell C. In contrast to the calculation of r on uniform grids given in Equation 12, Darwish and Moukalled[10] accounted for the nonuniform distance between the cell centers. Instead of determining an additional point U, the value at node U is interpolated based on the gradient of cell C, as shown in Figure 2 (left). This is, in fact, a very local calculation of r. If the gradient in cell C does not represent the overall behavior of the variable, the calculated r differs significantly from a 3-point stencil.
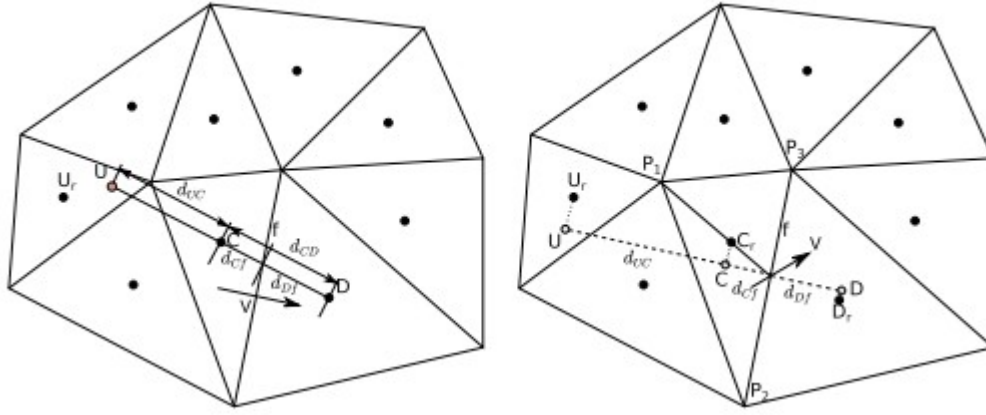
Figure 2

Three-point stencil for Darwish and Moukalled[10] (U is not taken into account), Li and Liao,[11] and Hou et al[6] (left) and 3-point stencil for Hou et al[12] (right)

Li and Liao[11] defined the stencil for calculating r using the cell centers of 2 adjacent cells C and D and constructed a virtual node U that is located in the upstream direction on the line connecting C and D, such that all nodes are equidistant (cf Figure 2, left). U_r is the cell center wherein node U is located. The value at U is interpolated based on the gradient of the cell containing U( ), ie,

$$q_U = q_{U_r} + \vec{d}_{U_r, U} \cdot (\nabla q)_{U_r},\ (14)$$

and Equation 12 becomes

$$r = \frac{q_C - \left(q_{U_r} + \overrightarrow{r_{U_r, U}} \cdot (\nabla q)_{U_r}\right)}{q_D - q_C}.\ (15)$$

The method by Li and Liao[11] contains more upstream information and is not as local as Darwish's scheme.

All of these TVD schemes neglect the interface position and the distance of the cell centers to the interface. Hou et al[6] proposed a reconstruction (Hou's first scheme) that includes the interface position in the interpolation. It can be written as

$$q_{CD} = q_C + \frac{d_{Cf}}{d_{Cf} + d_{Df}} \psi(r_f)(q_D - q_C),\ (16)$$

$$r = \frac{(q_D - q_C)/d_{CD}}{(q_C - q_U)/d_{UC}}.\ (17)$$

Here, f denotes the interface, and d_{UC}, d_{CD}, d_{Cf}, and d_{Df} are the distances from node U to C, from C to D, from C to the interface, and from D to the interface, respectively (cf Figure 2, left).

Hou et al[12] noted that in most reconstruction techniques, the value at node U is interpolated from the cell center using the gradient of the variable in the cell. Thus, the cell gradient has to be calculated first. It is then argued that in advection-dominated flows, the flux can be split into a component normal to the interface and a component tangential to the interface. The tangential component is transported by the normal component as a passive scalar. Therefore, instead of connecting the cell centers of the adjacent cell and locating the upstream node U on this line, Hou et al[12] suggested drawing a line that goes normal to the interface, through the center of the interface, and locating all 3 nodes of the stencil on this line (cf Figure 2, right). C, D, and U are projections of the cell centers C_r, D_r, and U_r, respectively. U_r is determined as the nearest cell center around the vertex P_1 of cell C to the line. The values at the points C, D, and U are not interpolated but shifted directly from the cell centers, ie, the value at node U is equal to the value at node U_r. Then, Equations 16 and 17 are used to reconstruct the values at the interface.

3.2 Vector manipulation methods

Buffard and Clain[8] developed an approach to construct the upwind and downwind slopes of edges in the considered cells by manipulating the geometric directional unit vector. Based on this approach, Hou et al[13] developed a more straightforward scheme for the calculation of the upwind and downwind slopes.

As shown in Figure 3, dimensional unit vectors are calculated as

$$\vec{r_k} = \frac{\overrightarrow{iM_k}}{|\overrightarrow{iM_k}|} \quad (18)$$

$$\vec{t_k} = \frac{\overrightarrow{ij_k}}{|\overrightarrow{ij_k}|}. \quad (19)$$

M is the center of the edge, $\vec{r_k}$ and $\vec{t_k}$ are the unit directional vectors from the considered cell center to the edge center and the neighbor cell center, respectively. It can also be easily shown that the opposite direction of $\vec{r_k}$ will pass by the node coordinate that belongs to the considered cell but not the vertex of the edge.
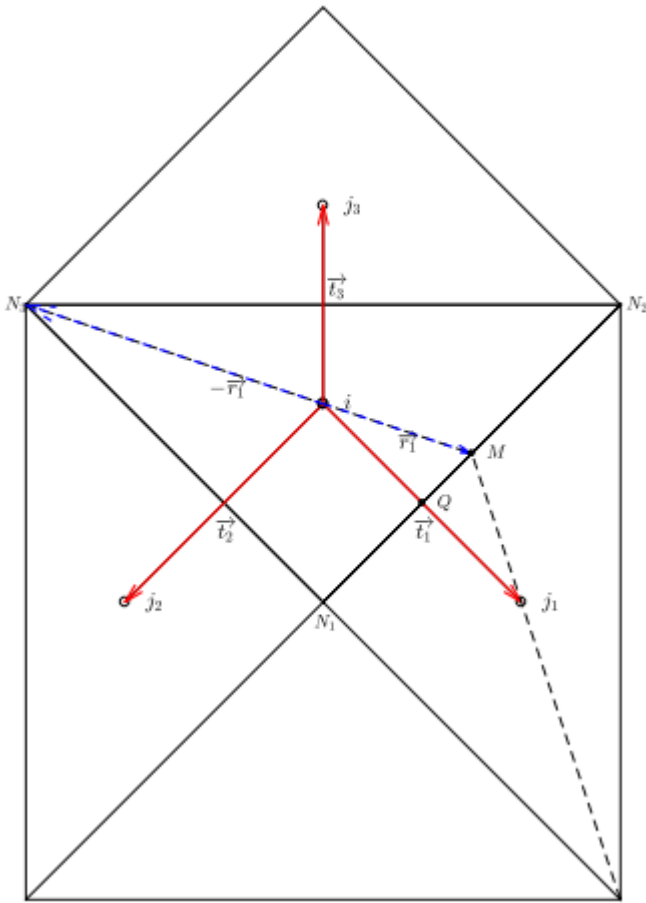
Figure 3

Vector notations at the considered and neighboring cells

The slope along $\vec{t_k}$ can be calculated as

$$\nabla q_{ij} = \frac{q_{j_k} - q_i}{|ij_k|}. \quad (20)$$

By following the approach of Buffard and Clain,[8] the slopes along the line connecting the cell center with the edge center in the upwind and downwind directions can be thought of as the slopes from the far node N_m, with $m=1,2,3$, to the cell center i and from the cell center i to the considered edge center M, respectively, ie, the slopes along and in Figure 3, respectively.

In the scheme by Hou et al,[13] the vectors and are obtained by solving a set of linear equations, which can be obtained by geometric considerations as

$$\vec{r_1} = \alpha_1 \vec{t_1} + \alpha_2 \vec{t_3} \quad (21)$$

$$-\vec{r_1} = \beta_1 \vec{t_2} + \beta_2 \vec{t_3}. \quad (22)$$

Here, $\alpha_{1,2}$ and $\beta_{1,2}$ are coefficients for linear construction and can be calculated from Equations 21 and 22. The slopes along are obtained by Equation 20. The gradients along and are independent from each other and , so that slopes at the upwind and downwind directions can be calculated as

$$\nabla q_{iM} = \alpha_1 \nabla q_{i,j_1} + \alpha_2 \nabla q_{i,j_2} \quad (23)$$

$$\nabla q_{N_1,i} = -(\beta_1 \nabla q_{i,j_2} + \beta_2 \nabla q_{i,j_3}). \quad (24)$$

Values at the cell edge are defined as q_M^i and q_M^o, where the superscripts i and o denote that the variable is defined at the inside or at the outside of the cell under consideration, respectively. It is noted that the number of variables outside of the cell under consideration is equal to that inside of the neighbor cell. q_M^i can be calculated as

$$q_M^i = q_i + |iM_k|\psi\left(\nabla q_{iM_k}, \nabla q_{N_j i}\right), \quad (25)$$

where k is the local index of the edge, and j is the local index of the node along the opposite direction of the considered edge. ψ is the limiter function as defined in the previous section, with the difference being that the parameters here are using the upwind and downwind slopes instead of the slope ratio r.

3.3 Methods of improving the TVD property

One aspect that, to the authors' knowledge, has not been discussed in the literature is a special case of MUSCL reconstruction that violates monotonicity without creating new extrema. Consider the case illustrated in Figure 4. If $r_{(C,D)} > 1.0$ and $r_{(D,C)} > 1.0$, where $r_{(C,D)}$ is the slope ratio on the left side of the edge and $r_{(D,C)}$ is the slope ratio on the right side of the edge, the reconstructed values give $q_{eL} > q_{eR}$ even though $q_C < q_D$. The classical TVD limiter does not prevent this case because no new extremum is created. However, the numerical flux based on q_{eL} and q_{eR} in this case is physically not feasible.

Figure 4

Monotonicity-violating reconstruction

In this work, it is ensured that the following condition is satisfied, so that the case discussed above is always prevented:

$$(q_{i+1} - q_i)(q_{i+1,i} - q_{i,i+1}) \geq 0.0. \quad (26)$$

This condition preserves the monotonicity for both cells and edges. The derivation of Equation 26 is briefly presented in the following.

The TVD condition for the 1D case is given in the work of Harten[5] as

$$q_i^{n+1} = q_i^n + C_{+,(i,i+1)}\Delta q_{i,i+1}^n - C_{-,(i-1,i)}\Delta q_{i-1,i}^n. \quad (27)$$

Here, C is a variable-dependent coefficient, subscripts "+" and "−" mean the flux flow into and out of cell i, respectively, . The sufficient TVD conditions are expressed in a series of inequalities, ie,

$$C_{+,(i,i+1)} \geq 0, \quad C_{-,(i,i+1)} \geq 0, \quad C_{+,(i,i+1)} + C_{-,(i,i+1)} \leq 1. \quad (28)$$

A TVD scheme needs to satisfy the conditions given in Equations 27 and 28. For 1D grids, the variable at time step $n+1$ in cell i can be written as

$$q_i^{n+1} = q_i^n + C_{+,(i,i+1)}(q_{i+1} - q_i) - C_{-,(i-1,i)}(q_i - q_{i-1}), \quad (29)$$

and the reconstructed variables read

$$q_{i+1,i} = q_{i+1} + \Upsilon_{(i+1,i)}(q_i - q_{i+1}) \quad (30)$$

$$q_{i,i+1} = q_i + \Upsilon_{(i,i+1)}(q_{i+1} - q_i) \quad (31)$$

$$q_{i,i-1} = q_i + \Upsilon_{(i,i-1)}(q_{i-1} - q_i) \quad (32)$$

$$q_{i-1,i} = q_{i-1} + \Upsilon_{(i-1,i)}(q_i - q_{i-1}) \quad (33)$$

Y can be seen as the distance multiplied with the slope limiter. Then, it is easy to get

$$q_{i+1} - q_i = \frac{q_{i+1,i} - q_{i,i+1}}{1.0 - \Upsilon_{(i+1,i)} - \Upsilon_{(i,i+1)}} \quad (34)$$

$$q_i - q_{i-1} = \frac{q_{i,i-1} - q_{i-1,i}}{1.0 - \Upsilon_{(i,i-1)} - \Upsilon_{(i-1,i)}} \quad (35)$$

so that Equation 29 can be rewritten as

$$q_i^{n+1} = q_i^n + \frac{C_{+,(i,i+1)}}{1.0 - \Upsilon_{(i+1,i)} - \Upsilon_{(i,i+1)}}\left(q_{i+1,i}^n - q_{i,i+1}^n\right) - \frac{C_{-,(i-1,i)}}{1.0 - \Upsilon_{(i,i-1)} - \Upsilon_{(i-1,i)}}\left(q_{i,i-1}^n - q_{i-1,i}^n\right),$$
$$(36)$$

$$\Delta q_{i,i+1}^{n+1} = q_{i+1}^{n+1} - q_i^{n+1}$$

$$= q_{i+1}^n + \frac{C_{+,(i+1,i+2)}}{1.0 - \Upsilon_{(i+2,i+1)} - \Upsilon_{(i+1,i+2)}}\left(q_{i+2,i+1}^n - q_{i+1,i+2}^n\right) - \frac{C_{-,(i,i+1)}}{1.0 - \Upsilon_{(i+1,i)} - \Upsilon_{(i,i+1)}}\left(q_{i+1,i}^n - q_{i,i+1}^n\right)$$
$$- \left(q_i^n + \frac{C_{+,(i,i+1)}}{1.0 - \Upsilon_{(i+1,i)} - \Upsilon_{(i,i+1)}}\left(q_{i+1,i}^n - q_{i,i+1}^n\right) - \frac{C_{-,(i-1,i)}}{1.0 - \Upsilon_{(i,i-1)} - \Upsilon_{(i-1,i)}}\left(q_{i,i-1}^n - q_{i-1,i}^n\right)\right)$$

$$= \frac{q_{i+1,i}^n - q_{i,i+1}^n}{1.0 - \Upsilon_{(i+1,i)} - \Upsilon_{(i,i+1)}} + \frac{C_{+,(i+1,i+2)}}{1.0 - \Upsilon_{(i+2,i+1)} - \Upsilon_{(i+1,i+2)}}\left(q_{i+2,i+1}^n - q_{i+1,i+2}^n\right)$$
$$- \frac{C_{-,(i,i+1)}}{1.0 - \Upsilon_{(i+1,i)} - \Upsilon_{(i,i+1)}}\left(q_{i+1,i}^n - q_{i,i+1}^n\right)$$
$$- \left(\frac{C_{+,(i,i+1)}}{1.0 - \Upsilon_{(i+1,i)} - \Upsilon_{(i,i+1)}}\left(q_{i+1,i}^n - q_{i,i+1}^n\right) - \frac{C_{-,(i-1,i)}}{1.0 - \Upsilon_{(i,i-1)} - \Upsilon_{(i-1,i)}}\left(q_{i,i-1}^n - q_{i-1,i}^n\right)\right)$$

$$= \frac{1.0 - C_{-,(i,i+1)} - C_{+,(i,i+1)}}{1.0 - \Upsilon_{(i+1,i)} - \Upsilon_{(i,i+1)}}\left(q_{i+1}^n - q_{i,i+1}^n\right) + \frac{C_{+,(i+1,i+2)}}{1.0 - \Upsilon_{(i+2,i+1)} - \Upsilon_{(i+1,i+2)}}\left(q_{i+2,i+1}^n - q_{i+1,i+2}^n\right)$$
$$+ \frac{C_{-,(i-1,i)}}{1.0 - \Upsilon_{(i,i-1)} - \Upsilon_{(i-1,i)}}\left(q_{i,i-1}^n - q_{i-1,i}^n\right).$$
$$(37)$$

For TVD schemes, it is not difficult to get $1.0 - C_{-,(i,i+1)} - C_{+,(i,i+1)} \geq 0, C_{+,(i+1,i+2)} \geq 0, C_{-,(i-1,i)} \geq 0$. If Equation 37 satisfies the TVD property, the coefficients should be nonnegative, which gives that $1.0 - \Upsilon_{(i,i-1)} - \Upsilon_{(i-1,i)}, 1.0 - \Upsilon_{(i+1,i)} - \Upsilon_{(i,i+1)}, 1.0 - \Upsilon_{(i+2,i+1)} - \Upsilon_{(i+1,i+2)}$ should all be positive values and thus

$$\left|\Delta q_{i,i+1}^{n+1}\right| \leq \frac{1.0 - C_{-,(i,i+1)} - C_{+,(i,i+1)}}{1.0 - \Upsilon_{(i+1,i)} - \Upsilon_{(i,i+1)}}\left|q_{i+1}^n - q_{i,i+1}^n\right| + \frac{C_{+,(i+1,i+2)}}{1.0 - \Upsilon_{(i+2,i+1)} - \Upsilon_{(i+1,i+2)}}\left|q_{i+2,i+1}^n - q_{i+1,i+2}^n\right|$$
$$+ \frac{C_{-,(i-1,i)}}{1.0 - \Upsilon_{(i,i-1)} - \Upsilon_{(i-1,i)}}\left|q_{i,i-1}^n - q_{i-1,i}^n\right|.$$
$$(38)$$

Summing up Equation 38 for $-\infty < i < \infty$, the total variation at time $n+1$ gives

$$
\begin{aligned}
TV\left(q^{n+1}\right) = \sum_{i=-\infty}^{i=\infty}\left|\Delta q_{i,i+1}^{n+1}\right| &\leq \sum_{i=-\infty}^{i=\infty}\frac{1.0 - C_{-,(i,i+1)} - C_{+,(i,i+1)}}{1.0 - \Upsilon_{(i+1,i)} - \Upsilon_{(i,i+1)}}\left|q_{i+1}^{n} - q_{i,i+1}^{n}\right| \\
&+ \sum_{i=-\infty}^{i=\infty}\frac{C_{+,(i+1,i+2)}}{1.0 - \Upsilon_{(i+2,i+1)} - \Upsilon_{(i+1,i+2)}}\left|q_{i+2,i+1}^{n} - q_{i+1,i+2}^{n}\right| \\
&+ \sum_{i=-\infty}^{i=\infty}\frac{C_{-,(i-1,i)}}{1.0 - \Upsilon_{(i,i-1)} - \Upsilon_{(i-1,i)}}\left|q_{i,i-1}^{n} - q_{i-1,i}^{n}\right| \\
&= \sum_{i=-\infty}^{i=\infty}\left|\Delta q_{i,i+1}^{n}\right| = TV\left(q^{n}\right).
\end{aligned}
$$

(39)

It is noted that Y depends on the slope limiter Ψ and the position information $\Delta x_{i,i+1/2}/(\Delta x_{i,i+1/2}+\Delta x_{i+1/2,i+1})$; hence, Equations 30 to 36 are also valid for unstructured grids. If $1.0 - Y_{(i+1,i)} - Y_{(i,i+1)}$ and $1.0 - Y_{(i,i-1)} - Y_{(i-1,i)}$ are zero, which means the reconstructed values along the edge for both neighbors are the same, the absolute value for the value difference is zero and does not influence the TVD property of this scheme, as

$$1.0 - \Upsilon_{(i+1,i)} - \Upsilon_{(i,i+1)} \geq 0,$$ (40)

if $q_{i+1} \geq q_i$, Equation 40 can be rewritten as

$$q_i + \gamma_{i,i+1}(q_{i+1} - q_i) - \left(q_{i+1} + \gamma_{i+1,i}(q_i - q_{i+1})\right) \leq 0.0,$$ (41)

it can be obtained that if $q_i \leq q_{i+1}$, then $q_{i,i+1} \leq q_{i+1,i}$, and vice versa, it will be very easy to get the relationship for the variables along the edges, ie,

$$(q_{i+1} - q_i)(q_{i+1,i} - q_{i,i+1}) \geq 0.0,$$ (42)

which is the condition given by Equation 26.

If the TVD condition is satisfied for the reconstruction method, the scheme should fullfill the relationship given in Equation 28, but if the upwind and downwind neighbor cells are defined as $i-2$ and $i+2$, respectively, Equation 28 becomes

$$q_i^{n+1} = q_i^n + C_{+,(i,i+2)}\Delta q_{i,i+1}^n - C_{-,(i-2,i)}\Delta q_{i-1,i}^n.$$ (43)

The value difference between cells i and $i+1$ on the new time step $n+1$ can be stated as

$$\Delta q_{i,i+1}^{n+1} = \left(1 - C_{+,(i,i+2)} - C_{-,(i,i+2)}\right)\Delta q_{i,i+1}^n + C_{+,(i+1,i+3)}\Delta q_{i+1,i+2}^n + C_{-,i-2,i}\Delta q_{i-1,i}^n.$$ (44)

If all the coefficients satisfy the inequalities

$$C_{+,(i,i+2)} \geq 0, \quad C_{-,(i,i+2)} \geq 0, \quad C_{+,(i,i+2)} + C_{-,(i,i+2)} \leq 1,$$ (45)

the total variation can be summed up for $-\infty < i < +\infty$, and the following expression is obtained:

$$TV\left(q^{n+1}\right) = \sum_{i=-\infty}^{i=\infty} \left|\Delta q_{i,i+1}^{n+1}\right| \le \sum_{i=-\infty}^{i=\infty} \left(1.0 - C_{-,(i,i+2)} - C_{+,(i,i+2)}\right)\left|\Delta q_{i,i+1}^{n}\right|$$

$$+ \sum_{i=-\infty}^{i=\infty} C_{+,(i+1,i+3)}\left|\Delta q_{i+1,i+2}^{n}\right| + \sum_{i=-\infty}^{i=\infty} \left(C_{-,(i-2,i)}\right)\left|\Delta q_{i-1,i}^{n}\right| . \tag{46}$$

Changing the indices of the last 2 terms to $i+2$ and $i-2$, respectively, it is seen that the resulting equation is not TVD anymore, ie,

$$TV\left(q^{n+1}\right) = \sum_{i=-\infty}^{i=\infty} \left|\Delta q_{i,i+1}^{n+1}\right| \le \sum_{i=-\infty}^{i=\infty} \left(1.0 - C_{-,(i,i+2)} + C_{-,(i-1,i+1)}\right)\left|\Delta q_{i,i+1}^{n}\right|$$

$$\ne TV\left(q^{n}\right). \tag{47}$$

It can be concluded that the coefficients that get the influence from $i-2$ and $i+2$ will not lead to a TVD scheme. As shown in Figure 5, the situation can be thought about as a dam-break problem. From the time level t to $t+1$, if $i \pm 2$ are chosen to be the upwind and downwind cells, the slope for the water depth (left) is calculated properly, but the slope for the discharge (right), $i+2$, will yield a wrong interpolation if cell i is used as the upwind cell instead of $i+1$.
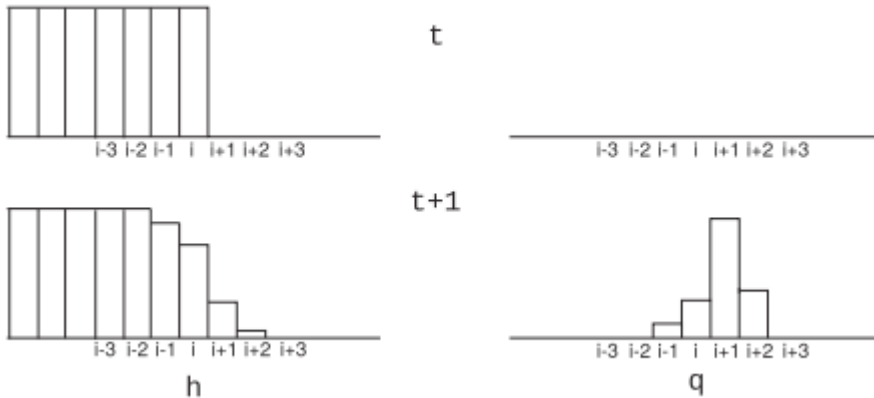
Figure 5

Illustration of the interpolation error in 2-dimensional node evaluation methods

Considering the previous 2D node evaluation methods, only Darwish's scheme satisfies the condition presented in this section. In contrast, the other 2D node evaluation schemes using a certain distance or preferential direction may set $i \pm (n \ne 1)$ as the upwind cell, thus leading to a nonstable scheme. However, Darwish's scheme consists of a very localized stencil and, therefore, has lower-order accuracy.

In the works of Hou et al[26] and Park et al,[27] the stability condition of 2D SWEs is extended to an interval from the minimum and maximum values of the cell-center values at both sides of the considered edge and the values of the 2 vertices of that edge, so that it will allow that the interpolated values of the

edge can be the values larger or lesser than the values at the cell centers of both adjacent cells.

The TVD condition for the vector manipulation method is obtained by verifying that the scheme satisfies Equation 26. As an illustrative example, a Scottish mesh is chosen to calculate the stencil of the reconstruction method, as shown in Figure 6. α_1, α_2 and β_1, β_2 are calculated to be 1.34164, 0.632456 and 1.26491, 1.34164, respectively. Because the vectors and are located between the vectors, they will be used for constructing the linear system later; it can be shown that all of the values are nonnegative numbers, so that the slope for upwind and downwind depends on . Let h_D=1.1 m and h_E=0.2 m, then the reconstruction of $h_{D,M}$=1.1 m remains of first order, but for $q_{C,M}$, the slopes for upwind and downwind in cell C can be calculated as $\nabla q_{N,C}$=0.7589 and $\nabla q_{C,M}$=0.7589, which means there will be a positive value calculated by the limiter function and multiplied with the distance, which may create a local extremum value larger than $q_{D,M}$. If h_D=1.0, the overestimated interpolation will also happen and will lead to the unphysical flux.
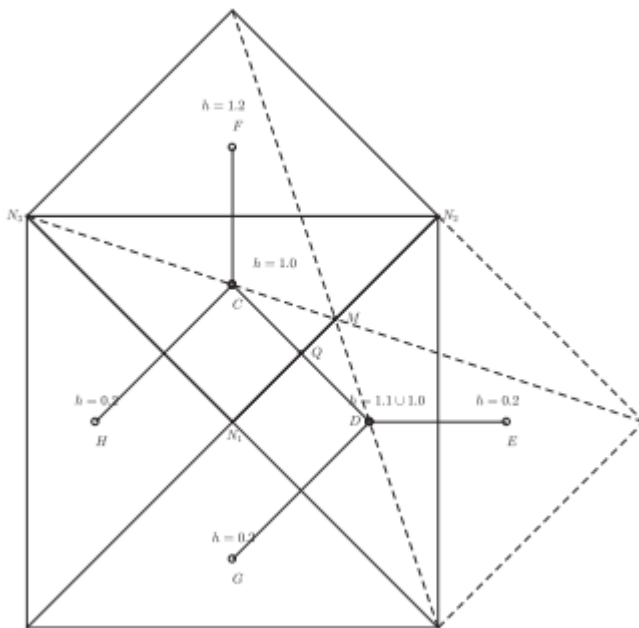
Figure 6

Stencils for the vector manipulation method

4 IMPROVED MULTISLOPE MUSCL RECONSTRUCTION METHODS

For multislope methods on unstructured grids, the 3-point stencil can be considered as a 1D interpolation stencil problem in the local coordinate system of the edge, whereby the main challenge becomes the definition of the upstream node. In this work, it is suggested that the upwind values interpolated or extrapolated by the surrounding neighbor cell-center values be used. The details are given in the following.

In the following, it is assumed that all the multislope MUSCL methods require that the cell center of the considered cell be located inside the triangle constructed by the cell centers of the surrounding neighbors (cf Figure 7).
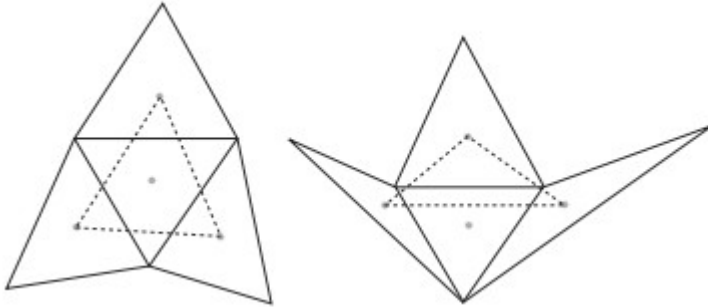
Figure 7

Configuration that satisfies (left) and does not satisfy (right) the assumption

4.1 Improved 2D node evaluation method

Motivated by the reconstruction method proposed in the work of Hou et al,[12] which is based on the fact that in advection-dominated problems, the flux at the interface mainly depends on the variables transported normal to this interface, the following TVD reconstruction is suggested.

1. Draw a line perpendicular to the considered edge, which passes through the center of the edge.

2. Find the intersection node U of the normal line and the line connecting the neighboring cell centers that are the outside neighbors of the edges in the considered cell except the considered edge (cf Figure 8A).

3. Interpolate (if U is located between E and F) or extrapolate (if U is not located between E and F).

4. Reconstruct the values.

(A) UEM   (B) UEQ
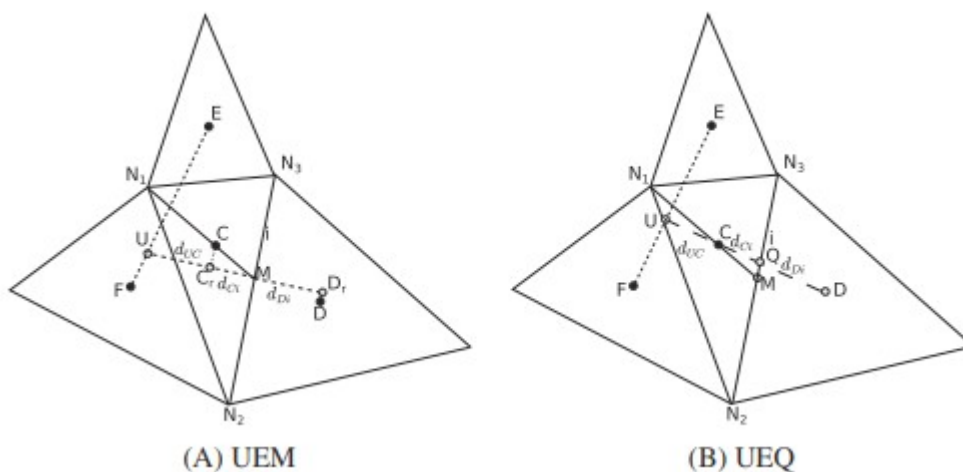
Figure 8

Stencils for (A) the 2-dimensional (2D) nodal evaluation method along the edge normal direction (UEM) and (B) the 2D nodal evaluation method along the cell centerline (UEQ)

The main novelty of this method is the choice of the upstream node U. The upwind cell of the considered cell for the considered edge is chosen from the neighboring cells of the considered cell. As illustrated in Figure 8A, these are cells E and F; thus, both of these 2 cells can be thought of as the upwind cell, but both of these 2 cell centers are far from the normal line of the considered edge. Thus, the value of the upstream node is calculated by interpolating or extrapolating the values at the center of E and the center of F. There are 2 ways to draw a line to determine the position of the upstream node U: (i) the line is drawn through the edge center M in the normal direction of the edge (cf Figure 8A), so-called UEM; or (ii) the line is drawn through the cell center of C to the projection point $C_r(|\overrightarrow{CC_r}|)$ and the distance from cell center D to the projection point $D_r(|\overrightarrow{DD_r}|)$ may be large in comparison to the distances $|\overrightarrow{C_rD_r}|$ and $|\overrightarrow{C_rU}|$, leading to numerical diffusion and increased mesh dependency.

4.2 Improved vector manipulation method

In Section 3.3, the discussion of occurring local extrema is based on Hou's vector manipulation method.[13] The MUSCL reconstruction interpolates the cell-center values to the edge along the gradient of the edge direction or, in other words, along the outside neighbor cells. Special attention has to be given to the downwind slope calculation in order to preserve the TVD property. In this study, it is suggested to use the geometric vector relationship to calculate the downwind slope by connecting both cells across the considered edge and construct the vector from the outside cell center to this edge center, as shown in Figure 9. Then, we have

$$\overrightarrow{CM} = \overrightarrow{CD} + \overrightarrow{DM} \quad (48)$$

so that

$$\overrightarrow{r_{CM}} = \frac{|\overrightarrow{CD}|}{|\overrightarrow{CM}|}\overrightarrow{r_{CD}} + \frac{|\overrightarrow{DM}|}{|\overrightarrow{CM}|}\overrightarrow{r_{DM}} \quad (49)$$

$$\nabla q_{CM} = \frac{|\overrightarrow{CD}|}{|\overrightarrow{CM}|}\nabla q_{CD} + \frac{|\overrightarrow{DM}|}{|\overrightarrow{CM}|}\nabla q_{DM}. \quad (50)$$

(A) VEM   (B) VEQ
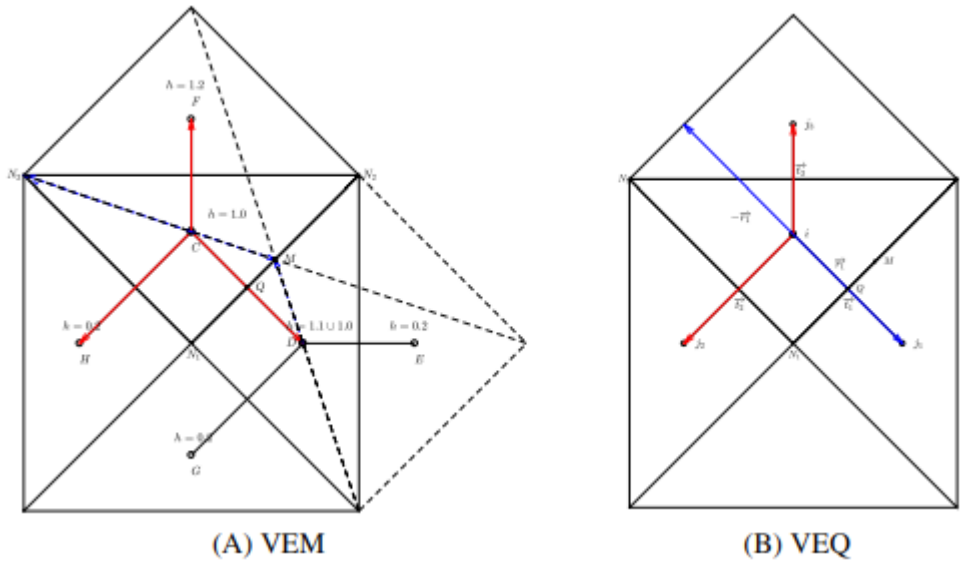
Figure 9

Stencils for (A) the improved vector manipulation method (VEM) and (B) Buffard and Clain's vector manipulation method (VEQ)

The additional computational effort is very small compared to the scheme of Hou et al,[13] but the slope for calculating the downwind solely depends on the variables of the downwind direction (outside cell). Recalculating the water depth in Section 3.3, if $h_D=1.0$, the downwind slope is calculated as $\nabla q_{CM}<0.0$, and hence, q_{CM} will be reconstructed first-order accurate. If $h_D=1.1$, the downwind slope is calculated by Equation 49, giving $\nabla q_{CM}=-0.8064$. Then, q_{CM} will also be reconstructed first-order accurate. When we observe the meshes, it can be seen that the line CM will pass through the center of cells D and E, and cell E shows a decreased slope from cells C and D. Therefore, it is meaningful to use a first order for the reconstruction of the values at the considered edge.

In comparison, consider the method by Buffard and Clain,[8] where the slope is calculated along the line connecting the center of the considered cell with the center of the neighbor cell (cf Figure 9B). While this method is able to accurately calculate downwind slopes, the extrapolated values at the considered edge are calculated not in the center of this edge M but rather at the intersection point Q and, therefore, may not represent the averaged values at the considered edge. Hereinafter, the improved vector manipulation method is referred to as VEM, and the method by Buffard and Clain is referred to as VEQ.

4.3 Comparison of the multislope MUSCL reconstruction procedures in a cell

The procedures of MUSCL reconstruction methods used in this work are summarized and compared in Table 1.

TABLE 1 Procedures of multislope monotonic upstream-centered scheme for conservation laws reconstructions

Steps	2-Dimensional Nodal Evaluation Q Method (UEQ); Choose Figure 8B as Legend	Vector Manipulation Extrapolate Q Method (VEQ); Choose Figure 9B as Legend
1	Calculate the coordinate of intersection point U of line CD and the neighbor cell centers connecting line EF	Compute $\vec{r}_k$ and $\vec{k}$
2	Interpolate value U from values at centers E and F	Solve Equation (22) to get coefficients β_1 and β_2
3	Extrapolate the edge value by following Equations (16) and (17)	Calculate $\nabla q_{C,(D,F,H)}$ from Equation (20)
4	–	Extrapolate $\nabla q_{N,C}$ from Equation (23)
5	–	Extrapolate the edge value by following Equation (25)

Steps	Vector Manipulation Extrapolate M Method (VEM); Choose Figure 9A as Legend	Old Vector Manipulation Extrapolate M Method (HOU); Choose Figure 9A as Legend
1	Compute $\vec{r}_k$ and $\vec{t}_k$	Compute $\vec{r}_k$ and $\vec{t}_k$
2	Solve Equations (21) and (22) to get coefficients α_1, α_2, β_1, and β_2	Solve Equations (21) and (22) to get coefficients α_1, α_2, β_1, and β_2
3	Calculate $\nabla q_{C,(D,F,H)}$ and $\nabla q_{D,(C,E,G)}$ from Equation (20)	Calculate $\nabla q_{C,(D,F,H)}$ and $\nabla q_{D,(C,E,G)}$ from Equation (20)
4	Evaluate ∇q_{CM} and $\nabla q_{N,C}$ from Equation (23)	Evaluate ∇q_{CM} and $\nabla q_{N,C}$ from Equation (23)
5	Update ∇q_{CM} with Equation (50)	Extrapolate the edge value by following Equation (25)
6	Extrapolate the edge value by following Equation (25)	–

In this work, a modified van Leer's limiter function from the work of Hou et al[6] is adopted for 2D nodal evaluation methods, ie,

$$\Psi(r) = \frac{0.5Rr + 0.5R|r|}{R - 1 + r}. \quad (51)$$

Here, $R = (d_{Cf} + d_{Df})/d_{Cf}$. Moreover, a limiter function from the work of Delis and Nikolos[29] is adopted here for vector manipulation methods, ie,

$$\Psi(a, b) = \begin{cases} \frac{(a^2 + e)b + (b^2 + e)a}{a^2 + b^2 + 2e}, & \text{if } ab > 0 \\ 0, & \text{if } ab \le 0. \end{cases} \quad (52)$$

4.4 Boundary treatment

In this work, the boundary conditions are treated by following the description in the work of Hou et al[16]; in order to maintain the high order inside the computational domain, a ghost cell technique is applied here. We will individually discuss the different MUSCL methods presented in Table 1 based on the legend of Figure 10.

• UEQ: The ghost cells are used here for storing the values of the boundary information. In order to make sure that all internal cells have neighboring cells out of the local internal edge, the geometric information is stored by mirroring the internal neighboring cell. After storing the geometric information in the ghost cells, a 2-step boundary interpolation will be carried out to preserve high-order accuracy for the whole computational domain.

o   The internal neighboring cells *ACD* and *BDE* will use cell-center values for calculating the ghost cells' values by using the boundary conditions from the work of Hou et al.[16]

o   The interpolation of edge values at edges *CD* and *DE* will be calculated based on the UEQ method, and the ghost cells' value will be chosen as the downwind cell value. After that, a new loop calculation for boundary edge values at *DC* and *ED* will be updated by the boundary conditions.[16] The ghost cells' values will remain the same as the boundary edges', and thus, all internal cells and edges have neighboring cells for high-order reconstruction.
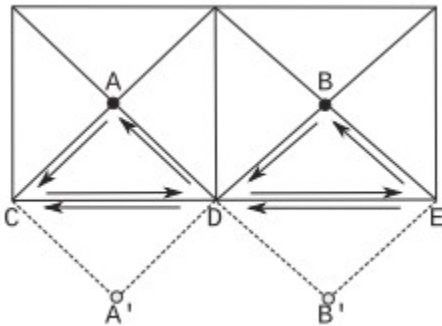
Figure 10

Ghost cell techniques used for the monotonic upstream-centered scheme for conservation laws reconstructions

- VEM, VEQ, and HOU: With the ghost cell technique, the slope from the center of the ghost cell to the edge center will be thought of as 0.0.

o   Compute the ghost cells' values using the same method as UEQ. The downwind slope of the internal cell will be thought of as the value difference between the ghost cell and the internal cell divided by the distance from the internal cell center to the edge center and the edge perpendicular point for the VEM (same as HOU) and VEQ, respectively.

4.5 A novel approach for restraining the reconstruction instability

The instability of MUSCL reconstructions is mainly caused by overestimated velocities,[30] which will yield an overestimated flux across the cell edges. In order to avoid numerical instability, the velocities at the edge should satisfy the following monotonicity condition:

$$\max(u_C, u_D) \geq u_M^L \geq \min(u_C, u_D). \quad (53)$$

Here, u_C, u_D, and u_M^L mean the velocities at the cell inside and outside and the velocity at the edge after interpolation.

If the case that creates local maxima in velocity is considered, if the unit discharges have the same sign, ie, $q_C q_D \geq 0$, then , and Equation 53 can be rewritten as

$$\min\left(|u_C|, |u_D|\right) \leq |u_M^L| \leq \max\left(|u_C|, |u_D|\right) \quad \text{if} \quad q_C q_D \geq 0. \tag{54}$$

This means that if the discharges have the same sign, the absolute value of the discharge can be considered for simplification. If the slope of the absolute value of the discharge and the slope of the water depth have different signs, no local maximum will be created, and therefore, we have

$$\min\left(|u_C|, |u_D|\right) \leq |u_M^L| \leq \max\left(|u_C|, |u_D|\right) \quad \text{if} \quad \frac{d\left(|q|\right)}{dx} * \frac{dh}{dx} \leq 0 \land q_C q_D \geq 0. \tag{55}$$

It is always true for Equation 55 that, besides this, if discharge with the same signs is considered, 2 distinct cases can be identified, which will create local extrema in the velocity: (i) increasing water depth and (ii) decreasing water depth.

For the increasing water depth, if the velocity is beyond the range of the adjacent velocities, again, 2 distinct cases have to be considered.

1.   For the larger velocity interpolation: Velocities are calculated by using the discharge divided by the water depth. If the velocity is bigger than the adjacent velocities (absolute values), we can conclude that the discharge is overestimated. In this case, we will use the water depth h_M^L multiplied with the velocity with the larger absolute value. This means that the increasing estimated situation (slope) of discharge is not changed, but adjusted by imposing a limit to the MUSCL reconstruction of the discharge, as shown in Figure 11A.

2.   For the smaller velocity interpolation: As before, we want to limit the MUSCL reconstruction values but do not want to change the slope sign for the MUSCL reconstruction. In other words, the aim of the additional limitation is just to limit the relatively bigger overestimate. For this smaller velocity interpolation, we think about the modification of water depth, increasing the absolute value of discharge may give local extrema, as shown in Figure 11B.

(A)   (B)   (C)   (D)
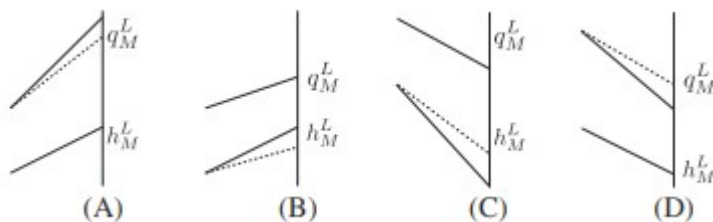
Figure 11

When $q_C q_D \geq 0$, the local extrema for velocities created by monotonic upstream-centered scheme for conservation laws reconstruction: (A) dh/dx>0, $|u_M^L| > \max(|u_C|, |u_D|)$, the overestimated result is controlled by limiting the q_M^L; (B) dh/dx>0, $|u_M^L| < \min(|u_C|, |u_D|)$, the overestimated result is controlled by limiting the h_M^L; (C) dh/dx<0, $|u_M^L| > \max(|u_C|, |u_D|)$, the overestimated result is controlled by limiting the h_M^L; and (D) dh/dx<0, $|u_M^L| < \min(|u_C|, |u_D|)$, the overestimated result is controlled by limiting the q_M^L

For the decreasing water depth, if the velocity is beyond the range of velocities, we also give 2 situations for consideration.

•   For the larger velocity interpolation: If the velocity is larger than the range of velocities (absolute values), the water depth is underestimated. The water depth will be calculated as the edge discharge q_M^L divided by the larger velocity of both sides, as shown in Figure 11C.

•   For the smaller velocity interpolation: If the velocity is underestimated for the conditions of the water depth and the absolute discharge is decreasing, the discharge is underestimated, so that the discharge will be calculated as the water depth multiplied with the smaller velocity of both sides, as shown in Figure 11D.

We can summarize as follows:

$$
\text{if } q_C * q_D \geq 0
\begin{cases}
q_M^L = h_M^L \max\left(|u_C|, |u_D|\right) q_M^L / \left|q_M^L\right|, & \text{if } dh/dx \geq 0, \left|u_M^L\right| > \max\left(|u_C|, |u_D|\right) \\
h_M^L = q_M^L / \min\left(|u_C|, |u_D|\right) q_M^L / \left|q_M^L\right|, & \text{if } dh/dx \geq 0, \left|u_M^L\right| < \min\left(|u_C|, |u_D|\right) \\
h_M^L = q_M^L / \max\left(|u_C|, |u_D|\right) q_M^L / \left|q_M^L\right|, & \text{if } dh/dx \leq 0, \left|u_M^L\right| > \max\left(|u_C|, |u_D|\right) \\
q_M^L = h_M^L \min\left(|u_C|, |u_D|\right) q_M^L / \left|q_M^L\right|, & \text{if } dh/dx \leq 0, \left|u_M^L\right| < \min\left(|u_C|, |u_D|\right).
\end{cases}
$$

(56)

The case for both sides of the considered edge with different signs for the discharge is limited in a similar way. We analyzed the situation with the increasing and decreasing water depths, and as shown in Figure 12A-D, all kinds of situations are listed, which may occur for both sides with different signs for the discharges: the left side is the condition that will not create an extreme velocity, and the right side is the one where this situation may happen. The limitation can be summarized as follows:

$$
\text{if } q_C * q_D < 0
\begin{cases}
q_M^L = h_M^L \max(u_C, u_D), & \text{if } dh/dx \geq 0, u_M^L > \max(u_C, u_D) \\
q_M^L = h_M^L \min(u_C, u_D), & \text{if } dh/dx \geq 0, u_M^L < \min(u_C, u_D) \\
h_M^L = q_M^L / \max(u_C, u_D), & \text{if } dh/dx \leq 0, u_M^L > \max(u_C, u_D) \\
h_M^L = q_M^L / \min(u_C, u_D), & \text{if } dh/dx \leq 0, u_M^L < \min(u_C, u_D).
\end{cases}
$$
(57)
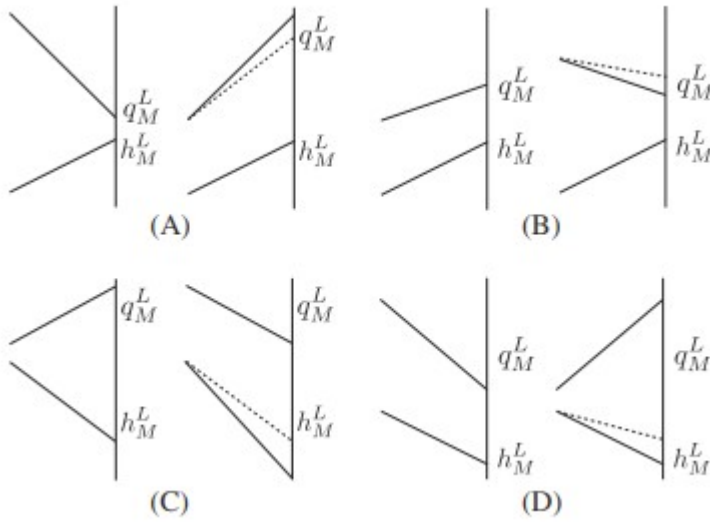
Figure 12

When $q_c q_D < 0$, the left-hand side of images (A)-(D) shows the situations that cannot create the local extrema velocities. For the right-hand side, we have that: (A) dh/dx>0, $u_M^L > \max(u_C, u_D)$, the overestimated result is controlled by limiting the q_M^L; (B) dh/dx>0, $u_M^L < \min(u_C, u_D)$, the overestimated result is controlled by limiting the q_M^L; (C) dh/dx<0, $u_M^L > \max(u_C, u_D)$, the overestimated result is controlled by limiting the h_M^L; and (D) dh/dx<0, $u_M^L < \min(u_C, u_D)$, the overestimated result is controlled by limiting the h_M^L

It can be observed that the water depth is modified during the limitation of the velocity. In 2D SWEs, the unit discharge is usually split into x- and y-directions, which may make the limitation process more challenging, because the unit discharge in both directions can be limited independently. The modified water depth may lead to a local extremum in velocity in the direction that is not being limited, eg, if the x-direction is limited, a local extremum in the velocity in the y-direction may occur. However, the treatment described in Equations 56 and 57 restricts the slope of the water depth while its sign is ensured to stay the same. Thus, the method for reconstructing the discharges in the x- and y-directions will automatically select the smaller slope and, therefore, always satisfies the condition in both x- and y-directions. An additional challenge is that the variables have to be also reconstructed at the vertices of the cell; hence, the limiting process is applied to the whole cell. Consequently, the order of accuracy of the MUSCL reconstruction decreases, and if the values of discharge at the vertices differ significantly from the values at the cell edge, the interpolation of the discharge will be wrong. In order to control this issue, we propose to use a threshold value ε_+ for limiting very high velocities at the wet and dry interfaces. We choose this value to be $\varepsilon_+ = 100\varepsilon$, where $\varepsilon = 10^{-6}$ m is the threshold that determines whether a cell is wet (if the water depth in the cell is larger than ε) or dry (if the water depth in the cell is less than ε); compare, for example, with the works of Simons et al,[19] Liang,[20] and Hou et al.[30]

5 COMPUTATIONAL EXAMPLES

We present 5 computational test cases, 2 of which with analytical solutions. The performance of the improved MUSCL reconstructions is compared regarding accuracy, efficiency, and ease of implementation.

The first test case is the well-known Thacker's solution benchmark, which is chosen to demonstrate accuracy and the capability to deal with wet and dry interfaces of the MUSCL schemes. The second test case is chosen to verify the capability of the scheme to capture shock waves for high-speed discontinuous flow conditions and shows a steady-state oblique jump. The third test case shows a radial dam break, and the fourth test case shows a 2D Riemann problem, to demonstrate the accuracy of the presented reconstruction methods. Finally, in a last test case, a Tsunami experiment is replicated to demonstrate a near–real-world application.

Three types of meshes are employed in this work, namely, the diagonal mesh, the Scottish mesh, and the Delaunay mesh,[8] as shown in Figure 13.
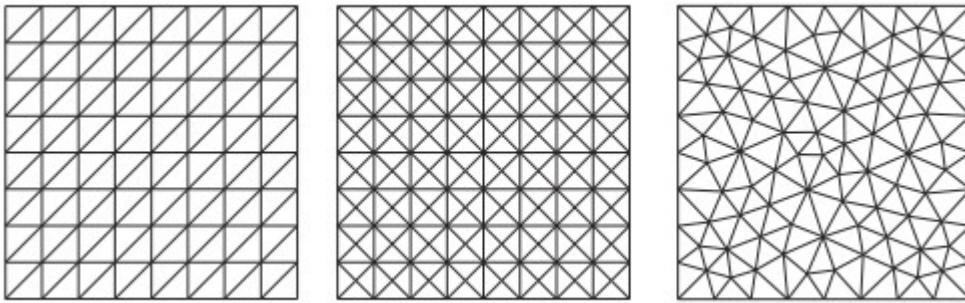
Figure 13

The 3 types of mesh employed to evaluate the effective accuracy of the schemes: the diagonal mesh (left), the Scottish mesh (center), and the Delaunay mesh (right)

5.1 Thacker's planar rotation benchmark

Thacker's planar rotation benchmark is considered to be one of the most challenging test cases for numerical codes, because it involves moving wet and dry fronts inside a parabola. The bottom topography is defined as

$$z_b(x, y) = -h_0 \left[1 - \frac{(x - x_0)^2 + (y - y_0)^2}{a^2} \right], \quad (58)$$

where (x_0, y_0) represent the coordinates of the parabola center, h_0 is the water depth at the parabola center, and a is the distance from the center point to the zero elevation of the shoreline. The analytical solution of this test case is given as

$$\eta_t = \frac{\sigma h_0}{a^2} [2(x - x_0) \cos(\omega t) + 2(y - y_0) \sin(\omega t)], \quad (59)$$

$$u(t) = -\omega \sin(\omega t), \quad v(t) = \omega \sigma \cos(\omega t), \quad (60)$$

where σ is a constant value, and $\omega = \sqrt{2gh_0}/a$ is the angular velocity of the rotation. In this work, parameters are set to be the same as in the work of Hou et al[13] with h_0=0.1 m, a=1.0 m, and σ=0.5, and the computational domain is set to be 4×4 m² with the domain center at (2 m, 2 m). All boundaries are closed boundaries.

The characteristic length $\Delta x = \sqrt{A/N}$ is used to set the resolution of the meshes, where A is the area of the computational domain and N is the total number of cells.

The accuracy is expressed as the L_1-error, which can be calculated as

$$L_1 = \frac{\sum_1^n |q_i - q_{i,\text{exact}}| A_i}{\sum_1^n A_i}. \tag{61}$$

The upper part of Figures 14-18 plots the water depth contour of FOU, HOU, UEQ, VEQ, and VEM compared to the analytical solution, respectively, at t=3.5T and 4T, where T represents 1 period. It can be observed that VEM reaches the best agreement. HOU also yields good agreement, which is slightly less accurate than VEM. FOU yields the worst agreement and has more diffusion compared to the other schemes. The lower part of Figures 14-18 shows the velocity field plot, where it can be observed that FOU gives the most drag effect of velocity; HOU shows a significantly drag tail, which may be due to too much information from upwind, leading to more diffusive behavior; and VEM shows the best velocity field, where the vectors of the velocity almost coincide with the water depth contour. In order to compare the accuracy of different methods, we consider the worst grids to test the schemes. Cut section plots are shown in Figures 19 and 20 for Delaunay and Scottish grids presented in Figure 13. It is seen that VEM can obtain good accuracy on both grids. The difference between VEM and the other methods is not very large on the Delaunay grid; however, on the Scottish grid, VEM yields much better agreement than the other methods. HOU always gives the second best agreement. The time step size is set adaptively using the CFL criterion; therefore, the velocity influences the time step size. Even though the algorithm of VEM is slightly more complex than HOU, its computational effort is about 10% less than that of HOU, because no high velocities are reconstructed.
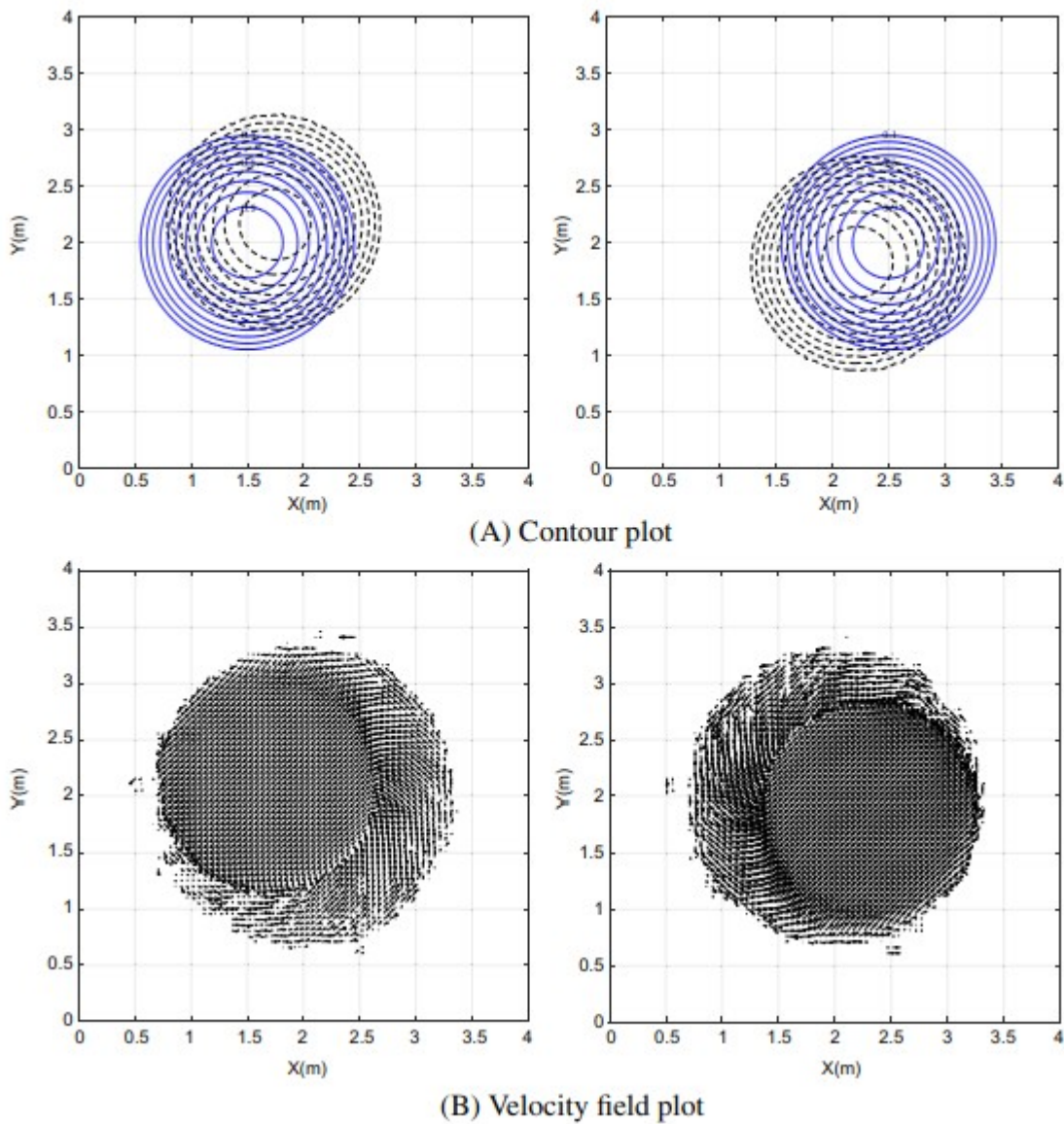
Figure 14

Thacker's planar solution: contours of water depth computed by the FOU scheme (dashed line) and the analytical solution (solid line) from Thacker at (A, left) $t = 3.5T$ and at (A, right) $t = 4.0T$, with unit (m) and with 16 364 cells of the Delaunay mesh, and (B) the corresponding velocity field

(A) Contour plot

(B) Velocity field plot
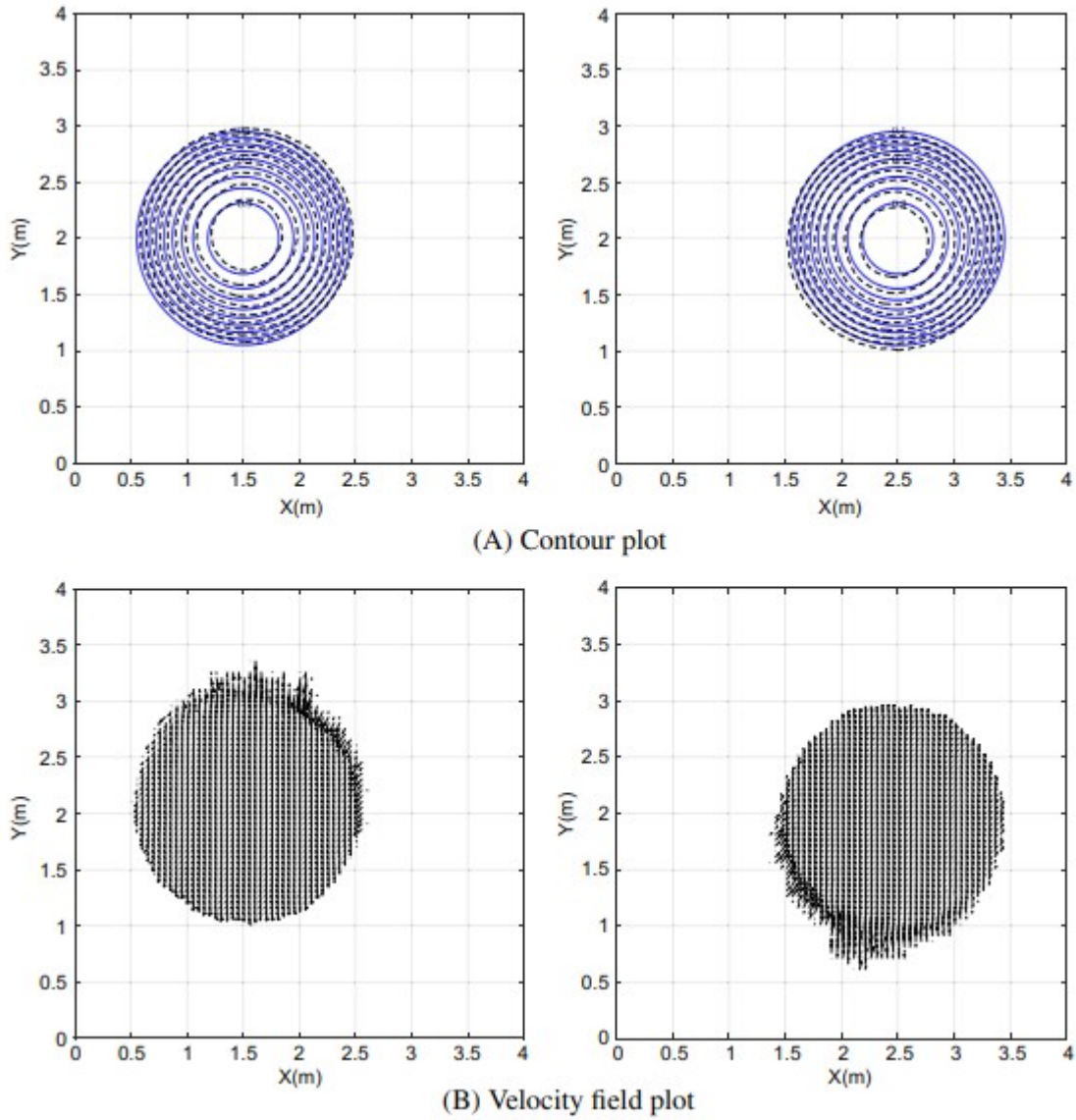
Figure 15

Thacker's planar solution: contours of water depth computed by the HOU scheme (dashed line) and the analytical solution (solid line) from Thacker at (A, left) $t = 3.5T$ and at (A, right) $t = 4.0T$, with unit (m) and with 16 364 cells of the Delaunay mesh, and (B) the corresponding velocity field
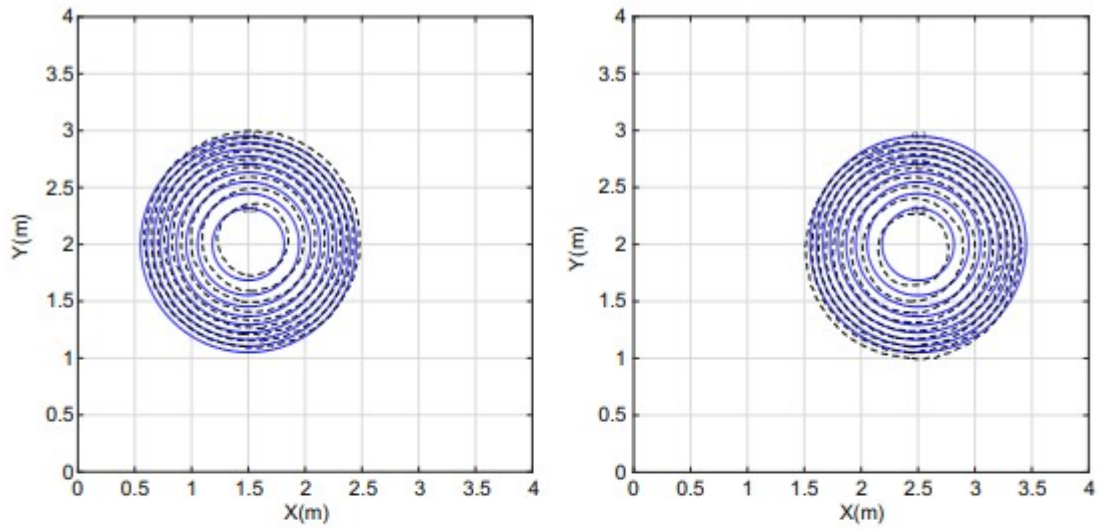
(A) Contour plot
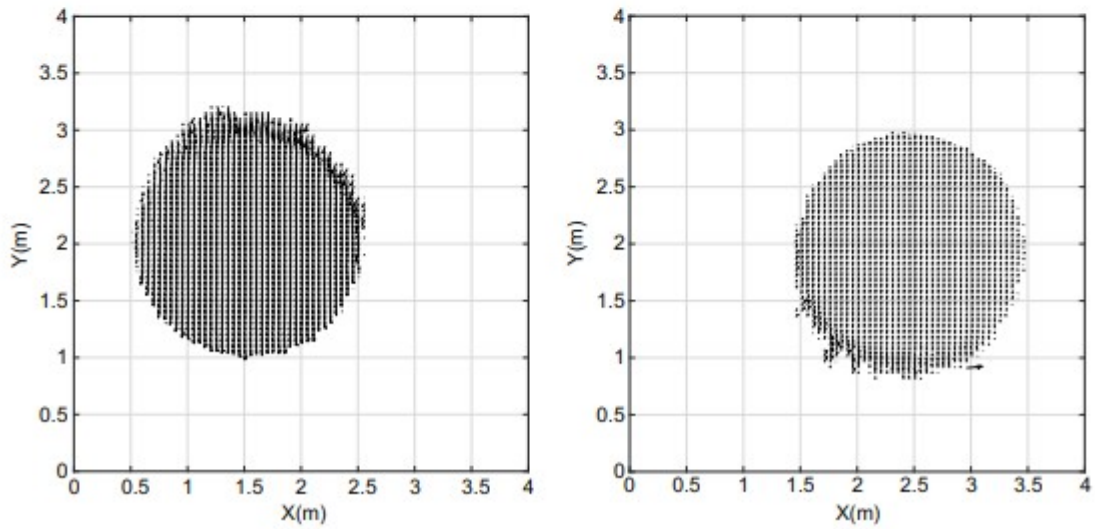
(B) Velocity field plot

Figure 16

Thacker's planar solution: contours of water depth computed by the UEQ scheme (dashed line) and the analytical solution (solid line) from Thacker at (A, left) $t = 3.5T$ and at (A, right) $t = 4.0T$, with unit (m) and with 16 364 cells of the Delaunay mesh, and (B) the corresponding velocity field

(A) Contour plot

(B) Velocity field plot
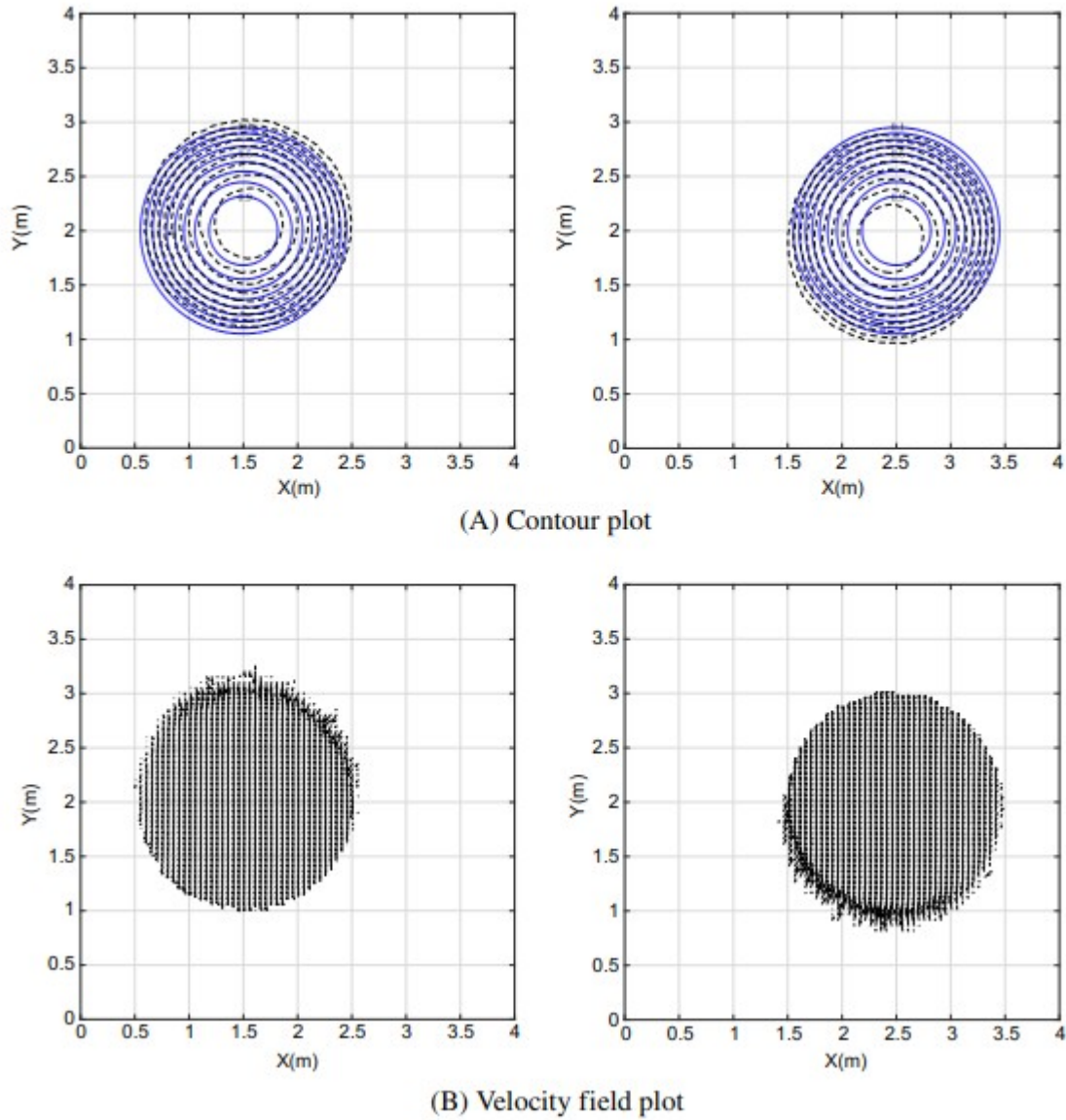
Figure 17

Thacker's planar solution: contours of water depth computed by the VEQ scheme (dashed line) and the analytical solution (solid line) from Thacker at (A, left) $t = 3.5T$ and at (A, right) $t = 4.0T$, with unit (m) and with 16 364 cells of the Delaunay mesh, and (B) the corresponding velocity field
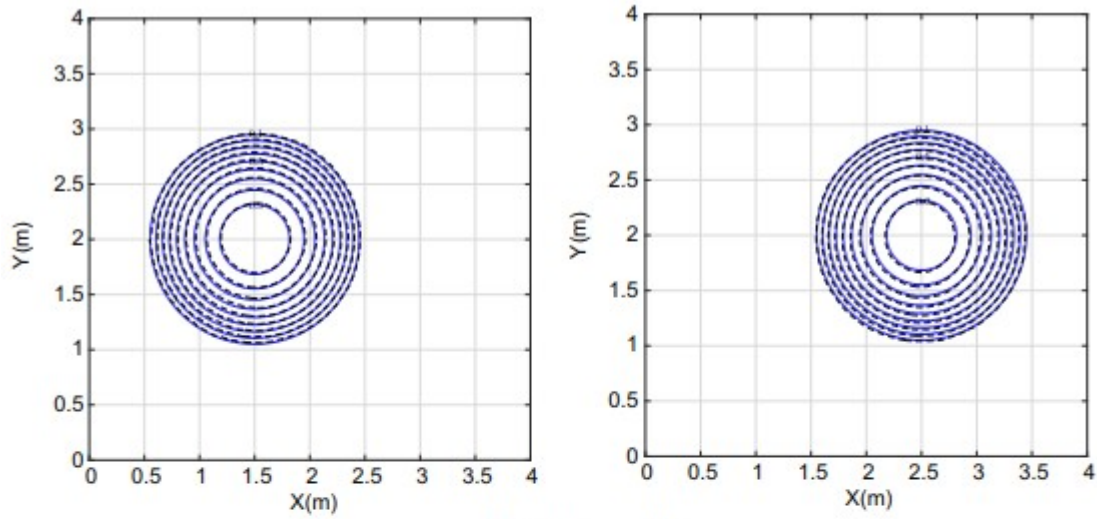
(A) Contour plot
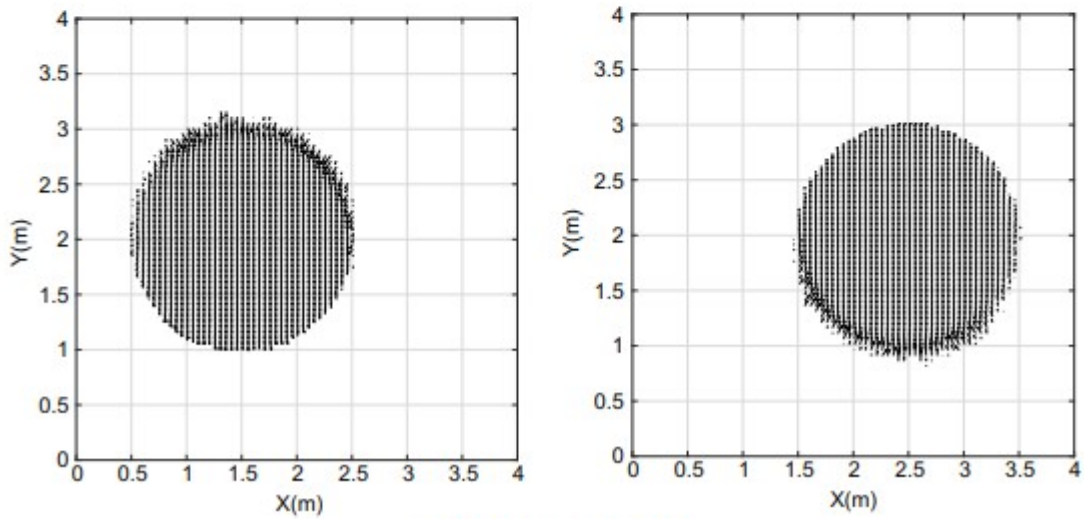
(B) Velocity field plot

Figure 18

Thacker's planar solution: contours of water depth computed by the VEM scheme (dashed line) and the analytical solution (solid line) from Thacker at (A, left) $t = 3.5T$ and at (A, right) $t = 4.0T$, with unit (m) and with 16 364 cells of the Delaunay mesh, and (B) the corresponding velocity field
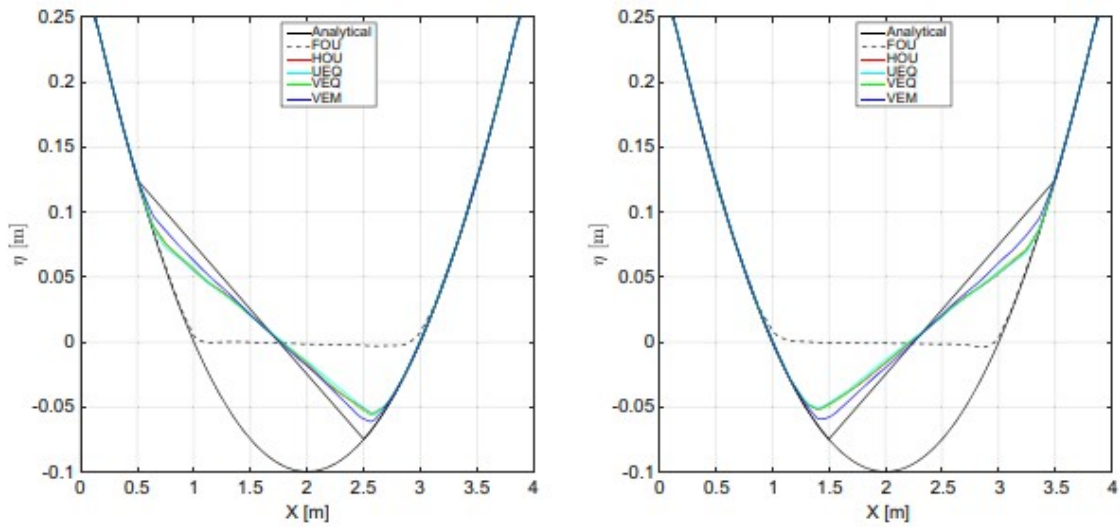
Figure 19

Thacker's planar solution: computed and analytical water levels at the cross section of $y=2$ m at $t = 3.5T$ (left) and at $t = 4.0T$ (right), with 2062 Delaunay meshes
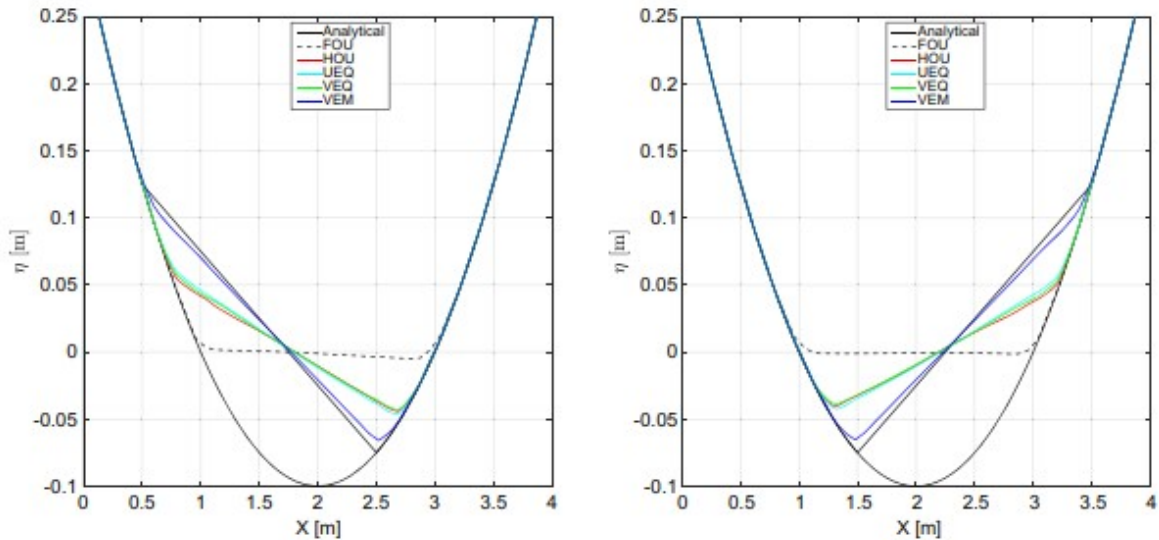
Figure 20

Thacker's planar solution: computed and analytical water levels at the cross section of $y=2$ m at $t = 3.5T$ (left) and at $t = 4.0T$ (right), with 2116 Scottish meshes

We present a mesh convergence study for this test case in Figures 21-23. The L_1-error at $t=4T$ for h and q_y is plotted in these Figures for different grids, in which the lines with symbols represent the numerical results varying with the refining of the mesh level (represented by characteristic length Δx shown in Table 2) and the solid line is of order 2. It can be seen that the results for VEQ and HOU are the same for the diagonal grid, as the middle point and the intersection point will reach the same coordinate on the diagonal grid; this also can be found in the work of Buffard and Clain.[8] For the numerical results

on the diagonal grid, all numerical results give a relatively low order; however, the VEM still can obtain the highest accuracy and numerical order. For the Scottish and Delaunay grids, the HOU scheme increases the accuracy significantly, which means that the scheme is more or less mesh dependent. Furthermore, the VEM scheme keeps the highest accuracy and numerical order. UEQ and VEQ schemes give almost slightly different results for different grids; the advantage of each scheme is related to the mesh type but is quite small. The presented MUSCL schemes add a limitation presented in Section 4.5 at wet and dry fronts. This is to prevent the occurrence of negative water depths. Consequently, wet and dry fronts reduce the overall accuracy of these schemes, and this can be thought of as the explanation as to why the accuracy order cannot reach 2 in this test case.
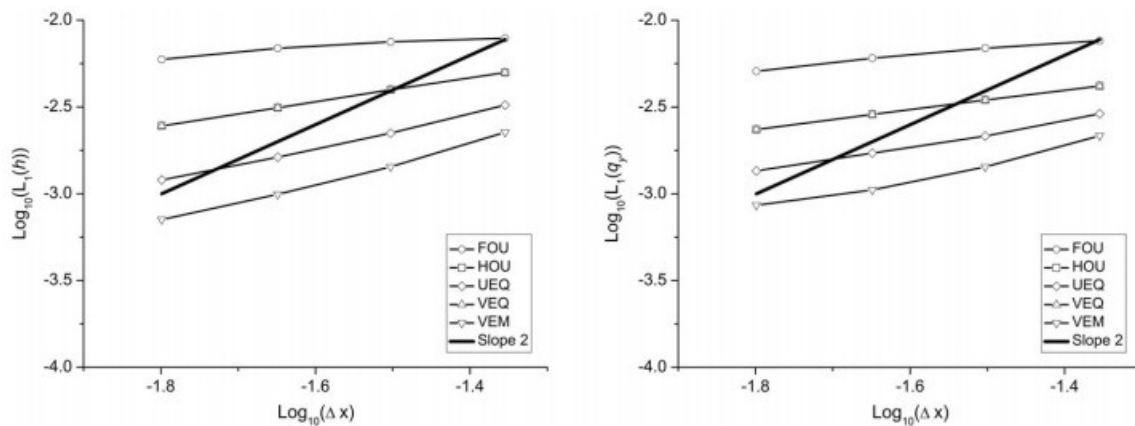
Figure 21

Thacker's planar solution: h (left) and q_y (right) convergence results for numerical schemes used on the diagonal grid at $t = 4T$
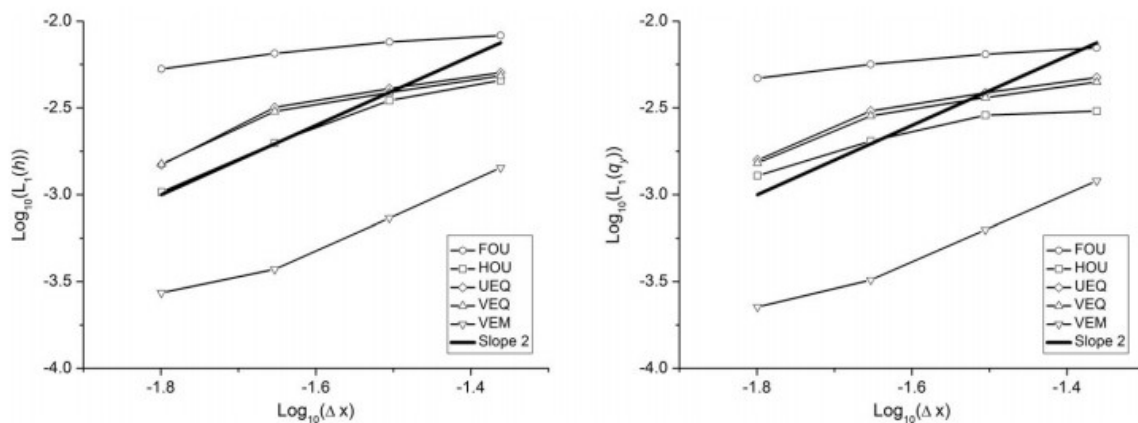
Figure 22

Thacker's planar solution: h (left) and q_y (right) convergence results for numerical schemes used on the Scottish grid at $t = 4T$
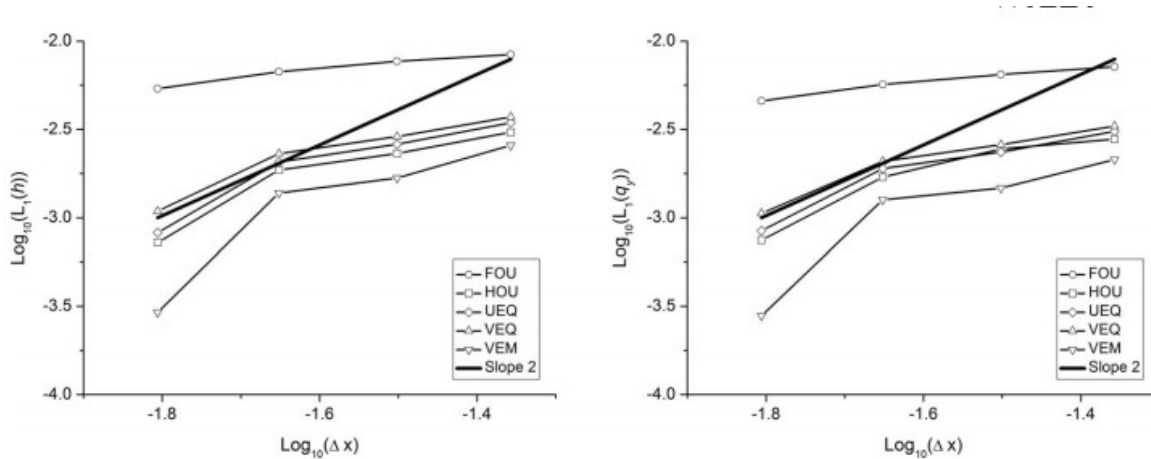
Figure 23

Thacker's planar solution: h (left) and q_y (right) convergence results for numerical schemes used on the Delaunay grid at $t = 4T$

TABLE 2 Characteristic length Δx used for the mesh convergence test

Mesh Level	Diagonal Mesh, m	Scottish Mesh, m	Delaunay Mesh, m
1	0.0442	0.0435	0.0439
2	0.0314	0.0313	0.0315
3	0.0224	0.0222	0.0223
4	0.0159	0.0159	0.0156

In summary, it can be observed that VEM yields the lowest error and the highest order in all Figures, HOU yields a slightly lower error than the other methods for the water depth, but also yields the most diffusive velocity field, except FOU. VEQ and UEQ are similar, because their algorithms are both based on using a line connecting both cell centers. However, from the results, we can conclude that the difference is negligibly small. The implementation of UEQ is more straightforward. The present results are compared to the results reported in the works of Hou et al[16] and Delis and Nikolos.[29] The cell numbers in these test cases are similar to those in the present one; therefore, a comparison is meaningful. The VEQ yields less accurate results, and the VEM yields almost similar accuracy. It is noted that both VEQ and VEM are computationally more efficient and more straightforward to implement than the approaches in the aforementioned works.([16, 29])

5.2 Steady oblique hydraulic jump

A steady-state hydraulic jump that develops when a unidirectional supercritical flow in an open channel hits an inclined solid wall is investigated. A 40 m × 30 m frictionless domain with a flat bed is used, and a uniform supercritical velocity with a Froude number of 2.7 is applied at the western inflow boundary. The initial water depth and velocity for the whole domain is set to 1 m and 8.57 m/s, respectively. The eastern boundary is a

free outflow boundary, and the northern and southern boundaries are closed boundaries. The southern wall is given an inclined angle of 8.95°, beginning from $x=10$ m to the northeastern direction. The supercritical velocity will reflect, thus creating an oblique hydraulic jump along the southern wall. Theoretically, the water will jump from 1 to 1.5 m and, starting from $x=10$ m, form a 30° angle to the x-direction. We enforce a constant time step for all methods for comparison purposes. The simulation runs for 10 seconds until the flow is steady and the hydraulic jump is formed. The simulation is carried out on a Delaunay grid (Figure 13) with 3029 cells.

Results are shown in Figures 24-28. The dashed line in the water level represents the 30° angle of the analytical solution. Except FOU, all methods diminish the numerical diffusion. On the left of the Figures, the cut sections show that the water depth increases from 1 m sharply to 10 m, and the water level almost reaches 1.4 m at $x=5$ m. All MUSCL reconstruction methods yield good accuracy for this test case.
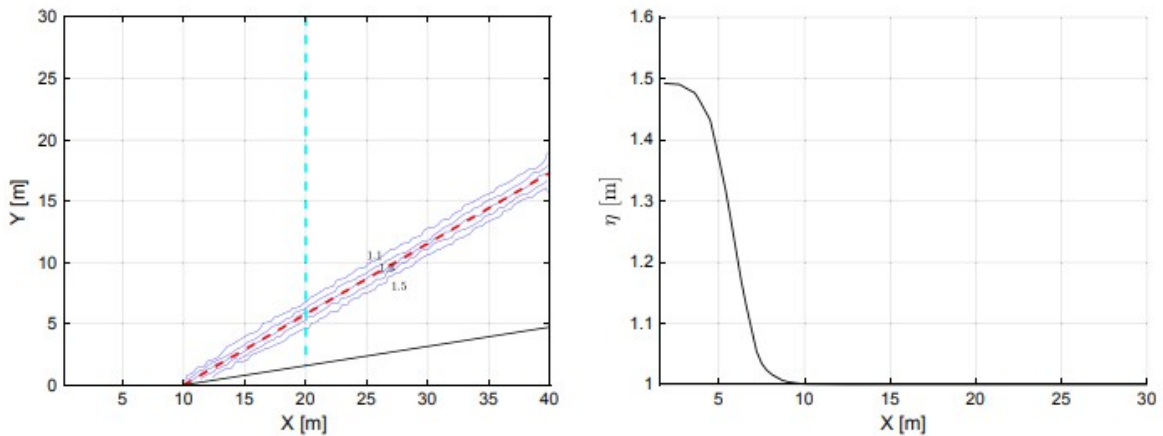
Figure 24

Water level contour (left) and cut section plot for the FOU scheme (right)
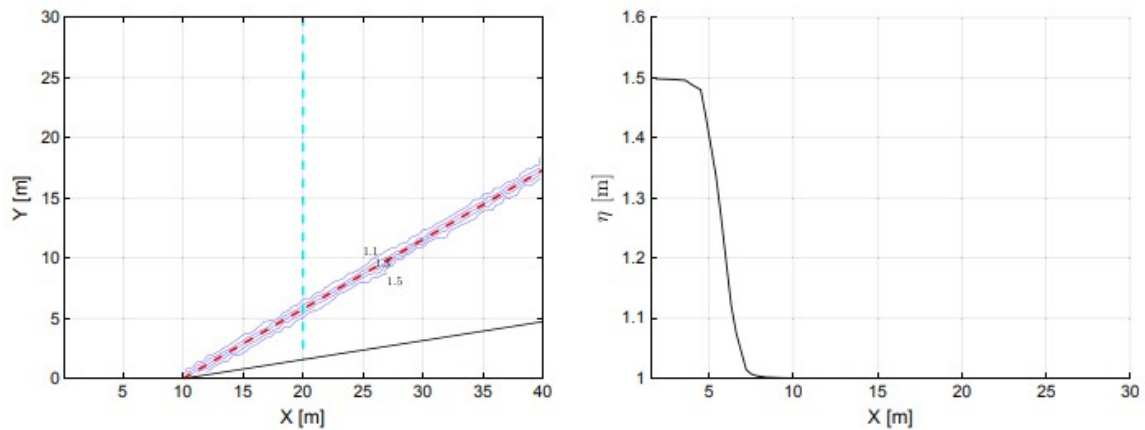
Figure 25

Water level contour (left) and cut section plot for the HOU scheme (right)
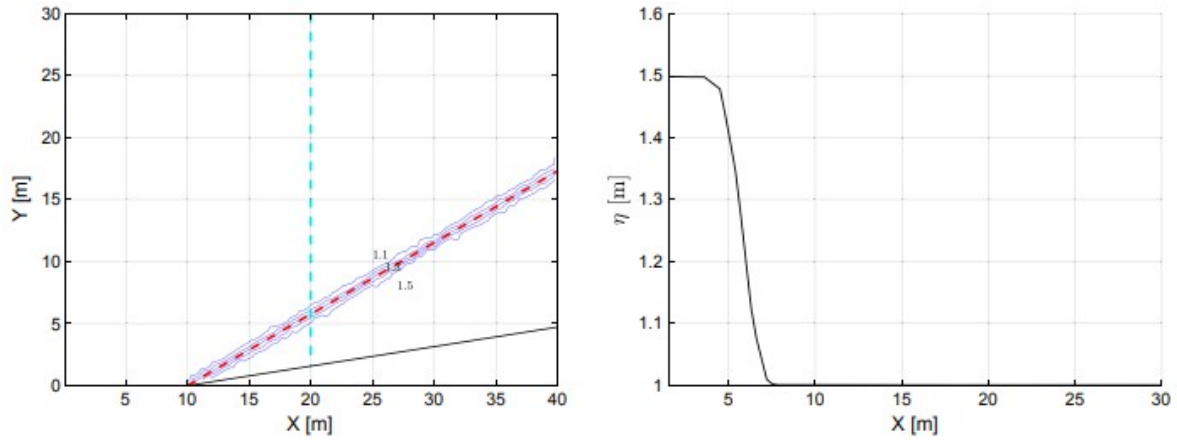
Figure 26

Water level contour (left) and cut section plot for the UEQ scheme (right)
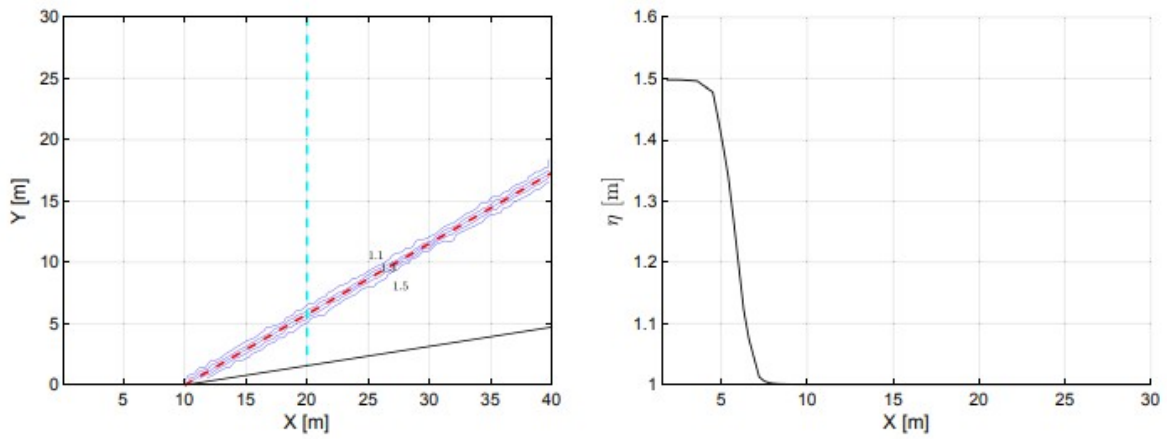
Figure 27

Water level contour (left) and cut section plot for the VEQ scheme (right)
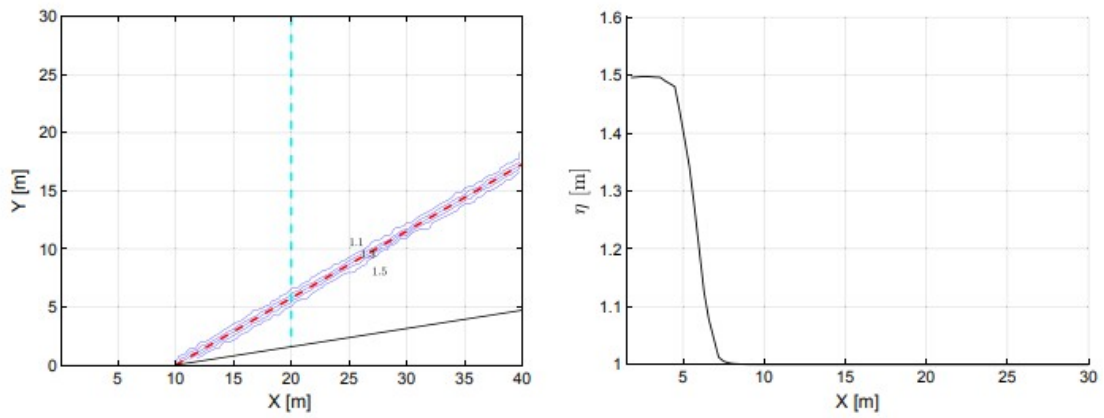
Figure 28

Water level contour (left) and cut section plot for the VEM scheme (right)

As shown in Table 3, FOU is the fastest scheme. For the MUSCL reconstructions, UEQ is the most efficient method due to the straightforward implementation and simple algorithm. Comparing to HOU, VEQ only needs to compute the linear system along the upwind direction, where we can improve the computational efficiency. The VEM takes the most computational time, but compared to HOU, the difference is quite small; the additional treatment for including the downwind information does not influence the computational efficiency too much.

TABLE 3 Oblique hydraulic jump: relative computational cost

Schemes	FOU	UEQ	VEQ	HOU	VEM
Relative computational time	1.00	1.35	1.40	1.47	1.50

5.3 Radial dam break

A 2D radial dam-break case from the work of Canestrelli et al[31] is simulated to test the capability of the reconstruction methods to preserve the symmetry of the problem. The initial conditions are

$$h(0, x, y) = \begin{cases} 2 \, \text{m}, & \text{if } \sqrt{(x - 25)^2 + (y - 25)^2} < 20.0 \\ 0.5 \, \text{m}, & \text{otherwise.} \end{cases} \tag{62}$$

The computational domain is 50 m × 50 m. The computational mesh is a Delaunay mesh (Figure 13) with 11 932 cells. The simulation is run for 3 seconds.

A reference solution is obtained using a high-resolution simulation with 1 000 000 cells on a structured grid using the high-resolution Godunov-type scheme of Clawpack.[32]

Figure 29 shows a 3-dimensional plot of VEM results at $t=1.0$ s and $t=2.5$ s. The cut section along $y=25$ m is shown in Figure 30, where the numerical results are plotted over a section that goes from (25,50) m to (50,50) m at times $t=1$ s, $t=2$ s, and $t=3$ s.
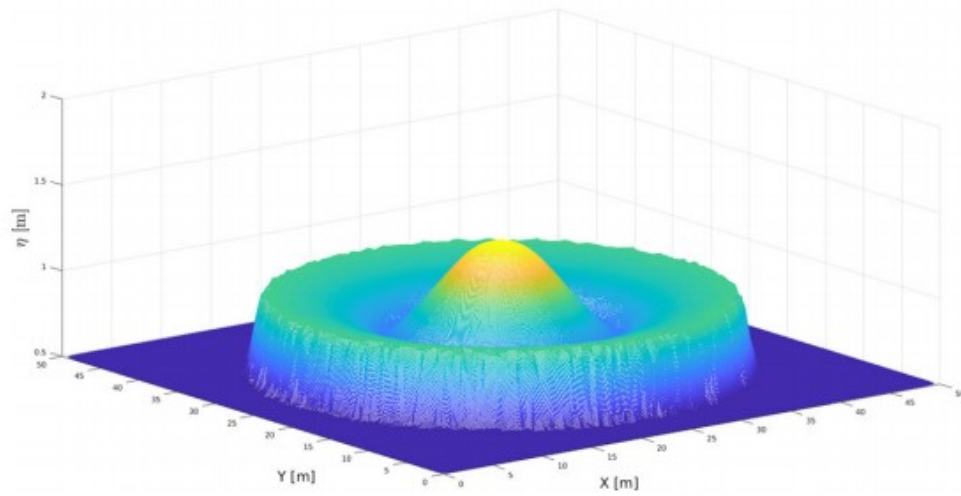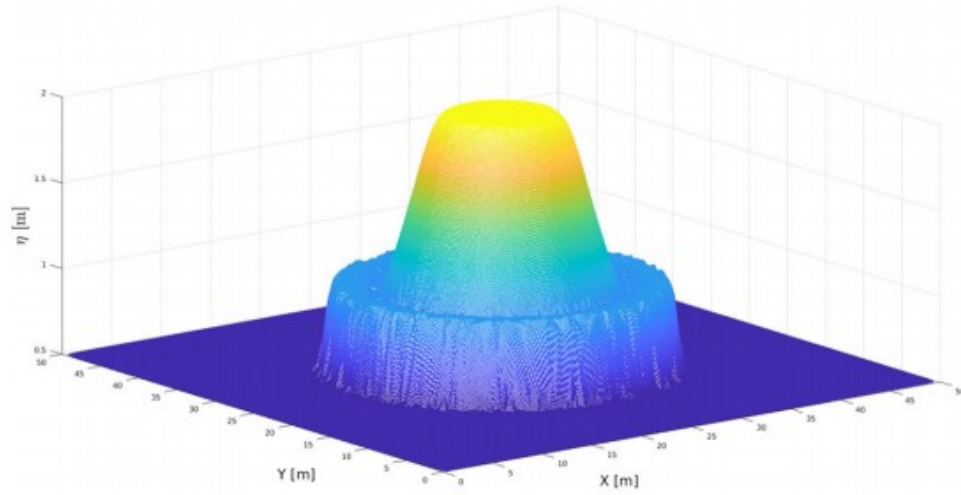
Figure 29

Radial dam-break problem: 3-dimensional view of the water level computed by VEM at $t = 1.0$ s (upper) and at $t = 2.5$ s (lower)
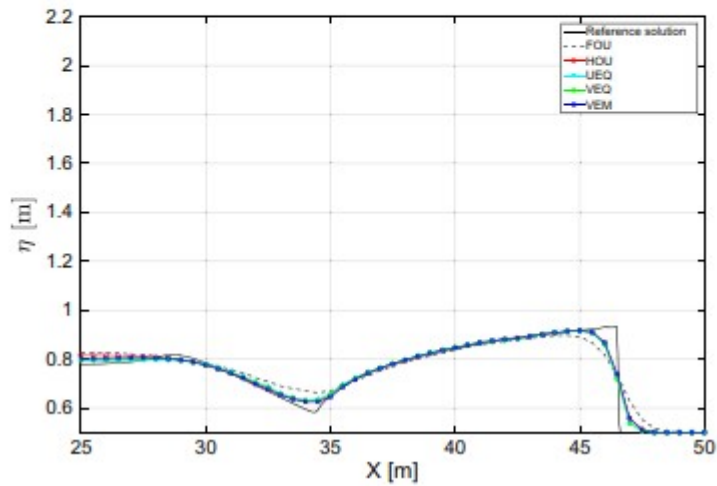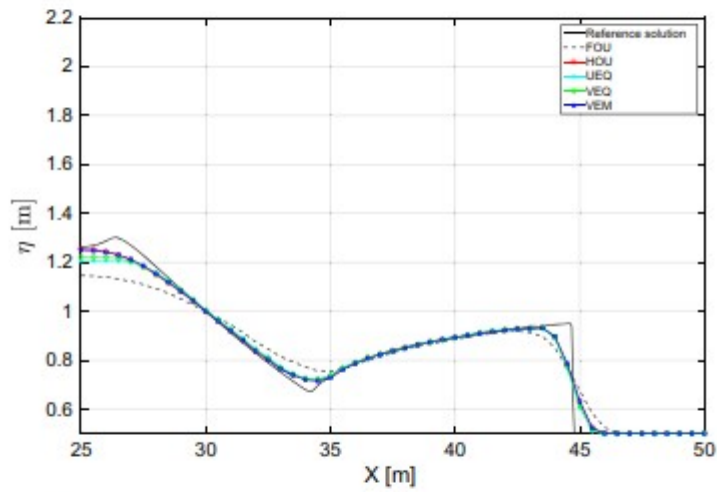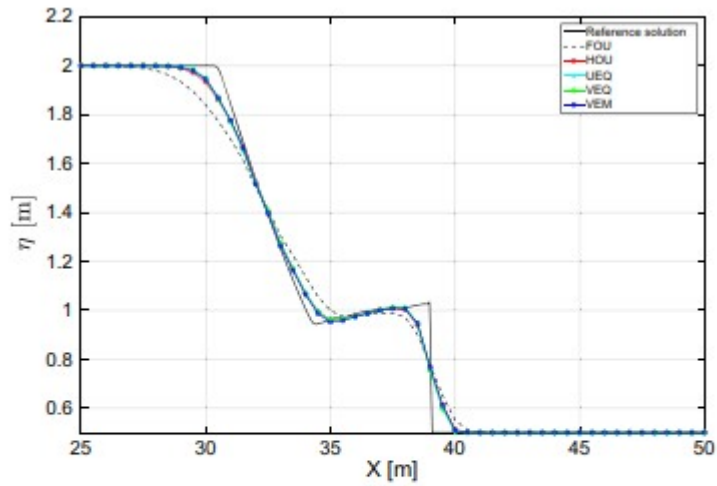
Figure 30

Radial dam-break problem: section view of the computed water level at $t = 1.0$ s (upper), $t = 2.5$ s (middle), and $t = 3.0$ s (lower)

The difference between the different MUSCL reconstructions are quite small, but at t=1 s, it can be observed that UEQ shows the best agreement, and VEM is slightly better than VEQ and HOU. The shock wave position is correctly captured, but due to numerical diffusion, the shock wave front smeared. This is because of the low resolution of the grids. For t=2 s, the water level at the domain center is only well captured by HOU and VEM, whereas UEQ shows the lowest water level (about 0.05 m lower than the reference solution). Except at the center position, the MUSCL reconstructions are near the reference solution, in which the VEM shows the best agreement, followed by HOU. For t=3 s, the shock wave is still well captured, but for the water level at the domain center, HOU shows the most numerical diffusion, and VEQ gives the best agreement with the maximum water level of the reference solution, followed by VEM.

5.4 2D Riemann problem

This test case is originally presented in the work of Guinot.[4] A frictionless [0,200]×[0,200] computational domain with an initial condition set as

$$h = \begin{cases} 10, & \text{if } x \geq 100 \wedge y \geq 100 \\ 1, & \text{else} \end{cases} \tag{63}$$

$$u = \begin{cases} 10, & \text{if } x \leq 100 \\ 0, & \text{else} \end{cases} \tag{64}$$

$$v = \begin{cases} 10, & \text{if } y \leq 100 \\ 0, & \text{else} \end{cases} \tag{65}$$

is used. The Delaunay computational mesh (Figure 13) consists of 6552 cells. In order to investigate the accuracy, a mesh that can be considered "poor" is used.

Figure 31 shows the flow pattern calculated by VEM at 1, 3, and 5 seconds. It can be observed that the shock wave positions are well captured. Due to the poor grid, the front of the shock wave is smeared. The rarefaction wave at the northeast of the domain is well resolved by the numerical scheme. The water depth contour plot and the velocity field are shown in Figure 32. As there is no analytical solution for this test case, quantitative analysis cannot be further conducted, but it can be obviously found that the present MUSCL schemes produce less diffusion than the FOU scheme.
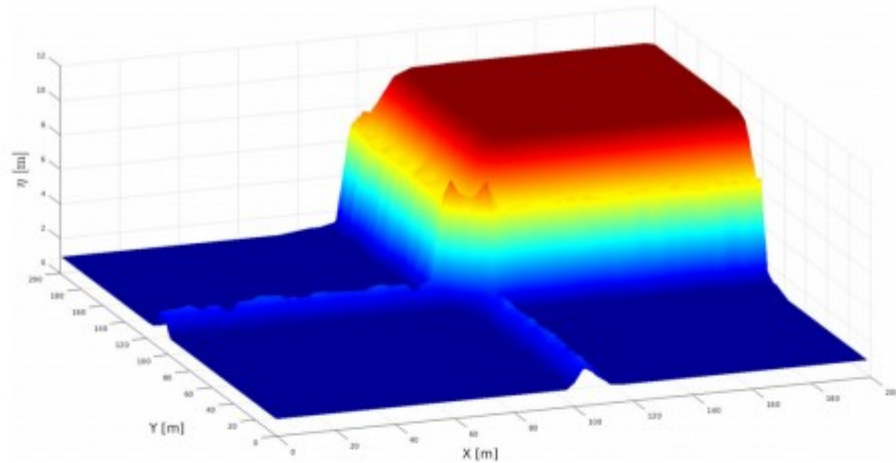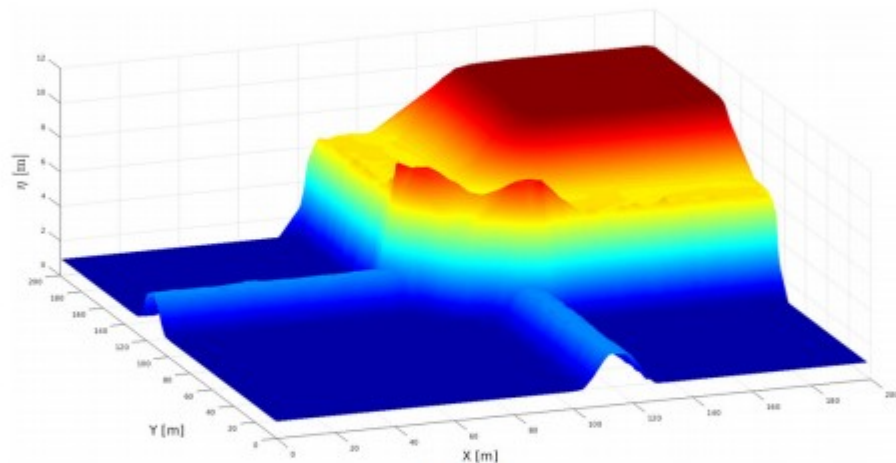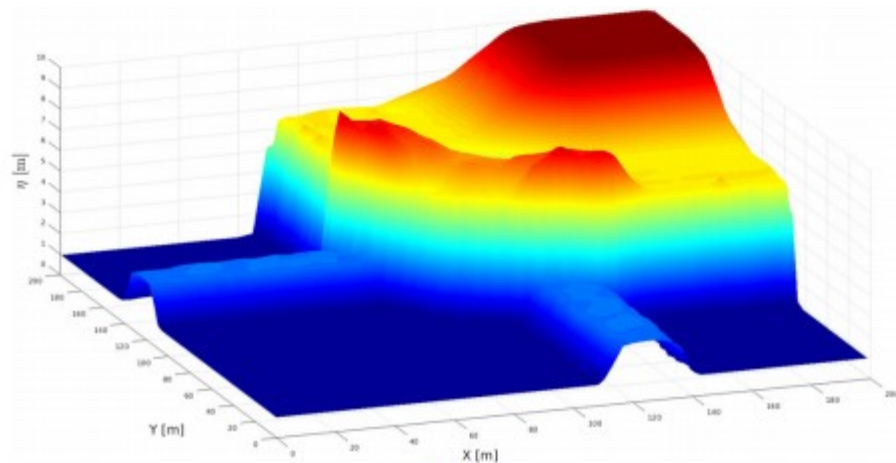
Figure 31

Two-dimensional Riemann problem: 3-dimensional views of the computational flow pattern by the VEM scheme at (A) $t = 1$ s, (B) $t = 3$ s, and (C) $t = 5$ s
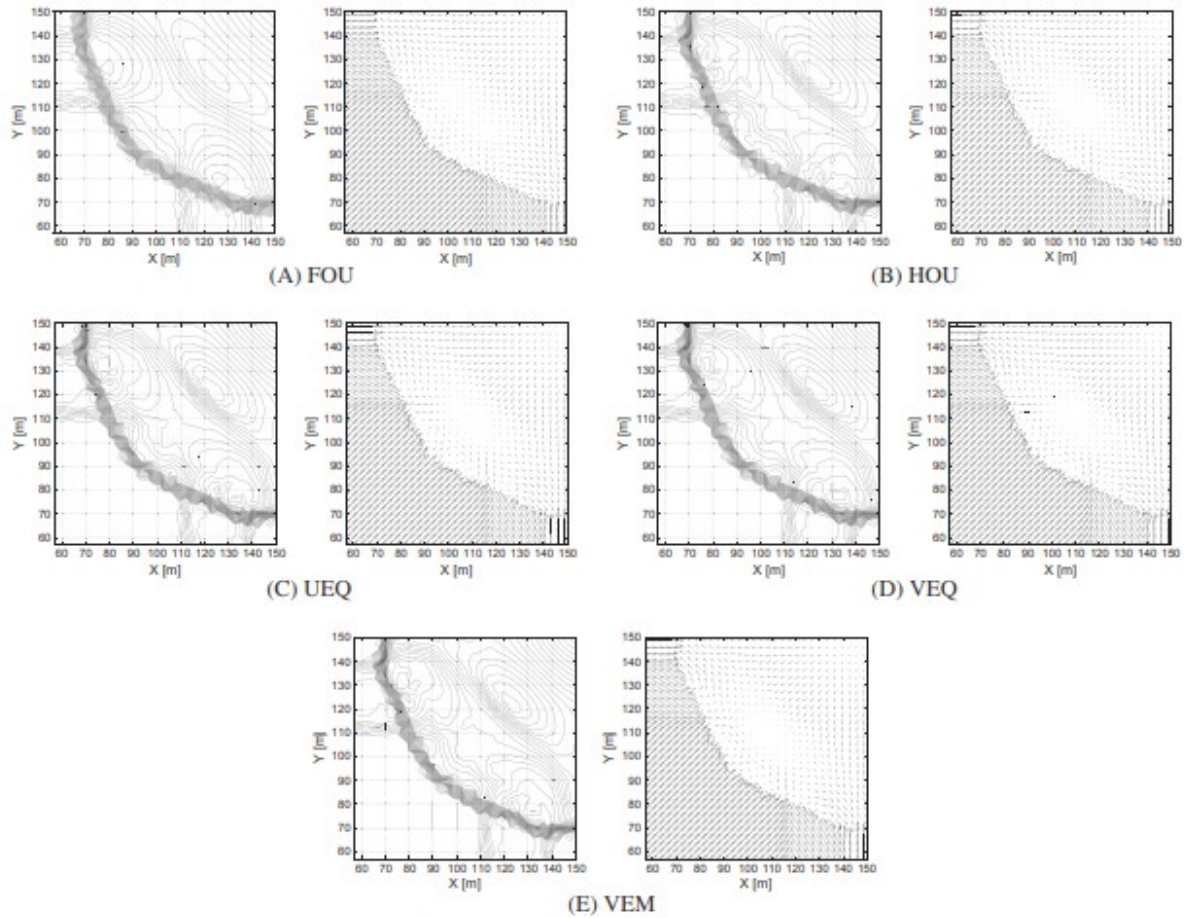
Figure 32

Two-dimensional Riemann problem: contour plots of water depths (left) and velocity vector fields (right) for the different numerical schemes (A) FOU, (B) HOU, (C) UEQ, (D) VEQ, and (E) VEM

In order to verify the order of accuracy of the methods, a cut section along [0,0] [200,200] is set inside the domain. The water depth along this cut section is shown in Figure 33. We can observe that the MUSCL reconstruction methods are quite similar, but UEQ captured a slightly steeper rarefaction wave compared to the others. According to the description in the work of Hou et al[26] and the theoretical and numerical analysis in the work of Toro,[2] the rarefaction waves are likely to be dampened by low-order schemes. Then, we can conclude that UEQ has the best performance in modeling shock problems on unstructured grids.
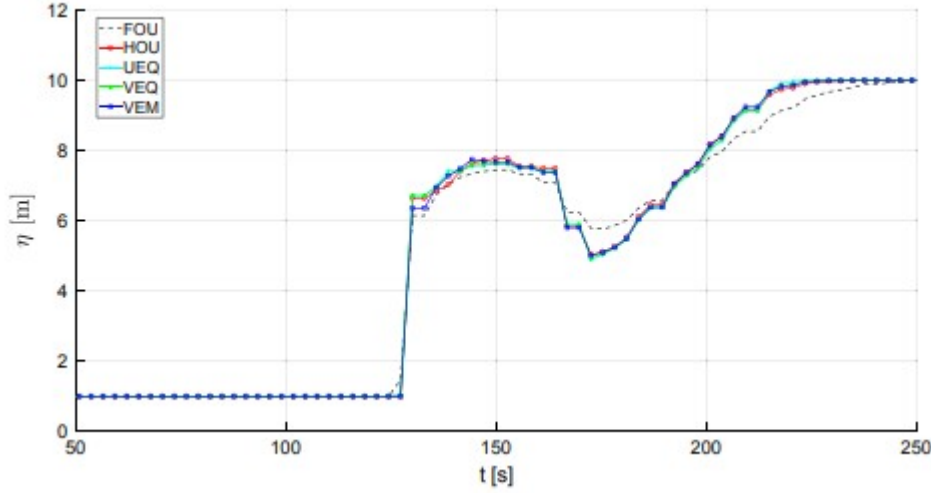
Figure 33

Two-dimensional Riemann problem: diagonal section of the water depth plot at $t = 5$ s for different numerical schemes

5.5 Tsunami wave impact on a conical island

We replicate the laboratory experiment from the work of Briggs et al[33] using the presented numerical schemes and the MUSCL reconstructions. The experiment features wet and dry fronts, uneven topography, and very shallow water depths, which are challenging for a numerical method.

The experiment domain is a [0.0, 25.92] m ×[0.0, 27.6] m rectangle (Figure 34). A Delaunay grid of 15 692 cells is used for discretization. An ideal conical island with the center located at $[x_0,y_0] = [12.96, 13.8]$ m is defined as

$$z(x, y) = \max \left[\max \left(0.625, 0.9 - 1/\left(4\sqrt{(x - x_0)^2 + (y - y_0)^2} \right) \right), 0.0 \right].$$ (66)

The initial still water level of the domain is 0.32 m, the island is partially submerged inside the water, a wave maker is set at the inflow boundary with a varying water level relative to the initial still water level, and the velocity is set as

$$\eta(t) = H\,\text{sech}^2[\sqrt{3H/4\eta_0^3} \sqrt{g(\eta_0 + H)}(t - T)] + \eta_0$$ (67)

$$u(t) = \frac{C(\eta(t) - \eta_0)}{\eta(t)}$$ (68)

$$v(t) = 0,$$ (69)

where H is the amplitude of the wave and η_0 is the still water depth; T denotes the time for the wave crest to reach the domain. The wave working condition is chosen as η_0=0.32 m, T=2.45 s, and H=0.064 m. Friction is not taken into account in this case. The simulation run time is 20 seconds.
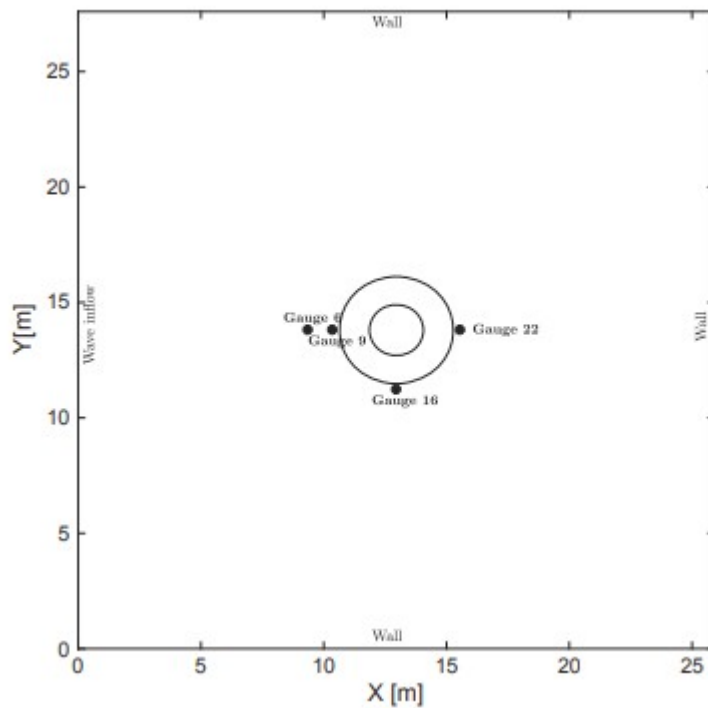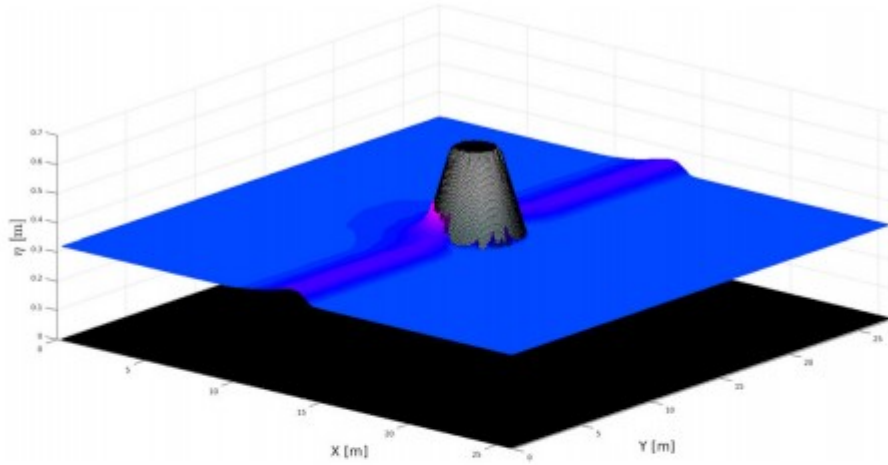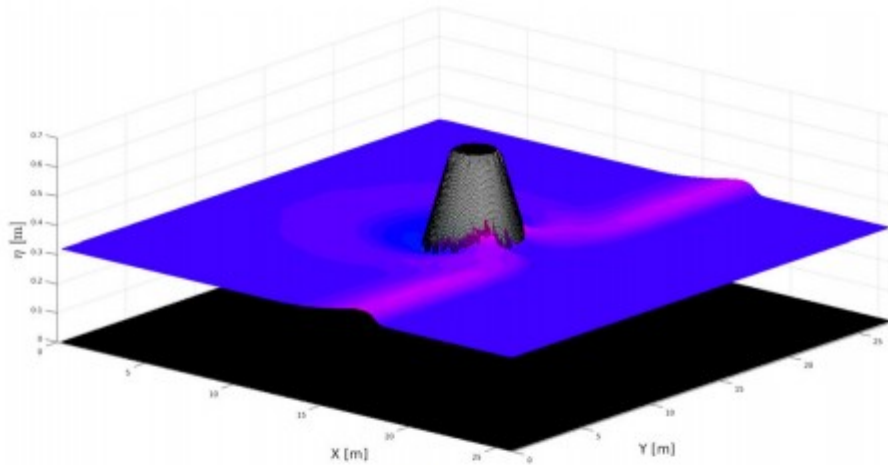
Figure 34

Tsunami wave impact on a conical island: computational domain, boundary conditions, and locations of selected gauges

Figure 35 shows the maximum wave run-up at the front, the side, and the back of the island at $t=9$ s, $t=11$ s, and $t=13$ s. Figure 36 shows oscillating solutions computed by VEM and HOU without the treatment for the velocity, respectively, which means that the velocity limitation for the wetting and drying front significantly influences the stability of the numerical scheme.

(A) t = 9s

(B) t = 11s
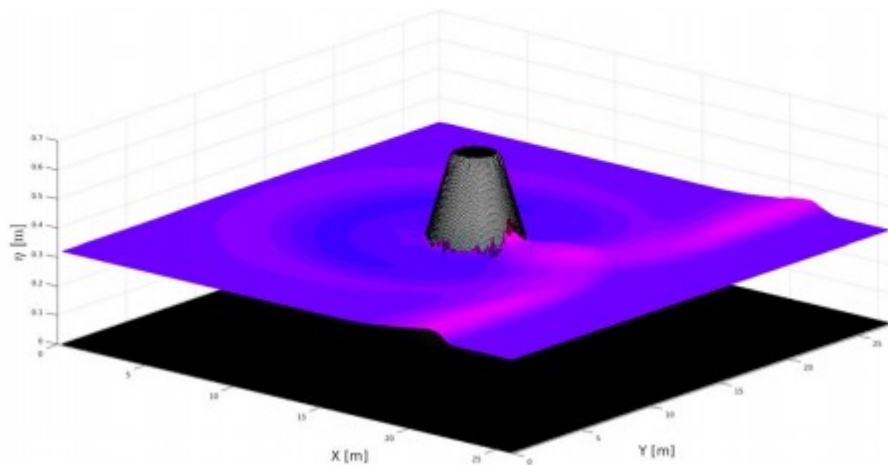
(C) t = 13s

Figure 35

Tsunami wave impact on a conical island: 3-dimensional view of the water depth and the bottom topography calculated by the UEQ scheme at (A) t = 9 s, (B) t = 11 s, and (C) t = 13 s
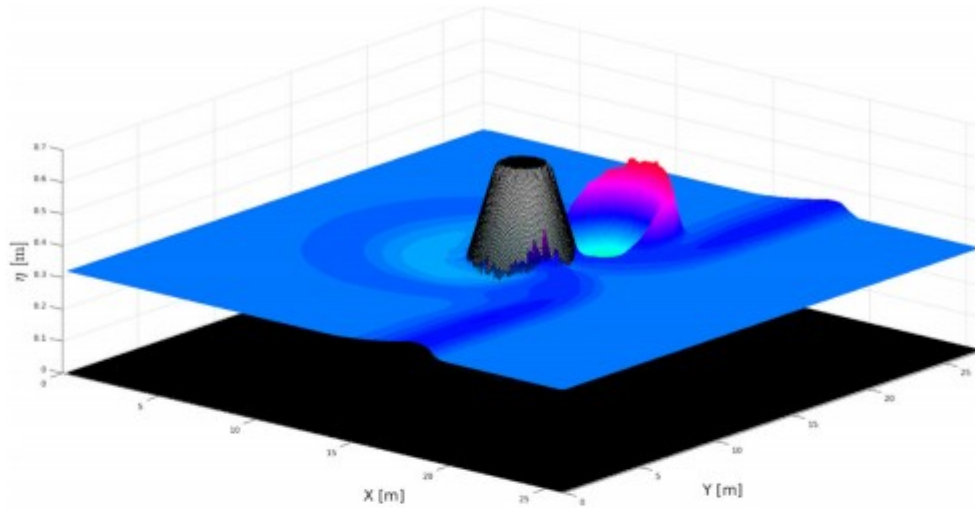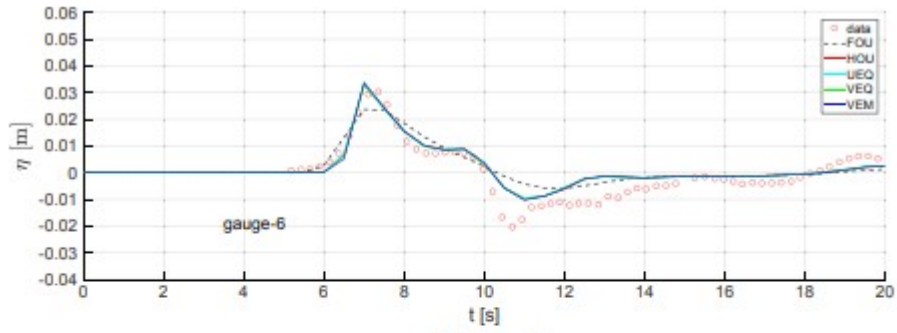
(A) VEM
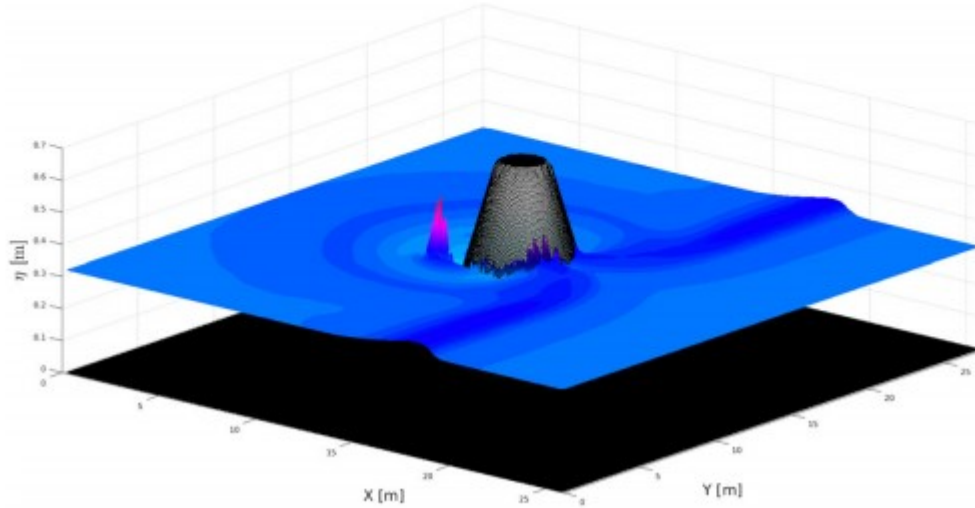
(B) HOU

Figure 36

Tsunami wave impact on a conical island: 3-dimensional view of the water depth and the bottom topography calculated by the (A) VEM and (B) HOU schemes without a limitation for the velocity at 11.5 s

Figure 37 shows the comparison of measurement data from the experiment with the numerical results from different reconstruction methods at gauges located approximately at the run-up area. The gauge numbers 6, 9, 16, and 22 are located at [9.36, 13.80] m, [10.36, 13.80] m, [12.96, 11.22] m, and [15.56, 13.80] m, respectively, as shown in Figure 34. Numerical results capture the peak of the water level at gauge 6 and gauge 9, but for gauge 6, the trough is overestimated by all methods. For gauge 16 and gauge 22, both peak and trough are slightly smaller compared to the measurement data. This is because of the 3-dimensional effects of the wave propagation. The position of the wave is well captured by all the MUSCL reconstructions;

FOU shows the most diffusion. Furthermore, the computed maximum run-up on the island is compared with the measurement data in Figure 38. A slightly overestimated run-up can be observed for the MUSCL schemes at the front face to the wave come direction; this may be the case because of the mesh size relative to the wetting and drying interfaces, which leads to a numerical error, whereas the backwash direction is well captured except for FOU. It can be observed that the difference in the results between the different MUSCL schemes is quite small, but such results are much better than those of FOU. All MUSCL reconstructions are capable of handling wet and dry fronts over uneven terrain.
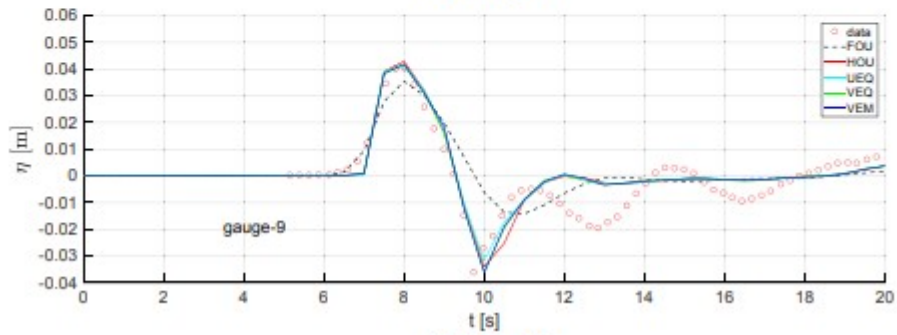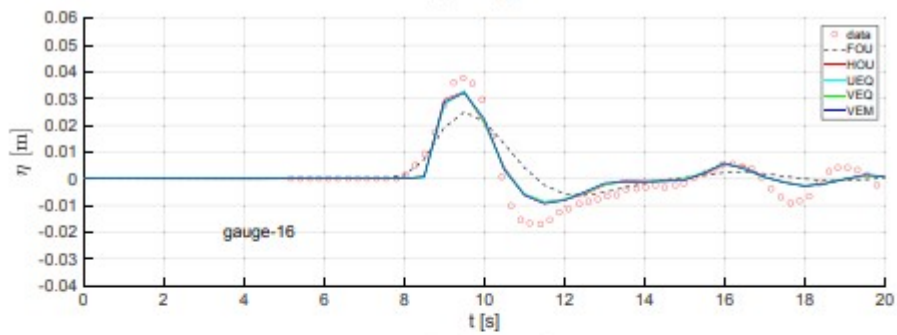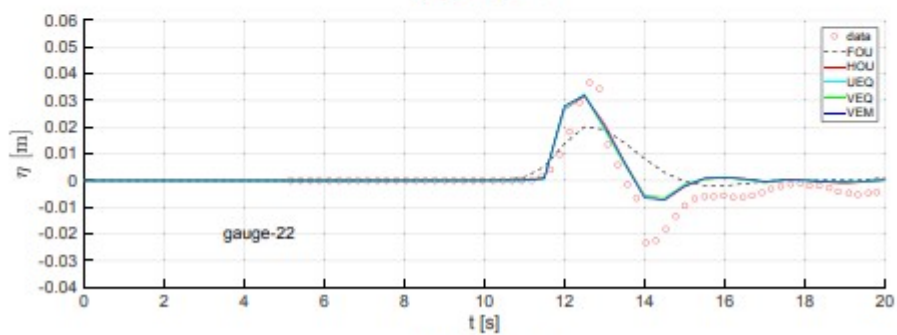
Figure 37

Tsunami wave impact on a conical island: time series of the water level at (A) gauge 6, (B) gauge 9, (C) gauge 16, and (D) gauge 22
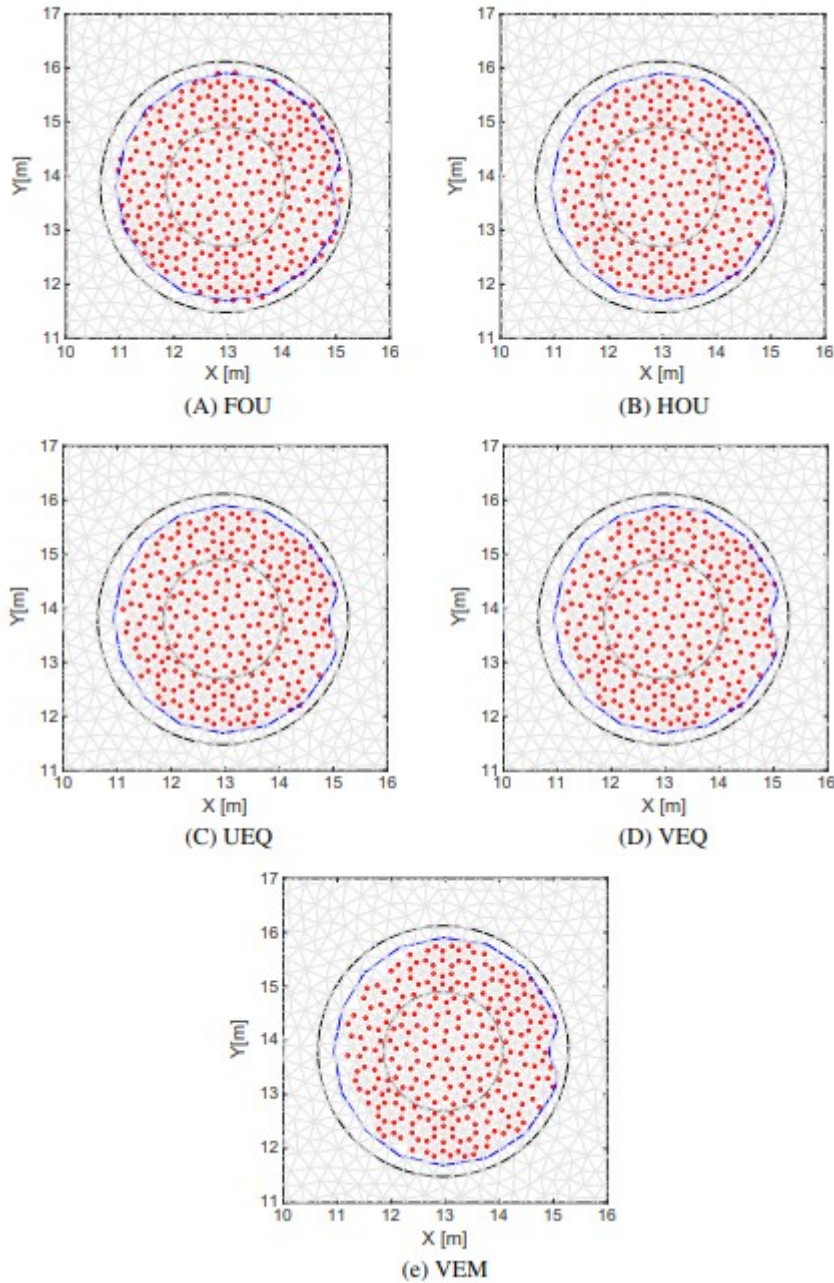
Figure 38

Tsunami wave impact on a conical island: measured maximum run-up (blue lines) and computed dry area (red dot) for schemes (A) FOU, (B) HOU, (C) UEQ, (D) VEQ, and (E) VEM

6 CONCLUSIONS

This paper has presented 2 novel MUSCL reconstruction methods on unstructured grids: UEQ and VEM. Based on the TVD approach, the search for the upwind information is ambiguous. Therefore, we developed improved ways to determine the point from which the upwind information can be obtained. We derived an additional TVD condition, which limits the edge values based on the variables of the cells at the considered edge and

showed that existing MUSCL reconstruction methods do not satisfy this condition. In order to avoid spurious velocity oscillations at the edge, we proposed a treatment for limiting the velocities.

The derived reconstruction methods are tested in 5 test cases. We compared the results with the methods from the works of Buffard and Clain[8] and Hou et al.[13] In most cases, we demonstrate that the novel VEM method is superior to the existing methods, especially on Scottish meshes (Figure 13). The presented numerical scheme is able to handle wet and dry fronts, where the advantage and necessity of the proposed velocity treatment is significant.

ACKNOWLEDGEMENT

REFERENCES

1 van Leer B. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *J Comput Phys*. 1979; 32(1): 101- 136. https://doi.org/10.1016/0021-9991(79)90145-1

2 Toro EF. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. 3rd ed. Berlin, Germany: Springer-Verlag, Berlin Heidelberg; 2009.

3 Godunov SK. A difference method for numerical calculation of discontinuous equations of hydrodynamics [in Russian]. *Matematicheskii Sbornik*. 1959; 47: 271- 300.

4 Guinot V. *Godunov-type Schemes: An Introduction for Engineers*. 1st ed. Amsterdam, The Netherlands: Elsevier Science B.V.; 2003.

5 Harten A. High resolution schemes for hyperbolic conservation laws. *J Comput Phys*. 1983; 49(3): 357- 393. https://doi.org/10.1016/0021-9991(83)90136-5

6 Hou J, Simons F, Hinkelmann R. Improved total variation diminishing schemes for advection simulation on arbitrary grids. *Int J Numer Meth Fluids*. 2012; 70(3): 359- 382. https://doi.org/10.1002/fld.2700

7 Le Veque RJ. *Finite Volume Methods for Hyperbolic Problems*. 1st ed. New York, NY: Cambridge University Press; 2002.

8 Buffard T, Clain S. Monoslope and multislope MUSCL methods for unstructured meshes. *J Comput Phys*. 2010; 229(10): 3745- 3776. https://doi.org/10.1016/j.jcp.2010.01.026

9 Venkatakrishnan V. Convergence to steady state solutions of the Euler equations on unstructured grids with limiters. *J Comput Phys*. 1995; 118(1): 120- 130. https://doi.org/10.1006/jcph.1995.1084

10 Darwish M, Moukalled F. TVD schemes for unstructured grids. *Int J Heat Mass Transf*. 2003; 46(4): 599- 611. https://doi.org/10.1016/S0017-9310(02)00330-7

11 Li X, Liao H. An improved *r*-factor algorithm for TVD schemes. *Int J Heat Mass Transf*. 2008; 51(3-4): 610- 617. https://doi.org/10.1016/j.ijheatmasstransfer.2007.04.051

12 Hou J, Simons F, Hinkelmann R. A new TVD method for advection simulation on 2D unstructured grids. *Int J Numer Meth Fluids*. 2013; 71(10): 1260- 1281. https://doi.org/10.1002/fld.3709

13 Hou J, Liang Q, Zhang H, Hinkelmann R. Multislope MUSCL method applied to solve shallow water equations. *Comput Math Appl*. 2014; 68(12): 2012- 2027. https://doi.org/10.1016/j.camwa.2014.09.018

14 Liang Q, Borthwick AG. Adaptive quadtree simulation of shallow flows with wet–dry fronts over complex topography. *Comput Fluids*. 2009; 38(2): 221- 234. https://doi.org/10.1016/j.compfluid.2008.02.008

15 Liang Q, Marche F. Numerical resolution of well-balanced shallow water equations with complex source terms. *Adv Water Resour*. 2009; 32(6): 873- 884. https://doi.org/10.1016/j.advwatres.2009.02.010

16 Hou J, Simons F, Mahgoub M, Hinkelmann R. A robust well-balanced model on unstructured grids for shallow water flows with wetting and drying over complex topography. *Comput Methods Appl Mech Eng*. 2013; 257: 126- 149. https://doi.org/10.1016/j.cma.2013.01.015

17 Bussing TRA, Murman EM. Finite-volume method for the calculation of compressible chemically reacting flows. *AIAA Journal*. 1988; 26(9): 1070- 1078. https://doi.org/10.2514/3.10013

18 Hou J, Liang Q, Simons F, Hinkelmann R. A 2D well-balanced shallow flow model for unstructured grids with novel slope source term treatment. *Adv Water Resour*. 2013; 52: 107- 131. https://doi.org/10.1016/j.advwatres.2012.08.003

19 Simons F, Busse T, Hou J, Özgen I, Hinkelmann R. A model for overland flow and associated processes within the Hydroinformatics Modelling System. *J Hydroinf*. 2014: 1- 26. https://doi.org/10.2166/hydro.2013.173

20 Liang Q. Flood simulation using a well-balanced shallow flow model. *J Hydraul Eng*. 2010; 136(9): 669- 675. https://doi.org/10.1061/(ASCE)HY.1943-7900.0000219

21 Guinot V, Delenne C. MUSCL schemes for the shallow water sensitivity equations with passive scalar transport. *Comput Fluids*. 2012; 59: 11- 30. https://doi.org/10.1016/j.compfluid.2012.02.001

22 Toro EF, Spruce M, Speares W. Restoration of the contact surface in the HLL-Riemann solver. *Shock Waves*. 1994; 4(1): 25- 34. https://doi.org/10.1007/BF01414629

23 Audusse E, Bouchut F, Bristeau MO, Klein R, Perthame B. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water

flows. *SIAM J Sci Comput*. 2004; 25(6): 2050- 2065. https://doi.org/10.1137/S1064827503431090

24 Roe PL. Some contributions to the modeling of discontinuous flows; 1985.

25 Venkatakrishnan V, Barth T. Application of direct solvers to unstructured meshes for the Euler and Navier-Stokes equations using upwind schemes. Paper presented at: 27th Aerospace Sciences Meeting; 1989; Reston, VA. https://doi.org/10.2514/6.1989-364

26 Hou J, Liang Q, Zhang H, Hinkelmann R. An efficient unstructured MUSCL scheme for solving the 2D shallow water equations. *Environ Model Software*. 2015; 66: 131- 152.

27 Park JS, Yoon SH, Kim C. Multi-dimensional limiting process for hyperbolic conservation laws on unstructured grids. *J Comput Phys*. 2010; 229(3): 788- 812. https://doi.org/10.1016/j.jcp.2009.10.011

28 Bruner C, Walters R. Parallelization of the Euler equations on unstructured grids. Paper presented at: 13th Computational Fluid Dynamics Conference; 1997; Snowmass Village, CO. https://doi.org/10.2514/6.1997-1894

29 Delis A, Nikolos I. A novel multidimensional solution reconstruction and edge-based limiting procedure for unstructured cell-centered finite volumes with application to shallow water dynamics. *Int J Numer Meth Fluids*. 2013; 71(5): 584- 633.

30 Hou J, Liang Q, Simons F, Hinkelmann R. A stable 2D unstructured shallow flow model for simulations of wetting and drying over rough terrains. *Comput Fluids*. 2013; 82: 132- 147.

31 Canestrelli A, Dumbser M, Siviglia A, Toro EF. Well-balanced high-order centered schemes on unstructured meshes for shallow water equations with fixed and mobile bed. *Adv Water Resour*. 2010; 33(3): 291- 303. https://doi.org/10.1016/j.advwatres.2009.12.006

32 Clawpack Development Team. Clawpack Software Version 5.3.1. 2014. http://www.clawpack.org; https://doi.org/10.5281/zenodo.50982

33 Briggs MJ, Synolakis CE, Harkins GS, Green DR. Laboratory experiments of tsunami runup on a circular island. *Pure Appl Geophys*. 1995; 144(3-4): 569- 593.